The University of Hong Kong
Faculty of Engineering
Department of Mechanical Engineering

Data-driven Control Strategies for Precise Manipulation of Continuum Robots

Wang Xiaomei

Oct 2020

A thesis submitted in partial fulfilment of the requirements of
the Degree of Doctor of Philosophy
at The University of Hong Kong

Abstract of thesis entitled

"Data-driven Control Strategies for Precise Manipulation of Continuum Robots"

Submitted by

Wang Xiaomei

for the degree of Doctor of Philosophy
at The University of Hong Kong
in October 2020

Continuum robots are designed to inherently possess dexterity and adaptability, as bio-inspired by elephant trunks and octopus tentacles. This compliance with surroundings endows them the advantage of safe operation in confined space, even in medical applications. However, the tradeoff between manipulator flexibility and precise control for the infinite degrees of freedom requires particular approaches concerning modeling and sensing. Different from conventional rigid-link robots whose kinematics/dynamics could be certainly determined, modeling uncertainties due to intrinsic and extrinsic factors would apparently deteriorate the model-based control performances. Under such circumstances, data-driven strategies making use of learning algorithms would be a promising way out for continuum robot control.

The main focus of this thesis is to propose appropriate control schemes and sensing modalities for continuum robots, utilizing data-driven algorithms to resolve or enhance the precise manipulation under various environments. Visual servoing control is a significant focus of this thesis, where the self-contained camera on the robot end-effector can provide intuitive perception and positional feedback. A novel visual servoing framework utilizing localized online learning enables stable path following in the camera view, even under external forces and varying tip loading. Optical fiber inscribed with fiber Bragg gratings (FBGs) can enhance the feature tracking accuracy in the 2D visual servoing. Integrated with the flexible fiber, the vision-based sensing can possess higher reliability under poor or even extreme visual conditions (e.g., full shielding, absolute darkness). The proposed

online learning-based sensing fusion of camera and FBG measurements can also provide robust 6D pose estimation of the end-effector, expanding the task space filed for continuum robot control. Referring to the convergence guaranteed by kinematic model-based control and the accommodation accredit to learning-based control, a hybrid controller is designed to incorporate their strengths. A kinematic model without the need for fine tuning and an online-updated error compensator can increase the control accuracy, as well as circumvent prior data exploration and training at the same time. The assistance of FBG is also investigated on shape tracking and learning-based modeling of slim continuum robots, which are validated on a robotic catheterization platform. Continuum robot visual servoing has excellent potential in robotic endoscopy, while the optical fiber's compatibility to magnetic resonance imaging (MRI) also indicates the future clinical application of the proposed shape sensing and control.

(Word count: 373 words)

Declaration

I declare that this thesis represents my own work, except where due acknowledgment is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed _____

Wang Xiaomei

Dedication

To my parents
who always trust and encourage me unconditionally
no matter where I am and what I decide to do

To my friends
who have inspired and supported me
in countless anxious and tough moments

To my teammates
who have accompanied and helped me
to keep enthusiastic about research and life

Acknowledgements

My Ph.D. journey gave me an unprecedented growth experience, mostly accrediting to my supervisor, Dr. Ka-Wai Kwok. He has brought me into the field of robotics, which was a brand new domain for me four years ago. His dynamism, motivation, and vision have deeply inspired me, teaching me to "create" rather than "fabricate". His comprehensive training method has urged me to break my limitation and pursue excellence, as well as guided me to communicate and collaborate. I would like to express my sincere gratitude to him, for anything he motivated and taught me.

I am also grateful to my co-supervisor, Prof. James Lam, for his patient help and encouragement. I also want to extend my gratitude to Dr. Alex Pui-Wai Lee, Prof. Kaspar Althoefer, and Prof. Yun-Hui Liu, for their valuable advice in my research projects.

I am also thankful to all my teammates in IRIS, who accompanied me for the longest time in these four years. I enjoy the time with them for experiments, discussion, writing, and even hiking on Victoria Peak. The crazy "fighting" experiences before deadlines will be unforgettable memories for me. I would like to thank Dr. Ziyan Guo, Dr. Kit-Hang Lee, and Dr. Denny Kin-Chung Fu, who enlightened me a lot particularly during my early stage of Ph.D. study; Dr. Ziyang Dong, Ge Fang, Kui Wang, Justin Di-Lang Ho, Zhuoliang He, Alan Wai-Lun Tang, Jing Dai, Chris Hon-Sing Tong, Jacky Hin-Choi Fu, Chim-Lee Cheung, Owen Yui-Lun Ng, Zhiyu Liu, Martin Chun-Wing Leong, Dr. Xiaochen Xie, Dr. Peggy So-Sum Leung, Dr. Angel Siu-Yan Ng, and Becky Tik-Sze Chung, for their bountiful help in my research projects and daily life. It is my great honor to be in the same group with you all.

Last but the most important, I want to thank my parents and best friend, Yin, for their considerate and unlimited supports. Without you, I may not be strong enough to keep on.

Contents

Abstract **1**

Acknowledgements **7**

List of Figures **13**

List of Tables **23**

Nomenclatures **25**

1 Introduction **29**

1.1 Motivation and Objectives 29

1.2 Structure and Contributions of Thesis 32

1.3 Publications, Patents and Awards during Ph.D. Study 35

2 Control, Sensing and Applications of Continuum Robots **37**

2.1 Introduction 37

2.2 Eye-in-hand Visual Servoing and Applications 39

2.3 Fiber Bragg Grating (FBG) in Soft Robot Sensing 41

2.4 Control Algorithms for Continuum Robots 44

2.4.1 Kinematic/dynamic Model-based Algorithms 44

2.4.2 Learning-based Algorithms 45

2.5 Conclusion 47

3 Kinematic Modeling and Learning Approaches for Continuum Robots **48**

3.1 Introduction 48

3.2 Piecewise Constant Curvature (PCC) Kinematics 49

3.2.1 Robot-independent Properties 50

3.2.2 Robot-specific Properties 53

3.2.2.1 Two-dimension Actuation 53

3.2.2.2 Three-dimension Actuation 54

3.2.3 Inverse Kinematics for Model-based Control 56

3.3 Simulation Results of PCC Model 57

3.3.1 Two-channel Tendon-driven Mechanism 57

3.3.2 Three-chamber Pneumatic-driven Mechanism 59

3.4 Data-driven Control Algorithms 60

3.4.1 Convex Optimization-Based Method 60

3.4.2 Gaussian Process Regression (GPR) 61

3.4.3 Conclusion 63

4 Eye-in-hand 2D Visual Servoing and 6D Pose Estimation 64
4.1 Introduction . . . 64
4.2 Online Learning for 2D Visual Servoing Control . . . 65
4.2.1 Task Space Definition . . . 66
4.2.2 Motion Estimation on Image Plane . . . 67
4.2.3 LGPR-based Online-learning Control . . . 69
4.2.3.1 Localized GPR Model . . . 69
4.2.3.2 Incremental Learning . . . 71
4.3 Experiments of 2D Visual Servoing . . . 73
4.3.1 Experimental Setup . . . 73
4.3.2 Pre-training of Local GPR Inverse Model . . . 74
4.3.3 Experiments and Results . . . 76
4.3.3.1 Point-to-point Tracking . . . 76
4.3.3.2 Target Tracking under External Forces . . . 78
4.3.3.3 Path Following with a Scarce Pre-trained Model . . . 79
4.3.3.4 Path Following under Varying Load . . . 81
4.4 SLAM-based 6D Pose Estimation of Robot End-effector . . . 83
4.4.1 Task Space Definition . . . 83
4.4.2 Camera Pose Estimation via ORB-SLAM2 . . . 84
4.5 Conclusion . . . 86
5 FBG-aided Modeling and Control for Tendon-driven Catheters 87
5.1 Introduction . . . 87
5.2 MRI Guidance in Robotic Interventions . . . 89
5.3 Clinical Motivation of Cardiac EP Catheterization . . . 90
5.4 Teleoperation Platform of Robotic Catheterization . . . 92
5.4.1 Generation of Endoscopic View . . . 93
5.4.2 Control Methodologies . . . 94
5.4.2.1 Kinematic Model-based Control . . . 95
5.4.2.2 Model-free Control . . . 97
5.4.3 Experimental Setup . . . 98
5.4.4 Evaluations via Simulated Ablation Task . . . 101
5.5 Shape Tracking with FBG Sensors . . . 104
5.5.1 Shape Sensing with Multi-core FBG Fiber . . . 105
5.5.2 Fusing and Calibration of Shape and Positional Feedback . . . 107
5.5.3 Towards Shape Tracking in MRI . . . 109
5.6 Learning-based Modeling and Control of Cardiac Catheter . . . 111
5.6.1 Modeling of Catheters . . . 111
5.6.2 System Characterization with a Learning-based Method . . . 113
5.6.3 Autonomous Motion Control and Ablation . . . 116
5.7 Performance Evaluation of Shape Tracking and Autonomous Control . . . 117
5.7.1 MR Safe Robotic Catheter Platform . . . 117
5.7.2 Shape Sensing Performance of the Multi-core FBG . . . 117

5.7.3 Shape Tracking under Active Bending . . . 120
5.7.4 Learning-based PCC vs CC . . . 122
5.7.5 Autonomous Targeting . . . 124
5.7.6 Path Following . . . 126
5.7.7 LA Simulator with Pulsatile-like Flow and *Ex-Vivo* Tissue . . . 129
5.7.8 *Ex-vivo* Tissue Ablation with Simulated Pulsatile Flow . . . 131
5.8 Conclusion . . . 133
6 Optical-FBG-Enhanced Visual Servoing 135
6.1 Introduction . . . 135
6.2 FBG Enhanced 2D Visual Servoing Control . . . 137
6.2.1 Task Space Definition . . . 137
6.2.2 Learning-based Motion Estimation Combined with FBG . . . 138
6.2.2.1 Data-driven Motion Estimation with FBG . . . 138
6.2.2.2 Camera-FBG-combined Motion Estimation . . . 140
6.2.3 Model-less Visual Servoing Based on 2D Motion Estimation . . . 141
6.3 Experiments of Enhanced 2D Visual Servoing . . . 144
6.3.1 Continuum Endoscopic Robot with FBG Sensor Device . . . 144
6.3.2 Experimental Setup . . . 145
6.3.3 Visual Servo in Feature-deficient Scenes . . . 147
6.3.4 Visual Servo under External Disturbances . . . 151
6.4 FBG Enhanced SLAM Pose Estimation . . . 156
6.4.1 Task Space Definition . . . 157
6.4.2 Learning-based Pose Estimation using FBGs . . . 158
6.4.3 Camera-FBG Sensing Fusion . . . 161
6.5 Hybrid Position Control of Continuum Robots . . . 162
6.5.1 Kinematics Initialization by CC Model . . . 163
6.5.2 Online Data-driven Error Compensator using GP . . . 164
6.6 Experiments of 6D Pose Estimation and Hybrid Control . . . 166
6.6.1 Pose Estimation by ORB-SLAM2 . . . 166
6.6.2 Sensor Fusion Pose Estimation . . . 169
6.6.2.1 Pose Estimation under Moving Obstacles . . . 170
6.6.2.2 Pose Estimation under Varying Lighting Condition . . . 172
6.6.3 Tracking: Hybrid Control vs Model-based Control . . . 175
6.6.4 Hybrid Control with Sensing Fusion . . . 180
6.7 Conclusion . . . 182
7 Conclusion 184
7.1 Summary of Thesis Achievements . . . 184
7.2 Future Work . . . 187
References 187

List of Figures

2.1 Continuum robot examples with different actuation mechanisms. **(a)** A (intrinsic) pneumatic-driven robot enabling omni-directional bending. **(b)** A (extrinsic) tendon-driven manipulator in Hansen Medical Sensei system targeting catheterization. **(c)** Air-Octor multisection continuum robot driven by tendons and stiffened by pneumatic-pressured backbone. **Image source**: [1–3]. 38

2.2 Eye-in-hand and eye-to-hand modalities of visual servoing, sorted by the camera's position. *Eye-in-hand* indicates the camera embedded on the robot end-effector while *eye-to-hand* stands for a statically-fixed camera providing a global sight of the scene. **Image source**: [4]. 39

2.3 Experimental setups of existing continuum-robot visual servo control. **(a)** A cable-driven robot commanded for targeting a black point on white background in free space and **(b)** constrained by a static physical restriction. **(c)** An sPAMs equipped with a commercial image processing system aiming at visual servoing [5]. **(d)** A cable driven robot validated in underwater scenario. **Image source**: [5–8]. 40

2.4 Working principle of FBG technology. FBG is a permanent periodic variation in the refractive index value of the fiber core. It can reflect a particular wavelength of light and transmit others. The reflected wavelength is determined by the periodicity of the grating and will shift with the fiber axial elongation or temperature change. **Image source**: [9]. 42

2.5 Distributions of single-core fiber triplets in model-based shape reconstruction. **(a)** Three fibers are fixated along the continuum manipulator parallelly and evenly, where a corresponding geometry model is used for shape reconstruction. **(b)** Three fibers are helically-wrapped along the manipulator thus a model considering torsion besides curvature is established. **Image source**: [10, 11]. 43

3.1 Mappings between three spaces defined in the kinematics of constant-curvature model. Robot-specific mapping f_s, means the relation from actuation space $\boldsymbol{u}$ to configuration space r, θ and ϕ; robot-independent mapping f_i transforms the parameters from configuration space to task space, that is, the workspace of end-effector. 49

3.2 Configuration of the robot backbone under the constant curvature assumption, where the robot base is regarded as the coordinate origin. **(a)** The angle ϕ represents the rotation/bending direction angle of arc out of $\boldsymbol{x}$-$\boldsymbol{O}$-$\boldsymbol{z}$ plane. **(b)** The plane that the spatial arc lies in is represented as $\boldsymbol{x}'$-$\boldsymbol{O}$-$\boldsymbol{z}$. The radius, center angle and total length of this arc is defined as r, θ and l, respectively. 50

3.3 Configuration of PCC-constructed backbone on the bending plane $\boldsymbol{x}'$-$\boldsymbol{O}$-$\boldsymbol{z}$. The robot backbone is approximated by n curve segments with different curvatures, which curvature of the i^{th} curve represented by κ_i. 51

3.4 **(a)** Schematic diagrams of a typical tendon-driven continuum manipulator in CC configuration. CC geometry is characterized by configuration space parameters as in **Fig. 3.2**. Insertion distance and rotation angle of the manipulator body are represented by d and ϕ. The deformation is determined by the relative position of two tendons, marked by ΔL. The constant distance between two tendons is ΔR. **(b)** PCC (**Fig. 3.3**) geometry of a tendon-driven continuum manipulator on the bending plane, with the tendon-related variables illuminated. 54

3.5 CC model of continuum robots. **(a)** Configuration-space parameters that are defined to describe the spatial arc. **(b)** A top view of the robot base. 55

3.6 Simulation performance of a two-channel tendon-driven continuum robot, in the task of tracking a moving point along a square path. **(a)** The desired trajectory (10×10 mm^2) is defined on the $\boldsymbol{x}$-$\boldsymbol{O}$-$\boldsymbol{z}$ plane, while the robot initial backbone (65-mm length) is along the z axis. **(b)** Tracking errors on different directions. **(c)** A simulated sketch of the robot backbone during the tracking task. 58

3.7 Simulation performance of a three-chamber pneumatic-driven continuum robot, in the task of tracking a moving point along a circle. **(a)** The desired trajectory (30×30 mm^2) was defined on the $\boldsymbol{x}$-$\boldsymbol{O}$-$\boldsymbol{y}$ plane, while the robot initial backbone (90-mm length) was along the z axis. **(b)** Lengths of three pneumatic chambers enabling the end-effector to follow the circle path. **(c)** Parameters $\boldsymbol{u}$ in the configuration space during the tracking. 59

4.1 Schematic diagram of motion estimation. **(a)** Camera coorindate frames at time step k and $k+1$; **(b)** Incremental motion in image plane can be acquired based on the displacement from the template pattern (in red block) to the matched block (in yellow) that is searched by block sliding. 66

4.2 **(a)** Motion estimation errors sampled at angular resolution of 0.25°. **(b)** Twisting angle of robot end-effector (camera) under varying actuation inputs when one chamber is inflated. All three chambers are pre-pressured with actuation input $\boldsymbol{u}_0$. 68

4.3 Proposed learning-based control architecture. Parameters $\boldsymbol{u}$ and $\boldsymbol{z}$ denote, respectively, the actuation command and the position of captured image features. The input unit provides the positional command in image domain, where the target position $\boldsymbol{z}^*_{k+1}$ can be selected manually or predefined by a reference trajectory. The LGPR-based control unit generates the actuation command $\Delta\boldsymbol{u}_k$, referring to the desired displacement $\Delta\boldsymbol{z}^*_k$ and current state $\boldsymbol{u}_k$. The image processing unit esitmates the real-time displacement $\Delta\boldsymbol{z}_k$ for the online update of LGPR models and the feedback control. 72

4.4 Experimental setup in a scene of LEGO®. The soft manipulator was made of silicone rubber, and which was driven by three fiber-constrained air chamber. A endoscopic camera and five LEDs were mounted at the tip. . . . 74

4.5 Three thousand sample pairs of robot actuation and tip position collected for pre-training of the inverse model. **(a)** Using the k-means algorithm, all these training points were divided into six (colored) clusters based on their actuation inputs. **(b)** Corresponding tip position distribution of six clusters is also shown. 75

4.6 Five targets manually selected on the image plane for robot tracking. **(a)** Mosaic image obtained during multiple target tracking. A template pattern (in red block) was form and centered at the target point selected. A series of green bubbles represent the instantaneous camera centers, showing the trajectory travelled along the matched template patterns. **(b)** Corresponding robot configurations while tracing targets successively at its image plane center. **(c)** Absolute tracking errors throughout the journey. 77

4.7 Target tracking experiment involving external disturbance. The upper pictures (from left to right) show three robot configurations before, during and after the external force application. The correspoding tracking errors are shown below. The 10-pixel tolerance level is marked (dash lines in orange). 78

4.8 Performance of tracking on a predefined "∞" trajectory. **(a)** Robot tracing the reference trajectory in its endoscopic view using the pre-trained LGPR models. The red-dashed block indicates the initial view point centered at the intersection of the "∞" trajectory. **(b)** Motions depicted in two separated coordinates, w and v. **(c)** Corresponding tracking errors throughout the 135-second journey. **(d)** Tracking performances in general. 80

4.9 Tracking performance under variable tip loading. A (6-gram) balloon cap pumped with water in-and-out was mounted at the robot tip, introducing a variable load ranging from 6 to 21 g. **(a)** Tracked trajectory in three cycles. Three substantial deviations corresponded to the errors of pre-training (1-st cycle in red), injecting and removing water (2-nd cycle in blue); **(b)** Corresponding tracking error in time domain; **(c)** Tracking error *vs* variable load varied below 21 g. The load is represented as % relative to the mass of robot itself, 20 g; **(d)** Robot configuration and its balloon shape at three-time steps. 82

4.10 Footprint of a soft continuum robot constructing the 3D position workspace. The pneumatic-driven robot was actuated by a pre-defined input sequence with increasing values. 83
4.11 Module diagram of ORB-SLAM2, where the tracking, local mapping, and loop closing threads run in parallel. In the sparse map, the red points are recorded landmarks while the blue and green markers represent the camera poses corresponding to key frames and camera accumulated trajectory, respectively. 85

5.1 Schematic diagram of RF ablation in LA. **(a)** Sketch diagram of RF ablation in LA, LA, where the catheter passes through IVC to RA, punctures the atrial septum, and reaches LA for RF ablation. **(b)** Atrial fibrillation corrected with ablation surgery, where non-conductive scars on PVO isolate the abnormal EP signals. **Image source**: [12] 91
5.2 MRI visibility of RF and cryo ablated lesions imaged using non-contrast **(a)** T1w, **(b)** T2w, and contrast-enhanced **(c)** T1w and **(d)** late gadolinium enhancement techniques. Wide arrows mark RF ablation results, while narrow arrows represent cryo-ablation. All RF ablated lesions could be readily detected on non-contrast images. **Image source**: [13] 92
5.3 MR safe robotic catheter platform providing the 3-DoF manipulation (bending, rotation and insertion) of standard cardiac electrophysiology (EP) catheter. 93
5.4 **(a)** Virtual lesion targets near PVO pre-defined in the EP roadmap, with successfully ablated targets displayed in yellow and to-be-completed ones in red. The normal direction of endoscopic view was set to follow the axial direction of catheter tip; **(b)** Diagram illustrating the incremental movement of catheter tip from time step k to $(k+1)$. 94
5.5 Schematic diagram illustrating the manipulation mechanism of a standard catheter. **(a)** CC geometry of the catheter distal section. **(b)** 3-DoF manipulation of the catheter handle, including the axial translation d, knob twisting ϕ, and handle rotation α. 95
5.6 LA simulator and EP ablation catheter. A PTFE pipeline guided the catheter to approaching the PVO in a similar position and direction to cardiovascular interventions. EM tracking system was used to provide positional measurement. 99
5.7 Virtual visualization of the catheter tip providing with simulated LA roadmap displayed in three camera perspectives, including two virtual overall views and one virtual endoscopic view. Simulated roadmap was indicated with circular lesion targets. Successfully ablated targets would turn yellow. 100
5.8 Human-robot control interface and realization of hand-to-eye coordination. The motion of catheter tip reflected in the endoscopic view was designed to be synchronized with the input motion from the operator. 101

5.9 Actual footprint of the catheter tip recorded by the EM tracking system. The LA model was overlaid in the same coordinate. 102

5.10 Two major performance indices, namely accuracy and efficiency defined based on **(a)** proximity distance measured from tip to lesion target around the PVO and **(b)** total length of incomplete lesion segments (red). 102

5.11 Performance comparison of kinematic model-based and model-free control methods regarding the accuracy and effectiveness indices. The lower the p-value (≤ 0.05), the higher the probability. 104

5.12 Diagram showing the architecture of shape sensing for robotic catheter system. **(a)** Catheter distal bending section integrating a multi-core FBG fiber (12-m long) and the tracking marker for shape tracking; **(b)** Inner structure of distal bending section, with multi-core FBG fiber put throughout the water channel; **(c)** Handle of cardiac catheter, where the FBG fiber can be fixed at the water channel entrance; **(d)** Real-time shape reconstruction can be achieved by continuously acquiring the strain data from FBGs. The 3D curvature of the catheter bending section can then be feedback to the robotic catheter platform for autonomous control. 106

5.13 Diagram illustrating the shape tracking method by *fusing* the 3D curvature from FBG and positional sensing from the tracking coil. 108

5.14 **(a)** MR image revealing the MR tracking markers and the reconstructed 3D positions from the 1-D projection pulse-sequence; **(b)** 1D projection of the three tracking coils along the orthogonal axes. Three RF signal peaks can be found in each projection. 110

5.15 Configuration of the catheter approximated by several curve segments. The curvature of i^{th} segment can be represented by κ_i. A straight section is added at the distal end on top of the PCC segments, which is the rigid tip of catheter. 112

5.16 Control architecture of the autonomous ablation procedure. Parameters of the PCC model can be automatically tuned by a learning-based algorithm, which enables accurate positional control of the catheter end-effector. The 3D curvature of catheter can be obtained in real-time from the FBG feedback. RF ablation will be triggered when the catheter tip reaches the target and contacts tissue. 116

5.17 **(a)** Bending curvature fixtures used to evaluate the shape sensing performance of the multi-core FBG fiber. Seven curvatures were included at each side of bending, with an absolute value from 0° to 105°; **(b)** Reconstructed catheter shapes compared with the ground-truth curves. The starting positions are all aligned at $(0,0)$ in the plot; **(c)** Bending-angle errors along the shape sensing section. 119

5.18 **(a)** Top and side views of catheter shapes obtained from FBG shape sensing (red), EM tracking system (black), as well as model prediction (green). The catheter was actuated by the robotic platform with seven knob steering angles, which were also used for model input; **(b)** Spatial differences along the shape sensing section by comparing curves obtained from FBGs and the EM tracking system. 121

5.19 **(a)** Simulated catheter bending curves by the CC model and the learning-based PCC model with five knob steering angles (I1-I5); **(b)-(c)** Diagrams showing the curvature and tip bending angle variation trends for CC model and learning-based PCC model. The learning-based PCC model can predict the actuation hysteresis by considering the actuation direction. . 123

5.20 **(a)** Diagrams showing the catheter tip trajectory towards the five targets during the autonomous targeting. The reconstructed catheter shape and position are plotted for showcasing the instances when the tip reached the targets; **(b)** Deviation from the tip to the targets during the five stages. . . . 125

5.21 **(a)** The rendering color (cool to warm) represents the tip deviation (small to large) from the desired path. The reconstructed catheter shapes and positions were obtained from the proposed shape tracking approach; **(b)** Deviation of the catheter tip from the desired path during 5 cycles. The catheter was controlled to quickly trace the path during the approaching stage; **(c)** Top view showing the catheter bending shapes at the five instances. 127

5.22 **(a)** Catheter tip footprint under autonomous control with the CC model. Large deviation could be found at segments with red color; **(b)** Tip deviation from the desired path. Slow and delayed retracement could be observed at several segments due to the inaccurate estimation of kinematic mapping done by the CC model. 128

5.23 **(a)** Left atrial (LA) simulator filled with liquid, of which the simulated pulsatile flow was generated by a hydraulic pump. A slice of swine tissue was attached on the inner surface at the target ablation area (around PVO); **(b)** Simulated liquid pressure compared to the human ECG and LA pressure; **(c)** Pressures in LA simulator with fast and slow rates. The simulated pulsatile rate could be adjusted within the range. **Image Source**: [14] 130

5.24 Results of the PVI task conducted in the LA simulator. A standard EP catheter was autonomously controlled to reach a series of targets around the simulated PVO. **(a)** Front view showing the catheter tip footprint, the targeted ablation points (red sphere), and the desired lesion path (yellow). The footprint (blue line) indicates three selective tip trajectories towards the targets; **(b)** Tissue with ablated points. 132

6.1 **(a)** Example FBG wavelength shift changes from $\lambda(k)$ to $\lambda(k+1)$, **(b)** due to the corresponding robot deformation/bending, the difference of camera coordinate frames at k and $k+1$ can be seen; **(c)** Incremental motion in the image plane acquired based on the displacement from the template pattern (yellow dotted block) to the matched block (red) that is found by block sliding in the search window (purple block). 137

6.2 Finite element modeling (FEM) 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. 138

6.3 Control architecture of the proposed control method. Parameters $\boldsymbol{u}$ and $\boldsymbol{z}$, respectively, denote the actuation command and the position of the captured feature in camera view. The input unit provides the positional command in the image view. A model-free feedback controller is responsible for generating the inverse solution to the actuator, causing the object movement in the camera view as well as the change of FBG wavelength. The actual displacement after the execution of actuation command is obtained by the weighted sum of FBG-based motion estimation and image processing. . . . 143

6.4 Pneumatic-driven 3-chamber robot was used. An optical fiber with multiplexing 16 FBGs was helically wrapped around the manipulator for sensing feedback of robot configurations in real-time. A monocular endoscopic camera and a LED module were fixed on the tip cap of robot. Cross-section view of the robot could show the silicone chambers constrained individually with helical Kevlar strings. 144

6.5 Endoscopic scene in abdominal surgery simulated with swine viscera. The robot was pre-bended to approximate insertion angle of a laparoscope. EM tracking coils were attached on the tip as the ground truth of end-effector pose. 145

6.6 Robot following of a predefined "Batman" path in the scene of **Fig. 6.5**. **(a)** The motion estimated by image processing of the endoscopic view *alone*; **(b)** Features in the red block were selected by the user *before* the motion displacement. *After*, such a block was expected to keep matching/tracking at the same square of features (black dotted block), acting as a static reference for robot to "draw" the path; **(c)** Actual path of end-effector recorded by EM tracking coils, which was projected on the same *u*-*v* coordinates. 148

6.7 FBG-enhanced tracking performance of the same "Batman" path. **(a)** The trajectory in the endoscopic view; **(b)** The deviation between the selected feature after three cycles. **(c)** Actual trajectory recorded by EM trackers. . . 149

6.8 Tracking error analysis of the "Batman" path in the laparoscopic scene (**Fig. 6.5**). **(a)** Tracking achieved by image processing, e.g. mean error: 94.45 (SD: 53.55) induced in Cycle 1; **(b)** Tracking achieved by FBG-enhanced estimation. Curves represent the absolute tracking error in **Fig. 6.6c**, and **Fig. 6.7c** and red points illustrate the maximum error in each cycle. 150

6.9 Setup in a scene of LEGO®. Robot was fixed downward viewing the workspace scene. An air nozzle was fixed facing the robot to generate force disturbances. EM tracking coils were attached as the ground truth. 151

6.10 Performance of robot following on the predefined "Batman" path in the scene of **Fig. 6.9**. Force disturbances are applied to the end-effector from the nozzle, with 4 bar to 6 bar pressure and released in the 3rd cycle. **(a)** The motion estimated by image processing *alone* vibrates during and after the disturbances applied; **(b)** Features in red block was selected by user *before* the motion displacement. *After* then, yellow dotted block indicates the position in $before$; **(c)** Actual path of end-effector deviates gradually after three cycles. 153

6.11 Performance of robot following on the "Batman" path in the scene of **Fig. 6.9** with the motion estimation by enhanced method. **(a)** Robot motion in the endoscopic view maintains stable under and after external forces; **(b)** Feature deviation is well solved; the actual path in **(c)** demonstrates the improvement. 154

6.12 Tracking error analysis in the LEGO® scene (**Fig. 6.9**). **(a)** Tracking achieved by image processing; **(b)** Tracking achieved by FBG-enhanced estimation. Curves represent the absolute error in (**Fig. 6.10c** and **Fig. 6.11c**). 155

6.13 Soft continuum robot wrapped with a helically-wound single-core FBG fiber. **(a)** Tip camera poses obtained at each time step, k, based on SLAM algorithm. **(b)** FBG wavelengths shifted correspondingly, i.e., from $\boldsymbol{\lambda}(k)$ to $\boldsymbol{\lambda}(k+1)$. 157

6.14 Robot control architecture. Hybrid controller combining kinematic model and data-driven trained compensator is implemented, with the pose feedback obtained by the sensing fusion of strain measurement and visual feedback. Mono-camera 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 proportion in the sensing fusion will be more heavily weighted in visual feature-deficient scenarios. 165

6.15 Command sequence $\boldsymbol{U}$ to the actuation unit (three sets of stepper motors adjusting the volume of pneumatic chambers via cylinders) during the data-exploration. 166

6.16 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 3D, which are reconstructed using the SLAM pose estimation and image feedback. 168
6.17 Sensor fusion under 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. **(c)** Trajectories of fusion-, SLAM- and tracker-obtained camera positions. 171
6.18 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-based and fusion-based pose estimation compared with the ground truth obtained by EM tracking. 172
6.19 Scenario reconstructions with disturbances of **(a)** moving obstacle (**Fig. 6.17**) and **(b)** varying lighting condition (**Fig. 6.18**). Several blurs due to the obstacle or varying lighting are indicated by dotted outlines. 174
6.20 Control performance of tracking a moving target 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 2,000 time steps (each 0.05s), afterwards another 2,000 steps run with the hybrid control. Two series of 2,000 time steps data are plotted and overlaid for ease of comparison. **(c)** Zoom-in of the square blocks in **(a)** and **(b)**. 177
6.21 Performance of tracking on a circle path, where K_p were set as **(a)** 0.1 and **(b)** 0.06. Subfigures are arranged as same as in **Fig. 6.20**. 179
6.22 Tracking performance along an elephant contour. The sensing feedback was provided by visual-strain pose fusion. CC model-based control was conducted for the 1st-cycle tracking, after which the learning-based error compensator was included. In the 3rd cycle, a moving obstacle applied visual disturbances in the camera view. Trajectories of the three cycles were plotted in **(a)**, as well as the tracking error **(b)** and the path deviation **(c)** with mean values indicated. 181

List of Tables

5.1 Evaluation results of 10 sujects in performing the teleoperation task using model-based and model-free control. 103

6.1 Mean tracking errors and path deviations in *pentagram* tracking with four different gain K_p. 176
6.2 Mean tracking errors and path deviations in *circle* tracking with four different gain K_p. 178

Nomenclature

Number Sets

$\mathbb{N}$	Natural Numbers
$\mathbb{R}$	Real Numbers
$\mathbb{U}$	Actuation Space

Acronyms

2D	2-dimensional
3D	3-dimensional
6D	6-dimensional
AR	Augmented reality
BA	Bundle adjustment
CC	Constant curvature
CNN	Convolutional neural network
CT	Computed tomography
DLL	Dynamic link library
DoF	Degree of freedom
EAM	Electro-anatomical map
ECG	electrocardiogram
ELM	Extreme learning machine
EM	Electromagnetic
EP	Electrophysiology
FBG	Fiber Bragg grating
FEM	Finite element modeling

IMU	Inertial measurement unit
Intra-op	Intraoperative
IVC	Inferior vena cava
LA	Left atrial/atrium
LGPR	Localized Gaussian process regression
LSTM	Long short-term memory
MIP	Minimally invasive procedures
MR	Magnetic resonance
MRI	Magnetic resonance imaging
NN	Neural network
OFDR	Optical frequency domain reflectometry
OFS	Optical fiber sensors
PCC	Piecewise constant curvature
PET	Positron emission tomography
PnP	Perspective-n-Point
Pre-op	Preoperative
PTFE	Polytetrafluoroethylene
PVI	Pulmonary vein isolation
PVO	Pulmonary vein ostia
RA	Right atrium
RBF	Radial basis function
RF	Radio-frequency
RMSE	Root mean square error
ROS	Robot Operating System
SfM	Structure from motion
SLAM	Simultaneous localization and mapping
SLFNs	Single-hidden-layer feedforward networks
sPAMs	Series pneumatic artificial muscles

UDP	User Datagram Protocol
US	Ultrasound
VO	Visual odometry
w.r.t.	With respect to
WDM	Wavelength division multiplexing

Chapter 1

Introduction

1.1 Motivation and Objectives

Continuum robots are naturally inspired by the dexterity and adaptability exhibited via snakes, elephant's trunks, and octopus' tentacles [15]. In contrast to conventional rigid-link manipulators, "continuum" mechanisms leverage a series of continuous arcs without skeletal structure to produce bending motion [16]. Such design initially focuses on large-scale grasping, locomotion, and positioning in industrial applications [16], or even urban search and rescue operations in confined environments [17]. The tradeoff between high flexibility and low payload induces strict structural requirements. As for small-scale continuum robots, the concerns also divert to the delicate steering of the slim robot body. The flexible characteristics of continuum robots are appropriate for medical-field applications. Enabling infinite degree of freedoms (DoFs)-manipulation within small scales, continuum robots endow flexible access to the target and less invasion to the patient [18]. Moreover, pliable medical devices such as catheters with broad-range functions, afford much inspiration to the development of robotic continuum manipulators. Besides the mechanism of a robot, sensors and controllers are also necessary to guarantee accurate control performance.

Sensing systems are an essential part of robotic platforms in the interaction with environments, based on which well-designed control strategies are capable of completing the whole manipulation loop. *Vision* is the most intuitive perception, potential to derive further quantitative information such as positional details to facilitate control. In specific applications, especially medical procedures, various categories of vision become gradually

essential. Image-guided interventions have developed as recognized standard medical procedures, applying computer-based imaging modalities to navigate surgical instruments in assorted diagnoses or treatments [19]. The imaging feedback would also provide positional information to soft robots as a feedback in the closed loop control. In addition to these virtual imaging patterns of visualization, the optic camera would be a more direct device to "see" the surroundings. Such design has been widely used in various kinds of endoscopes. Besides the medical application, camera is also a generally used sensory device in detection-related industrial applications, such as grabbing. Utilizing image-processing algorithms to provide effective feedback for robot control, and improve the sensing accuracy, play an essential role in the control scheme.

Challenges arise with the chances of continuum robots. One central aspect is that the whole manipulator will share the elastic deformation, and no discrete joints are available to be controlled or measured [16]. This is a novel problem compared to conventional rigid-link robots. The major actuation modes for continuum robots are fluid-driven and tendon-driven, where the former one could be achieved by hydraulic or pneumatic transmission. Whether prototypes or commercial robots, the kinematic/dynamic modeling and real-time operating for reliable control of robot end-effector remain an open research area [20]. With kinematic redundancy and nonlinearity, the analytical modeling is complex in continuum robots. Data-driven methods provide a breakthrough in the control of such robots. Utilizing the sensory data related to the robot end-effector state (i.e., task space), the kinematic mapping between task space and actuation space could be "learned" or optimized. The combination of kinematic model-based and model-free approaches would also boost the balance between performance stability and accuracy.

Although the potential use of continuum robots has been widely investigated, the gap between proof of concept and applications needs to be narrowed by effective control and sensing methods. Particularly in the fields demanding accuracy requirements, such as clinical catheterization, intelligent modeling approaches and precise sensing measurements can significantly enhance the safety and efficacy of continuum robots. It is timely to promote versatile control frameworks which are transferable to practical problems, as well as to provide user-friendly operation solutions for existing continuum manipulators. To this end, the main focus of this thesis is to address data-driven methods to facilitate continuum robot control and enhance the sensing modality, which would accommodate the special flexible inherence of them.

The main objectives of this thesis include:

1. To investigate efficient modeling and learning-based control algorithms for continuum robot kinematics;

2. To exploit localized online learning for delicate 2-dimensional (2D) visual servoing of hyper-elastic continuum robots without prior knowledge of camera and robot structural parameters, achieving robot adaptation to external force disturbances;

3. To apply simultaneous localization and mapping (SLAM) on eye-in-hand camera to assist the continuum robot pose estimation, closing the loop for positional control;

4. To construct evaluation platforms for the comparison of kinematic model-based and model-free control methods, aiming at lab-based pre-clinical validations for teleoperated catheterization;

5. To design and implement a continuum-robot shape-tracking system by integrating multi-core fiber Bragg grating (FBG) sensor and develop a corresponding learning-based kinematic modeling method;

6. To enhance the feature tracking accuracy during visual servoing using sparse strain measurement from a single-core FBG fiber, improving the real-time tracking performance in dim/feature-lacking environments;

7. To develop visual-strain fusion modality for robot pose estimation and 6-dimensional (6D) image stitching under extreme deterioration of image quality, as well as combine kinematic model and data-driven compensator for robust positional control.

1.2 Structure and Contributions of Thesis

Chapter 2 provides an overview of continuum robot control, sensing, and applications. The characteristics of continuum robots, as well as difficulties in their manipulation and measurement, are explained. Existing works of eye-in-hand visual servo control are discussed, which belong to one of the focuses of this thesis. The high-flexible optic fiber inscribed with FBG is also reviewed regarding its utilization in shape sensing. A general introduction of the kinematic model-based and model-less control methods for continuum robots are given. Applications of continuum robots in industrial and medical fields are summarized.

Chapter 3 focuses on the introduction of commonly-used the kinematic model-based and model-less approaches for continuum robot control. A representative kinematic modeling approach based on piecewise constant curvature (PCC) assumption is described in detail. The modeling procedure is explained in robot-independent and robot-specific levels. Machine learning algorithms that could be used to optimize or train the inverse kinematics are illustrated. The performance using PCC model-based control is also tested via simulation, guaranteeing the convergence of tracking tasks. PCC is also the model to be compared and improved via learning-based approaches in the following chapters.

Chapter 4 proposes an adaptive eye-in-hand visual servoing framework based on the localized online learning technique. The controller is constructed by directly learning the inverse mapping from collected camera images to robot actuations, without any prior knowledge of the camera and robot parameters. Promising accuracy can be assured without the need to tune any hyperparameters in the training model. Localized Gaussian process regression (LGPR) models enable rapid and effective online updates in order to accommodate the latest robot status via new input data. As a result, precise manipulation in the camera view can be achieved even when the robot encounters unknown and varying external disturbances. SLAM is also applied for the acquisition of robot pose, which will be utilized in **Chapter 6**.

Chapter 5 develops an evaluation platforms for lab-based pre-clinical validations of teleoperated catheterization, and a shape tracking method to provide both positional and morphological sensing of flexible cardiac catheters. Catheters are used as an example of continuum manipulators to validate the shape tracking and learning-based modeling method. To achieve accurate and effective feedback control, shape information obtained

from the FBGs are used for kinematics characterization as well as tip localization in real-time autonomous control. The proposed learning-based modeling method is implemented on a robotic platform to achieve feedback control of a cardiac catheter. The shape tracking performance was experimentally verified, as well as the control results in autonomous targeting and path following tasks.

Chapter 6 proposes an optic-FBG based 2D motion-sensing modality to enhance the visual servoing performance under feature-deficient and force-disturbed conditions. 6-dimensional (6D) pose estimation can also be improved using this sensing fusion modality. This work is the first to consider and solve the feature "drifting" problem in soft robot visual servoing. Besides the eye-in-hand camera, the soft continuum robot is equipped with a helically-wrapped FBG fiber which can be trained online. Learning-based method is inherently advantageous in soft robot sensing using optical fibers, circumventing modeling uncertainties due to elastic materials. A hybrid controller combined with a kinematic model and a learning-based error compensator is then applied for position control of the soft robot, employing the fusion result as feedback. Experimental validations demonstrate the improvement of fusion-based pose estimation and the efficiency of hybrid control.

The main contributions of the thesis are summarized as:

1. Development of the *first* online learning-based soft robot visual servoing architecture. Inverse kinematics of fluid-driven soft robots can be directly approximated by LGPR. Efficient update of inverse motion mapping compensates *dynamic* disturbances by adjusting the most relevant LGPR model. Novel experimental validations demonstrated precise point tracking and path following of a hyper-elastic low-stiffness soft robot with *variable* tip load.

2. Design and implementation of a shape tracking system and a learning-based kinematic modeling method. Integrated multi-core FBG fiber and tracking coils enable real-time shape reconstruction of a standard cardiac catheter. A learning-based modeling method aided by FBG-reconstructed curves is proposed and implemented on a robotic platform to accomplish feedback control of a cardiac catheter. An autonomous pulmonary vein isolation (PVI) task was conducted with *ex-vivo* tissue ablation inside a left atrial (LA) phantom model with simulated pulsatile liquid flow.

3. Development of an enhanced eye-in-hand tracking algorithm by incorporating optical FBG fiber. The prototype endoscopic soft robot is helically wrapped with FBG fiber, which improves the real-time tracking performance in dim/feature-lacking environments. The learning-based motion estimation approach alleviates the need for precise FBG placement on the soft robot. Experiments validate the enhanced motion estimation method by path tracking even under disturbance.

4. Development of an vision-FBG sensing fusion modality for 6D pose estimation and a hybrid control scheme. An FBG-related model can be trained online and fused with SLAM for robust pose estimation and 6D image stitching. Hybrid control combines model-based and data-driven methods on tracking tasks of soft robots, without the need for accurate parameter tuning or data collection. Experimental validations demonstrated of the pose estimation under extreme visual conditions and robust control performance.

5. Experimental validation of a teleoperated robotic catheterization platform to simulate RF ablation. Control mapping for hand-to-eye coordination can be constructed based on the virtual endoscopic view utilizing model-based and model-free control methods. A simulated ablation task was conducted by ten subjects for quantitative evaluation and comparison.

1.3 Publications, Patents and Awards during Ph.D. Study

Journals:

1) X. Wang, J. Dai, H.S. Tong, K. Wang, X. Xie, Y.H. Liu, K.W. Kwok, *"Learning-based Visual-Strain Fusion for Eye-in-hand Soft Robot Pose Estimation and Control,"* **IEEE Transactions on Robotics**, 2020 (under review).

2) X. Wang, G. Fang, K. Wang, X. Xie, K.H. Lee, J.D.L. Ho, W.L. Tang, J. Lam, K.W. Kwok, *"Eye-in-hand Visual Servoing Enhanced with Sparse Strain Measurement for Soft Continuum Robots,"* **IEEE Robotics and Automation Letters**, vol. 5, no. 2, pp. 2161-2168, 2020.

3) G. Fang*, X. Wang* (*co-first authorships), K. Wang, K.H. Lee, J.D.L. Ho, H.C. Fu, D.K.C. Fu, K.W. Kwok, *"Vision-based Online Learning Kinematic Control for Soft Robots using Local Gaussian Process Regression,"* **IEEE Robotics and Automation Letters**, vol. 4, no. 2, pp. 1194-1201, 2019.

4) X. Wang, K.H. Lee, D.K. Fu, Z. Dong, K. Wang, G. Fang, S.L. Lee, A.P. Lee, K.W. Kwok, *"Experimental Validation of Robot-assisted Cardiovascular Catheterization: Model-based versus Model-free Control,"* **International journal of computer assisted radiology and surgery**, vol. 13, no. 6, pp. 797–804, 2018.

5) Z. Dong, X. Wang, J.D.L. Ho, K.H. Lee, H.C. Chang, D.T.M. Chan, K.W. Kwok, *"Shape Tracking and Feedback Control of Cardiac Catheter,"* **IEEE Transactions on Robotics**, 2020 (in the 1st round of revision).

6) G. Fang, X. Wang, K. Wang, C.K. Chow, K.H Lee, J.D.L. Ho, X. Xie, W.L. Tang, K.W. Kwok, *"Visual Servoing of an MR-safe Soft Manipulator for Transoral Laser Microsurgeries,"* **IEEE Transactions on Robotics**, 2020 (in the 1st round of revision).

7) J. Dai, Z. He, G. Fang, X. Wang, Y. Li, C. L. Cheung, L. Liang, I Iordachita, H. C. Chang, K. W. Kwok, *"A Robotic Platform to Navigate MRI-guided Focused Ultrasound System,"* **IEEE Robotics and Automation Letters**, 2020 (under review).

8) X. Xie, J. Lam, C. Fan, X. Wang, K.W. Kwok, *"Energy-to-peak Output Tracking Control of Actuator Saturated Periodic Piecewise Time-varying Systems with Nonlinear Perturbations,"* **IEEE Transactions on Systems, Man and Cybernetics: Systems**, 2020 (under review).

9) J. Liu, J. Lam, B. Zhu, X. Wang, Y. Cui, Z. Shu, K.W. Kwok, *"Leader-follower Consensus of Positive Multi-agent Systems via Static Output-feedback Control,"* **IEEE Transactions on Neural Networks and Learning Systems**, 2020 (under review).

Patents:

K.W. Kwok, J.Y.K. Chan, H.C. Chang, G. Fang, X. Wang, J.D.L. Ho, C.K. Chow, *"Visual Servoing of An MR-safe Soft Manipulator for Transoral Laser Microsurgeries,"* **US Prov. Patent**: US 63/021,692. [Filed on 8 May 2020]

International awards:

Best Workshop Paper Award (Second Place) (*Sponsored by Intuitive Surgical Inc.*)
Z. Dong, X. Wang, J. D.L. Ho, Z. He, G. Fang, W. L. Tang, A. P.W. Lee, K.W. Kwok, *"Experimental validation of autonomous motion control with standard cardiac electrophysiology catheter,"* **IEEE International Conference on Robotics and Automation (ICRA) Workshop** – Open Challenges and State-of-the-Art in Control System Design and Technology Development for Surgical Robotic Systems, 2019.

Chapter 2

Control, Sensing and Applications of Continuum Robots

2.1 Introduction

Continuum robots could be featured as a continuously bending manipulator with infinite DoFs, constructed by elastic materials/structure [15]. Different from the hyper-redundant robot which comprises finite short rigid links, the whole continuum robot body is continuously and smoothly connected. To facilitate the accurate control of continuum robots and fully employ it into practical, effective models are thus expected. The special structure requires more complicated modeling approaches than in conventional rigid-link robotics, in aspect to both kinematics and dynamics. These difficulties mainly result from the actuation mechanism and deformation principle of the continuum body. Generally speaking, the actuation mechanism of continuum robots can be categorized as intrinsic and extrinsic [16], based on the location of actuator. The intrinsic [21–23] means the actuators are located inside and form as part of the mechanism. One example could be pneumatic-driven robots [24, 25], whose deformation is induced by the inflation of internal elastic chambers (**Fig. 2.1a**). Extrinsic mechanisms [1, 26, 27] use external components to distort the robot body (**Fig. 2.1b**), such as tendons/cables dragged by motors. The choice of intrinsic and extrinsic design is decided in light of factors such as the manipulator scale (especially diameter), output force range, response speed, and compatibility with the environment. Extrinsic can minimize the diameter to accommodate the deformation in confined space but increase the hysteresis and friction; the intrinsic is the opposite. These two actuation mechanisms could also be combined [3, 20, 28], such as in Air-Octor (**Fig. 2.1c**), where

a pneumatically-pressurized backbone is inflated to produce necessary support to form constant curvature structures by tendon-driven [3].

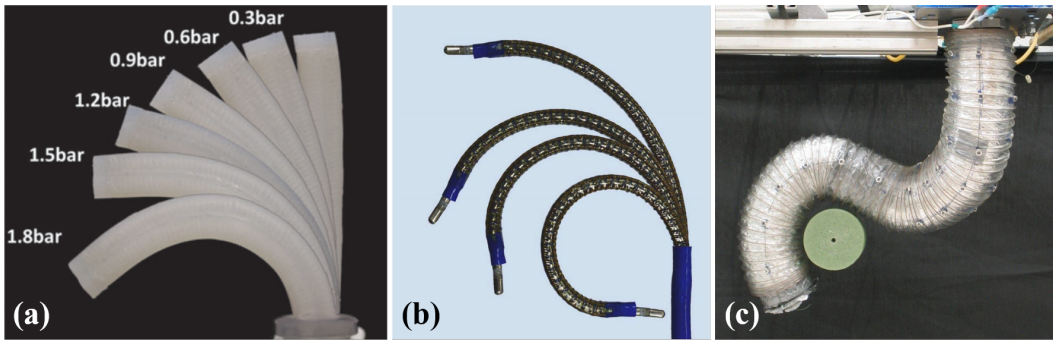


Fig. 2.1: Continuum robot examples with different actuation mechanisms. **(a)** A (intrinsic) pneumatic-driven robot enabling omni-directional bending. **(b)** A (extrinsic) tendon-driven manipulator in Hansen Medical Sensei system targeting catheterization. **(c)** Air-Octor multisection continuum robot driven by tendons and stiffened by pneumatic-pressured backbone. **Image source**: [1–3].

The applications of continuum robots have been investigated or demonstrated in plenty of industrial manufacturing fields. As bio-inspired by the octopus, underwater manipulation [29–31] has been studied. In addition to common grasping operation, several robots are designed and developed in consideration of safety and health, such as nuclear decontamination [32], nuclear reactor repair [33], or search and rescue [34, 35]. Continuum robots also have great potential in medical-related applications, especially in the view that a lot of medical devices have already been designed in the configuration of a continuum manipulator. The robotization of these devices, e.g. catheters [36–40], and endoscopes [41–44], have been developed and even commercialized. With the assistance of a camera, the robot is capable of intuitive operation. As for configuration sensing, although conventional angular encoders could not resolve the shape reconstruction of a continuum body, strain measurement is a substitute in the configuration sensing.

This chapter will first introduce the visual servoing control scheme of continuum robots, which can promote the robotization of endoscope [45] and detection-related applications. Given the consideration of specific continuum configuration, a sensing device named FBG, acting as force or shape sensors, is reviewed and summarized. Soft continuum robots cannot use rotary encoders as sensing devices, since there are no distinctive joints to measure. In contrast, the self-contained camera and FBG fiber can be employed to reflect soft robot configurations, making them the major focus of sensing devices in this thesis. Existing model-based and data-driven control methods are also investigated to clarify the research background of this thesis in the aspect of control.

2.2 Eye-in-hand Visual Servoing and Applications

Vision plays an essential role in robotic sensing, as it provides a non-invasive and intuitive measuring modality [46]. Visual servoing is a control strategy that utilizes visual feedback to close the control loop of robotic systems [47]. It is able to resist distance-related limitations in robot end-effector control, with applications extending to the manufacturing industry, military field, automobile steering, and even aircraft landing [48]. Visual servoing can be classified into eye-in-hand and eye-to-hand configurations based on the camera's location (**Fig. 2.2**) [47]. Eye-in-hand cameras are embedded at the robot end-effector and enable more flexible viewing of the workspace. Due to the limited selection of sensors that are compatible with the highly deformable structures, soft robots [2, 49] can take advantage of compact and self-contained camera feedback. Validation of visual servoing has been conducted widely in rigid-link robots, where teleoperated object/feature tracking and obstacle avoidance in imperfectly-modeled environments can be achieved [46, 48].

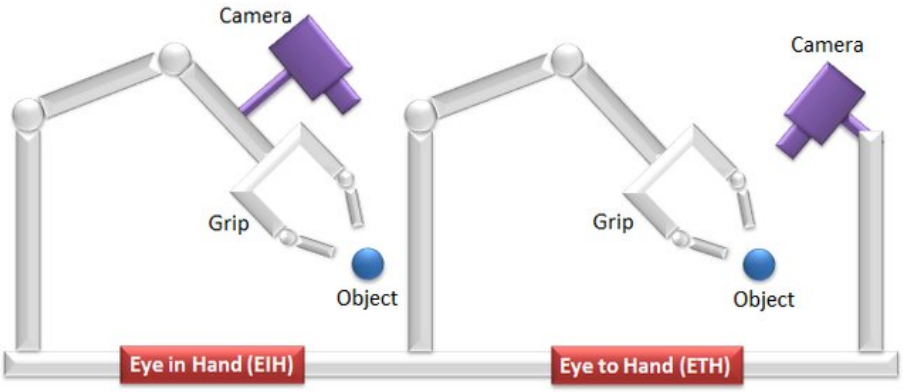


Fig. 2.2: Eye-in-hand and eye-to-hand modalities of visual servoing, sorted by the camera's position. *Eye-in-hand* indicates the camera embedded on the robot end-effector while *eye-to-hand* stands for a statically-fixed camera providing a global sight of the scene. **Image source**: [4].

With the rising prevalence of soft robotics, researchers also sought to implement visual servoing to enhance their feedback control. Vision-based systems are an appropriate selection of sensors to be integrated with soft robots, as they can be small and self-contained. Sometimes conventional sensors could not be integrated on soft continuum robots suitably since the rigid components cannot fit the varying curved shape. For example, the infinite degrees of robots prohibit the use of encoders for the acquisition of state feedback. Therefore, vision-based control methods have attracted more and more research interests in the field of soft robotics accredit to the convenience integration of camera. Making use of the camera feedback, visual servoing has been extensively studied over the last

decades, with many approaches proposed [50, 51]. The research on visual servoing could be divided into two main aspects, the image processing part for feedback procurement and the design of control method. As for the visual sensing and analyzing part, offline camera calibration would induce inevitable errors in the final control performance. Liu *et al.* [51] introduced the depth independent interaction matrix for parameter linearization of a projection matrix and updated the parameters by Slotine–Li adaptive algorithm in real-time. This adaptive algorithm was further validated in an eye-in-hand modality setup via simulation [52]. In studies of underwater continuum robots with eye-in-hand setting, the disturbances in camera view were gradually investigated. Xu *et al.* [53] considered the refraction effects in underwater environment and analogized this effect to radial distortion. In their recent work [54, 55], online correction of the image distortion was implemented to compensate for the refraction. In the above adaptive algorithms, the intrinsic and extrinsic camera parameters are calibrated in real-time; the control scheme would still depend on the modeling of robot.

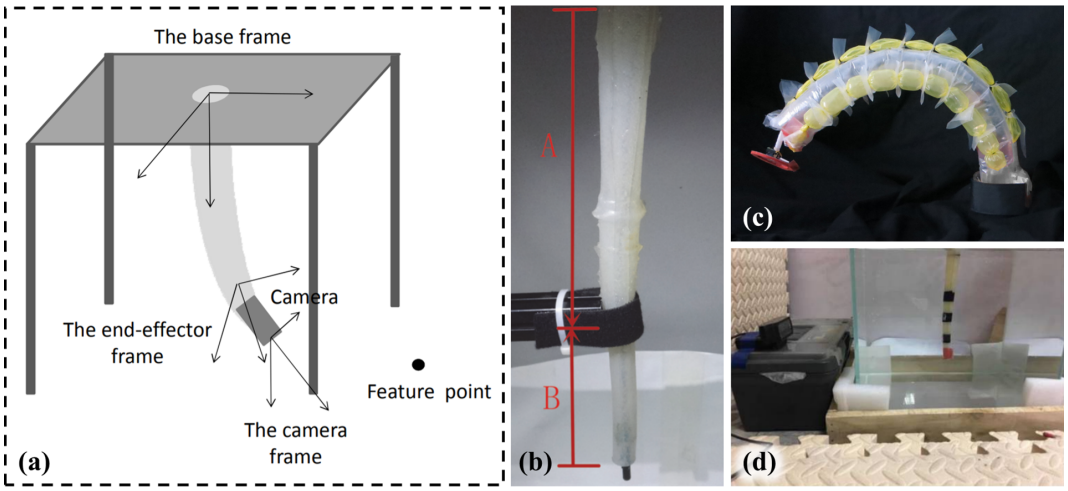


Fig. 2.3: Experimental setups of existing continuum-robot visual servo control. **(a)** A cable-driven robot commanded for targeting a black point on white background in free space and **(b)** constrained by a static physical restriction. **(c)** An sPAMs equipped with a commercial image processing system aiming at visual servoing [5]. **(d)** A cable driven robot validated in underwater scenario. **Image source**: [5–8].

As for control methods, Wang *et al.* [6] first applied eye-in-hand visual servo control on a cable-driven soft robot based on the analytical modeling of kinematics, where the intrinsic and extrinsic camera parameters needed to be estimated in advance. They also extended their work to handle static physical constraints using a model-based adaptive controller [7], taking the interaction with environment into account. However, their work focused on analytical kinematic modeling based on piecewise constant curvature (PCC) without considering visual sensing accuracy. Additionally, they only applied the visual

servoing to a simplified non-surgical scene containing distinct features. Similar scenarios were also used in studies that performed visual servoing with concentric-tube robots [56] and series pneumatic artificial muscles (sPAMs) [5]. Greer *et al.* [5] used a commercial image processing system (SightlineApplications Inc.) for feature identification. These two controllers had only been validated in free-space manipulation. None of those above researches on soft robot visual servoing have considered the effect of poor feature qualities in camera view or external disturbances. In the study [8], hydrodynamic forces in the underwater environment were considered in dynamic modeling using Kane's method.

For large-scale soft robots, visual servoing could be applied in search, rescue, and space exploration, where the restrictions on volume and weight were rigorous [57]. The most potential transition may arise in the medical field, considering the excellent compliance of the continuum robot body. Such applications include robotic minimally invasive procedures (MIP), where surgical instruments and specific tissues or organs can be tracked intra-operatively [58], reducing the burden of manual endoscope control. Inserted into the openings (e.g., mouth or anus) and channeled through exiting tracts (e.g., digestive tract), endoscopes are capable of MIP to observe internal organs or tissues in detail. Gastroscopy and colonoscopy are representative instances. Alternatively, they can also be inserted via small incisions, such as the laparoscope accessed in the abdomen. In the long and thin shape, endoscopes are in accord with the design concept of continuum robots. This inherent similarity provides ample space for the development of robotic endoscopes, which have the potential to relieve the surgeon's workload and improve manipulation efficiency [59]. The robotic prototype proposed in **Chapter 4** is designed based on this eye-in-hand setting. I intend to accomplish intuitive endoscopic control directly in the camera view. It is expected that the mono-camera is sufficient for providing visual information in control. Therefore, I propose a learning-based control method that can be operated without kinematic modeling nor visual depth estimation. Image-based feedback is sensitive to visual feature qualities, which may deteriorate due to poor lighting conditions and camera shielding, somehow. A type of optic-based strain measurement is thus applied to resolve this drawback.

2.3 Fiber Bragg Grating (FBG) in Soft Robot Sensing

Researchers have proposed several ideas of additional sensors to assist vision-based localization or motion estimation. The inertial measurement unit (IMU) is a common choice for conventional rigid robots. IMUs offer a high tracking rate without the requirement of

external landmark references [60]. Since they can only keep track of orientation at discrete nodes, the continuous morphology of the robot must be predicted by extrapolation. IMUs are embedded with magnetometers to determine the 3D orientation and compensate gyro drift. However, the accuracy of magnetometers is sensitive to ferromagnetic materials, such as surgical instruments. Special design considerations should be taken into account because an IMU would be placed at the robot end-effector and packed with the camera and LED. In interventional operations, additional water-proofing for the IMUs would also be required. Other sensors such as electromagnetic (EM) trackers are restricted to the workspace of the EM field generator and may suffer from EM interference.

Optical fiber sensor (OFS) is a kind of flexible device functioning to measure specific physical parameters, such as strain, temperature, electric current, voltage, pressure, and vibration [61]. Originally, OFS was exploited and prevalent in the electrical power industry, with advantages of distinguished immunity to electrical insulation and EM interference, no need for metallic components, lightweight and compactness, work over long distances without the requirement of local electrical power, and capability of multiplexing [61].

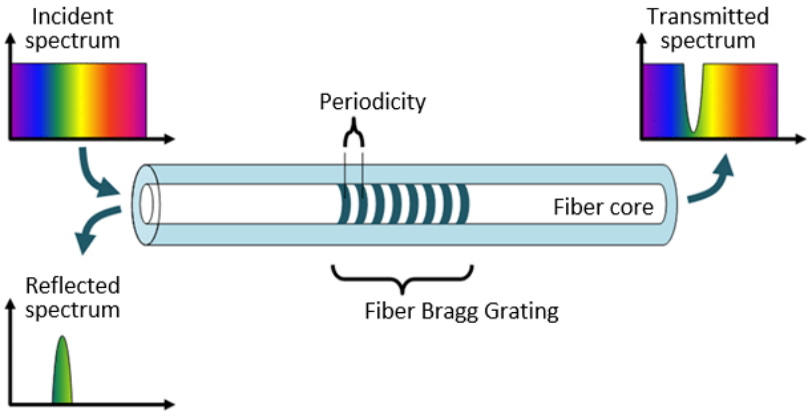


Fig. 2.4: Working principle of FBG technology. FBG is a permanent periodic variation in the refractive index value of the fiber core. It can reflect a particular wavelength of light and transmit others. The reflected wavelength is determined by the periodicity of the grating and will shift with the fiber axial elongation or temperature change. **Image source**: [9].

FBG is a representative sensor in OFSs, available for strain or temperature measurements with a relatively strong reflected signal. FBG is a permanent periodic variation in the refractive index value of the fiber corem [62, 63]. This periodic pattern (i.e., FBG) can reflect a particular wavelength of light, which is determined by the grating's periodicity. Fiber elongation or temperature change will be sensitively detected at the FBG location. This will change the periodicity and thus bring about a variation of the reflected

wavelength (**Fig. 2.4**). Optical fibers with FBGs have intrinsic advantages in strain-related state detection and are one of few sensing modalities particularly suited for soft robots. FBGs can measure strain along with optical fiber at high frequencies (>100 Hz) based on the change of reflected wavelengths. The fiber can be combined with soft elastic substrates to form a standalone sensor [64] or attached directly to soft manipulators [11]. FBGs have been investigated for shape and tip sensing of continuum manipulators, even in surgical tools like biopsy needles [65, 66]. Different from sensors like IMUs, optical fibers could reflect morphology information and are unaffected by many environmental effects such as EM fields and water submersion.

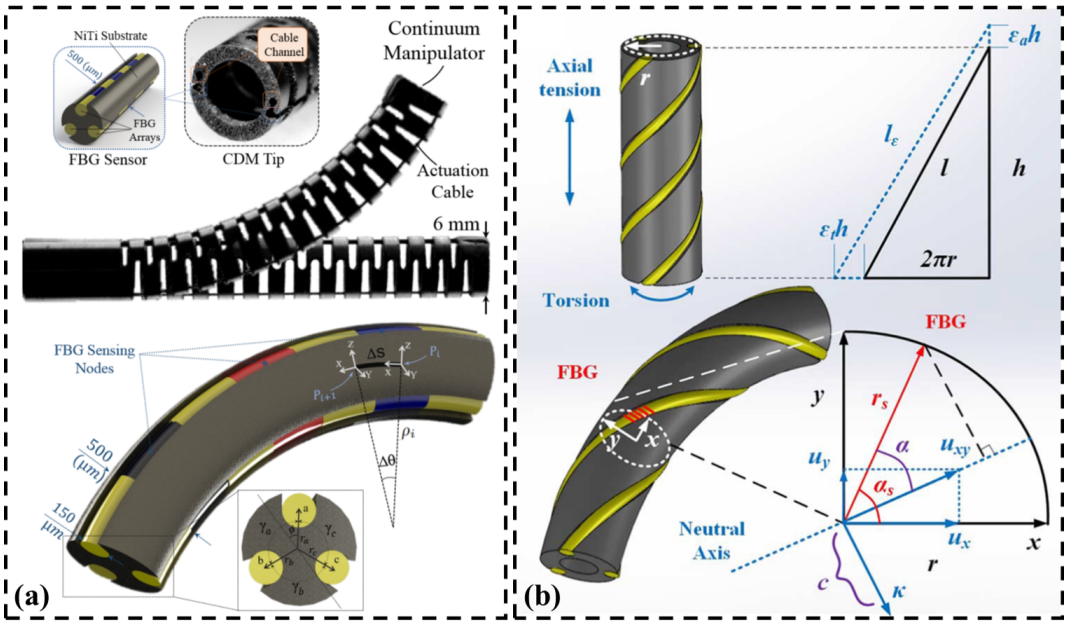


Fig. 2.5: Distributions of single-core fiber triplets in model-based shape reconstruction. **(a)** Three fibers are fixated along the continuum manipulator parallelly and evenly, where a corresponding geometry model is used for shape reconstruction. **(b)** Three fibers are helically-wrapped along the manipulator thus a model considering torsion besides curvature is established. **Image source**: [10, 11].

Both model-based and learning-based position/shape estimation approaches have been investigated in FBG-based shape reconstruction [10, 67]. Model-based algorithms using single-core FBG fibers need the coordination of at least three fibers to reconstruct the omni-directional bending [68–70]. In the study [10], the parallel distributed three fibers in triangular configuration (**Fig. 2.5a**) can reconstruct the curvature in 3D space based on accumulation, while the helically-wrapped fiber triplet (**Fig. 2.5b**) is potential to consider the torsion of manipulator [11]. Learning-based methods use additional positional sensors as the ground truth to train a mapping between the FBG wavelength and position of interested points [10, 71]. Multi-core FBG fiber contains multiple channels of FBG strings.

This compact integration is usually incorporated in the setting with diameter requirements [72, 73], such as thin catheters. Alambeigi *et al.* used an eye-to-hand camera as the external calibration device to improve the FBG sensing accuracy [74]. Similarly, as an eye-in-hand monocular camera already enables 2D motion estimation with slow robot motion and abundant features, it could be employed as the ground truth to train FBGs as a motion estimator. Therefore the estimation results obtained by the camera in scenarios with poor vision can be enhanced.

2.4 Control Algorithms for Continuum Robots

Now it can be seen that continuum robots made of elastomeric materials [46, 48] have great potential in various manipulations. This is accredited to not only their high-power density [58] actuation, but also their passive adaptability with confined space and unstructured surroundings. These advantages can resolve challenges that are commonly encountered in robotic manipulations demanding safe interaction with human, e.g., manufacturing [75], exoskeleton/wearable devices [76], and minimally invasive surgery [77]. However, their high flexibility, as well as the nonlinear deformation, usually hinder their uses in precise manipulation, compared to their rigid counterparts.

2.4.1 Kinematic/dynamic Model-based Algorithms

To conduct effective control on soft robots, various kinds of models have been investigated. Such approaches aim to approximate the kinematics/dynamics performance of the flexible manipulators without rigid-linked backbones [78]. PCC assumption is a generally accepted method to simplify the bending kinematics of continuum robots with uniform shape and symmetrical actuation [1, 15, 20]. With the feedback provided by positional sensors (e.g., EM trackers [79]), the PCC-based geometric solution can empower real-time closed-loop control of the end-effector pose in free space. There was also recent work utilizing the PCC model and a self-contained curvature sensor to control the locomotion of a robotic snake [80]. Parallel kinematics has also been examined to accomplish positional control of continuum robots with elastomer strain sensors [81]. PCC model is one of the most widely used kinematic models, accrediting to its concise parameter definition and fast computation speed. Details of the modeling procedure will be introduced in **Section 3.2**. Geometrically exact models are another popularly applied modeling approach. Those based on the Cosserat

rod theory [82–86], have been used to investigate the kinematics by establishing force equilibrium with gravity and external load accounted. However, unknown disturbance applied on the robot, such as interaction with surroundings or unpredictable payloads, would deteriorate the model promptly. Based on the defined vector field, this method allowed continuous locating of points along the Cosserat curve, but its application would suffer from high computational loads.

Several dynamic modelings for specific actuation mechanisms were proposed, such as for dielectric elastomer actuators deformed by applying voltage dropout [87,88]. With prior knowledge concerning system dynamics, model-based feedforward control was designed to handle the nonlinearity of actuator [87]. Finite element modeling (FEM) was also applied to accurately simulate complex robot deformations, by which the kinematic mapping could be generated numerically and incorporated into the soft robot control [89]. Largilliere *et al.* proposed asynchronous FEM could be combined with a quadratic programming algorithm for real-time control of soft robots [90]. Nevertheless, the modeling accuracy would be sensitive to the defined geometric and material parameters in FEM, the searches of which would also be a heuristic process. Moreover, the models mentioned above were specifically designed for particular robot structures. Actual robot kinematics is not feasible to be modeled accurately. A constant simple model will suffer from large errors in open-loop control. If effective sensing feedback can be used in the updated model, it is expected that the model can perform enhanced adaptability with unknown external disturbances. Therefore, I propose to initialize a simple kinematic model capable to be updated online (**Chapter 6**). Such a model update should be autonomous that it does not require the user to put much effort on pre-calibration.

2.4.2 Learning-based Algorithms

Data-driven control approaches can circumvent the analytical modeling procedure by *learning* the kinematic mapping or control policy from acquired sensing data. Neural networks (NNs) have been adopted to approximate the global inverse kinematics of nonredundant elastic continuum robots [91–94]. Rolf *et al.* solved the redundancy using weighting-guided goal-directed exploration in controlling an elephant trunk robot via simulation [95]. NNs could be specifically designed to learn global kinematic mapping accurately. However, it would not be efficient for online updating because all parameters contained have to be renewed in every iteration. Novel data-based approach [96] could also

be applied to estimate the kinematic Jacobian matrix online based on the optimization of an objective function. Yip *et al.* demonstrated stable control of a tendon-driven continuum robot in a 2D statically constrained environment. Recently, Lee *et al.* have also utilized a locally weighted model on 3D orientation control of a pneumatic-driven soft robot, with the model update online [2]. It could overcome externally unmodeled disturbance; however, the requirements for heuristic tuning of multiple data-dependent parameters appeared to be the main defect of this approach [97]. Ansari *et al.* proved in a simulation that the reinforcement learning method was available for the position control of a continuum robot [98]. However, simulation ignores a lot of nonlinear transmission factors and idealizes the training data. The time-consuming exploration procedure and strategy definition are still problems to be solved in reinforcement learning.

Learning-based modeling methods can accommodate to robot characteristics as the training data are specifically collected. However, this will also be the drawback, which indicates that the trained model may not be suitable for widely used. Data collection and model training will also be time-consuming before the online use. To make the most of data in real-time, I use online-update approaches to refine the trained model, therefore accomplishing the model's adaptability to disturbances from the environment. The learning-based method could also be assistance for kinematic model-based control, to reduce or even eliminate the steady errors caused by insufficiently-calibrated model, particularly in tracking tasks (**Chapter 6**).

2.5 Conclusion

This chapter presents several basic definitions in continuum robots, including the categorization and characteristics of each type of actuation mechanism. Intrinsic actuators such as pneumatic-driven robots have an advantage in output force and response speed. At the same time, extrinsic actuated manipulators could be designed in a compact scale to accommodate the deformation in confined space. Sometimes their combination is employed for balancing both of their properties. Eye-in-hand visual servoing enables to overcome distance-related limitations in robotic end-effector control, providing flexible viewing of the workspace and image analyzed positional data. Its suitability in MIP (e.g., endoscopy), and large-scale applications such as space exploration prompt the development of soft-robot visual servo controllers. FBG fiber endowing inherent advantages in morphology monitoring and immunity to numerous environmental effects implies its matched integration with continuum robots. Further combination of FBG and visual servoing is one primary subject of this thesis. Representative existing modeling approaches and data-driven methods are summarized. In the following chapters of this thesis, model-based, learning-based, or even hybrid controllers are all proposed or enhanced, and validated via abundant experiments with quantitative analysis.

Chapter 3

Kinematic Modeling and Learning Approaches for Continuum Robots

3.1 Introduction

Conventional rigid-link robots could be controlled directly by commands of motors on each joint. Once the joint angles are available as well as the link lengths, the pose of all points, including the end-effector, can be fully determined. Different from rigid-link mechanisms which have definite kinematic mapping, soft robots could not be modeled in such approaches, considering the inherent compliance due to elasticity and also nonlinear actuated deformation. In order to obtain the accurate pose information of any interested points along with the robot, including the end-effector, characterization of internal and external forces on the robot body is in need. The advantage of adaptiveness induces the difficulty in precise control of the continuum body. Accurate kinematics mapping will be required to accomplish soft robot control. Model-based methods are the most conventional approach, where analytical relations are derived to approximate the kinematics/dynamics performance. As introduced in **Section 2.4.1**, the PCC model is one of the most simplified by effective kinematic model for continuum robots. In this chapter, it will be introduced as an example to illustrate the establishment of analytical kinematics.

3.2 Piecewise Constant Curvature (PCC) Kinematics

Constant curvature (CC), due to its simplifications in shape-related modeling, has been widely regarded as an approximation to continuum robot characteristics. Elastic structures with variable/inconstant curvatures [99] can be described by integral-based functions. Under CC assumption, the robot backbone can be considered as an arc comprising finite curved links. Each of these curved links is described by a set of (usually three) structural parameters. With the combination of CC arcs, the arc geometry and end-effector position will be available. Thus, CC can distinctly facilitate the kinematical analysis and circumvent complicated computations such as differential operation.

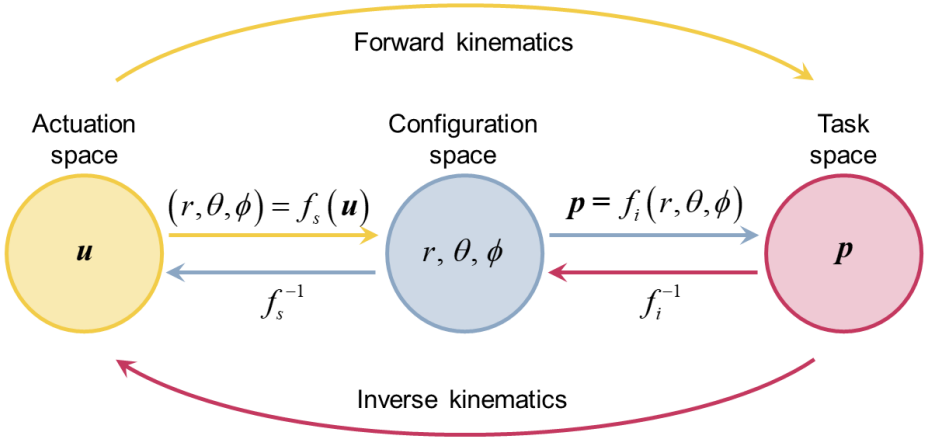


Fig. 3.1: Mappings between three spaces defined in the kinematics of constant-curvature model. Robot-specific mapping f_s, means the relation from actuation space $\boldsymbol{u}$ to configuration space r, θ and ϕ; robot-independent mapping f_i transforms the parameters from configuration space to task space, that is, the workspace of end-effector.

As the name expressed, PCC assumption regards the backbone of continuum robot as a combination of several spatial arcs with constant curvatures. It usually decomposes the kinematics into two mappings [20], namely robot-independent mapping and robot-specific mapping. The robot-independent mapping depicts the relation between the configuration and task spaces, while the robot-specific mapping represents the relation between actuation and configuration spaces (**Fig. 3.1**). Task space means the workspace of end-effector $\boldsymbol{p}$ which is the objective to be controlled. Actuation space represents the command $\boldsymbol{u}$ on motors or other actuators which could be instructed accurately. In rigid-link robots, the kinematic mapping could be definitely established between actuation and task spaces, as the configuration space would be linearly related to actuation space. However in soft robots, the configuration parameters describe the characteristics of arcs. The elasticity or fluid

dynamics would induce large nonlinearity between actuation and configuration spaces. That is also why the actuation-related part is called robot-specific mapping.

3.2.1 Robot-independent Properties

The robot-independent mapping (**Fig. 3.2**) is from the configuration space to task space, characterizing the pose of corresponding spatial curve by a triple of parameters in configuration space. These parameters contain the curvature κ (or radius r), the rotation angle ϕ of x'-O-z plane containing the arc, and the center angle (i.e. the bending direction) θ [100] as shown in **Fig. 3.2**. Alternatively, as the relation $l = \theta r = \theta/\kappa$ can be established, the parameter triple in configuration space also allows other combinations including the arc length l. This robot-independent mapping applies to any continuum robots that modeled under constant curvature assumption.

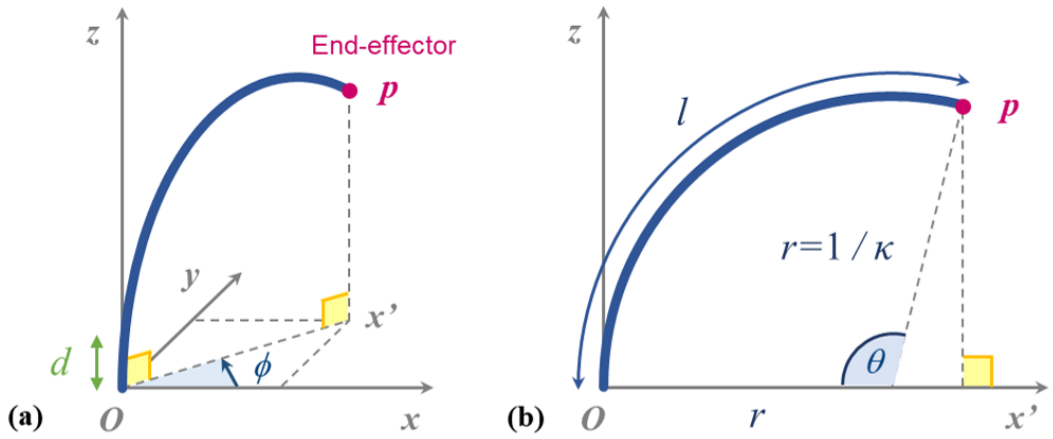


Fig. 3.2: Configuration of the robot backbone under the constant curvature assumption, where the robot base is regarded as the coordinate origin. **(a)** The angle ϕ represents the rotation/bending direction angle of arc out of x-O-z plane. **(b)** The plane that the spatial arc lies in is represented as x'-O-z. The radius, center angle and total length of this arc is defined as r, θ and l, respectively.

As in **Fig. 3.2b**, the position of robot end-effector on the x'-O-z plane could be represented as

$$\begin{cases} x' = r(\boldsymbol{u})\left(1-\cos\left(\theta(\boldsymbol{u})\right)\right) \\ z = r(\boldsymbol{u})\sin\left(\theta(\boldsymbol{u})\right) \end{cases}. \tag{3.1}$$

Taking the rotation/bending direction angle ϕ in to consideration, the 3D position of

end-effector could be obtained as

$$\boldsymbol{p}\left(r\left(\boldsymbol{u}\right),\theta\left(\boldsymbol{u}\right),\phi\left(\boldsymbol{u}\right)\right)=\begin{bmatrix} r\left(\boldsymbol{u}\right)\left(1-\cos\left(\theta\left(\boldsymbol{u}\right)\right)\right)\cos\left(\phi\left(\boldsymbol{u}\right)\right) \\ r\left(\boldsymbol{u}\right)\left(1-\cos\left(\theta\left(\boldsymbol{u}\right)\right)\right)\sin\left(\phi\left(\boldsymbol{u}\right)\right) \\ r\left(\boldsymbol{u}\right)\sin\left(\theta\left(\boldsymbol{u}\right)\right) \end{bmatrix}, \tag{3.2}$$

where $\boldsymbol{u}$ is for representing the parameters in the actuation space.

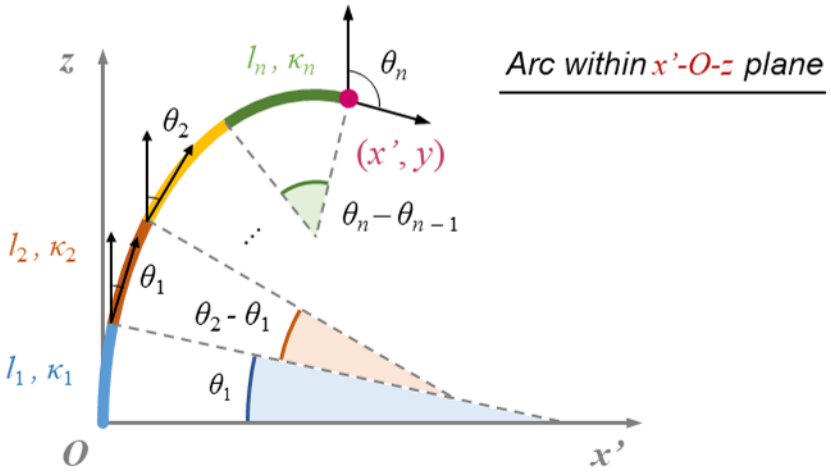


Fig. 3.3: Configuration of PCC-constructed backbone on the bending plane $\boldsymbol{x}'$-$\boldsymbol{O}$-$\boldsymbol{z}$. The robot backbone is approximated by n curve segments with different curvatures, which curvature of the i^{th} curve represented by κ_i.

The CC modeling method of arc in the $\boldsymbol{x}'$-$\boldsymbol{O}$-$\boldsymbol{z}$ plane regards the arc as one segment arc with the same curvature. This method has been widely used [15]. Another modeling method supposing the robot backbone as a combination of piecewise arcs with different curvatures (**Fig. 3.3**) could be obtained utilizing integral. This idea was first proposed in [101] for the use of catheter force estimation. Here I utilize and modify it for accurate shape modeling of continuum robots. The robot backbone is modeled as a combination of n piecewise circular arcs, with curvatures represented by κ_i, $i=1,2,\cdots,n$. Regarding the base of robot as the origin, define the arc length from origin to a point on the bendable section as s, and the angle between axis z and tangent at that point as $\theta\left(s\right)$ (**Fig. 3.3**). The arc length of each segment is l_i, $i=1,2,\cdots,n$. According to [101], point position along the bendable section can be described by

$$\begin{cases} x_p'\left(s\right)=\int_0^s \sin\theta\left(\zeta\right)d\zeta \\ z_p\left(s\right)=\int_0^s \cos\theta\left(\zeta\right)d\zeta \end{cases}, \tag{3.3}$$

where the angle $\theta_i\left(s\right)$ on the i^{th} segment is

$$\theta_i\left(s\right)=\kappa_i s+\sum_{j=1}^{i-1} l_j\left(\kappa_j-\kappa_i\right),\ \kappa_0=0. \tag{3.4}$$

The position (x',z) of arc endpoint on the bending plane without rotation can be calculated based on **Eq. 3.3** as

$$\begin{cases} x' = \sum_{i=1}^{n} \int_{a_i}^{b_i} \sin\theta_i(\zeta)\, d\zeta \\ z = \sum_{i=1}^{n} \int_{a_i}^{b_i} \cos\theta_i(\zeta)\, d\zeta, \end{cases} \tag{3.5}$$

where the lower and upper limits of integration are respectively

$$a_i = \sum_{j=0}^{i-1} l_j \quad b_i = \sum_{j=0}^{i} l_j, \quad i = 1,2,\ldots,n. \tag{3.6}$$

Substituting **Eq. 3.4** into **Eq. 3.5** and defining $l_0 = 0$, the final expression of position (x',z) can be obtained as

$$\begin{cases} x' = \sum_{j=1}^{n-1} \left[\frac{1}{\kappa_{j+1}} \cos\left(\sum_{p=1}^{j} \kappa_p l_p \right) - \frac{1}{\kappa_j} \cos\left(\sum_{p=1}^{j} \kappa_p l_p \right) \right] + \frac{1}{\kappa_1} - \frac{1}{\kappa_n} \cos\left(\sum_{i=1}^{n} \kappa_i l_i \right) \\ z = \sum_{j=1}^{n-1} \left[\frac{1}{\kappa_j} \sin\left(\sum_{p=1}^{j} \kappa_p l_p \right) - \frac{1}{\kappa_{j+1}} \sin\left(\sum_{p=1}^{j} \kappa_p l_p \right) \right] + \frac{1}{\kappa_n} \sin\left(\sum_{i=1}^{n} \kappa_i l_i \right) \end{cases}. \tag{3.7}$$

Utilizing **Eq. 3.7**, the 3D position $\boldsymbol{p}$ would be

$$\boldsymbol{p} = \begin{bmatrix} x'\cos\phi \\ x' \cdot \sin\phi \\ z \end{bmatrix}. \tag{3.8}$$

In **Eq. 3.7**, the acquisition of configuration space parameters would be different, details about how to utilize this modeling algorithm will be introduced in **Section 5.6.1** of **Chapter 5**.

Soft robot body could be constructed in different dimensions, which will be analyzed in detail in the following **Section 3.2.2**. Representative actuation dimension number inside the robot body is 1 to 4. For the robots consisting at least 3 actuation channels (tendons or fluidic chambers) inside, they will enable an omni-direction bending itself when the robot base is fixed. For those with 1 or 2 actuation channels, an additional rotation DoF would be required to act as the parameter ϕ if targeting at 3D task-space control. This is because that the robot itself could only achieve planar bending with existing channel(s). For some robots that require larger workspace comparing to the robot size, especially the space along z axis,

an extra insertion DoF is sometimes designed to adjust the position of whole robot body.

3.2.2 Robot-specific Properties

Representative examples of actuation variable forms include lengths of tendons, flexible push rods or fluid chambers. Tendon-driven and pneumatic-driven robots (as introduced in **Section 2.1**) are the most common design in existing researches. The modeling difference between such two mechanisms is mainly about the transmission approach from the actuation unit. In aspect to how the elongation/contraction of internal channels would act on the robot configuration change, the principle is the same. In this section, I will discuss the robot-specific mappings of manipulators with two-dimension and three-dimension actuation channels, which will be applied investigated in the following chapters of this thesis.

3.2.2.1 Two-dimension Actuation

For continuum manipulator with two actuation channels inside, additional rotation and insertion DoFs are usually supplemented to enable sufficient 3D workspace manipulation (**Section 3.2.1**). I would take the tendon-driven robot as an example to generally indicate the relation between tendon length and configuration space parameters. In the tendon-driven mechanism as **Fig. 3.4a** shows, the two tendons are parallel distributed on a plane through the central axis and capable of manipulator bending on this plane. These two tendons could be pulled/pushed by a single or separate actuation unit(s).

Referring to **Eq. 3.2**, the mapping from actuation space parameters $\boldsymbol{u}$ to r, θ and ϕ is needed to construct the complete forward kinematics. The axial translation d of robot end-effector values approximately equal to the insertion distance of robot base, so does the rotation angle ϕ. With the constant curvature assumption applied in such a two-tendon driven mechanism, the bending motion of robot is related to the tendons' axial difference ΔL,

$$\Delta R \cdot \theta = \Delta L. \tag{3.9}$$

Slimilarly in the PCC geometry (**Fig. 3.4b**), although the curve is separated by several segments, the relation is also satisfied, since

$$\Delta R \cdot [\theta_1 + (\theta_2 - \theta_1) + \cdots + (\theta_n - \theta_{n-1})] = \Delta L. \tag{3.10}$$

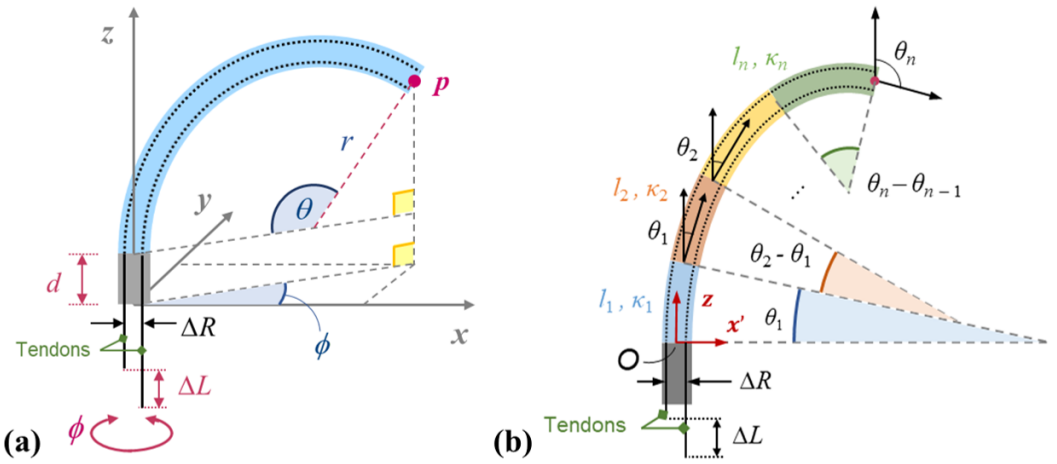


Fig. 3.4: **(a)** Schematic diagrams of a typical tendon-driven continuum manipulator in CC configuration. CC geometry is characterized by configuration space parameters as in **Fig. 3.2**. Insertion distance and rotation angle of the manipulator body are represented by d and ϕ. The deformation is determined by the relative position of two tendons, marked by ΔL. The constant distance between two tendons is ΔR. **(b)** PCC (**Fig. 3.3**) geometry of a tendon-driven continuum manipulator on the bending plane, with the tendon-related variables illuminated.

Thus the three parameters r, θ and ϕ could all be obtained by linear correlation with the length of continuum body, rotation angle and tendons state.

3.2.2.2 Three-dimension Actuation

In this section for analyzing the three-dimension actuation mechanism, I would take the pneumatic-driven robot as an example, of which the PCC-based kinematic model has been well-established [15]. In order to obtain the accurate mapping from actuation space to configuration space, this relation is usually evolved by measuring the moments or forces applied by the actuator and conveyed to the robot manipulator. It has been known that in CC model, the three parameters in the configuration space are r, θ and ϕ, where ϕ means the bending direction (**Fig. 3.5a**) of arc instead of an additional rotation.

Suppose the robot comprises three fluidic chambers (denoted by $i = 1,2,3$), lengths of which could be represented as $\boldsymbol{u} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^{\mathrm{T}}$. **Fig. 3.5b** shows the base of the bendable segment viewed down along the z-axis. The arc radius r corresponding to the robot backbone relates to the radius of each chamber according to

$$r_i = r - d\cos\phi_i, \tag{3.11}$$

where d is the distance between the centers of robot body and each chamber, and ϕ_i

particularizes the angle between the robot bending direction and the corresponding chamber as

$$\phi_1 = 90^\circ - \phi,\ \phi_2 = 210^\circ - \phi,\ \phi_3 = 330^\circ - \phi,$$
$$\sum_{i=1}^{3} \cos\phi_i = 0. \tag{3.12}$$

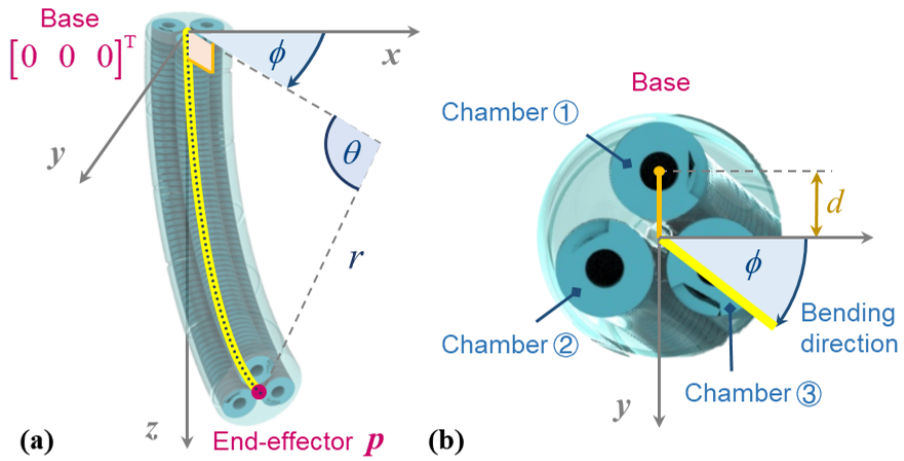


Fig. 3.5: CC model of continuum robots. **(a)** Configuration-space parameters that are defined to describe the spatial arc. **(b)** A top view of the robot base.

The arc lengths of the robot and each chamber could be calculated by

$$l = \theta r,\ l_i = \theta_i r_i. \tag{3.13}$$

The relation between these arc lengths can be found as

$$l = l_i + \theta d\cos\phi_i. \tag{3.14}$$

Based on **Eq. 3.13**, the relationship **Eq. 3.14** yields that

$$l = \frac{l_1 + l_2 + l_3}{3}. \tag{3.15}$$

From **Eq. 3.14**, the angle ϕ representing the bending direction could be obtained as

$$\phi = \tan^{-1}\left(\frac{\sqrt{3}\,(l_2 + l_3 - 2l_1)}{3\,(l_2 - l_3)}\right). \tag{3.16}$$

The radius of each chamber arc with constant curvature r_i could be obtained using the relation $\theta = \kappa l = l/r = l_i/r_i$, therefore $r_i = l_i/\kappa l$. Substituting this into **Eq. 3.13**, the curvature κ can be calculated as

$$\kappa = \frac{l - l_i}{ld\cos\phi_i}. \tag{3.17}$$

Using the relation in **Eq. 3.12** when $i = 1$ and **Eq. 3.13**, the above expression of κ yields

$$\kappa = \frac{l_2 + l_3 - 2l_1}{(l_1 + l_2 + l_3)\, d \sin\phi}, \tag{3.18}$$

where $\delta = \left(l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_1 l_3 - l_2 l_3\right)^{1/2}$. Therefore, the expression of three parameters r, θ and ϕ in the configuration space can be represented as

$$\begin{cases} r(\boldsymbol{u}) = d(l_1 + l_2 + l_3)/2\delta \\ \theta(\boldsymbol{u}) = 2\delta/3d \\ \phi(\boldsymbol{u}) = \tan^{-1}\left(\sqrt{3}(l_2 + l_3 - 2l_1)/3(l_2 - l_3)\right) \end{cases}. \tag{3.19}$$

Substituting these above expressions into **Eq. 3.2**, the position of end-effector $\boldsymbol{p}$ in the coordinate as shown in **Fig. 3.5a** can be thus obtained. This result is also applicable for tendon-driven robots, where the effective lengths of three tendons are equal to $\boldsymbol{u}$.

3.2.3 Inverse Kinematics for Model-based Control

The final expression of robot end-effector $\boldsymbol{p}$ represented by $\boldsymbol{u}$ is the forward kinematics of controlled robot. The corresponding differential function could be represented as

$$\dot{\boldsymbol{p}} = \boldsymbol{J}\dot{\boldsymbol{u}}, \tag{3.20}$$

where the Jacobian matrix $\boldsymbol{J}$ could be calculated by differentiating the position $\boldsymbol{p}$ with respect to the input variable $\boldsymbol{u}$ in the following form.

$$\boldsymbol{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial l_1} & \frac{\partial x}{\partial l_2} & \frac{\partial x}{\partial l_3} \\ \frac{\partial y}{\partial l_1} & \frac{\partial y}{\partial l_2} & \frac{\partial y}{\partial l_3} \\ \frac{\partial}{\partial l_1} & \frac{\partial z}{\partial l_2} & \frac{\partial z}{\partial l_3} \end{bmatrix}. \tag{3.21}$$

With the Jacobian matrix $\boldsymbol{J}$ of above forward kinematics, we could establish its inverse kinematic function

$$\dot{\boldsymbol{u}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{p}}, \tag{3.22}$$

where $\boldsymbol{J}^{\dagger}$ is the pseudoinverse of $\boldsymbol{J}$. When the Jacobian matrix $\boldsymbol{J}$ is a square and invertible matrix, the pseudoinverse could also written as its original inverse $\boldsymbol{J}^{-1}$. Up to know, the model utilizing CC/PCC assumption has been given. These results would be compared,

utilized or improved in the following chapters.

3.3 Simulation Results of PCC Model

3.3.1 Two-channel Tendon-driven Mechanism

The validity of model can be tested via simulation before its use in practical. **Fig. 3.6** shows the performance of a tendon-driven continuum robot (**Section 3.2.2.1**), in the task of tracking a moving target along a square shape. The robot length was set as 65 mm, with its initial backbone position defined along the z axis and base on the coordinate origin. The desired trajectory was defined in the $\boldsymbol{x}$-$\boldsymbol{O}$-$\boldsymbol{z}$ plane, with a size of 10×10 mm^2. The robot tip (end-effector) could follow the target accurately, with the mean tracking error of 0.0443 mm. This result demonstrates that the CC model and its analytical solution of Jacobian matrix are accurate. Although in the control of a real robot, the factors such as robot material, friction, hysteresis would disturb the final performance, this simulation result could provide a baseline for further parameter tuning.

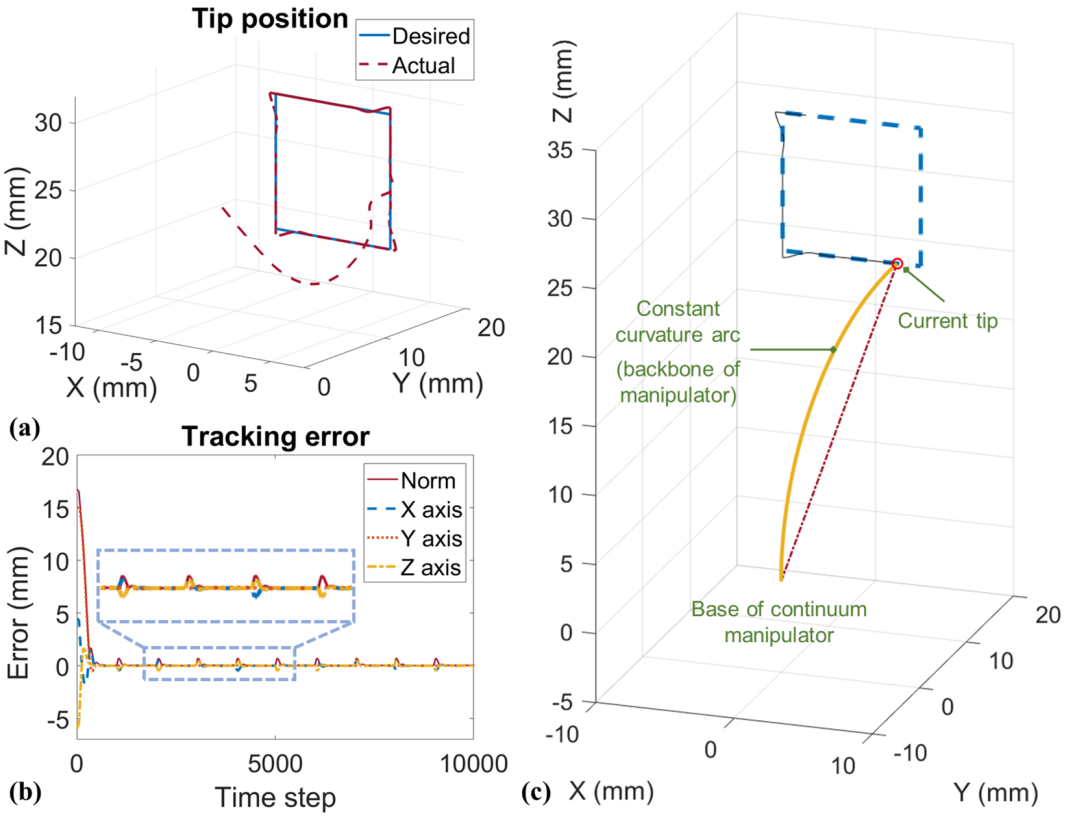


Fig. 3.6: Simulation performance of a two-channel tendon-driven continuum robot, in the task of tracking a moving point along a square path. **(a)** The desired trajectory (10×10 mm^2) is defined on the x-O-z plane, while the robot initial backbone (65-mm length) is along the z axis. **(b)** Tracking errors on different directions. **(c)** A simulated sketch of the robot backbone during the tracking task.

3.3.2 Three-chamber Pneumatic-driven Mechanism

Similarly, for pneumatic-driven robots, the CC model also performs well in the simulation. As the robot would deform due to the inflation of chambers, the chamber lengths will always no less than its initial length. In this simulation, it was assumed that the increment of length and its corresponding actuation unit (e.g. motor) has a linear relation. In practice, however, this relation could never satisfy this assumption. The performance of different robots varies a lot, even in the same structural design. This is also the reason why learning-based control has large potential in the soft manipulators.

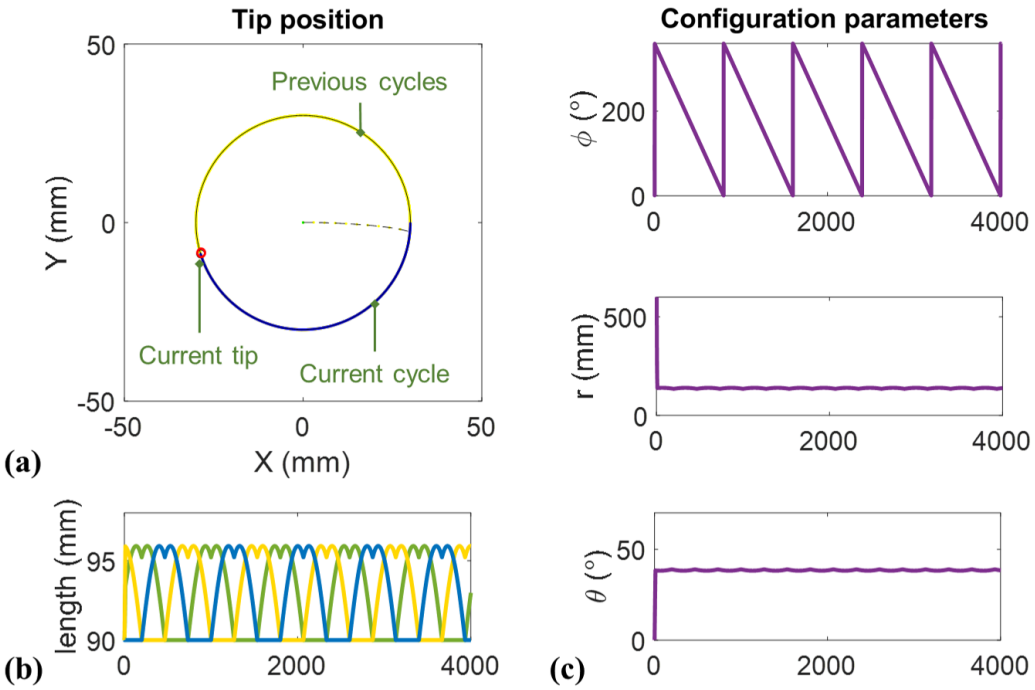


Fig. 3.7: Simulation performance of a three-chamber pneumatic-driven continuum robot, in the task of tracking a moving point along a circle. **(a)** The desired trajectory (30×30 mm^2) was defined on the $\boldsymbol{x}$-$\boldsymbol{O}$-$\boldsymbol{y}$ plane, while the robot initial backbone (90-mm length) was along the z axis. **(b)** Lengths of three pneumatic chambers enabling the end-effector to follow the circle path. **(c)** Parameters $\boldsymbol{u}$ in the configuration space during the tracking.

3.4 Data-driven Control Algorithms

Modeling accuracy is sensitive to robot structural and material-related parameters. The tuning for these parameters would be a time-consuming heuristic process. Moreover, several models are designed explicitly to particular robot mechanisms, thus lack the potential of generalization. In this section, a simplified model-less control method utilizing optimization, which is inspired by the method in [96], and a conventional regression method [102] are introduced. These methods are capable of training the robot inverse kinematic mapping. The utilization of other NN-based learning based methods would be expanded in the chapters of their corresponding applications.

3.4.1 Convex Optimization-Based Method

This optimization-based method aims for initializing and updating the inverse kinematics **Eq. 3.22** using the sensory feedback, without the need for analytical solution of Jacobian matrix. This inverse kinematics can be discretized to

$$\Delta \boldsymbol{u} = \boldsymbol{J}^{-1} \Delta \boldsymbol{p}, \tag{3.23}$$

The core of obtaining $\Delta \boldsymbol{u}$ in **Eq. 3.23** for accurate motion is to find a proper Jacobian matrix $\boldsymbol{J}$. For the robot, the 3 degrees of freedom (DoFs) of the actuator are in the same unit and independent from each other. The initialization of $\boldsymbol{J}$ could be achieved by actuating the 3 DoFs in turn with an incremental amount $\Delta u_i, i = 1,2,3$ (i.e. the actuation commands are $\begin{bmatrix} \Delta u_1 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & \Delta u_2 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 0 & \Delta u_3 \end{bmatrix}$ successively), and measuring the corresponding displacements $\Delta \boldsymbol{z}_{pi}$. The initial Jacobian matrix could be constructed as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{J}_2 & \boldsymbol{J}_3 \end{bmatrix}, \tag{3.24}$$

where $\boldsymbol{J}_i = \Delta \boldsymbol{p}_i / \Delta u_i$. The obtained Jacobian matrix could also be updated online, relying on continuously solving a quadratic programming problem as follows

$$\begin{aligned} \text{minimize} \quad & \|\Delta \boldsymbol{J}(k)\| \\ \text{subject to} \quad & \Delta \boldsymbol{p}(k) = \boldsymbol{J}(k+1)\, \Delta \boldsymbol{u}(k) \\ & \boldsymbol{J}(k+1) = \boldsymbol{J}(k) + \Delta \boldsymbol{J}(k), \end{aligned} \tag{3.25}$$

where $\Delta \boldsymbol{J}(k)$ is the variable to be optimized, $\boldsymbol{J}(k)$ is the Jacobian matrix at time k, $\boldsymbol{J}(k+1)$

is the estimation for Jacobian at time $(k+1)$. After obtaining the latest Jacobian, the command to actuators could be calculated by

$$\Delta \boldsymbol{u}(k+1) = \boldsymbol{J}(k+1)^{-1} \Delta \boldsymbol{p}^*(k+1), \tag{3.26}$$

where $\Delta \boldsymbol{p}^*(k+1)$ is the desired motion in task space.

3.4.2 Gaussian Process Regression (GPR)

Training: The inverse mapping in the form of **Eq. 3.22** will be learned. To facilitate the data collection, the mapping should be rewritten since the Jacobian matrix would not be available. The input data should include the robot state and desired motion of end-effector. As introduced in **Section 3.2.2.2**, the 3 DoF actuation corresponding to three pneumatic chambers could reflect the configuration of robot, which could be recorded and utilized in the training of robot kinematic model. Generally, the mapping to be established could be written as

$$\Delta \boldsymbol{u}(k) = f(\boldsymbol{u}(k), \Delta \boldsymbol{p}^*(k)), \tag{3.27}$$

where the input and output of this mapping are $\begin{bmatrix} \boldsymbol{u}^{\mathrm{T}} & \Delta \boldsymbol{p}^{*\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ and $\boldsymbol{y} = \Delta \boldsymbol{u} \in \mathbb{R}^m$, respectively. The first step to establish this mapping is to collect a training dataset by actuating the real robot for model initialization. Consider there is a training set including input data $\boldsymbol{X} = \{\boldsymbol{x}_i\}$ and output data $\boldsymbol{Y} = \{\boldsymbol{y}_i\}$, $i = 1, 2, \ldots, N$. Although the variable dimension for output could be more than 1, each dimension $\boldsymbol{y}^s = \{y_i^s\}$, $s = 1, 2, ..., m$ will be independently trained. GPR model allows a nonlinear mapping $y_i^s = G(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}$ between input and output satisfied, where $\boldsymbol{\varepsilon}$ denotes a white Gaussian noise with zero mean value and a variance of σ_n^2. The output variable will be modeled as a Gaussian distribution $\boldsymbol{y}^s \sim N\left(0, \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}\right)$, where $\boldsymbol{I}$ represents the identity matrix and $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ means a covariance matrix. In the proposed application of kinematics learning, the zero-mean precondition will be adopted, as the change of actuation $\Delta \boldsymbol{u}$ should have a mean value with 0. The i^{th}-row, and j^{th}-column element $k_{ij} = k\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right)$ in covariance matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ can be a customized function, which the squared-exponential kernel function [102] is used in this scheme:

$$k_{ij} = k\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \sigma_s^2 \exp\left(-0.5\left(\boldsymbol{x_i} - \boldsymbol{x_j}\right)^{\mathrm{T}} \boldsymbol{\Lambda}\left(\boldsymbol{x_i} - \boldsymbol{x_j}\right)\right), \tag{3.28}$$

where σ_s^2 is the signal variance, and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ represents a diagonal matrix with characteristic length-scales $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^{\mathrm{T}}$ acting on each dimension of the input $\boldsymbol{X}$ correspondingly. There are several hyperparameters σ_s, σ_n and $\boldsymbol{\lambda}$ to be determined in this regression procedure, which can be optimized by maximizing the negative log marginal likelihood. For example, the hyperparameters can be found by standard optimization methods such as conjugate gradient. This will be an automatically seeking procedure without any requirement of manual heuristic intervention. After these above steps, GPR is capable to generate a global nonlinear mapping model used for prediction.

Prediction: Given a input set $\widehat{\boldsymbol{x}}$ for query, the joint distribution of the observed target values $\boldsymbol{y}^s$ and predicted value $g(\widehat{\boldsymbol{x}})$ will be expressed as [102]:

$$\begin{bmatrix} \boldsymbol{y}^s \\ g(\widehat{\boldsymbol{x}}) \end{bmatrix} \sim N\left(0, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2 \boldsymbol{I} & \boldsymbol{k}(\boldsymbol{X},\widehat{\boldsymbol{x}}) \\ \boldsymbol{k}(\widehat{\boldsymbol{x}},\boldsymbol{X}) & k(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{x}})) \end{bmatrix}\right). \tag{3.29}$$

The predicted mean $\bar{g}(\widehat{\boldsymbol{x}})$ and covariance $V(\widehat{\boldsymbol{x}})$ can be obtained by conditioning the above joint distribution:

$$\bar{g}(\widehat{\boldsymbol{x}}) = \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{X},\widehat{\boldsymbol{x}})\left(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{y}^s = \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{X},\widehat{\boldsymbol{x}})\boldsymbol{\beta}, \tag{3.30}$$

$$V(\widehat{\boldsymbol{x}}) = k(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{x}})) - \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{X},\widehat{\boldsymbol{x}})\left(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{k}(\boldsymbol{X},\widehat{\boldsymbol{x}}), \tag{3.31}$$

where $\boldsymbol{\beta}$ denotes the prediction vector.

This Gaussian model could also be updated online, as well as divided into several local models for the acceleration of computation speed. These method would be elaborated in the **Chapter 4** of visual servoing control. The mapping training using procedure NN is similar to GPR, the difference is only on the structure of network. In **Section 6.2.2**, NN is applied on the training of FBG sensor. Extreme learning machine (ELM) is also utilized in the fast training and online update of learning-based pose estimation of **Section 6.4.2**.

3.4.3 Conclusion

In this chapter, the CC and an integration-based PCC kinematics are elaborated, with the robot-independent and robot-specific parts. Based on such complete kinematics, a model-based controller could be implemented on the interested continuum robot for positional control. However, actual control performance would not be as ideal as in the simulation, due to the nonlinear actuation transmission and individual differences. To solve this problem, learning-based controller or error compensator is needed for fine performance adjustment. A lot of models for the learning procedure have been investigated in continuum robot control (summarized in **Chapter 2**). As examples, NN and GPR approaches are introduced, which will be introduced in the following chapters.

Chapter 4

Eye-in-hand 2D Visual Servoing and 6D Pose Estimation

4.1 Introduction

As introduced in **Chapter 2**, soft robots, composed of elastomeric materials, can ensure safe and mild interaction with their surroundings. These compliance characteristics unavoidably entail a trade-off against precise control of robot motion, as to which commonly-used model-based methods were offered to reflect the robot kinematics. However, too many parameters, concerning robot nonlinear deformation and external disturbance, may require time-consuming tuning and make the model complicated to be quantified. Choices of sensors integrated on the robot are expected to resolve such internal or external modeling uncertainties. The camera (eye) mounted on the robot end-effector (hand) is a typical sensor setting. Therefore, an eye-in-hand visual servo control framework is propose for continuum robots, and incorporated with a data-driven controller to perform precise robotic manipulation. The 2D image feedback can be analyzed to provide 2D relative motions in the camera view, along with 6D absolute pose of the camera itself. For the 2D incremental movement, it would be difficult to accurately and instantly model the correlation between 2D motion and robot actuation. The respective reasons are the nonlinearity induced by fluid-driven mechanism and the lack of real-time robot configuration feedback. In such circumstances, learning-based algorithms and online updates can be employed to construct the model. LGPR is used to initialize and refine the inverse kinematic mapping online. No prior knowledge will be needed in aspects of both robot- and camera-related parameters. Experimental validation was also conducted to demonstrate the performance of

a hyper-elastic robot, which could compensate for an external variable loading during path following. Utilizing the integrated camera, 6D motion/position detection algorithms could also be conducted. SLAM is one kind of multi-thread algorithm containing tracking, local mapping, and loop closing that can provide 6D pose estimation for robot sensing.

In this chapter, an online learning eye-in-hand visual serving framework is proposed. The controller is constructed by learning the localized inverse kinematic mapping directly from image-processing information, without the need for camera and robot parameters. The tuning of hyperparameters in the training model can be accomplished automatically. LGPR models enable rapid and effective online updates in order to accommodate the latest robot status via new input data. As a result, precise manipulation in the camera view can be achieved even when the robot confronts untrained external disturbances. I also implement monocular camera SLAM for the end-effector pose estimation for further robot control. The major contributions of the learning-based controller are:

1. *First* attempt to address a learning-based visual servo controller on a hyper-elastic soft robot such that the inverse kinematics can be initialized directly by LGPR;

2. Efficient update of the trained inverse kinematics mapping to allow rapid compensation of *dynamic* disturbance by refreshing the most relevant local GPR model;

3. Novel experimental validations that demonstrated precise point-to-point tracking and path following of a low-stiffness continuum robot with *varying* tip loading.

4.2 Online Learning for 2D Visual Servoing Control

The main focus of this chapter will be the 2D visual servoing control of continuum robots (even for those fabricated by hyper-elastic materials), thus the definition of task space (i.e., image frame) and control objective are introduced first. I utilize localized GPR models for the training and fast update of the inverse mapping in 2D visual servoing. This section will construct the whole control architecture of this project.

4.2.1 Task Space Definition

A camera mounted at robot end-effector allows image-based control strategy, namely eye-in-hand visual servo. Mappings from the spaces of actuation, configuration to task spaces have to be defined successively. The actuator input (at equilibrium) is represented as $\boldsymbol{u}(k) \in \mathbb{U}^m$ at time step k, where $\mathbb{U}^m$ denotes the m-dimensional actuation space. Let $\boldsymbol{s}(k)$ be the manipulator configuration space parameters under input $\boldsymbol{u}(k)$, which corresponds to an end-effector position $\boldsymbol{p}(k) \in \mathbb{R}^3$ and orientation normal $\boldsymbol{n}(k) \in \mathbb{R}^3$ in the Cartesian space. The collective variable $\boldsymbol{\theta}(k) = \begin{bmatrix} \boldsymbol{p}(k) & \boldsymbol{n}(k) \end{bmatrix} \in \mathbb{R}^6$ depends on robot configuration $\boldsymbol{s}(k)$:

$$\boldsymbol{\theta}(k) = h(\boldsymbol{s}(k)). \tag{4.1}$$

With quasi-static movement, the forward transition model can be expressed as:

$$\Delta \boldsymbol{s}(k) = f(\boldsymbol{s}(k), \Delta \boldsymbol{u}(k)) \ , \tag{4.2}$$

where $\Delta \boldsymbol{u}(k) = \boldsymbol{u}(k+1) - \boldsymbol{u}(k)$ is the difference of inputs between time step k and $k+1$, and $\Delta \boldsymbol{s}(k) = \boldsymbol{s}(k+1) - \boldsymbol{s}(k)$ represents the change of robot configuration due to the input difference $\Delta \boldsymbol{u}(k)$.

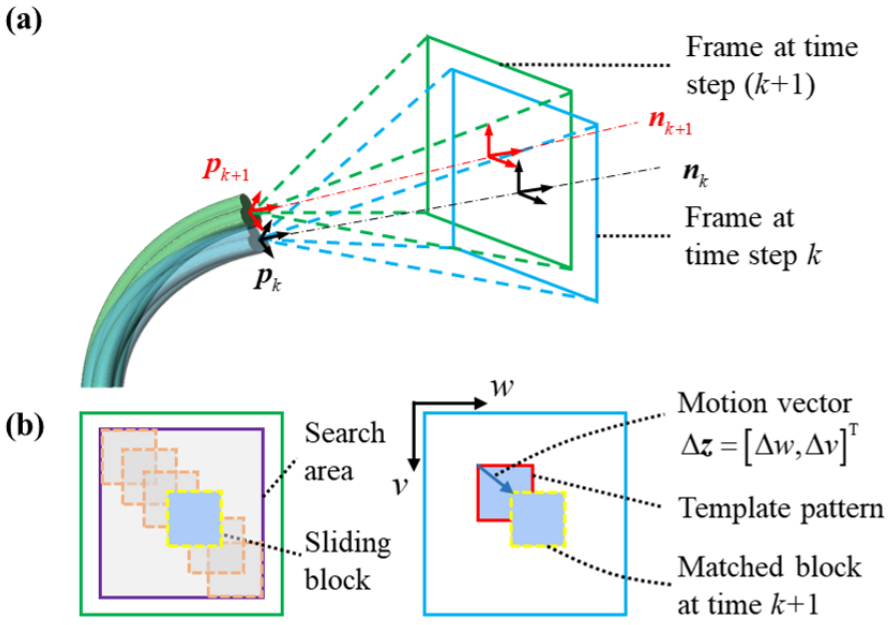


Fig. 4.1: Schematic diagram of motion estimation. **(a)** Camera coorindate frames at time step k and $k+1$; **(b)** Incremental motion in image plane can be acquired based on the displacement from the template pattern (in red block) to the matched block (in yellow) that is searched by block sliding.

The task space is defined in the camera frame (**Fig. 4.1a**), with the incremental

displacement denoted as $\Delta \boldsymbol{z}(k) \in \mathbb{R}^2$. The frame is always perpendicular to the robot tip normal. Combining with the mapping from end-effector states $\boldsymbol{\theta}(\cdot)$ to camera frames, **Eq. 4.1** can thereby be extended to:

$$\Delta \boldsymbol{z}(k) = g(\boldsymbol{s}(k), \Delta \boldsymbol{u}(k)), \tag{4.3}$$

The control objective is to generate the actuation command, achieving a desired movement $\boldsymbol{z}^*(k)$ in task space, mapping **Eq. 4.4** is necessary to approximate the inverse kinematics of **Eq. 4.3**, i.e.,

$$\Delta \boldsymbol{u}(k) = \widehat{\Phi}(\boldsymbol{s}(k), \Delta \boldsymbol{z}^*(k)). \tag{4.4}$$

The inverse transition $\widehat{\Phi}$ heavily depends on the current robot configuration that supposes to be an unknown without any sensing data of the end-effector pose. However, this can be resolved by a new inverse kinematic function of the actuator input $\boldsymbol{u}(k)$ that is defined based on a direct mapping from $\boldsymbol{s}(k)$ to $\boldsymbol{u}(k)$ during quasi-static movements. Hence, such an inverse mapping is presented as:

$$\Delta \boldsymbol{u}(k) = \Phi(\boldsymbol{u}(k), \Delta \boldsymbol{z}^*(k)), \tag{4.5}$$

which approximates the true inverse transition $\widehat{\Phi}$ as in **Eq. 4.4**. I proposed to employ image feedback $\Delta \boldsymbol{z}(\cdot)$, actuation input $\boldsymbol{u}(\cdot)$, $\Delta \boldsymbol{u}(\cdot)$ to directly estimate the robot inverse kinematics Φ, but without having to construct an analytical (kinematic) model.

4.2.2 Motion Estimation on Image Plane

To "learn" the inverse kinematic mapping using experimental data, an effective algorithm to measure the end-effector motion $\Delta \boldsymbol{z}$ with respect to camera frames, i.e. motion on the image plane, is in demand. To estimate the 2D incremental movement between two successive frames, a target square of image intensity features is defined as reference for comparison (**Fig. 4.1b**). Assume that the orientation change of the robot is small within a short time interval ($\sim$20 Hz). The change of camera orientation is primarily attributed to rotation about the camera normal, which was found to be $< 5^\circ$ between successive frames. This small range of rotation corresponds to a movement error of < 3 pixels when calculated from template matching. This orientation effect was evaluated by calculating the movement error of 100 images generated by rotating an original image from -10° to 10° (**Fig. 4.2a**). As the

displacement was calculated between successive frames at tiny time intervals, the twisting angle reflected in the camera view is $< 5^\circ$ referring to the recorded twisting angles during pre-training data exploration (**Fig. 4.2b**). Therefore, the movement error caused by the camera rotation would be < 3 pixels, which is acceptable relative to the 10-pixel tolerance. Besides, the orientation-induced error is generally correlated to the robot configuration, which could be accurately resolved by instant update of the actuation status in the learning model.

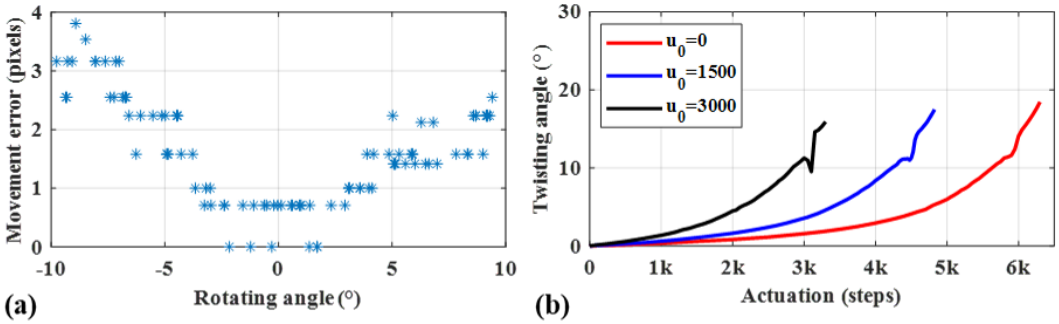


Fig. 4.2: **(a)** Motion estimation errors sampled at angular resolution of 0.25°. **(b)** Twisting angle of robot end-effector (camera) under varying actuation inputs when one chamber is inflated. All three chambers are pre-pressured with actuation input $\boldsymbol{u}_0$.

Therefore, in the cases of continuous robot movements, the camera frames would follow a more-or-less planar motion. The translational displacement in this image plane can be estimated using block template matching method (**Fig. 4.1b**), named "*matchTemplate*" in OpenCV [103]. A square block at time step k is first selected as the target template. At time $(k+1)$, the same size of block is sliding along the image plane to search for a block containing the intensity pattern that is coherent to the template. The coherence value is calculated referring to the metric function "*TM_CCORR_NORMED*" as:

$$R(\xi,\eta) = \frac{\sum_{i,j}\left(T(i,j)\cdot I(i+\xi,j+\eta)\right)}{\sqrt{\sum_{i,j}T(i,j)^2\cdot\sum_{i,j}I(i+\xi,j+\eta)^2}}, \tag{4.6}$$

where T and I represent the intensities of template and the sliding block, i and j are the indices of pixels in the local blocks, ξ and η are the movements of sliding block along w and v axes of camera view respectively. The motion vector $\Delta\boldsymbol{z}(k) = [\Delta w, \Delta v]^{\mathrm{T}}$ between two successive frames is therefore obtained using "*minMaxLoc*" function, by maximizing the coherence in **Eq. 4.6**:

$$[\Delta w, \Delta v] = \arg\max_{\xi,\eta} R(\xi,\eta). \tag{4.7}$$

We assume that axial movement of the camera has minimal effect on the template matching process, relative to lateral movement. This is because axial motion results in mainly

scaling of the tracked objects rather than translation. In addition, template matching is performed between successive frames, which means those differences in scale will be small while having a reasonable rate ($\geq$ 20 Hz) of camera imaging. Similarly, for optical aberration-caused geometrical distortions, its effect on relative motion estimation could be ignored, as the compared templates extracted from successive frames are supposed to be with little changes of distortion. Note that the absolute distortion of the template is not considered in this algorithm. During continuous motion, the template matching is iteratively updated referring to the previous frame, providing an accurate displacement estimation successively.

4.2.3 LGPR-based Online-learning Control

Learning the inverse mapping from the camera motion to actuator input is a regression problem. GPR is a nonparametric method by-design to approximate nonlinearity [102]. In this case, it would resolve the parametric uncertainties induced by soft robot fabrication and camera calibration, and even the noise of camera feedback. After the initialization using GPR, the inverse mapping model is updated online with the newly collected sensing data during the execution of the robot. Additionally, with a locally weighted learning scheme, the computational efficiency for both predicting and updating can be increased.

4.2.3.1 Localized GPR Model

For the establishment of inverse kinematics, GPR will be applied in two steps, separately related to training and learning. As given in **Section 4.2.1**, the inverse mapping $\Delta \boldsymbol{u} = \Phi(\boldsymbol{u}, \Delta \boldsymbol{z}^*)$ will be learned, where the input is defined as $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}^{\mathrm{T}} & \Delta \boldsymbol{z}^{*\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ and output as $\boldsymbol{y} = \Delta \boldsymbol{u} \in \mathbb{R}^m$. A training dataset is collected from the real robot for model initialization. Consider the training set with input data $\boldsymbol{X} = \{\boldsymbol{x}_i\}$ and output data $\boldsymbol{Y} = \{\boldsymbol{y}_i\}$, $i = 1, 2, \ldots, N$, where each dimension of the output $\boldsymbol{y}^s = \{y_i^s\}, s = 1, 2, ..., m$ is independently trained. Details of the training and prediction principle have been explained in **Section 3.4.2**. The model could be trained and applied by substituting the above input and output data.

The most time-consuming operation in GPR is the inversion of matrix $\left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right)$ with complexity of $O\left(N^3\right)$. To improve the computational efficiency for robot control, reducing the dimension of input matrix is an effective choice. Therefore, I partition the training data distributed in the whole workspace into M clusters,

$D_j, j = 1,2,\ldots,M$, using the k-means clustering algorithm, where the Euclidean distance is replaced by Gaussian kernel-based similarity measure as in **Eq. 3.30**. Each observation is assigned to the cluster that the similarity between the observation and cluster center reaches maximum. The actuator input performs as the clustering basis, since it could reflect the robot state. The center of j^{th} cluster could be represented as $\boldsymbol{c_j} \in \mathbb{U}^m$. Every cluster of training data containing N_j samples generates a local model Φ_j. Moreover, a maximum model size could be predefined as $N_j^{\max}$ for each cluster, thus simplifying the calculation. Each local model Φ_j will generate a relevant prediction $\widehat{\boldsymbol{y}}_j$ by **Eq. 3.30** at each step. The actuator input for the next step is determined upon the weighted average of M LGPR model predictions [104]:

$$\Delta \boldsymbol{u}(k) = \sum_{j=1}^{M} \omega_j \widehat{\boldsymbol{y}}_j \Big/ \sum_{j=1}^{M} \omega_j, j = 1,2,\ldots,M, \tag{4.8}$$

where ω_j quantifies the similarity between current actuator input $\boldsymbol{u}(k)$ and the j^{th} cluster center. Here, a Gaussian kernel is employed to measure this similarity. The procedures of clustering training data and initialization are summarized in **Algorithm. 1**.

Algorithm 1 Initialization of LGPR Models with Clustering

Inputs:

$\boldsymbol{X}$: input of the model;
$\boldsymbol{Y}$: observation;
$\omega(\cdot)$: similarity kernel function;

1: Partition the inputs samples into clusters using k-means clustering.
2: **for** each cluster $j = 1,2\ldots,M$ **do**
3: Train j^{th} Local GPR model **Eq. 3.28**;
4: **end for**

4.2.3.2 Incremental Learning

Online update of the inverse model enables the controller to adapt with various changes of robot interactions and mechanical property, e.g. loading on the end-effector. Once the new actuation $\Delta \boldsymbol{u}(k)$ is executed at each step, the corresponding actual motion vector $\Delta \boldsymbol{z}(k)$ could be obtained by the image processing unit. Thereby, a set of new sample data with input $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}(k)^{\mathrm{T}} & \Delta \boldsymbol{z}(k)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ and output $\boldsymbol{y} = \Delta \boldsymbol{u}(k)$ will be produced. This online sample could represent the latest working environment, and be added into the nearest cluster D_r, i.e., the one with maximal value of ω_j, for updating the corresponding local model Φ_r. The dataset update, including vector $\boldsymbol{Y}$ and input matrix $\boldsymbol{X}$, is straightforward. If the current model size $N_r \leq N_r^{\max}$, the samples in cluster $D_r = \{[\boldsymbol{x}_i, \boldsymbol{y}_i]\}, i = 1, 2, \ldots, N_r$ is retrained for a new inverse model Φ_r^{new}; otherwise $N_r > N_r^{\max}$, the oldest sample $[\boldsymbol{x}_1, \boldsymbol{y}_1]$ will be discarded and the cluster $D_r = \{[\boldsymbol{x}_i, \boldsymbol{y}_i]\}, i = 2, 3, \ldots, N_r$ is retrained. Under the size limitation, the point prediction can be kept fast and effective.

To update prediction vector $\boldsymbol{\beta}$ (**Section 3.4.2**), the Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ of matrix $\left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right)$ presented in [104] could be adjusted directly. Then the prediction vector can be solved from $\boldsymbol{y} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{\beta}$. A new point can be considered by adding a new row to the bottom of matrix $\boldsymbol{L}$:

$$\boldsymbol{L}_{new} = \begin{bmatrix} \boldsymbol{L} & \boldsymbol{0} \\ 1^{\mathrm{T}} & l_* \end{bmatrix}, \tag{4.9}$$

$$\boldsymbol{L}\mathrm{l} = \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_{new}), \quad l_* = \sqrt{k(\boldsymbol{x}_{new}, \boldsymbol{x}_{new}) - \|\mathrm{l}\|^2}. \tag{4.10}$$

Deleting the oldest data is achieved in two steps: Firstly, exchange the oldest data in $\boldsymbol{L}$ to the last row by multiplying permutation matrix

$$\boldsymbol{R} = \boldsymbol{I} - (\boldsymbol{\delta}_1 - \boldsymbol{\delta}_{Nr})(\boldsymbol{\delta}_1 - \boldsymbol{\delta}_{Nr})^{\mathrm{T}}, \tag{4.11}$$

i.e. $\boldsymbol{R}\boldsymbol{L}$, where $\boldsymbol{\delta}_i$ is a zero vector whose ith element is one; then $\boldsymbol{L}_{new}$ can be obtained by removing the last row of matrix $\boldsymbol{R}\boldsymbol{L}$. The incremental learning procedures are summarized in **Algorithm. 2**. The control block diagram in **Fig. 4.3** shows the key processing components including the aforementioned LGPR model prediction and motion estimation.

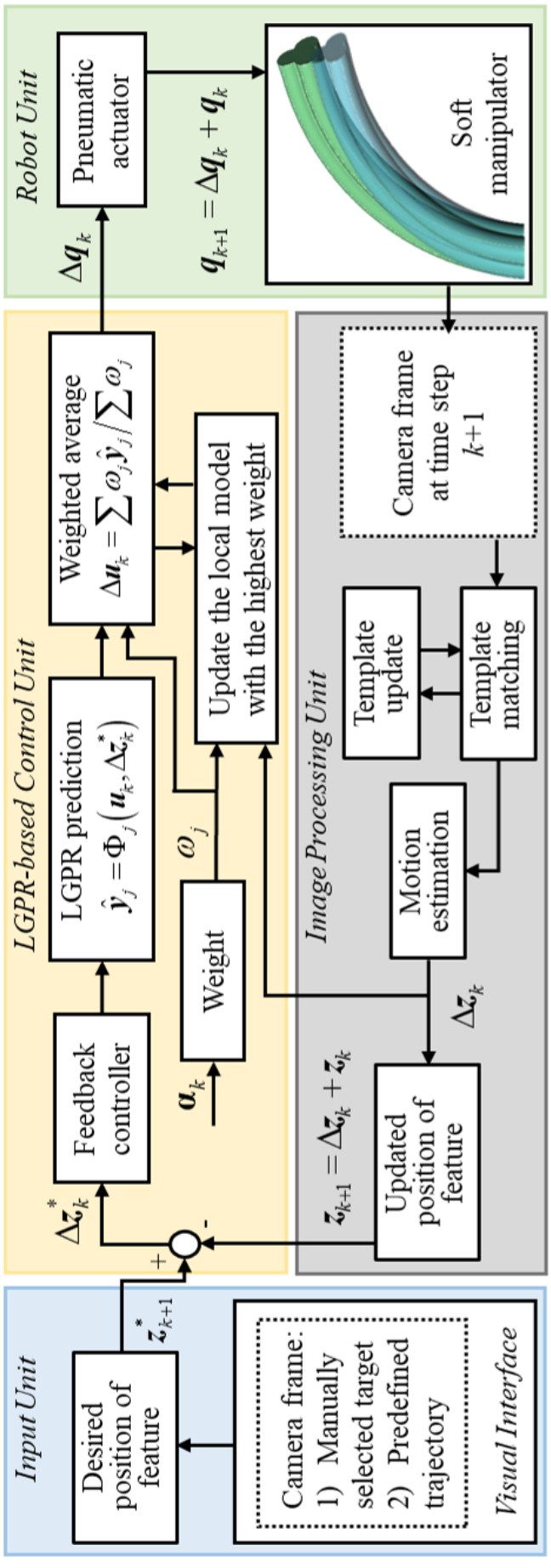


Fig. 4.3: Proposed learning-based control architecture. Parameters $\boldsymbol{u}$ and $\boldsymbol{z}$ denote, respectively, the actuation command and the position of captured image features. The input unit provides the positional command in image domain, where the target position $\boldsymbol{z}^*_{k+1}$ can be selected manually or predefined by a reference trajectory. The LGPR-based control unit generates the actuation command $\Delta\boldsymbol{u}_k$, referring to the desired displacement $\Delta\boldsymbol{z}^*_k$ and current state $\boldsymbol{u}_k$. The image processing unit esitmates the real-time displacement $\Delta\boldsymbol{z}_k$ for the online update of LGPR models and the feedback control.

Algorithm 2 Online Update of LGPR Models

for each new data point $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ **do**
  for each local GPR model $\Phi_j, j = 1, 2 \ldots, M$ **do**
    Compute similarity between the model center and input $\omega_j = \omega(\boldsymbol{x}_i, \boldsymbol{c}_j)$;
  end for
  Choose the closest model: $r = \arg\max_j \omega_j$;
  if $\omega_r >$ similarity threshold **then**
    if $N_r > N_r^{\max}$ **then**
      Delete the oldest point in model Φ_r;
    end if
    Insert $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ to the local model data set D_r: $\boldsymbol{X}_r = \boldsymbol{X}_r \cup \boldsymbol{x}_i$, $\boldsymbol{Y}_r = \boldsymbol{Y}_r \cup \boldsymbol{y}_i$;
    Update model center: $\boldsymbol{c}_r = \text{mean}(\boldsymbol{X}_r)$;
    Update the Cholesky matrix and the prediction vector of local model; **Eq. 4.9**;
  else
    Create a new local model $\boldsymbol{c}_{M+1} = \boldsymbol{x}_i, \boldsymbol{X}_{M+1} = \boldsymbol{x}_i, \boldsymbol{Y}_{M+1} = \boldsymbol{y}_i, M = M + 1$;
    Initialize new Cholesky matrix and new prediction vector;
  end if
end for

4.3 Experiments of 2D Visual Servoing

In this section, point-to-point tracking and path following tasks were conducted on a hyper-elastic pneumatic-driven continuum robot. External forces and varying tip loading were also applied as to validate the efficiency of the proposed online locally-updating method. The fast tracking convergence rate under external disturbances indicated the feasibility of using this method in other forms of continuum robot control.

4.3.1 Experimental Setup

As illustrated in **Fig. 4.4**, a soft manipulator was fixed downward, viewing the workspace scene built from LEGO®. A hyper-elastic soft manipulator in small size ($\varnothing 13 \times 67$ mm) was fabricated from room-temperature-vulcanization (RTV) silicone (Ecoflex 0050; Smooth-On, Inc.), which is a relatively low-stiffness rubber [2]. Such a "floppy" robot comprises three cylindrical air chambers that could be inflated individually. A layer of helical Kevlar string with 1-mm pitch was wrapped around each chamber to restrict its radial expansion, giving rise to pure elongation/shortening of chambers upon inflation/deflation. This provided the effective bending motion with a maximum bending angle larger than 90°. Cooperation of the three chamber pressures allowed the omni-directional servo of the camera (Depth of view: 8-150 mm) and LED illumination module mounted at the robot tip.

This LED module is for adjusting the brightness in the camera view when the operating environment is dark, since this lighting will move along with the robot end-effector and function the same role in each frame. All LEDs on the module should be in the same white color and brightness thus avoiding additional disturbances to original features. With a 90° diagonal field of view, the camera captures images of 400×400 pixels, indicating that a pixel translates to 0.16° field of view. The inflation volume of each chamber was controlled precisely with a pneumatic cylinder actuated by a stepper motor.

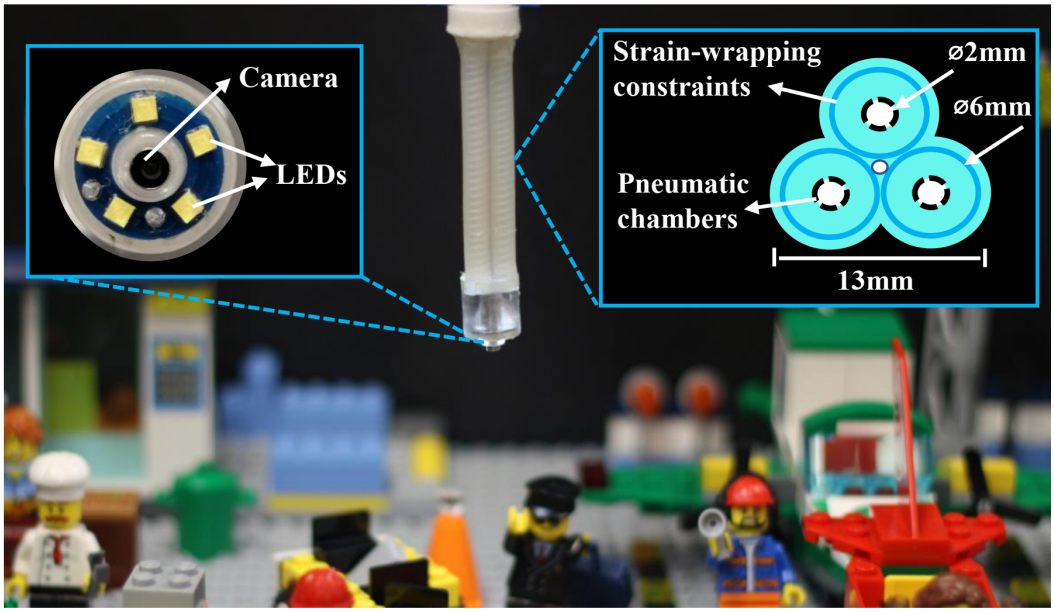


Fig. 4.4: Experimental setup in a scene of LEGO®. The soft manipulator was made of silicone rubber, and which was driven by three fiber-constrained air chamber. A endoscopic camera and five LEDs were mounted at the tip.

The major challenge that hurdles precise control of the soft robot would be attributed to its nonlinear kinematic behavior. The twisting of the continuum structure would also cause rotation of the camera view in unexpected directions.

4.3.2 Pre-training of Local GPR Inverse Model

Before operation, the LGPR models were first initialized using the data $\boldsymbol{D}' = \{\boldsymbol{u}_i, \Delta\boldsymbol{z}_i, \Delta\boldsymbol{u}_i\}|_{i=1}^{N}$ collected within a calibration environment, in which the robot base was fixed. An EM tracking coil (NDI Aurora®) was attached on the tip of the soft robot to record the 3D position. A uniformly distributed actuation input set $\boldsymbol{u}_i$ was used for exploration of the robot configuration space in this study. Camera image can be captured at every new actuation input after the equilibrium is reached.

The incremental change of $\Delta \boldsymbol{u}_i = \boldsymbol{u}_j - \boldsymbol{u}_i$ can be obtained from differencing of two robot configurations $\boldsymbol{u}_j$ and $\boldsymbol{u}_i$. To obtain $\Delta \boldsymbol{z}_i$, a visual feature appeared at the image center $\boldsymbol{z}_c$ corresponding to $\boldsymbol{u}_i$ was first detected, $\Delta \boldsymbol{z}_i = \boldsymbol{z}_{new} - \boldsymbol{z}_c$ was then obtained by detecting the new feature position $\boldsymbol{z}_{new}$ in the image corresponding to $\boldsymbol{z}_j$.

It is noteworthy that the collected data $\boldsymbol{D}'$ is generally reflecting the nonconvexity of the inverse mapping. This is because the actuation space ($\Delta \boldsymbol{u}_i$) has a higher dimension than the task space ($\Delta \boldsymbol{z}_i$). There may exist more than one $\Delta \boldsymbol{u}_i$ that corresponds to a data pair $(\boldsymbol{u}_i, \Delta \boldsymbol{z}_i)$, resulting in multi-valued mapping of $\Delta \boldsymbol{u}_i = \Phi(\boldsymbol{u}_i, \Delta \boldsymbol{z}_i^*)$. Learning such mapping directly from nonconvex set $\boldsymbol{D}'$ is an ill-posed problem. In this section, I constructed a nonredundant set of training data from $\boldsymbol{D}'$ by filtering out data pairs whose elements in $\boldsymbol{u}_i = \begin{bmatrix} u_{1i} & u_{2i} & u_{3i} \end{bmatrix}^{\mathrm{T}}$ are all nonzero:

$$\boldsymbol{D} = \{\boldsymbol{u}_i, \Delta \boldsymbol{z}_i, \Delta \boldsymbol{u}_i | u_{1i} \cdot u_{2i} \cdot u_{3i} = 0, \forall i = 1, \ldots, N\}. \tag{4.12}$$

The constraints in **Eq. 4.12** can ensure that at least one chamber among the three has zero pressure, thus every actuation command in **Eq. 4.12** will correspond to a unique tip position in the workspace. It indicates the actuation space is non-redundant, resulting in a single-valued mapping. In practice, due to approximation error, the controller initialized from the convex set D would output that violates the convexity. The constraint **Eq. 4.12** has also to be applied in order to maintain the convexity during the incremental update of the controller.

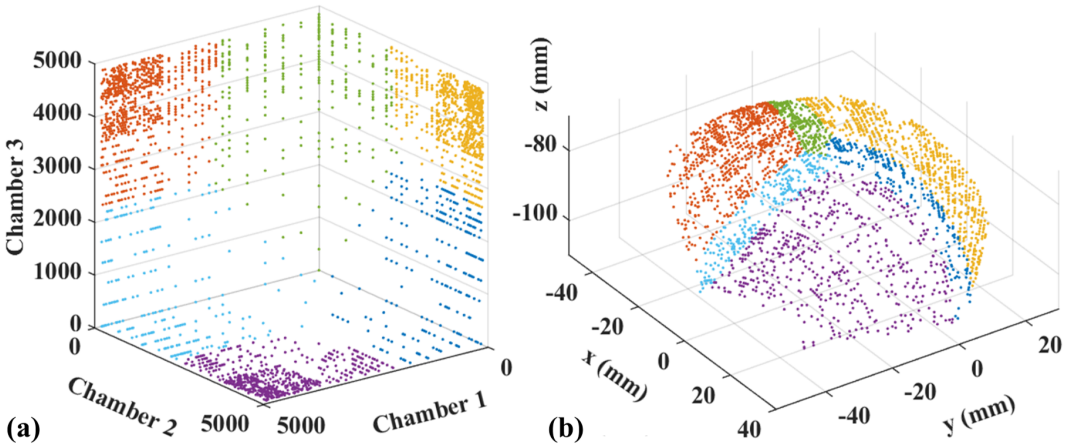


Fig. 4.5: Three thousand sample pairs of robot actuation and tip position collected for pre-training of the inverse model. **(a)** Using the k-means algorithm, all these training points were divided into six (colored) clusters based on their actuation inputs. **(b)** Corresponding tip position distribution of six clusters is also shown.

K-means clustering based on a Gaussian kernel was performed to partition the training data set $\boldsymbol{D}$ satisfying **Eq. 4.12** into 6 clusters, as shown in **Fig. 4.5**. A data set of 1000 samples are randomly selected for pre-training of LGPR models. The data size of 6 clusters are $[202, 257, 82, 294, 106, 59]$ respectively, with the upper limit of data size set to 300 for each local model.

4.3.3 Experiments and Results

Four manipulation tasks were conducted to evaluate the performance of proposed controller under various conditions, such as target tracking under external interactions and path following with changing load. The tracking error was defined as the shortest distance between the current position of target and the desired trajectory.

4.3.3.1 Point-to-point Tracking

In this task, the robot was commanded to aim the camera center at a series of target way-points in the image view (**Fig. 4.6**). Five target points (labels 1 to 5) were manually selected in series from the 400 image plane. Around the target points, a 100 template pattern centered was created and denoted by a red box in image frames. **Fig. 4.6a** shows the panorama image obtained by mosaicking the camera frames during the tracking task. The image center at each step was marked by a green circle. **Fig. 4.6b** presents the robot configurations when aiming at each target way-point. I set a 10-pixel tolerance for the accuracy of target matching. **Fig. 4.6c** provides the tracking error in unit of pixel, showing that the proposed controller can achieve precise targeting at each target way-point with an error less than 10 pixels. It demonstrates that the inverse mapping approximated by the local GPR model can accurately compensate the disorientation between input space and task space.

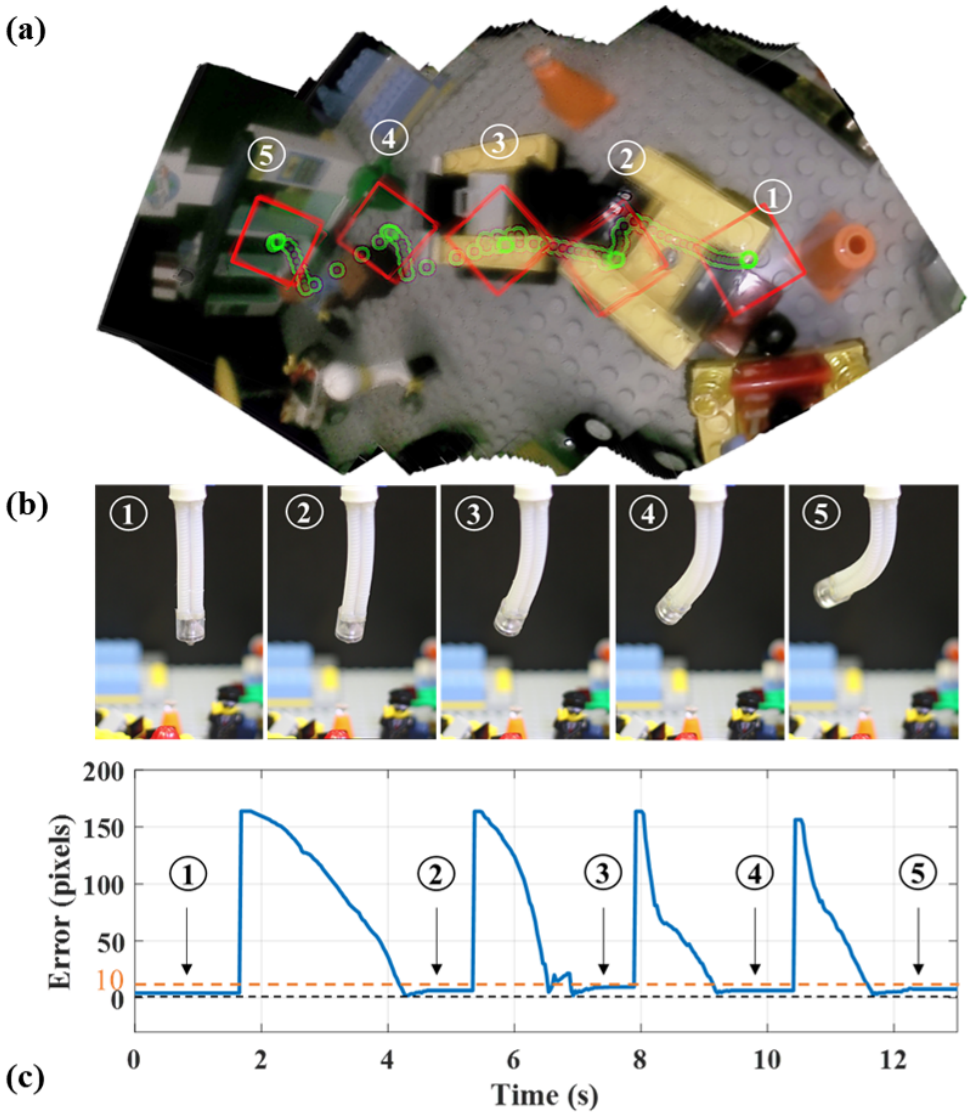


Fig. 4.6: Five targets manually selected on the image plane for robot tracking. **(a)** Mosaic image obtained during multiple target tracking. A template pattern (in red block) was form and centered at the target point selected. A series of green bubbles represent the instantaneous camera centers, showing the trajectory travelled along the matched template patterns. **(b)** Corresponding robot configurations while tracing targets successively at its image plane center. **(c)** Absolute tracking errors throughout the journey.

4.3.3.2 Target Tracking under External Forces

This section presents a tracking experiment with an aim to evaluate the feedback control performance in response to external disturbance. The robot was commanded to align its image center to a fixed target, while an unknown force pushed it away from its original configuration. The controller only utilized image as feedback. The robot configurations and target tracking errors are depicted in **Fig. 4.7**. It shows that the robot could fixate the desired target within an error < 20 pixels even under an unknown external force. The force was applied continuously by a finger within a certain time window (12.7∼31.0 sec). Feedback control achieved this enhanced accuracy by reducing the effective compliance of the robot, which may raise concerns about loss of adaptiveness to the environment. However, the force output of the system was still inherently limited by the low stiffness of the robot body.

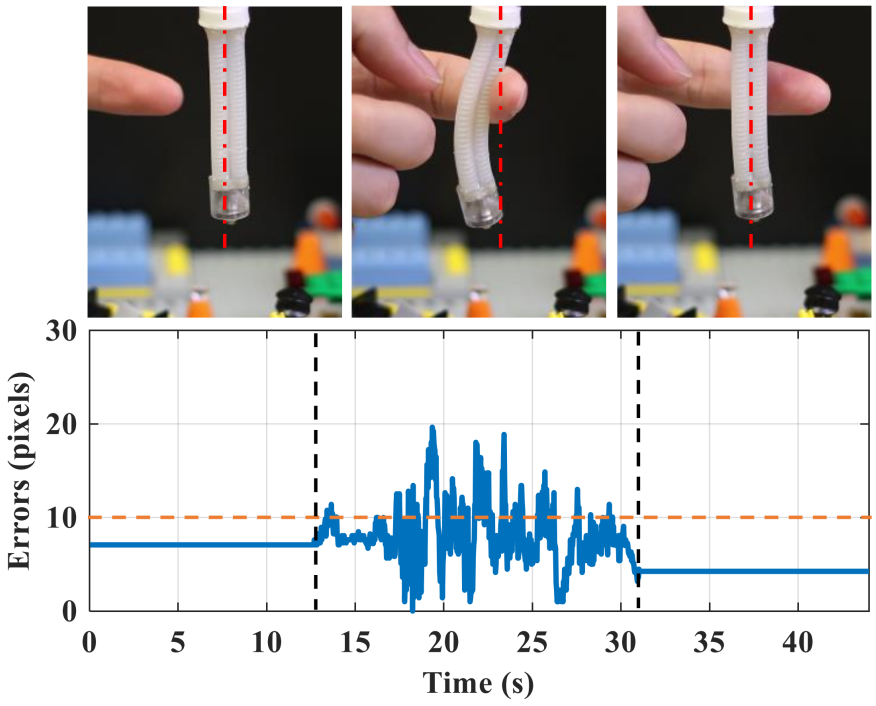
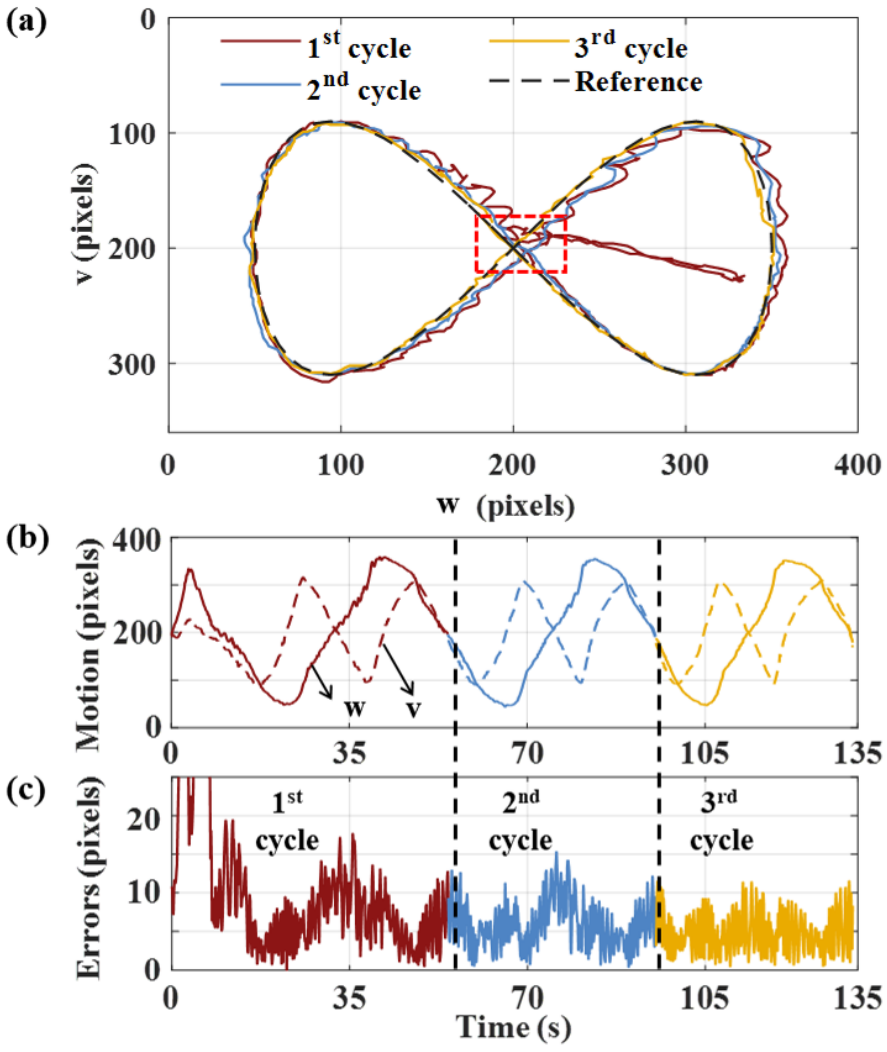


Fig. 4.7: Target tracking experiment involving external disturbance. The upper pictures (from left to right) show three robot configurations before, during and after the external force application. The correspoding tracking errors are shown below. The 10-pixel tolerance level is marked (dash lines in orange).

4.3.3.3 Path Following with a Scarce Pre-trained Model

The effect of online learning control was investigated via a path following task. The reference path was defined by a series of desired template center positions inside the 400×400 px camera frame. A purposely insufficiently trained LGPR model set was used at the start. Only a scarce dataset of 300 samples were presented to initialize the model, thus the controller must rely on the online collected data to refine the inverse mapping. The tracked trajectory and tracking errors are depicted in **Fig. 4.8**. In the first cycle, the robot could only roughly follow the reference path, with a root-mean-square error (RMSE) of 16.8 pixels and maximum of 64.5 pixels (**Fig. 4.8d**). Then the controller began to converge, keeping track of the reference path in the second and third cycles (**Fig. 4.8a**). The tracking error finally improved to an RMSE of 5.4 pixels and maximum of 11.5 pixels. **Fig. 4.8b** and **c** illustrate the w and v coordinates of image plane and the tracking error in pixels. The robot could follow the trajectory with a maximum error of 12 pixels after two cycles. This experiment demonstrates that the learning-based controller can achieve high accuracy path following through efficient online refinement of the inverse mapping, despite being initialized with a poorly trained model.



(d)

Cycle No.	1	2	3
RMSE (pixels)	16.8	6.6	5.4
Max. error (pixels)	64.5	15.3	11.5

Fig. 4.8: Performance of tracking on a predefined "∞" trajectory. **(a)** Robot tracing the reference trajectory in its endoscopic view using the pre-trained LGPR models. The red-dashed block indicates the initial view point centered at the intersection of the "∞" trajectory. **(b)** Motions depicted in two separated coordinates, w and v. **(c)** Corresponding tracking errors throughout the 135-second journey. **(d)** Tracking performances in general.

4.3.3.4 Path Following under Varying Load

The final experiment aims to study the online learning performance under varying tip load. In this regard, the LGPR models were first pre-trained at a vertical robot pose without any attachment. Then, an inflatable water balloon was wrapped around the robot tip, acting as a variable payload (**Fig. 4.9d**). Up to 15 grams of water could be inflated via a silicone tube. In addition to the 6 grams of balloon setup, it altogether corresponded to 105% of the robot original weight (20 g). The robot was also realigned to a horizontal pose after pre-training, meaning that the online learning controller was required to handle new tip load and gravity condition.

The tracking trajectory and errors in the three successive cycles are plotted in **Fig. 4.9a** and **Fig. 4.9b** respectively. In the first cycle, the controller quickly adapted to the additional 6 grams of balloon setup and the change in gravity direction. Despite the error peak at a $t = 22$ s, it was effectively compensated and eliminated in the following cycles. In the second cycle, the water balloon was inflated at $t = 89$ s to impose a fast additional payload of 12.5 g or 62.5% of the robot mass (dynamic loading), resulting in an error peak in 1 second (**Fig. 4.9c**). The configurations of robot with the inflated balloon at $t = 90.5$ s is shown in **Fig. 4.9d** (middle). The error was quickly diminished by the online learning controller at $t = 90.5 \sim 93.5$ s. The water balloon was further injected, increasing the tip load to a maximum of 15 g at $t = 94.8$ s (**Fig. 4.9d**, right). During this period where the loading changed slowly (quasi-static loading), a small tracking error could be maintained (< 12 pixels), which implies that the online learning controller can adapt to the additional tip load. Finally, the water balloon was deflated to empty at $t = 98$ s. The tracking error rose again due to the sudden change in payload, but then quickly converged to a small level within 1 second. In the third cycle, the tracking error maintained in a range less than 12 pixels. From this experiment, it could be analyzed that the closed-loop online learning controller could guarantee a stable path following under a variation (increment or decrement) of external loading within a 9.6%/s changing rate, where the percentage is referred to the added loading in the robot original weight. When the changing rate is in the range of 9.6$\sim$42.7%/s, the robot may not keep converging to the path. However, once the rate reduces less than 9.6%/s, the robot tracking error will converge to a steady range again, with an error reducing rate of $\sim$14 pixel/s. The proposed controller could handle the varying load, keeping track of a path with an error < 12 pixels, implying that the image motion feedback and actuation data can effectively update the kinematic models online.

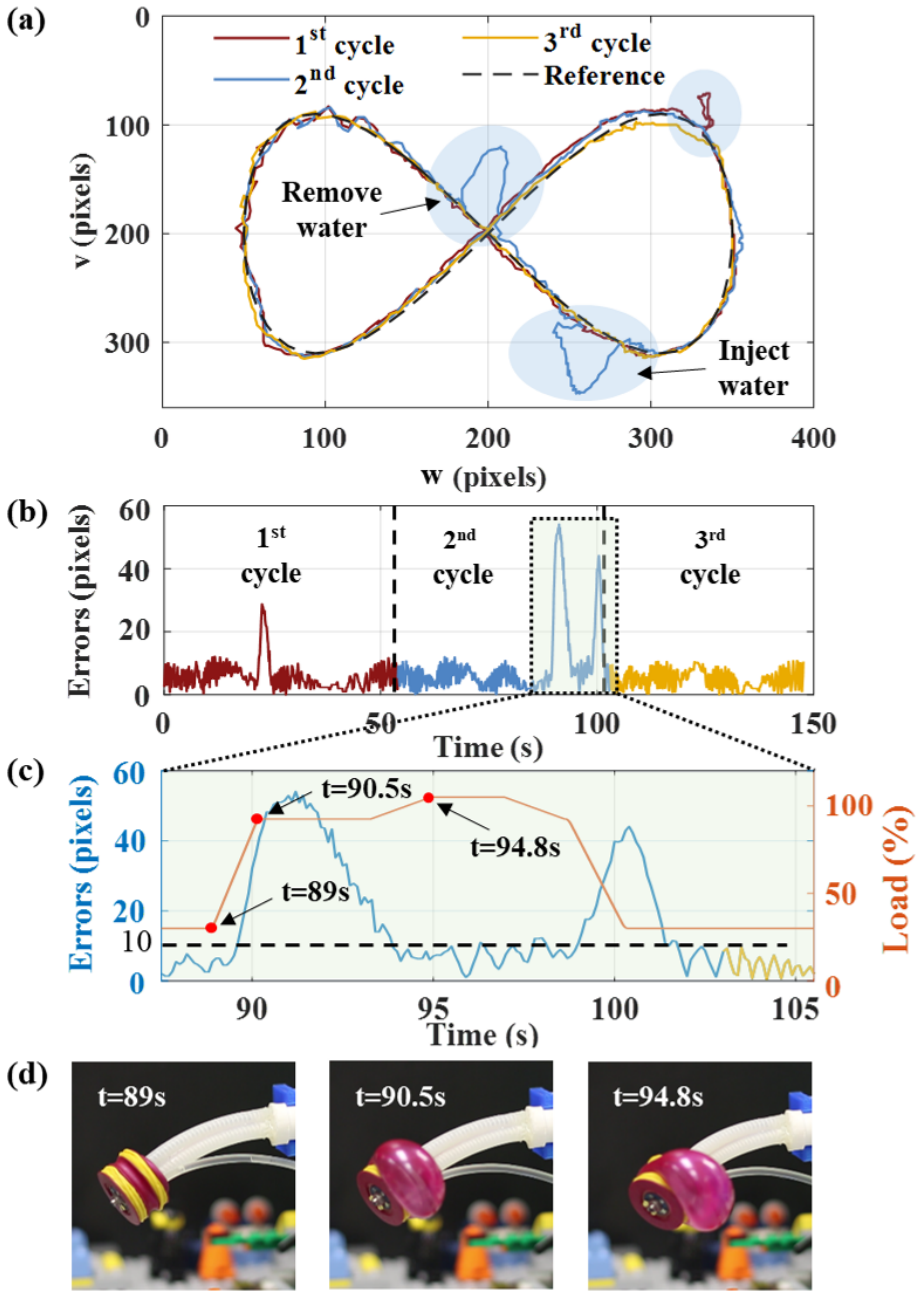


Fig. 4.9: Tracking performance under variable tip loading. A (6-gram) balloon cap pumped with water in-and-out was mounted at the robot tip, introducing a variable load ranging from 6 to 21 g. **(a)** Tracked trajectory in three cycles. Three substantial deviations corresponded to the errors of pre-training (1-st cycle in red), injecting and removing water (2-nd cycle in blue); **(b)** Corresponding tracking error in time domain; **(c)** Tracking error *vs* variable load varied below 21 g. The load is represented as % relative to the mass of robot itself, 20 g; **(d)** Robot configuration and its balloon shape at three-time steps.

4.4 SLAM-based 6D Pose Estimation of Robot End-effector

Utilizing the integrated eye-in-hand camera, the 6D camera pose estimation could also be conducted. As the robot end-effector shares the same pose with the camera, the vision-estimated result can be used as positional feedback for world-coordinate task-space control. Here the SLAM algorithm will be briefly introduced, the expanded implementations of which will be elaborated in **Chapter 6**.

4.4.1 Task Space Definition

The eye-in-hand camera is fixed on the robot end-effector, therefore sharing the same position with the robot tip. There have been various approaches utilizing visual feedback to analyze the camera pose. The end-effector moves along with the robot actuation. Suppose the initial position of the end-effector without robot inflated as the origin of a global coordinate system, and the robot central axis as the z-axis. The task space related to the end-effector position is defined in the 3D global frame **Fig. 4.10**. Represent the actuator input as well as the end-effector position and orientation (i.e. quaternion) as $\boldsymbol{u}(k) \in \mathbb{R}^m$, $\boldsymbol{p}(k) \in \mathbb{R}^3$ and $\boldsymbol{q}(k) \in \mathbb{R}^3$ at time step k (at equilibrium state) respectively, 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 $\Delta\boldsymbol{q}^*(k)$.

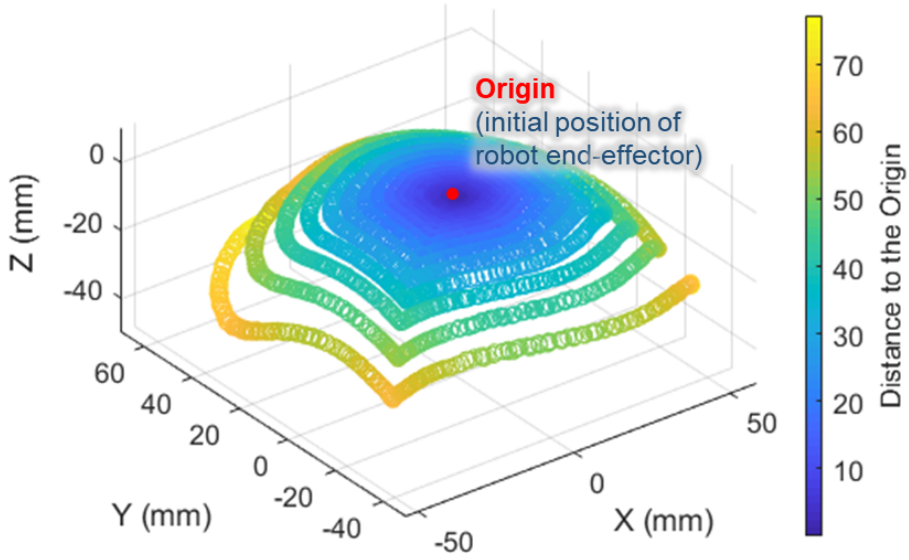


Fig. 4.10: Footprint of a soft continuum robot constructing the 3D position workspace. The pneumatic-driven robot was actuated by a pre-defined input sequence with increasing values.

4.4.2 Camera Pose Estimation via ORB-SLAM2

Camera pose estimation is one of the fundamental problems in computer vision, which has been investigated in structure from motion (SfM), SLAM, visual odometry (VO) [105], and can be further applied in augmented reality (AR) [106] and autonomous navigation [107]. Fiducial markers set in the scenario would facilitate optical cameras to determine the 3D position and orientation of robot end-effector or air vehicle [108, 109]. However, such marker-based algorithms are sometimes unable to meet scene diversity. Marker-less algorithms analyze the camera pose based on matches between 3D objects and 2D image features, namely feature correspondences [110]. These determinants of estimation accuracy are commonly ascertained using image matching techniques such as SIFT [111], SURF [112], and DAISY [113]. To improve the global consistency of robot trajectory and map building, multi-thread SLAM algorithms were proposed containing tracking, local mapping, and loop closing [114]. ORB-SLAM2 is a real-time SLAM system, providing accurate camera pose estimation solutions based on feature points with stereo, monocular, and RGB-D cameras [115]. In ORB-SLAM2, the camera pose is estimated by building a PnP (Perspective-n-Point) model in the tracking thread. As its calculation and ORB matching rate are extremely fast, ORB-SLAM2 could be chosen to achieve camera pose estimation in a feature-abundant environment.

Currently, the application of camera pose estimation in soft continuum robot centralized in the grasping manipulation using continuum fingers [116], space exploration, or search and rescue using large-scale arms [57]. For medical applications, pose estimation is usually investigated in the eye-to-hand modality for instrument pose estimation [117]. The eye-in-hand aspect for soft robots still has much room to be developed [118], which would facilitate the planar image stitching, 3D surface reconstruction, SLAM [118], and AR [106].

Like other SLAM systems, ORB-SLAM2 has three common modules: tracking, local mapping, and loop closing (**Fig. 4.11**). The camera pose can be obtained at each input image frame by building a PnP (Perspective-n-Point) model in the tracking thread. After ORB (Oriented FAST and Rotated BRIEF) features in the image are extracted, pose estimation can be conducted by matching features in two consecutive frames and refined by minimizing the re-projection error with motion-only bundle adjustment (BA) optimization. This re-projection error, represented as e_S here, is defined as the Euclidean distance between the image projection of 3D 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, quality of the pose estimation. The monocular camera was calibrated with OpenCV via Robot Operating System (ROS). Define the camera position and orientation estimated by the camera at time step k as $\boldsymbol{p}_c(k) \in \mathbb{R}^3$ and $\boldsymbol{q}_c(k) \in \mathbb{R}^4$, respectively. Here quaternion representation would be used for the orientation. This pose $\boldsymbol{z}_c(k) = \begin{bmatrix} \boldsymbol{p}_c(k) & \boldsymbol{q}_c(k) \end{bmatrix}$ estimated in feature-abundant camera views under stable and smooth movement could be considered as ground truth of the end-effector pose. 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. This SLAM-based pose estimation result will be applied in **Section 6.4** for the sensing fusion of 6D pose estimation.

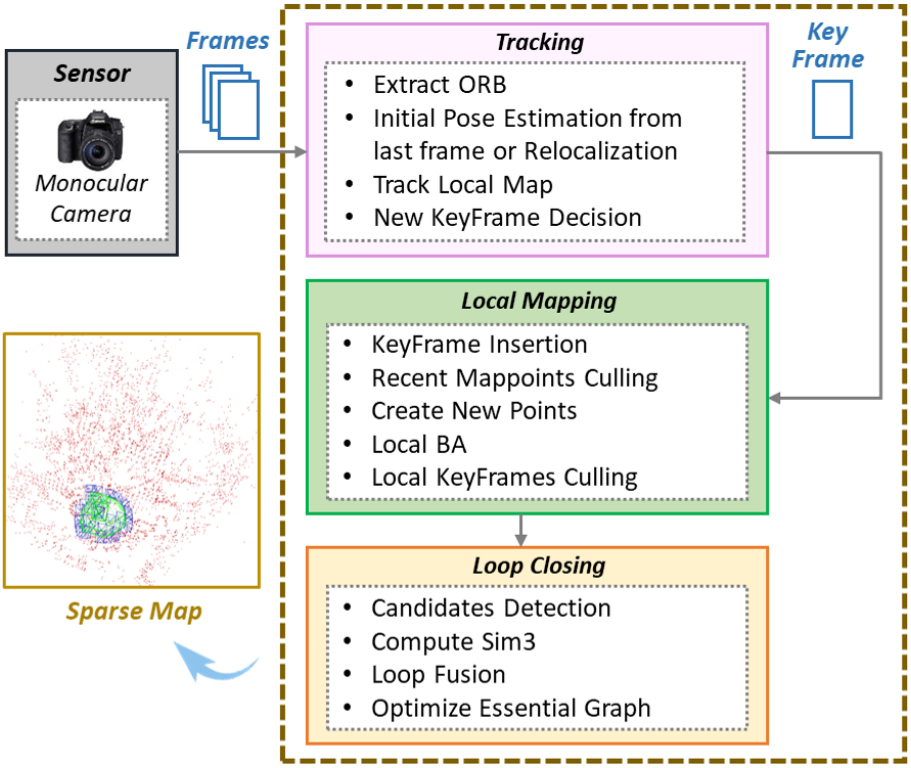


Fig. 4.11: Module diagram of ORB-SLAM2, where the tracking, local mapping, and loop closing threads run in parallel. In the sparse map, the red points are recorded landmarks while the blue and green markers represent the camera poses corresponding to key frames and camera accumulated trajectory, respectively.

4.5 Conclusion

This chapter presents a nonparametric online learning control framework, which enables eye-in-hand visual servo of a fluid-driven soft robot with very low stiffness. By estimating the inverse mapping solely from measured data, the proposed controller alleviates the need for a kinematic model or camera calibration, that may be challenging to acquire for soft manipulators. Compared to existing model-based visual servoing methods which did not consider unknown mechanical disturbances, my proposed learning-based method first resolves unknown variable external loading. The local weighted learning scheme supports efficient online update of the inverse mapping, thus enabling precise robot manipulation even under external interactions such as changing payload. Integrating image feedback with nonparametric learning can bring new opportunities to minimally invasive surgical applications, such as soft robotic endoscopy and laparoscopy [119, 120], as they can take advantage of existing camera feedback.

This first half part demonstrates vision-based path following for a hyper-elastic robot with heavy variable loading (up to 105% of the robot weight). In the cycle with addition or removal of the payload, it maintained an acceptable accuracy (RMSE 14.4 pixels and maximum 54.1 pixels). Even under a fast and heavy dynamic load ($\sim$ 62.5% of robot mass over 1.5 s), the controller could maintain stability. When the water load slowly changed in the period between 93.5 to 98 s (quasi-static loading), the controller could effectively reduce the error, in contrast to the initial dynamic loading. After removing the load, the tracking error converged to an even lower value (RMSE 5.6 pixels and maximum 10.8 pixels). In my future work, I will investigate the use of known robot or camera information in the algorithm to improve control performance. For example, Lee *et. al* [2] utilized FEM of their soft manipulator to initialize a learning-based model, reducing the need for time consuming random exploration. I also intend to extend the online learning controller to more dynamic tasks. Visual servo control of highly redundant robots will also be of my interest. This would enable more versatile manipulation in a confined space.

Chapter 5

FBG-aided Modeling and Control for Tendon-driven Catheters

5.1 Introduction

Optical fiber-based sensors using FBGs have attracted interest for their ability to measure morphological deformation of flexible surgical instruments. It can achieve real-time shape estimation with high sampling rates (>100 Hz) with thin and sub-millimeter diameters footprints. FBGs have excellent multiplexing capabilities, where many sensing points can be employed along a fiber without increasing the fiber diameter (**Section 2.3**) [121]. The use of multi-core fibers with FBGs can further increase the sensor packaging density and enable shape estimation of the fiber itself [122]. The flexibility of optical fiber also has minimal effect on structural stiffness of the instrument, allowing it to be integrated with delicate devices, such as needles and catheters. These distinctive advantages have prompted the use of FBG shape sensing in many applications, including navigation of medical instruments [123, 124], shape estimation and control of steerable interventional needles [125, 126], as well as force sensing of instruments [127]. Additionally, due to the inherent electromagnetic immunity [128], FBG fibers are entirely compatible with magnetic resonance imaging (MRI), which is a medical imaging modality prohibiting ferromagnetic materials. Details of MRI and requirements in magnetic resonance (MR) condition will be introduced in **Section 5.2**. The use of FBG-based sensing in MR safe/conditional instruments has been demonstrated in related researches [129, 130].

Despite the increasing use of FBG for shape sensing, few studies have applied

it to enhance the closed-loop control performance of surgical continuum manipulators [131,132]. In this chapter, the standard cardiac catheter for electrophysiology (EP) ($\sim$1.5 m) will be illustrated as an instrument example to validate the novel learning-based modeling method. In the cardiac EP procedure, complicated and repeated tasks introduce difficulties to manual operation of catheters, even for experienced operators. It is expected that fast and accurate robot-assisted catheter positioning could significantly reduce operation times and manipulation workload comparing to manual operation [133]. Previous studies have investigated various modeling/control methods to improve the accuracy and effectiveness of robotic catheter systems [101, 134]. However, most of these catheters were custom-made with their known or deterministic structural parameters that enable precise modeling. For a standard cardiac catheter, its mechanical characteristics will vary due to manufacturing tolerances, inducing uncertainties in modeling and control. It creates strong incentive for utilizing shape sensing in robotic catheter control to achieve dexterous and precise manipulation, especially under MRI guidance.

In this chapter, I aim to design a control interface and evaluation system for our customized teleoperated catheterization, and propose a shape tracking method that can provide both positional and morphological sensing of the flexible cardiac catheter. To achieve accurate and effective feedback control, shape information obtained from the FBGs are used for characterization of catheter kinematics as well as real-time autonomous control. The contributions are summarized below:

1. Design and implementation of a shape tracking system integrating a multi-core FBG fiber and tracking coils with a slim continuum manipulator (validated using a cardiac EP catheter). Both shape and positional tracking of the bendable section could be achieved;

2. Development of a learning-based modeling method for continuum robot integrated the optical fiber, which uses FBG-reconstructed 3D curvatures to initialize the model and update the motion mapping. The proposed modeling method was implemented on an MRI-guided robotic catheterization platform to achieve feedback control of a cardiac catheter;

3. Experimental validations of the shape tracking system and control algorithm. The overall robot control performance was demonstrated by point-to-point targeting and path following tasks.

5.2 MRI Guidance in Robotic Interventions

Image-guided interventions have developed as recognized standard medical procedures, applying computer-based imaging modalities to navigate surgical instruments in assorted diagnoses or treatments [19]. Preoperative (pre-op) or intra-op images or their combination visualize patient anatomies and relative movements of surgical instruments for precise and safer operation. This kind of technique enables MIP leading to fitter patient compliance and usually improved treatment results [135, 136]. MRI, fluoroscopy, ultrasound (US), computed tomography (CT), and positron emission tomography (PET) are representative modalities for different purposes. Image guidance leads to more accurate control and, therefore, effective treatment results. The imaging procedures usually have minimal invasions and bring improved patient compliance.

Clear visualization of targets and their surrounding tissues is fundamental to guarantee the safety. One essential capability of imaging techniques is to distinguish tissue states and types. CT is superior to X-rays in soft-tissue contrast, but inferior to MRI [136]. However, high-resolution imaging requires stricter working environment and more computational time. Conventional image-guided interventions would be accomplished in several steps: *1)* acquirement of pre-op images as preparation; *2)* intra-op tracking of surgical instruments using localizers; *3)* registration of patient anatomy to the pre-op image; *4)* augmentation of surgical instruments relative to the patient anatomy on the image; *5)* manipulation via the virtual display; *6)* generation of a configuration image upon procedure completion [19]. Typical applications have been conducted in the clinical areas of neurosurgery, orthopedics, cardiology, and abdominal interventions [19]. To maximize the synchronization as well as intuitive presentation, multi-modality image fusion has been developed. It is potential to improve the success of large-volume tumor ablation by overlapping composite treatment zones [137]. At the same time, fast intra-op MRI is also under investigation to achieve clear local visualization in real-time while avoiding registration errors.

MRI is the prime modality for visualizing tissue pathologic changes and guide instruments into target areas. Multiple contrast mechanisms available in MRI could be used not only to reveal tissue structure but also to probe the function of the tissue and examine physiology [138]. This high contrast in tissue is available even without the need for contrast medium [139] (**Fig. 5.2**). It can monitor and guide percutaneous procedures, operations, or therapies [136]. Positional, anatomical, and physiological changes could be monitored utilizing MRI. Recently, MRI-guided interventions have been rapidly developed,

while intra-operative MRI even shows the potential to bring breakthroughs in neurosurgical oncology. Besides, MRI performs well in the spatial and temporal resolution [140]. In addition to excellent imaging quality, MRI possesses an important advantage that it induces no ionizing radiation to the patient. As for the guidance role of MRI, comprehensive multi-planar imaging has been developed with at least 3D views [141]. Moreover, MRI can detect and quantify flow and temperature [142, 143], facilitating the guidance for thermotherapies. These above factors identify MRI as the optimal choice for accurate instrument guidance for interventions.

However, the working principle of such imaging modality also brings limitations and special requirements to the operating conditions. Conventional surgical instruments and standard operating room equipment are not compatible with MRI environment due to the effect of mid-to-high magnetic field [144]. Instrument-related safety issues in interventional MRI include imaging artifacts, passive instrument movements caused by the magnetic field, and unexpected heating [144]. Instruments and equipment are required to be MR-safe and MR-compatible, i.e., non-magnetic and to imaging disturbance [136]. Stipulations are also proposed in aspect to the actuation system.

5.3 Clinical Motivation of Cardiac EP Catheterization

Catheterization is an interventional procedure for the treatment of cardiovascular diseases. EP is a type of procedure to examine the electrical system of hearts and diagnose the source and nature of heart rhythm disorders (arrhythmias). Treatments could be conducted based on the diagnosing results utilizing EP. The areas causing arrhythmias may send abnormal impulses (triggers) or function as an electrical pathway for the impulses. In cardiac EP for diagnosing atrial fibrillation, a long ($\sim$1.5 m) and flexible EP catheter is inserted from the femoral vein to the heart chamber, e.g., LA, to obtain an electro-anatomical map (EAM) [145–147]. Treatment is delivered to create lesions inside the corresponding chamber using RF ablation or others. Taking the ablation in LA as an example, catheter is inserted at femoral vein, then passes through the inferior vena cava (IVC) to the right atrium (RA). After puncturing the atrial septum, it reaches LA for RF ablation [148] (**Fig. 5.1a**). The non-conductive scars created by RF ablation, usually on pulmonary vein ostia (PVO), will isolate the abnormal EP signals to treat arrhythmias [148–150] (**Fig. 5.1b**). In consideration of the operation safety and effectiveness, this MIP requires delicate and consistent motion of the catheter tip via the manipulation on the catheter handle. However, even provided with

X-ray visual guidance, maneuvering the distal tip to the desired location via the catheter handle on the other side is still a challenging task in manual manipulations.

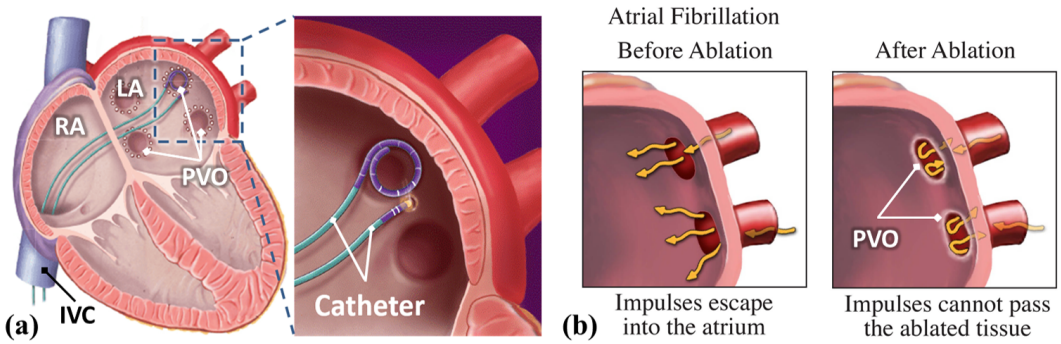


Fig. 5.1: Schematic diagram of RF ablation in LA. **(a)** Sketch diagram of RF ablation in LA, LA, where the catheter passes through IVC to RA, punctures the atrial septum, and reaches LA for RF ablation. **(b)** Atrial fibrillation corrected with ablation surgery, where non-conductive scars on PVO isolate the abnormal EP signals. **Image source**: [12]

To facilitate precise manipulation of the surgeons, robotic catheterization has attracted much attention. Despite the advent of several commercial robotic catheter systems, such as Sensei^{TM} X (Hansen Medical, Inc., Mountain View, CA, USA), Niobe (Stereotaxis, Inc., St. Louis, MO, USA) and Armigo^{TM} Remote Catheter System (Catheter Precision, Inc., Mount Olive, NJ, USA) [36, 133, 151–153], currently, there is not any study on MRI-guided tele-manipulation of the catheter. The conventional navigation relies on intra-op X-ray to provide a real-time navigation interface [154]. Compared with X-ray, intra-op MRI could offer superior details and differentiate soft tissues [155], which makes it a superior choice for cardiac EP catheterization. MRI-guided EP procedure has been validated by several research groups with patient trials [156–158], showing wide potential in practical.

However, there is neither commercial product nor research prototype of an MRI-guide robotic system for EP catheterization. Intra-op MRI provides superior soft tissue contrast [145], which facilitates the access to lesion locations and visualization of ablation progress (**Fig. 5.2**). It can be used to form a detailed 3D cardiac roadmap, visualize physiological changes of cardiac tissues and also to assess the ablation lesion formation intra-operatively [147, 157]. In this chapter, our customized MR Safe robotic platform [159] was integrated with a human-robot control interface to evaluate the control performance of kinematic model-based (**Section 3.2**) and model-free methods (**Section 3.4.1**), as well as how they influence the hand-to-eye coordination of operators. Another advantage of MRI-guided catheterization is the superior accuracy of MR-based tracking [160–162], in which catheter position could be measured under the same coordinate as imaging, eliminating potential registration error (**Section 5.5.3**). This characteristic also allows the generation of a

virtual endoscopic view from the catheter tip. The hand-to-eye coordination can hence be simplified, as the operator can control the catheter movement referring to the endoscopic view instead of conventional Cartesian space. Such an endoscopic visual guidance could improve accuracy and potentially reduce perforation risk in a simulated ablation task [163].

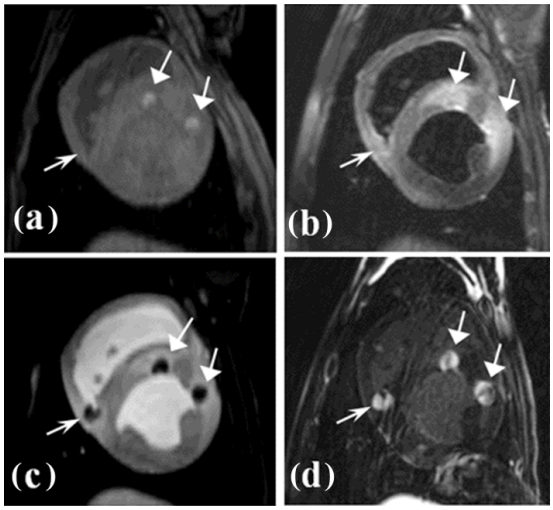


Fig. 5.2: MRI visibility of RF and cryo ablated lesions imaged using non-contrast **(a)** T1w, **(b)** T2w, and contrast-enhanced **(c)** T1w and **(d)** late gadolinium enhancement techniques. Wide arrows mark RF ablation results, while narrow arrows represent cryo-ablation. All RF ablated lesions could be readily detected on non-contrast images. **Image source**: [13]

5.4 Teleoperation Platform of Robotic Catheterization

A customized catheter robot [159] was used to replace manual control of the bi-directional EP catheter as shown in **Fig. 5.3**. Both MR-conditional catheter and standard EP catheters are compatible to this robot. The robot enables manipulation of three degrees of freedom (DoFs) for bending, rotation and insertion of the catheter. Details will be introduced in **Section 5.4.2.1**. As this robotic platform is a teleoperated system, the control strategy needs to map the robot actuation and movement of catheter tip. In this section, a control mapping for hand-to-eye coordinated teleoperation was proposed and implemented based on the virtual endoscopic view. Based on this generated endoscopic view, two representative control methods, as the kinematic model-based and the model-free control, were introduced for validation on the customized catheter robot prototype and comparison. The model-based controller (**Section 3.2**) was designed based on the CC kinematic model [164, 165]. The model-free controller (**Section 3.4.1**) utilized position feedback to update the kinematic Jacobian [96] by quadratic programming.

EP catheterization ablation requires skillful manipulation of a thin, flexible instrument

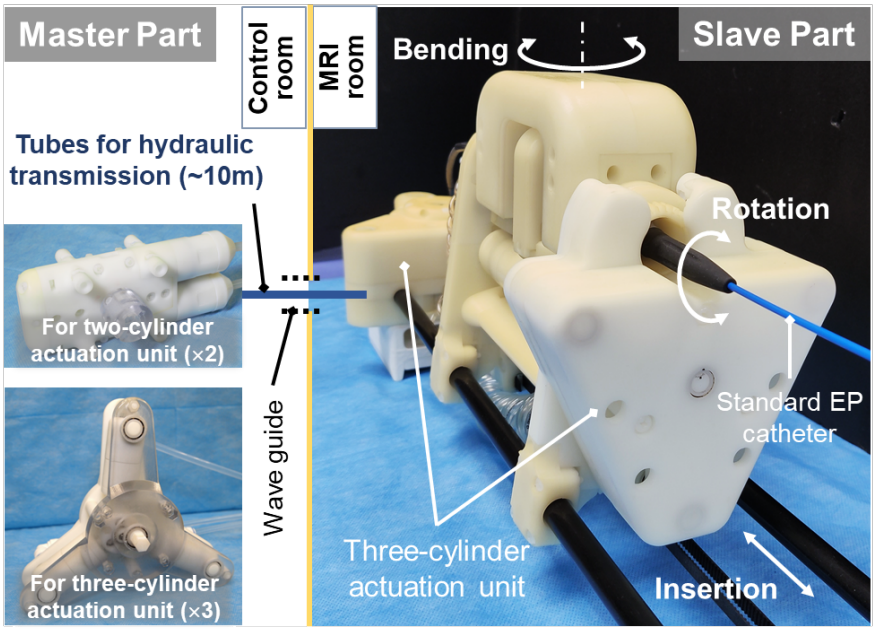


Fig. 5.3: MR safe robotic catheter platform providing the 3-DoF manipulation (bending, rotation and insertion) of standard cardiac electrophysiology (EP) catheter.

in a confined cardiac chamber. Even assisted by the navigation from a cardiac EP roadmap, manual manipulating the long catheter to target areas still remains challenging. The catheter tip motion will be affected by deforming cardiac tissue and pulsatile flow, such as in LA. The proposed experimental validation platform simulates the RF ablation procedure on the PVO inside the LA. The two control methods were implemented in this platform and validated their performances regarding accuracy and efficiency. The operator could manipulate the cardiac EP catheter via a 3D input device with the visualization of a real-time virtual endoscopic view. As for validation, quantitative evaluation indices were designed to grade the performances using subject test.

5.4.1 Generation of Endoscopic View

Fig. 5.4a demonstrates the definition of the proposed virtual endoscopic view. This additional view was defined and registered on the catheter tip, functioning as a camera perceiving the front view. The z-axis of the endoscopic view was aligned with the normal direction of the catheter tip. Denote a unit vector $\boldsymbol{z}_c \in \mathbb{R}^3$ to represent this direction. The pose (position and orientation) of virtual camera could be captured by a tracking system which is compatible with the MRI environment. In the interface design, an association between the catheter tip motion demonstrated in the endoscopic view and the operator's input, that is, to accomplish hand-to-eye coordination. To this end, the tip

pose and movement in the world coordinate obtained by the positional tracking system need to be transformed into the endoscopic frame (**Fig. 5.4b**). Denote the incremental movement of the attached virtual camera w.r.t. the world Cartesian coordinate $\{\boldsymbol{W}\}$ as $\Delta\boldsymbol{p}_w = \begin{bmatrix} \Delta x_w & \Delta y_w & \Delta z_w \end{bmatrix}^{\mathrm{T}}$, and the movement w.r.t. the endoscopic frame $\{\boldsymbol{C}\}$ as $\Delta\boldsymbol{p}_c = \begin{bmatrix} \Delta x_c & \Delta y_c & \Delta z_c \end{bmatrix}^{\mathrm{T}}$.

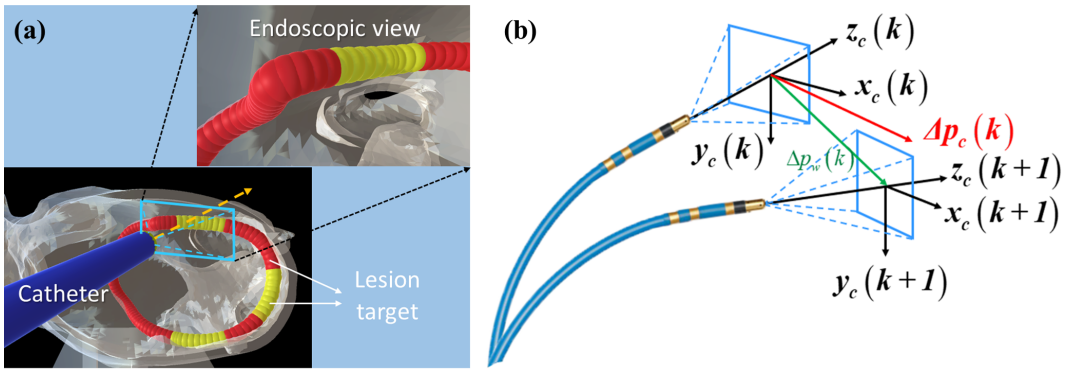


Fig. 5.4: **(a)** Virtual lesion targets near PVO pre-defined in the EP roadmap, with successfully ablated targets displayed in yellow and to-be-completed ones in red. The normal direction of endoscopic view was set to follow the axial direction of catheter tip; **(b)** Diagram illustrating the incremental movement of catheter tip from time step k to $(k+1)$.

The rolling of the camera along its z-axis of the endoscopic frame was fixed during the transition. This is for ensuring the consistency of catheter tip steering direction. The aforementioned $\boldsymbol{z}_c$ could represent the rolling axis. The horizontal and vertical axes x- and y-axis of 2D camera view could also be served as the direction of $\boldsymbol{x}_c$ and $\boldsymbol{y}_c$ of the virtual endoscopic view. Denote the rotation matrix from the world coordinate $\{\boldsymbol{W}\}$ to camera $\{\boldsymbol{C}\}$ coordinate frame as ${}^{C}_{W}\boldsymbol{R} = \begin{bmatrix} \hat{\boldsymbol{x}}_c & \hat{\boldsymbol{y}}_c & \hat{\boldsymbol{z}}_c \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{3\times3}$. If omitting the $\hat{\boldsymbol{z}}_c$ row, the matrix ${}^{C}_{W}\boldsymbol{R}$ would be ${}^{C}_{W}\widehat{\boldsymbol{R}} = \begin{bmatrix} \hat{\boldsymbol{x}}_c & \hat{\boldsymbol{y}}_c \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2\times3}$, which could transform the tip movement relative to its image plane coordinate, i.e. $\Delta\boldsymbol{p}_c = {}^{C}_{W}\boldsymbol{R}(\Delta\boldsymbol{p}_w)$.

5.4.2 Control Methodologies

With the variable $\begin{bmatrix} \Delta x_c & \Delta y_c \end{bmatrix}^{\mathrm{T}}$ to be controlled in the endoscopic view defined, the next main object is how to establish the connection with the operator's manipulation. Model-based control method is widely-used in the traditional rigid robots, the kinematics of which is relatively accurate and constant. However, for such a soft and long catheter, the unmodeled uncertainties will obviously affect the performances. The kinematic model-free control detours the modeling procedure, usually depends on the sensory data instead. This characteristic may accommodate the interactions with environments more compared with

the model-based control. But once getting sucked for some reasons, the update of sensory data will be inefficient for the generation of the to-be-learned model or the estimation using numerical calculation. This section will introduce the details of the model-based control under PCC assumption, and the model-free control using dynamic programming.

5.4.2.1 Kinematic Model-based Control

Model-based control relies on the analytical kinematic/dynamic model pre-established for the thin catheter. As introduced in **Chapter 5**, the catheter bendable distal can be regarded as a continuum manipulator. CC model [164, 165], beam theory model [166], n-rigid links model [167] have been investigated on catheters. A standard cardiac EP catheter can is capable of approximately planar bending in two opposite directions by pulling tendons **Chapter 5**. The forward kinematics of standard catheters could be established based on D-H parameter [164]. This model is constructed with the basis of CC and zero torsion assumptions. An brief schematic diagram of bending mechanism can be found in **Fig. 5.5**.

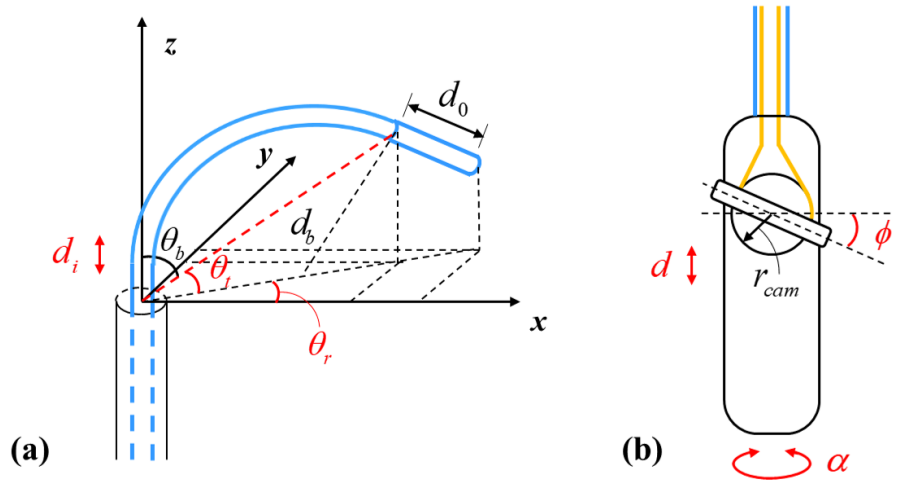


Fig. 5.5: Schematic diagram illustrating the manipulation mechanism of a standard catheter. **(a)** CC geometry of the catheter distal section. **(b)** 3-DoF manipulation of the catheter handle, including the axial translation d, knob twisting ϕ, and handle rotation α.

In **Fig. 5.5a**, the 3 DoFs of catheter distal section are represented as d_i, θ_r and θ_b, for the axial translation, rotation angle (bending direction) the bending angle, respectively. The active deformation of the catheter bendable part is actuated by manipulations on the catheter handle (**Fig. 5.5b**), where parameters regarding the 3 DoFs are insertion d, knob twisting α and handle rotation ϕ. The insertion d of catheter handle can be regarded approximately equal to the axial translation d_i of catheter tip, i.e., $d\,(d \approx d_i)$. The tip rotation angle θ_r can be calculated by multiplying a torsional transmission coefficient K to the

handle rotation angle α, as $\theta_r = K\alpha$. In the experiment, the catheter body was guided by Polytetrafluoroethylene (PTFE) pipeline to the LA simulator, and the friction between catheter and PTFE could be negligible. Thus parameter K could be valued as 1. According to the CC assumption, the catheter bending can be described utilizing the twisting angle ϕ of the knob as:

$$\Delta R\left(\pi - 2\theta_t\right) = r_{cam}\phi\left(1+k\right), \tag{5.1}$$

where a relation $\theta_t + \theta_b = \pi/2$ is satisfied referring to **Fig. 5.5**. r_{cam} is the cam radius, which is fixed with the steering knob to pull/push tendons, and ΔR is the difference of inner and outer arcs constructed by the tendons inside the bendable section. A backlash k is included to consider the tendons' additional displacement due to tension [165]. When the tendons are sufficiently pre-tensioned, this factor could be regarded as 1.

The position of catheter tip can be depicted based on variables d, ϕ, and α within the coordinate system of **Fig. 5.5a** as follows

$$\boldsymbol{p} = \begin{bmatrix} \mathrm{c}\left(K\alpha\right)\left(\dfrac{l}{M\phi}\mathrm{s}^2\left(M\phi\right) + d_0\mathrm{s}\left(2M\phi\right)\right) \\ \mathrm{s}\left(K\alpha\right)\left(\dfrac{l}{M\phi}\mathrm{s}^2\left(M\phi\right) + d_0\mathrm{s}\left(2M\phi\right)\right) \\ d + \dfrac{l}{2M\phi}\mathrm{s}\left(2M\phi\right) + d_0\mathrm{c}\left(2M\phi\right) \end{bmatrix}, M = \frac{r_{cam}\left(1+k\right)}{\Delta R}. \tag{5.2}$$

where l is the total length of catheter bendable section, and d_0 represents the length of rigid straight segment connected to the bendable part. The symbol $\mathrm{s}\left(\cdot\right)$ and $\mathrm{c}\left(\cdot\right)$ are abbrevations for operation $\sin\left(\cdot\right)$ and $\cos\left(\cdot\right)$, respectively. Length l is the sum of original length l_b and an additional length Δl which bring about the arc, respectively. In such a tendon-driven mechanism with small diameter, Δl can be ignored compared to the original l_b, i.e.,

$$l = \Delta l + l_b = 2r_{cam}\theta_t + l_b \approx l_b. \tag{5.3}$$

The actuation part of robot can be regarded to have a linear relation with the 3-DoF manipulation on the handle $\boldsymbol{u} = \begin{bmatrix} \phi & \alpha & d \end{bmatrix}^{\mathrm{T}}$. Denote the tip position in the virtual camera frame $\{\boldsymbol{C}\}$ as $\boldsymbol{p}_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^{\mathrm{T}}$. Therefore, the corresponding differential function could be represented as

$$\dot{\boldsymbol{p}}_c = \boldsymbol{J}\dot{\boldsymbol{u}}, \tag{5.4}$$

where the Jacobian matrix $\boldsymbol{J}$ could be calculated by differentiating the catheter tip position $\boldsymbol{p}_w$ w.r.t. the input variable $\boldsymbol{u}$. $\boldsymbol{p}_c$ and $\boldsymbol{p}_w$ could be transformed using the rotation matrix ${}^{C}_{W}\boldsymbol{R}$.

With the Jacobian matrix of forward kinematics, its inverse kinematic function $\dot{\boldsymbol{u}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{p}}_c$ can be established, which has the discretized form as

$$\Delta \boldsymbol{u} = \boldsymbol{J}^{-1}\Delta \boldsymbol{p}_c. \tag{5.5}$$

The focus of this teleoperated platform is to synchronize the operator's input motion with the catheter tip motion in the virtual endoscopic view. The user's input via a 3D input device is assigned to $\Delta \boldsymbol{p}_c$, which is the input to inverse kinematics in each step. The inverse Jacobian matrix $\boldsymbol{J}^{-1}$ can be calculated by substituting the variable values in the last time step. Then the actuation-related variable $\Delta \boldsymbol{u}$ could be calculated by **Eq. 5.5**.

5.4.2.2 Model-free Control

The kinematic model-free control method implemented here has been introduced in **Section 3.4.1**, which is simplified from the optimization approach used in [96]. This method initializes a Jacobian matrix for the use in inverse kinematics first, and then sets the variance of Jacobian as an objective function, to update it online. As same as in the model-based method, input and output are the actuation command $\boldsymbol{u}$ and catheter tip motion $\boldsymbol{p}_c$.

a. Jacobian Matrix Initialization: The robotic catheter has independent insertion, bending, and rotation DoFs, thus the initialization could be conducted by sequentially actuating the three DoFs of the robot with a discrete value Δu_i, $i = 1,2,3$, and recording the resultant positional movements $\Delta \boldsymbol{p}_{ci}$ of catheter tip tip. The Jacobian matrix could be therefore initialized by $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{J}_2 & \boldsymbol{J}_3 \end{bmatrix}$, where $\boldsymbol{J}_i = \Delta \boldsymbol{p}_{ci}/\Delta u_i$. A weighting matrix $\boldsymbol{W}$ is designed as $\boldsymbol{W} = \text{diag}\left(\|\boldsymbol{J}_1\|, \|\boldsymbol{J}_2\|, \|\boldsymbol{J}_3\|\right)$. Therefore, the kinematic function (5.4) could be represented as

$$\dot{\boldsymbol{p}}_c = \widehat{\boldsymbol{J}}\boldsymbol{W}\dot{\boldsymbol{u}},\ \widehat{\boldsymbol{J}} = \boldsymbol{J}\boldsymbol{W}^{-1}\ , \tag{5.6}$$

corresponding to a discrete form of

$$\Delta \boldsymbol{p}_c = \widehat{\boldsymbol{J}}\boldsymbol{W}\Delta \boldsymbol{u}. \tag{5.7}$$

b. Jacobian Matrix Updating: The update of Jacobian matrix utilizes the continuous

estimation of a quadratic programming process as below:

$$\begin{aligned} &\text{minimize} && \left\| \Delta \widehat{\boldsymbol{J}}(k) \right\| \\ &\text{subject to} && \Delta \boldsymbol{z}_p(k) = \widehat{\boldsymbol{J}}(k+1)\, \boldsymbol{W} \Delta \boldsymbol{u}(k) \ , \\ & && \widehat{\boldsymbol{J}}(k+1) = \widehat{\boldsymbol{J}}(k) + \Delta \widehat{\boldsymbol{J}}(k) \end{aligned} \tag{5.8}$$

where $\widehat{\boldsymbol{J}}(k)$ and $\widehat{\boldsymbol{J}}(k+1)$ are the known Jacobian matrix at time step k, and to be determined Jacobian $\widehat{\boldsymbol{J}}(k+1)$ at time step $(k+1)$. $\widehat{\boldsymbol{J}}(k+1)$ will be obtained by minimizing the Frobenius norm (L_2-norm) of $\Delta \widehat{\boldsymbol{J}}(k)$ to be added with $\widehat{\boldsymbol{J}}(k)$, aiming for smooth transition. $\Delta \boldsymbol{u}(k)$ and $\Delta \boldsymbol{p}_c(k)$ are the change of actuation and movement in the camera view at time k. Once $\widehat{\boldsymbol{J}}(k+1)$ is available, the new actuation command can be obtained as

$$\Delta \boldsymbol{u}(k+1) = \left(\widehat{\boldsymbol{J}}(k+1)\, \boldsymbol{W} \right)^{-1} \Delta {\boldsymbol{p}_c}^*(k+1), \tag{5.9}$$

where $\Delta {\boldsymbol{p}_c}^*(k+1)$ represents the target movement.

5.4.3 Experimental Setup

The validation platform was comprised of the customized catheterization robot, a standard cardiac EP catheter, a lab-based positional tracking systems as well as a human–robot control interface.

a) Catheterization Robot: The catheter robot was driven by hydraulic transmissions, which made it capable to be operated under intra-op MRI [159]. The robot comprises master and slave actuation, respectively, designed to locate in the control room and the MRI scanning room, respectively (**Fig. 5.3**). No metallic material is integrated into the slave actuation, ensuring the safety under the MR environment and the minimal interference to the MR images. The master units were actuated by electric DC motors in the control room. The master-slave motion transmission from control room and MRI room was conducted through long hydraulic pipelines ($\approx$ 10 meters). Such design separates the source of energy from the MRI scanner, ensuring negligible EM interference to the MR images. The three DoFs of the robot employed three-cylinder actuation units. A catheter holder was designed to enable being plugged-in with a standard RF ablation catheter (e.g. from Biosense Webster Inc. or St. Jude Medical), thus bent and steered the catheter distal section in a full range of $\pm 180^\circ$. The robot can provide sufficient workspace to perform RF ablation for PVI.

b) Left Atrial Model: The 3D phantom model of LA (**Fig. 5.6**) was reconstructed from

MRI scanning and 3D-printed with soft material (AgliusClear, Stratasys Inc.). The shore hardness of the material was A30-35 and the polymerized density was $1.14 - 1.15$ g/cm^3. The model comprised more than 5000 meshed, which was accurate and glossy enough to reproduce the geometry of real tissues. To conduct RF ablation inside LA, catheter will be needed to pass through femoral vein, IVC and RA, then approaches LA for the RF ablation around PVO [148]. In the experiment, a pathway sheath constructed by PTFE pipeline would guide the catheter to the LA simulator, mimicking the similar entering position and direction to actual cardiovascular interventions.

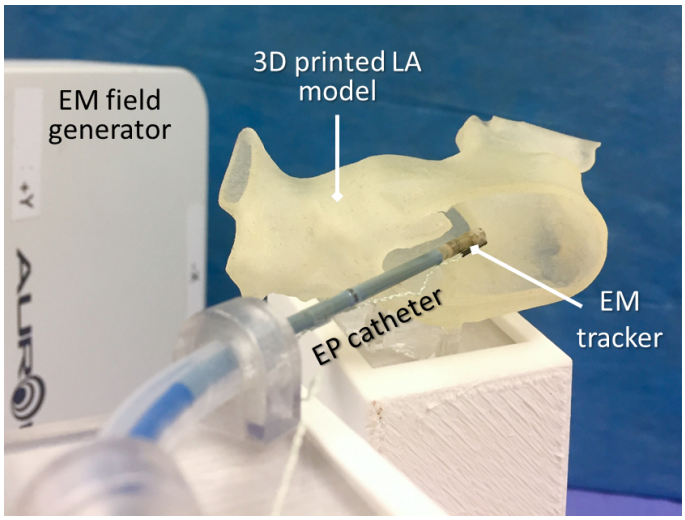


Fig. 5.6: LA simulator and EP ablation catheter. A PTFE pipeline guided the catheter to approaching the PVO in a similar position and direction to cardiovascular interventions. EM tracking system was used to provide positional measurement.

c) Position Tracking System: To feed back the catheter tip position and orientation in the world coordinate, an EM tracking system (Aurora V3, Northern Digital Inc.) (**Fig. 5.6**) was adopted in the lab-based experimental setup. A miniaturized tracking coil attached to the catheter tip would allow real-time pose measurement in six DoFs. For clinical applications, the MR-based active tracking approaches [161] can be utilized to acquire the position information with high frequency (40 Hz).

d) Robot Control Interface: As preparation, the LA simulator needed to be registered to the virtual LA model in Unity 3D. The LA simulator was made of soft material and anchored on a customized rigid pedestal. Six points on the pedestal were predefined and marked for registration between the world coordinated calibrated by EM tracking system and the virtual environment in Unity 3D. Based on the transformation established by these reference points, the LA simulator could be registered to the virtual LA. As introduced in "Position Tracking System", a 6-DoF EM tracking coil (Aurora # 610029, Northern Digital

Inc.) was attached to the distal end of the catheter, to capture the position and orientation. Hence, the relative geometric configuration between LA simulator and catheter could be measured in real-time for visualization and evaluation.

The human-robot control interface provided three virtual sub-views (**Fig. 5.7**), which included two overall views from different visual angles and one endoscopic view, and all these three sub-views were available to the subjects during the subject test. The overall views could visualize the interaction between the inner cavity environment and catheter tip from the exterior position. The generation of endoscopic view (**Fig. 5.4**) is introduced in Section **Section 5.4.1**. A 3D haptic device (Novint Falcon Haptics Controller) (**Fig. 5.8**) is set to read the operator's input. With the assistance of the endoscopic view, the operator can readily recognize the status of catheter tip and manipulate it at the same time. The hand-to-eye coordination can be built.

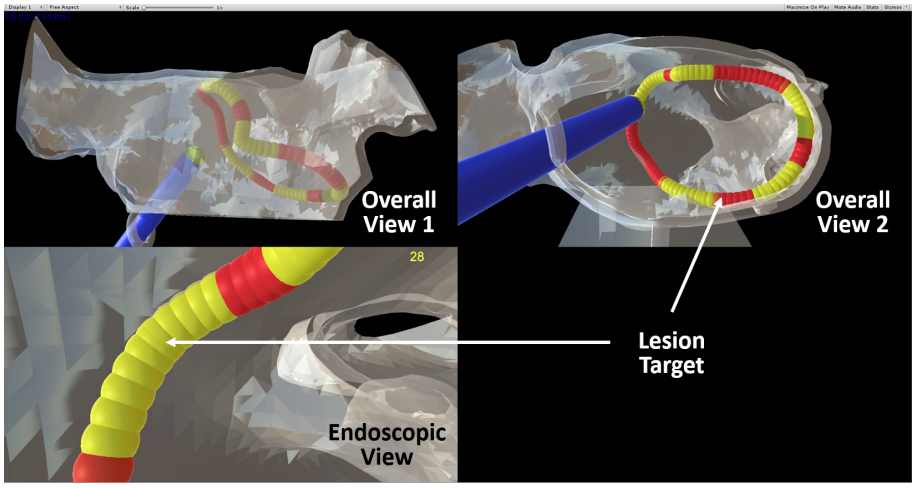


Fig. 5.7: Virtual visualization of the catheter tip providing with simulated LA roadmap displayed in three camera perspectives, including two virtual overall views and one virtual endoscopic view. Simulated roadmap was indicated with circular lesion targets. Successfully ablated targets would turn yellow.

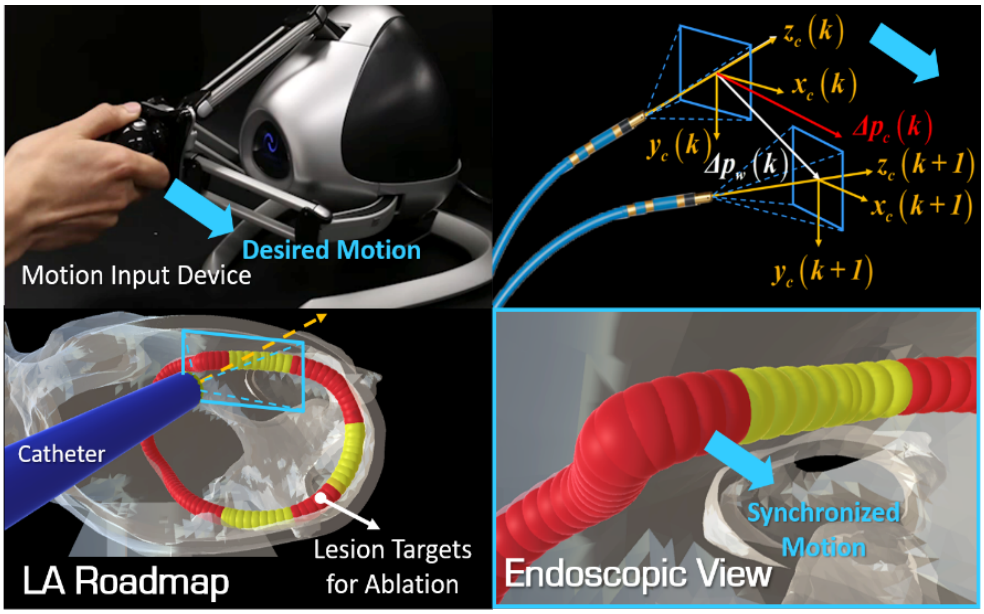


Fig. 5.8: Human-robot control interface and realization of hand-to-eye coordination. The motion of catheter tip reflected in the endoscopic view was designed to be synchronized with the input motion from the operator.

5.4.4 Evaluations via Simulated Ablation Task

In our navigation interface, a 3D roadmap of "lesion" targets (in red) composed by 121 spheres revealed around the PVO on the virtual LA model. The lesion targets were registered to the 3D-printed LA simulator together with its virtual model as mentioned in **Section 5.4.3**. The lesion targets were designed to consistently align with the inner surface of the LA model, since the limber catheter could only impose a limited-range of deformation ($\leq$ 2mm) on the LA simulator and even less on the "lesion" targets. This deformation induced by the force of catheter tip could be ignored in the evaluation of accuracy. Efficient ablation was judged when satisfying requirements on both tip-to-target distance and ablation duration. Valid operations performed on corresponding targets would turn their color from red to yellow gradually. The color could demonstrate ablation status. With a measuring frequency at 20 Hz, recorded data for further offline evaluations are: **1)** time stamp, **2)** catheter tip position (**Fig. 5.9**), **3)** distances from catheter tip to all lesion targets, **4)** ablation ON/OFF, and **5)** lesion target status denoted by values $0 \sim 50$ (0 for original status, 50 for successfully ablated).

Ten subjects (with ages $20 \sim 30$, 7 males and 3 females) were invited to join in this test. This evaluation followed the within-subjects design; thus all the subjects would conduct the

task using both two control methods in random order. In this way, the variance induced by individual differences could be minimized, since each one would act as her/his individual comparison standard [168]. All subjects had no practical background in EP catheterization. They were briefly introduced about the robot manipulation task before the operation. The time limit for this simulated RF ablation task was set as 3 minutes.

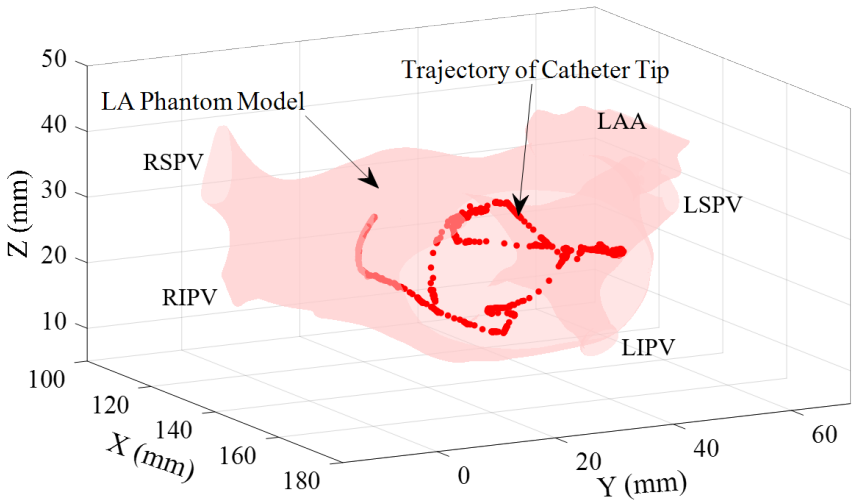


Fig. 5.9: Actual footprint of the catheter tip recorded by the EM tracking system. The LA model was overlaid in the same coordinate.

Accuracy and efficiency are the most important focus in clinical, thus corresponding evaluation indices for assessment were designed and analyzed (**Fig. 5.10**). The accuracy could be assessed in terms of **1)** the proximity distance measured from catheter tip to lesion target during the ablation, **2)** the number of times that ablation was turned on, and **3)** average ablation duration; and efficiency could be evaluated by **1)** the missed proportion of lesion targets and **2)** the maximum number of continuously missed lesion targets, and **3)** the total travel distance of the catheter tip.

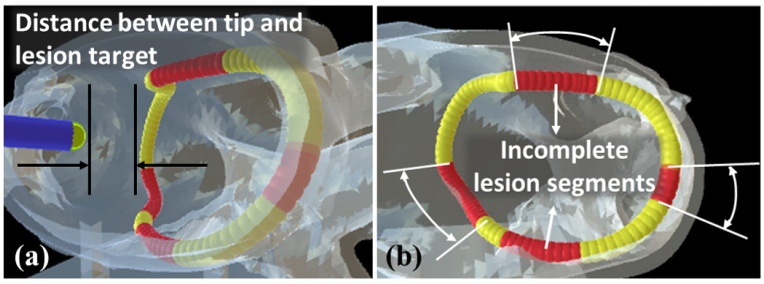


Fig. 5.10: Two major performance indices, namely accuracy and efficiency defined based on **(a)** proximity distance measured from tip to lesion target around the PVO and **(b)** total length of incomplete lesion segments (red).

Table 5.1 and **Fig. 5.11** show the performance evaluation result in aspects of accuracy

and efficiency/ Model-based and model-free control methods were compared regarding each index. Model-based method is set as the comparison baseline and the "improvement" means the increment of model-free control referring to the model-based method.

Table 5.1: Evaluation results of 10 sujects in performing the teleoperation task using model-based and model-free control.

Model	**Model-based**		**Model-free**		**Improvement**	
Accuracy	**Mean**	**STD**	**Mean**	**STD**	**%**	**p-val.**
Distance to target during ablation (mm)	6.94	1.34	5.61	1.31	19.1	0.08
Mean times of ablation turning on (sec)	23.7	10.9	24.1	7.2	-1.88	0.89
Ablation duration (sec)	1.49	0.36	1.76	0.48	-18.7	0.07
Efficiency	**Mean**	**STD**	**Mean**	**STD**	**%**	**p-val.**
Mean proportion of missed targets (%)	51.9	14.7	33.3	11.9	35.8	0.03
Maximum continuously missed targets	55.1	24.1	29.7	3.16	46.2	0.01
Total travel distance (mm)	649	176	699	320	-7.65	0.60

a) Accuracy: During the ablation (when ablation mode was turned on by foot pedal), the average tip-to-target distance using model-free control was 19.1% shorter than that in the model-based. This index indicated the advantage of the model-free control method to approach static targets. Besides, as for the comparison of ablation switch-on times and duration, the model-based method had fewer times and shorter duration. Such higher accuracy of the model-free control could be interpreted that the kinematic model was proposed without analyzing the interactions between the catheter and LA model. The simplified CC geometry could not depict and resolve the robot-specif factors sufficiently. Moreover, cardiac EP intervention is a task where complicated interactions with the endocardial environment are inevitable. However, the Jacobian matrix in model-free control could be updated online base on the actual robot kinematics and absorb the sensing data under soft contact with the simulator.

b) Efficiency: The completion percentage using model-free control was higher than that of the model-based control, with a 35.8% reduction in the missed lesion targets, and a 46.2% reduction in the maximum number of continuously missed lesion target. Both of these indices corresponded to a ≤ 0.05 p-value. At the same time, the model-free method demonstrated a shorter total tip footprint distance. These results indicate that the model-free method is possible to provide more sensitive and faster responses to the operator's input.

Fig. 5.11: Performance comparison of kinematic model-based and model-free control methods regarding the accuracy and effectiveness indices. The lower the p-value (≤ 0.05), the higher the probability.

This experimental platform using static LA model was the first step to validate the catheter control methods, as well as the robot platform. In the following section, the pre-clinical validation was enhanced by using a nonstatic LA simulator, of which the deformation can be induced by simulated pulsatile flow of liquid.

5.5 Shape Tracking with FBG Sensors

FBGs are sensors inscribed directly into optical fibers for measuring local, 1D strain. Multiple FBGs can be axially inscribed inside a single fiber to obtain strain sensing points without additional input or output connections. Lun *et al.* designed a sensor capable of surface reconstruction in a flexible substrate by embedding a single-core FBG fiber [169]. However, for flexible and thin surgical instruments that are designed to access deep regions in the body, e.g., biopsy needles and cardiac catheters, a fiber with only a single core of FBGs is insufficient for reconstructing its 3D curvature. Alternatively, multiple single-core fibers could be grouped [170] for co-located strain measurements. However, this would come at the cost of sensor size and ease of fabrication. A more reliable approach was to adopt a fiber fabricated with multiple optical cores (multi-core fiber) and FBGs [122]. It will allow shape sensing of the fiber geometry itself while maintaining a thin cross-sectional size. Here, shape sensing is referred to reconstructing the spatial configuration of the fiber. The shape can be regarded as a curve instead of a volume. Shape tracking is about the reconstruction of shape in real-time. The definitions of ”shape sensing” and ”shape tracking” both indicate the sensory measurement technology, but not robotic control tasks.

5.5.1 Shape Sensing with Multi-core FBG Fiber

A continuous-grating multi-core FBG fiber (FBGS International) with optical frequency domain reflectometry (OFDR) interrogation (RTS125+, Sensuron) is intend to reconstruct the shape of the cardiac catheter. The fiber was 12-m length with a diameter of $\varnothing$0.2 mm. As in **Fig. 5.12b**, seven cores run along the length of fiber, with one core centered around by other 6 cores equally spaced at 60 intervals. Shape sensing can be achieved by measuring off-axis strain from the FBG sensors located within the fiber cross-section. For the applied fiber in this chapter, strain measurements could be obtained at the distal 1-m section of the fiber with a spatial resolution of 3.17 mm. Note that the adopted multi-core fiber features with a high-density grating fiber structure; therefore, continuous strain measurement can be achieved. By contrast, the typical wavelength division multiplexing (WDM)-based fiber contains discretely-placed FBGs that are divided by bare fiber intervals.

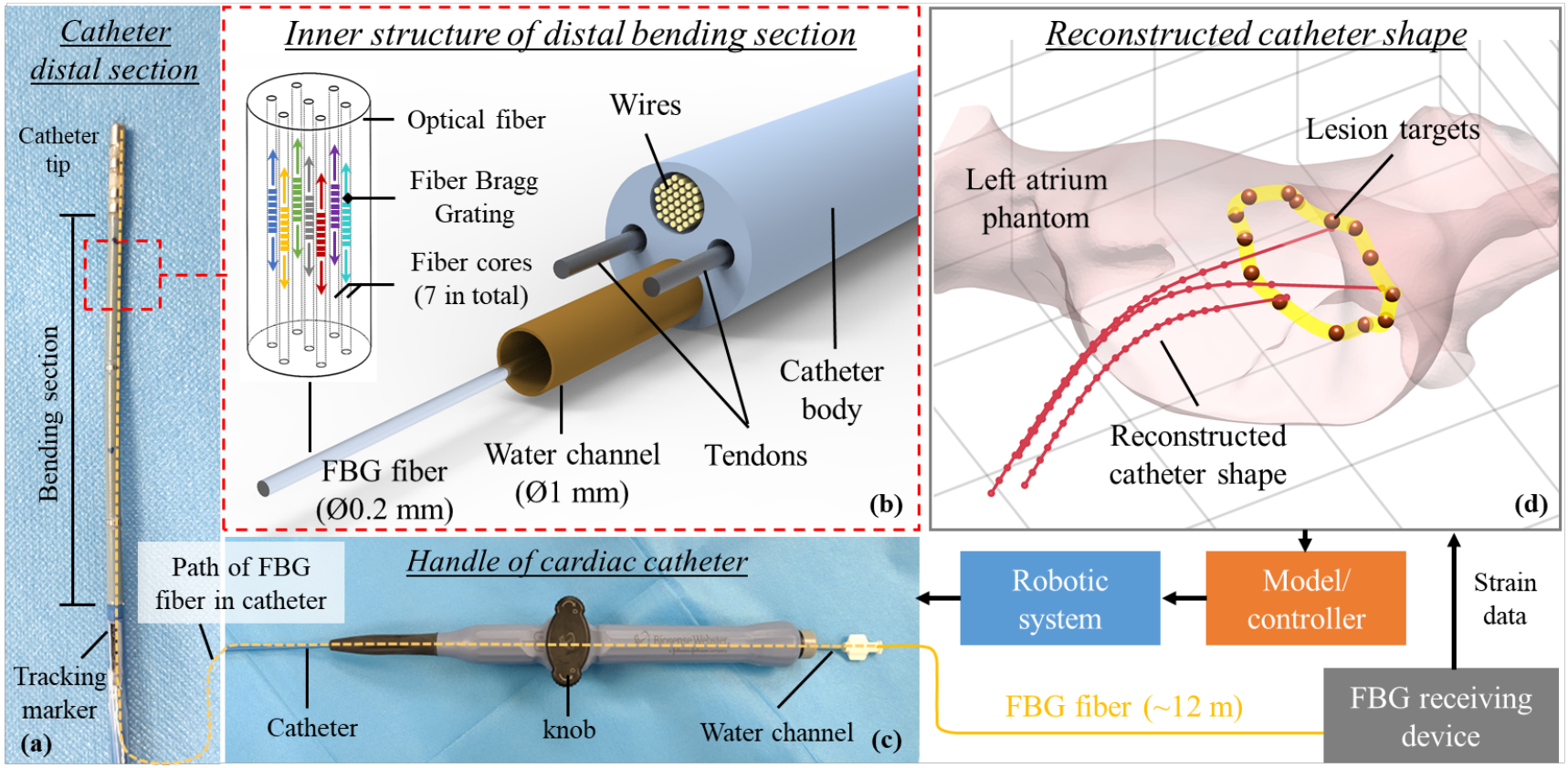


Fig. 5.12: Diagram showing the architecture of shape sensing for robotic catheter system. **(a)** Catheter distal bending section integrating a multi-core FBG fiber (12-m long) and the tracking marker for shape tracking; **(b)** Inner structure of distal bending section, with multi-core FBG fiber put throughout the water channel; **(c)** Handle of cardiac catheter, where the FBG fiber can be fixed at the water channel entrance; **(d)** Real-time shape reconstruction can be achieved by continuously acquiring the strain data from FBGs. The 3D curvature of the catheter bending section can then be feedback to the robotic catheter platform for autonomous control.

There are two common approaches towards multi-core FBG fiber shape reconstruction from strain data: *i)* Adopting the PCC model with the assumption of circular bending shape for each segment of FBG. The overall fiber shape can be constructed by accumulating the sum of discrete curvatures and corresponding bending directions, which are measured and calculated from a set of sensors at a particular cross-section of the fiber [171]; *ii)* Utilizing the Frenet-Serret formulas to describe 3D curves, which can take account of the torsion effect [172].

To optimize the reconstruction speed (i.e., computation time of shape reconstruction at each sensing step) and sensing performance (e.g., reconstruction accuracy), the PCC-based shape reconstruction approach is used in accredit to its faster convergence with respect to (w.r.t.) sensing segment length and superior noise handling ability [173]. Notice that this PCC model has the same principle as that in the continuum robot control, however, here this model is used to construct the shape of multi-core fiber. For 3D curvature reconstruction, at least three cores are needed to estimate the 2-degree-of-freedom (DoF) section bending along the fiber cross-section. More fiber cores can be employed to improve reconstruction accuracy and reduce overall sensing noise.

5.5.2 Fusing and Calibration of Shape and Positional Feedback

For shorter and more rigid instruments like biopsy needles, MR-based positional tracking markers can be located at a fixed base of the instrument to provide a reliable reference frame. However, this approach is not suitable for long ($\sim$1.5 m) and flexible cardiac catheters, since its handle would be placed far from the MR imaging volume. In order to monitor the 3D curvature of a standard cardiac catheter, as well as its tip pose, the coordinate frame of the fiber's reconstructed curvature has to be co-registered with MR images. Note that the 3D curvature reconstruction error will also be accumulated much along with the increasing sensing length.

To this end, the distal section of a multi-core fiber was incorporated to cover and feedback the bending of a 63.4-mm length catheter tip. Three cores were used, each containing 21 FBGs for strain measurement. In our preliminary test, the FBG fiber was put throughout the water channel of EP catheter (Thermocool® Bi-Directional Catheter, Biosense Webster Inc.) (**Fig. 5.12b-c**). The water irrigation function would not be affected as the fiber diameter ($\varnothing$0.2 mm) can be negligible relative to that of the water channel ($\varnothing$1 mm). Alternatively, the fiber could be integrated into the wire channel (**Fig. 5.12b**) of

catheter in the future manufacture. To minimize the noise induced by fiber tip contact, the fiber was offset approximately 2 mm from the distal end and fixed at the water channel entrance to prevent sliding (**Fig. 5.12c**).

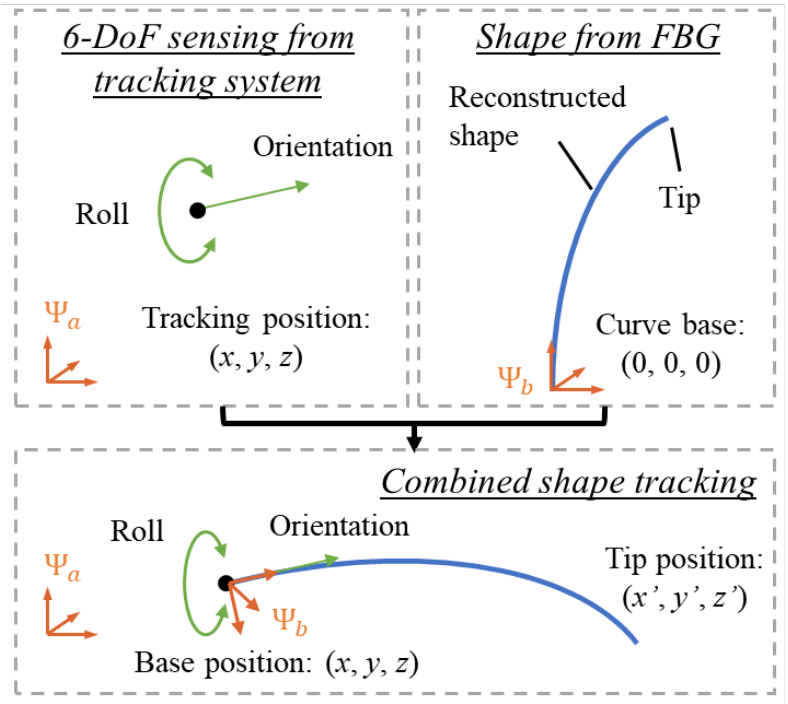


Fig. 5.13: Diagram illustrating the shape tracking method by *fusing* the 3D curvature from FBG and positional sensing from the tracking coil.

Although the fiber can be used to reconstruct its 3D curvature, the origin and orientation of reconstructed shape are not defined within the robot or MRI coordinate system. In addition, as the fiber was fixed at the catheter handle, the catheter rolling could not be measured by the shape reconstruction. Therefore, the use of real-time MR-based positional tracking coils is proposed, which will be integrated into the catheter to provide this information (**Fig. 5.13**). Three coils located closely before the active bending section are expected to provide 6-DoF information in the MR imaging coordinate, and also map it to the catheter model. MR-based coils were implemented to accommodate the specific MRI environment. In scenarios without the MR condition limitation, the coils could be replaced with other positional sensors. In our lab-based experimental validations, EM trackers and field generator (NDI Aurora) could be used as a comparable proxy for the MR-based tracking system. Preliminary validation of the MR tracking coils distributed along a catheter has been conducted and introduced in **Section 5.5.3** (**Fig. 5.14**).

Sensor calibration could be performed with the catheter bendable section placed in free space. The catheter was actuated to steer in opposite bending directions, while the

FBG-reconstructed positions along the catheter bendable section were collected. The bending plane $\mathscr{P}_m$ was generated by fitting through least squares of the normal distances. Similarly, the bending plane generated by FBG-reconstructed points can be obtained as $\mathscr{P}_w$. The transformation to align $\mathscr{P}_w$ to $\mathscr{P}_m$ can then be calculated by point-by-point optimization. In this way, the FBG-reconstructed shape can be registered with the MR tracking coordinates, thereby in the same domain as the imaging model, e.g. EAM.

5.5.3 Towards Shape Tracking in MRI

MR tracking markers/coils [174] can be used for 3D localization of catheters under MRI. Active coil wired to the receiver system could realize a tracking resolution as high as $0.6 \times 0.6 \times 0.6$ mm^3 [175]. MR-guided active tracking for cardiac EP catheters has been demonstrated in RF ablation human trials, where active markers enabled real-time tracking and overlay of EP catheters on MR images [176]. Additionally, wireless multilayer markers [177–179] have potential to further simplify integration with catheters. Fabricated with amplifying circuits, MR tracking markers can provide high-contrast MR signal against anatomical surroundings, thus allowing positional localization in the MR image coordinate (**Fig. 5.14a**). The use of 1D-projection pulse-sequence [180] allows acquisition of the marker positional signal along each axis, enabling fast localization up to 30 Hz. With an MRI real-time control interface, RTHawk (Heartvista™), the raw scanning data can also be streamed out at low latency ($<$20 ms). The 1D-projection signals of three markers attached longitudinally and at different separations along a catheter body can be seen in **Fig. 5.14b**. The peaks along each coordinate could be easily detected, and further back-projected to 3D marker positions by the geometry constraint matching, such as utilizing the distances between each marker pairs. An arc was fitted to pass through the three markers and approximate the 3D catheter shape (**Fig. 5.14a**). Benefiting from the small size and high accuracy ($\sim$0.48 mm), three markers could be mounted on the catheter in three separated circumferential directions in the future applications. The catheter orientation and rotation could be obtained by calculating the pose of cross-sectional triangle formed by the markers.

This **Section 5.5** depicts the sensing devices and algorithms used for the catheter shape acquisition, and proposes a solution for shape monitoring under MRI. This shape information will be utilized to establish a more detailed kinematic model, and offer real-time tip positional feedback for closed-loop control.

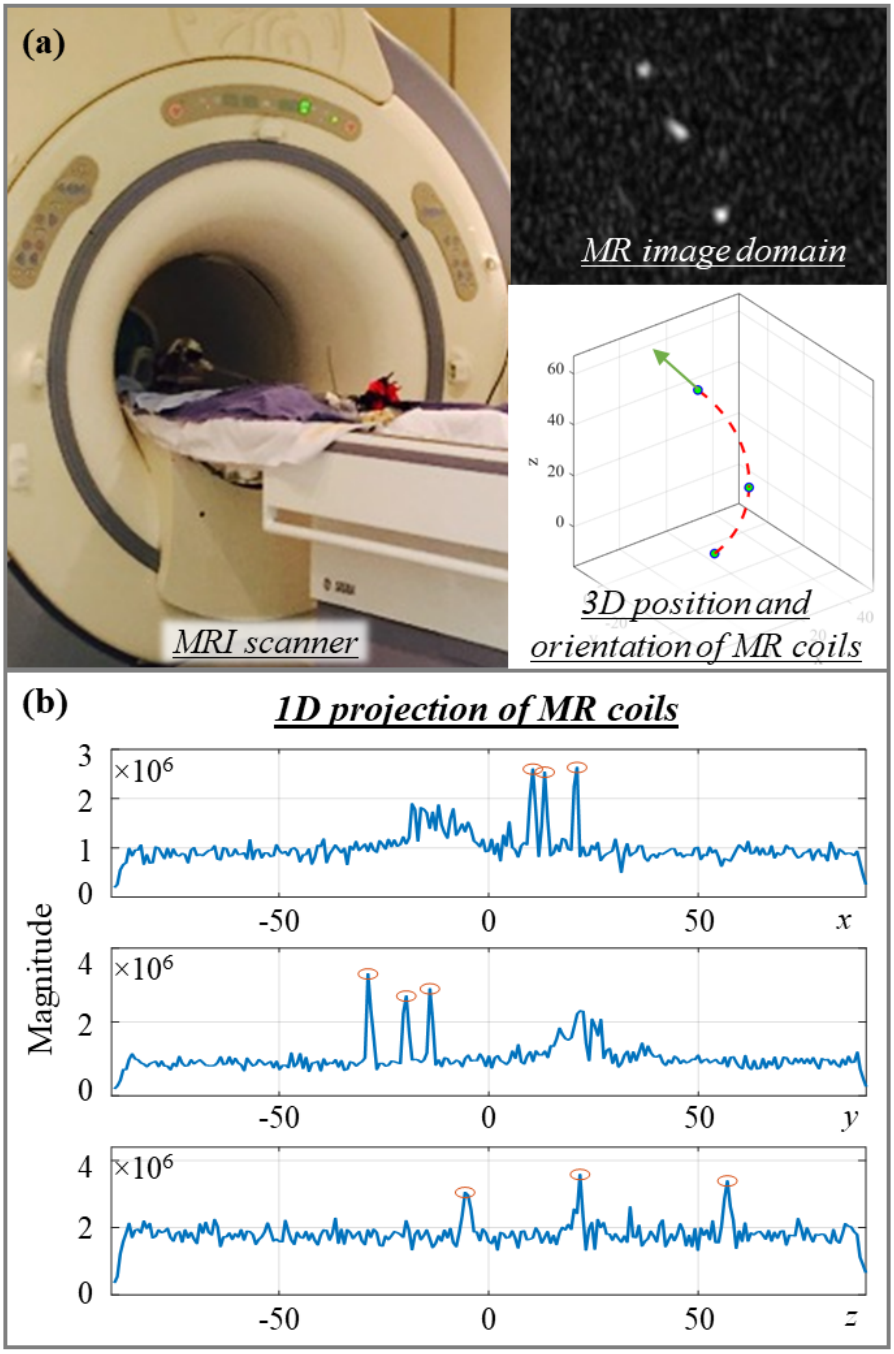


Fig. 5.14: **(a)** MR image revealing the MR tracking markers and the reconstructed 3D positions from the 1-D projection pulse-sequence; **(b)** 1D projection of the three tracking coils along the orthogonal axes. Three RF signal peaks can be found in each projection.

5.6 Learning-based Modeling and Control of Cardiac Catheter

Geometries of CC and integral-based PCC have been introduced in **Section 3.2**. However, the establishment of complete PCC kinematic mapping still requires information of curvatures. This information could be exactly obtained utilizing the FBG-based curvature reconstruction. In this section, the learning-based modeling method aided by FBG is explained, which can be applied further in the control continuum robot.

5.6.1 Modeling of Catheters

The catheter would be integrated on our previously developed MR safe catheter robot [159] (**Fig. 5.3**). The robot features a master-slave hydraulic transmission that can manipulate a standard cardiac catheter in three degrees of freedom (DoFs) for bending, rotation and insertion. The steerable catheter can be regarded as a typical robotic continuum manipulator after integrated into the actuation platform. To achieve accurate and effective autonomous control of catheters, particularly commercial ones, a reliable model is needed to provide precise motion mapping from the robotic actuator to the catheter tip.

Regarding modeling of the continuum manipulator, a common approach was to consider each of its bending sections as a constant-curvature arc [164]. The forward and inverse kinematic models can then be derived [181]. There are other more complex modeling approaches using beam theory [166] or Cosserat rod theory [182], mimicking a more realistic deflection but requiring longer computation time. However, for commercial cardiac catheters, the mechanical characteristics can vary due to manufacturing tolerances, which are difficult to be modeled using fixed parameters. It is also time-consuming and impractical to characterize the parameters of each cardiac catheter before use.

In view of this, a modeling approach based on the assumption of PCC is developed. The bendable distal shaft is modeled as a combination of n piecewise circular arcs, with curvatures represented by κ_i, $i = 1, 2, \cdots, n$. The distal section is assumed to deform in a plane determined by the two tendons. Regarding the base of bendable section as the origin, define the arc length, s, from the origin to a point on the bendable section, the angle between axis z and tangent at that point as $\theta(s)$ (**Fig. 5.15**). The arc length of each segment is l_i, $i = 1, 2, \cdots, n$. Without considering the rotation DoF, the catheter would bend in a fixed plane (x-z plane as in **Fig. 5.15**) w.r.t. the robot coordinate. Utilizing the modeling method as introduced in **Section 3.2.1**, the endpoint position of catheter bendable part could be

obtained. The constant length of distal straight tip is defined as d_0, and the insertion distance and rotation angle of end-effector as d and α, respectively. Thus the spatial position $\boldsymbol{p} = (x, y, z)$ of catheter end-effector in a coordinate as in **Fig. 5.15** could be represented as

$$\begin{cases} x = (x_p + d_0 \sin\theta_n)\cos\alpha \\ y = (x_p + d_0 \sin\theta_n)\sin\alpha \\ z = z_p + d_0 \cos\theta_n + d \end{cases}. \tag{5.10}$$

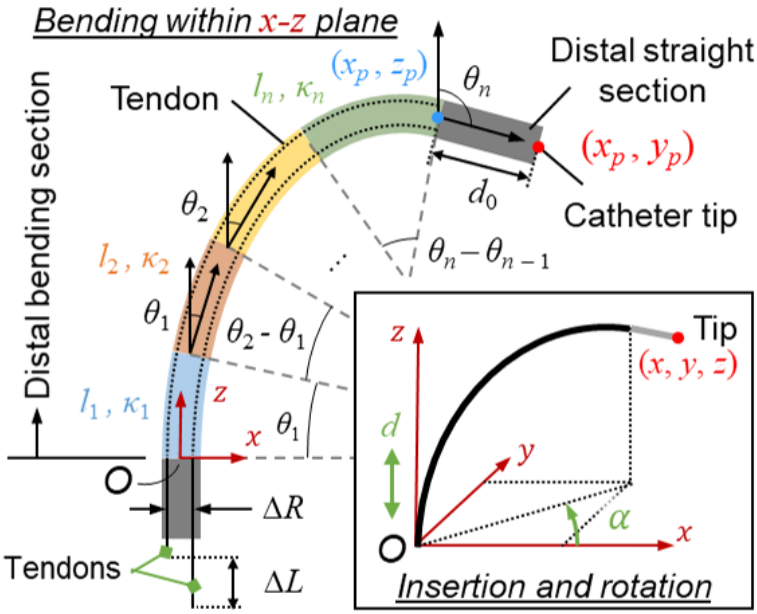


Fig. 5.15: Configuration of the catheter approximated by several curve segments. The curvature of i^{th} segment can be represented by κ_i. A straight section is added at the distal end on top of the PCC segments, which is the rigid tip of catheter.

The catheter deformation is determined by the relative position of two tendons. Here we represent the distance between two tendons as ΔR. The tendons are pulled and pushed by the knob on the catheter handle, whose radius is represented by r_{cam} and steering angle by ϕ (i.e., the rotation angle of steering knob [165]). A relationship based on the representation of tip bending angle θ_n can be obtained as

$$\theta_n = \sum_{i=1}^{n} \kappa_i l_i = \frac{2 \cdot r_{cam} \sin\phi}{\Delta R}. \tag{5.11}$$

The axial displacement difference between two tendons is

$$\Delta L = 2 \cdot r_{cam} \sin\phi. \tag{5.12}$$

To this end, the mapping between configuration and task spaces is derived [15], i.e.

the relationship between catheter tip position and bending curvatures κ_i, $i = 1,2,\cdots,n$. However, for the bending DoF of robotic catheter, the mapping between actuation space and configuration space is still unknown, that is, the relationship between knob rotation ϕ and bending curvatures κ_i, $i = 1,2,\cdots,n$. I aim to find the mapping from ϕ to each κ_i, as well as resolve the un-modeled characteristics by a learning-based method. Therefore, the kinematics between actuation and task spaces could be established.

5.6.2 System Characterization with a Learning-based Method

As described in **Section 5.6.1**, the forward model would be established only in the condition that all the segment curvatures or their ratios are available. Define the curvature ratio as

$$\boldsymbol{k}_c = \begin{bmatrix} k_{c1} & k_{c2} & \cdots & k_{cn} \end{bmatrix}^{\mathrm{T}}, \sum_{i=1}^{n} k_{ci} = 1. \tag{5.13}$$

In the PCC model, ΔL from **Eq. 5.12** could also be divided into n elements and represented as a vector in the following form

$$\Delta \boldsymbol{l} = \begin{bmatrix} \Delta l_1 & \Delta l_2 & \cdots & \Delta l_n \end{bmatrix}^{\mathrm{T}}, \Delta L = \sum_{i=1}^{n} \Delta l_i, \tag{5.14}$$

where $\Delta \boldsymbol{l}$ is proportional to the curvature ratio $\boldsymbol{k}_c$. A learning-based method is adopted to obtain the curvature ratio among all the segments.

Aforementioned in **Section 5.5.1**, the multi-core FBG fiber can reconstruct the curves by approximately consecutive points. The 3D positions and the radii between adjacent points can both be obtained, represented by $\boldsymbol{p}_j^s \in \mathbb{R}^{3\times 3}$ ($j = 1,2,\ldots,M$) and r_j^s ($j = 1,2,\ldots,M-1$), respectively, where M is the total number of effective points in the reconstruction. Positions and radii at time step k are $\boldsymbol{p}_j^s(k)$ and $r_j^s(k)$, respectively. Corresponding to the segmentation in PCC model, each segment's curvature $\kappa_i(k)$, $i = 1,2,\ldots,n$ (**Section 5.6.1**) can be obtained as the reciprocal of the average radius. The knob steering angle at time step k is $\phi(k)$, which drives the catheter tendon displacement. The curvatures from FBG shape reconstruction can be regarded as ground truth, and used to train the mapping between the steering angle $\phi(k)$ and the curvature ratio $\boldsymbol{k}_c$.

Besides the multi-segment curvatures, the hysteresis resulting from friction and/or tendon looseness can also be considered. The backlash will appear particularly when changing bending direction (i.e. steering direction of the knob). Differences between the analytical model and actual bending performance can be compensated by imposing

additional input l_c to the ΔL during the robot manipulation.

Therefore, the actual relative position after the learning-based compensation can be calculated by

$$\Delta \boldsymbol{l} = \boldsymbol{k}_c \cdot (\Delta L + l_c). \tag{5.15}$$

The parameters to be trained are $\boldsymbol{k}_c$ in **Eq. 5.13** and l_c, corresponding to the specific knob steering angle ϕ.

Training of motion mapping: In the pre-training procedure, the radii r_j^s, $j = 1,2,\ldots,M-1$ at each time step has to be collected, as well as the motor's position $u(t)$ that is correlated to the knob steering angle $\phi(t)$. To cover the whole bending workspace, the knob is rotated with a predefined actuation sequence

$$\boldsymbol{U} = \begin{bmatrix} u(1) & u(2) & \cdots & u(N) \end{bmatrix}, \tag{5.16}$$

where N is the sampling number. A value $s(t)$ is introduced to represent the actuation direction change between current and previous steps, which can be derived as:

$$s(t) = \text{sign}(u(t) - u(t-1)), t = 1,2,\ldots,N, \tag{5.17}$$

where the function $\text{sign}(\cdot)$ is to differentiate positive and negative values. The sequence of curvatures could be obtained as:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}(1) & \boldsymbol{c}(2) & \cdots & \boldsymbol{c}(N) \end{bmatrix}, \tag{5.18}$$

where $\boldsymbol{c}(t) = \begin{bmatrix} \kappa_1(t) & \kappa_2(t) & \cdots & \kappa_n(t) \end{bmatrix}^{\mathrm{T}}$ and $\kappa_i(t)$ is the curvature of the i^{th} segment in accord with **Section 5.6.1**. The curvature ratio $\boldsymbol{k}_c$ in **Eq. 5.13** could be obtained by normalizing $\boldsymbol{c}(t)$.

The parameter l_c could be derived based on the tip bending angle θ_n, which is independent of $\Delta \boldsymbol{l}$ and could be calculated from ΔL. For each actuation input, the actual tip bending angle θ_A can be obtained from the FBGs, while the predicted tip bending angle θ_P can be calculated from the forward PCC model. Thus, a compensation-related parameter f_c is available for training:

$$f_c = \exp(\Delta R \cdot (\theta_A - \theta_p)). \tag{5.19}$$

The input of training data is

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}(1) & \boldsymbol{v}(2) & \cdots & \boldsymbol{v}(N) \end{bmatrix}, \tag{5.20}$$

where

$$\boldsymbol{v}(t) = \begin{bmatrix} u(t) & s(t) \end{bmatrix}^{\mathrm{T}}, t = 1,2,\ldots,N, \tag{5.21}$$

The output of training data is

$$\boldsymbol{O} = \begin{bmatrix} \boldsymbol{o}(1) & \boldsymbol{o}(2) & \cdots & \boldsymbol{o}(N) \end{bmatrix}, \tag{5.22}$$

where

$$\boldsymbol{o}(t) = \begin{bmatrix} \boldsymbol{k}_c(t)^{\mathrm{T}} & f_c(t) \end{bmatrix}^{\mathrm{T}}, t = 1,2,\ldots,N. \tag{5.23}$$

Using the feedforward neural network in the deep learning toolbox of MATLAB, with $\boldsymbol{U}$ as input and $\boldsymbol{O}$ as output, we can train a mapping as

$$\boldsymbol{o}(t) = f(\boldsymbol{v}(t)),\ t = 1,2,\ldots,N. \tag{5.24}$$

Prediction of motion mapping: During robot motion, the curvature ratio $\boldsymbol{k}_c(k)$ among different segments and the compensation-related $l_c(k)$ at the k^{th} step can be calculated by

$$\begin{bmatrix} \boldsymbol{k}_c(k)^{\mathrm{T}} & f_c(k) \end{bmatrix} = f(u(k),\ s(k)),\ \ k = 1,2,\ldots. \tag{5.25}$$

The enhanced estimation of relative positions between two tendons $\Delta\boldsymbol{l}$ can be calculated by **Eq. 5.14**, where $l_c(k)$ can be derived as

$$l_c(k) = \ln(f_c(k)). \tag{5.26}$$

The predicted $\boldsymbol{k}_c$ and l_c would link up ϕ and κ_i, $i = 1,2,\ldots,n$, completing the mapping between actuation and configuration spaces of bending DoF. Combined with **Eq. 5.10**, the complete forward kinematics are available for robotic control

$$\dot{\boldsymbol{p}} = \boldsymbol{J}\begin{bmatrix} \dot{\alpha} & \dot{\phi} & \dot{d} \end{bmatrix}^{\mathrm{T}}, \tag{5.27}$$

where $\boldsymbol{J}$ is the Jacobian matrix.

5.6.3 Autonomous Motion Control and Ablation

For the task of autonomous ablation, targets in the model coordinates $\{\boldsymbol{C}\}$ are predefined around the pulmonary vein ostium in the simulator. The real-time tip position $\boldsymbol{p}_w$ can be provided by the shape tracking system in the world coordinates $\{\boldsymbol{W}\}$. $\boldsymbol{p}_c$ and $\boldsymbol{p}_w$ are transformed using the rotation matrix $\boldsymbol{R}_w^c$ and translation vector calculated by registration approach.

During each targeting cycle, the target can be appointed either automatically in sequence or selected by the operator. The error between current tip position $\boldsymbol{p}_c$ and desired position $\boldsymbol{p}^*$ will be calculated and normalized with constant step size as $\Delta\boldsymbol{p}_c$. The robot actuation is derived by multiplying inverse Jacobian matrix and $\Delta\boldsymbol{p}_c$. A tolerance distance from the catheter tip would be set for each target. The ablation process would be automatically triggered by simultaneously satisfying the two conditions: **i)** tissue contact is detected with an appropriate impedance for ablation (e.g. $< 200\ \Omega$ [183]); **ii)** targeting error is within the tolerance distance. During the ablation process, the robot would keep still for RF ablation on the tissue. The control block diagram in **Fig. 5.16** shows the key processing components including the kinematic control and autonomous ablation.

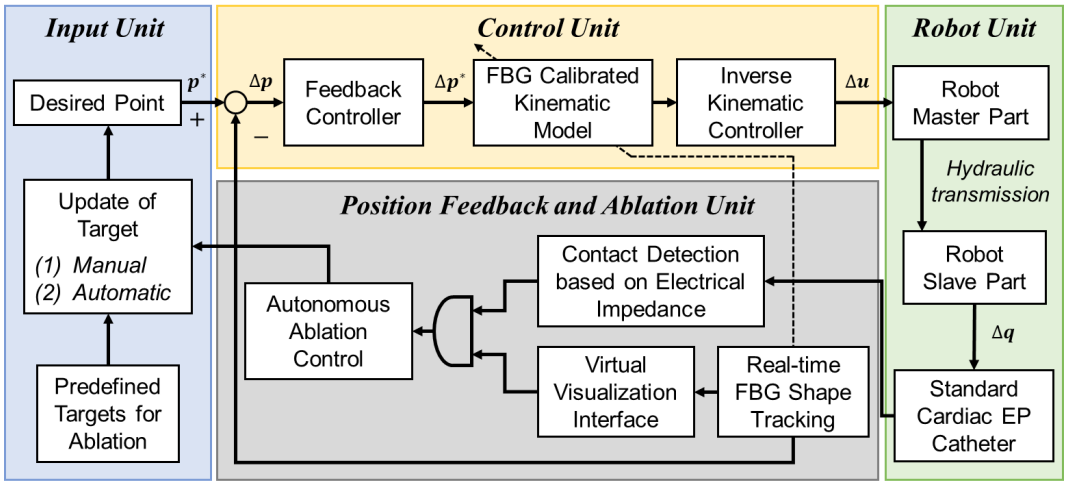


Fig. 5.16: Control architecture of the autonomous ablation procedure. Parameters of the PCC model can be automatically tuned by a learning-based algorithm, which enables accurate positional control of the catheter end-effector. The 3D curvature of catheter can be obtained in real-time from the FBG feedback. RF ablation will be triggered when the catheter tip reaches the target and contacts tissue.

5.7 Performance Evaluation of Shape Tracking and Autonomous Control

The proposed shape sensing method and control algorithm are evaluated on an MR-safe robotic catheter system [159, 184] (**Fig. 5.3**). The shape sensing performance will be analyzed first, based on the accuracy tests of passive and active catheter deformations. For the controller, point-to-point tracking and path following tasks were validated. In the path following task, a predefined series of 3D points (M in total) with incremental indices ($1 \sim M$) constructs a spatial path. For each actuation step, the target for catheter tip will be one point (index k) on this path, and the target point for the next actuation step will switch to the one with index $k+1$. As the target path is a closed contour, when the target index exceeds the total number of targets, the target point will reset to the point with index 1.

5.7.1 MR Safe Robotic Catheter Platform

The MR Safe catheterization robot is capable of teleoperating the catheter inside MRI scanner bore. The robot is designed for the teleoperation under intra-operative (intra-op) MRI, enabling to accommodate standard cardiac EP catheters. The robot features a master-slave hydraulic transmission system in order to provide high-accuracy and low-latency actuation. The master units were driven by electric motors located in the control room. Actuation from master units can be transmitted to the slave units through long (e.g., 10 m) hydraulic pipelines. The robot can manipulate a standard EP catheter (Thermocool® Bi-Directional Catheter, Biosense Webster Inc.) in 3 DoFs, namely bending, insertion and rotation. Incorporating our previously designed three-cylinder actuation units [185] for insertion and rotation DoFs, it enables an extensive motion range for catheter advancement (340 mm) and rotation ($\pm$360), as well as high-fidelity catheter manipulation. Such robotic actuation performance would enable feedback control implementation for effective autonomous catheter manipulation, thereby delicate cardiac EP catheterization tasks, such as EAM and RF ablation.

5.7.2 Shape Sensing Performance of the Multi-core FBG

To validate the FBG shape sensing performance when integrated with the cardiac catheter, a fixed curvature test was conducted on the two symmetric bending curvature fixtures, as

shown in **Fig. 5.17a**. The fiber can be maintained in consistent performance after multiple uses. Calibration for the fiber only needs to be done once, tuning OFDR parameters such as update rate, center wavelength, and resolution.

The mechanical fixtures were 3D-printed with constant-curvature grooves of 2.7-mm width, which is approximately equal to the outer diameter of EP catheter. They are used to guide the catheter bending in planar direction. The bending angles of the curved section were from 0° to 105° on each side. The length of the constant-curvature arc section was 55 mm. The integration of FBG and cardiac catheter followed the method in **Section 5.5.1**. The shape of 63.4-mm distal catheter section was reconstructed by three cores in the fiber, with each core containing 21 FBG segments. In this curvature test, only 2D fixtures were used because the tendon-driven catheter itself could only enable planar bending; and in our proposed modeling method, only 2D piecewise curvature relations are in demand to be learned. The 3D curvature measurements are collected for bending plan registration. For each measurement, the location of first FBG segment was aligned to the start position of the constant-curvature arcs. The reconstructed 3D curvature was obtained by taking average of 100 consecutive data captured for the entire sensing section.

The reconstructed catheter shapes are plotted in **Fig. 5.17b**, with the position of the first FBG section set as the origin $(0,0)$. Curves for the ground truths are also plotted, following the same shape of corresponding grooves. The overall average angular error of all FBG sensors was 2.33°. And the average tip bending angle was 2.35° for all curvatures. **Fig. 5.17c** illustrates the average errors of the tip bending angle in different groups of predefined bending angle. Other than the sensor error of FBGs, the misalignment between the reconstructed shape and ground truth could be partially attributed to the large gap between the FBG fiber ($\varnothing$0.2 mm) and the water channel ($\varnothing$1 mm). In future manufacture, the fiber could be put through a sheath with slightly larger diameter to minimize such interspace, thus to improve the overall sensing accuracy. The sheath could also be integrated into the wire channel of catheter to free up the water channel for irrigation function.

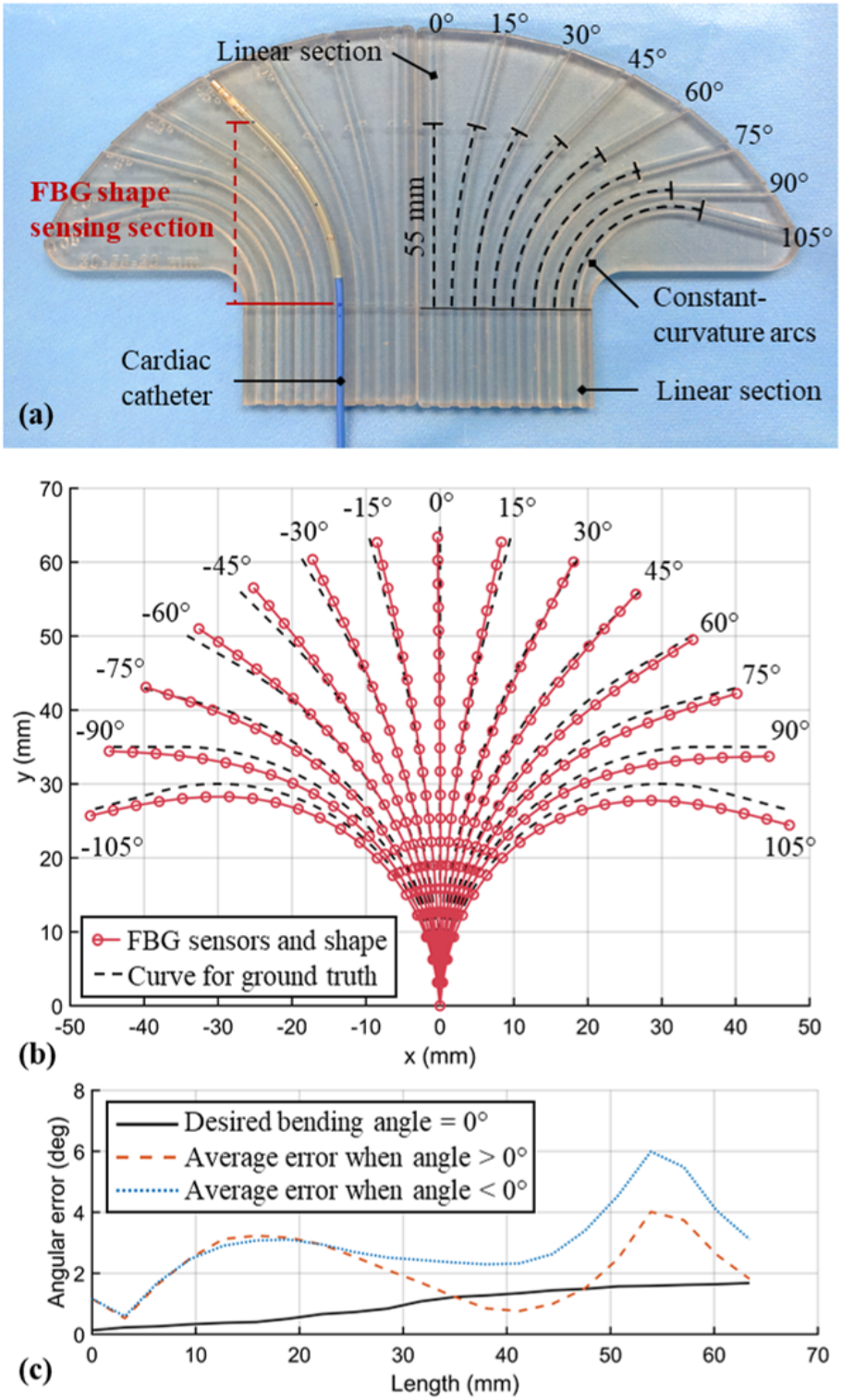


Fig. 5.17: **(a)** Bending curvature fixtures used to evaluate the shape sensing performance of the multi-core FBG fiber. Seven curvatures were included at each side of bending, with an absolute value from 0° to 105°; **(b)** Reconstructed catheter shapes compared with the ground-truth curves. The starting positions are all aligned at $(0,0)$ in the plot; **(c)** Bending-angle errors along the shape sensing section.

5.7.3 Shape Tracking under Active Bending

The shape tracking performance was further evaluated with active catheter manipulation. The integration method of FBG fiber with the catheter remained the same as in **Section 5.7.2**. A 6-DoF EM positional sensor ($\varnothing 0.8 \times 9$ mm, NDI Medical Aurora) was attached to the catheter at the location of first FBG sensor. Before data collection, the coordinate frames of the reconstructed shape and EM tracking system were aligned following the procedure in **Section 5.5.2**. During the test, the cardiac catheter was actuated by the robotic platform for bending in two directions. The shape and positional data of the catheter were recorded under seven knob steering angles, which are summarized in the table in **Fig. 5.18a**. For each input angle, the 3D curvature for ground truth was represented by 25 evenly distributed points, which were captured along the catheter by an EM tracking probe (NDI Medical Aurora).

The diagrams in **Fig. 5.18a** depict the shape of catheter bending section under the seven knob steering angles. To obtain the catheter tip position, a 14-mm straight section was added to the tip of reconstructed 3D curvature based on the orientation of last FBG segment. Simulated shapes deduced by the learning-based model in **Section 5.6.2** are also overlaid in the diagram, predicting similar shapes and positions to the measured results. The side view of the shape tracking curves is plotted in **Fig. 5.18a**, indicating that the catheter used for the experiment had a nearly planar bending behavior. The reconstructed shapes from FBG fiber can be consistent with the real shape, implying there is no significant twist between the FBG fiber and the catheter. The spatial positional errors of shape tracking for the bending section are shown in **Fig. 5.18b**. The average positional error of 21 FBG sensing segments was 0.63 mm, and 1.64 mm at the tip. The largest tip error was 2.46 mm for A7, which was within the tolerance of targeting error ($\sim$5 mm [186]) in EP procedure.

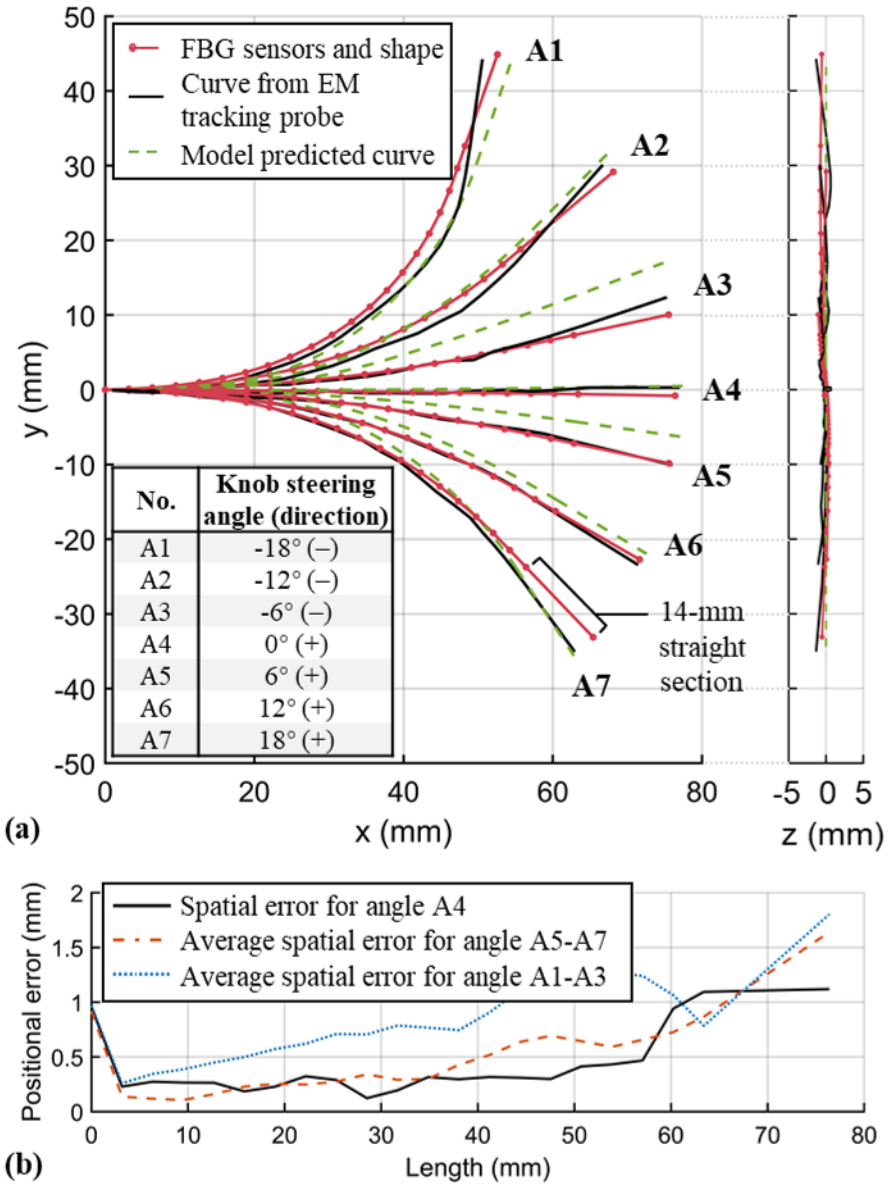


Fig. 5.18: **(a)** Top and side views of catheter shapes obtained from FBG shape sensing (red), EM tracking system (black), as well as model prediction (green). The catheter was actuated by the robotic platform with seven knob steering angles, which were also used for model input; **(b)** Spatial differences along the shape sensing section by comparing curves obtained from FBGs and the EM tracking system.

5.7.4 Learning-based PCC vs CC

The proposed learning-based PCC model was compared to the conventional CC model by simulation. The structure-related parameters of the two models (e.g. length, diameter) were set to be identical. In the PCC model, we divided the bendable section into three segments (i.e. $n = 3$ in **Section 5.6.1**) for ease of calculation and analysis. The knob steering angles ϕ in a range of -20° to 20° were input to both models. In the proposed learning-based PCC model, the actuation direction $s(t)$ is also considered as an input, as introduced in **Eq. 5.17**.

Fig. 5.19a illustrates the simulated catheter planar curves with five inputs of knob steering angle, A1-A5. For the learning-based PCC model, each knob angle could generate two curves in two actuation directions, namely positive (+) and negative (–). The variation trends of curvature and tip bending angle are depicted in **Fig. 5.19b-c** respectively, w.r.t. the knob steering angles. The CC model exhibits a linear increase with a larger magnitude in both cases. As for the learning-based PCC model, a nonlinear relation between the knob input and curvatures of three segments could be predicted, as well as the tip bending angle. These predictions conform with the observed bending behavior of commercial cardiac catheters, which could be attributed to the tendon-sheath friction, manufacturing tolerance and non-ideal material conditions.

The learning-based PCC model could also reveal the bending hysteresis of catheter by involving the actuation directions. As illustrated in **Fig. 5.19c**, the tip bending angle would follow the solid (/dashed) curve when the knob steering angle is increasing (/decreasing), corresponding to the + (/–) actuation direction. The transition between the two curves occurs when the actuation direction changes, which were at $\pm 20°$ in this case. The tip angle would remain unchanged while the input angle increases/decreases, which could represent the bending hysteresis of catheter. The hysteresis can be predicted for every knob input within the actuation range, leading to a more accurate mapping between the catheter task space and actuation space.

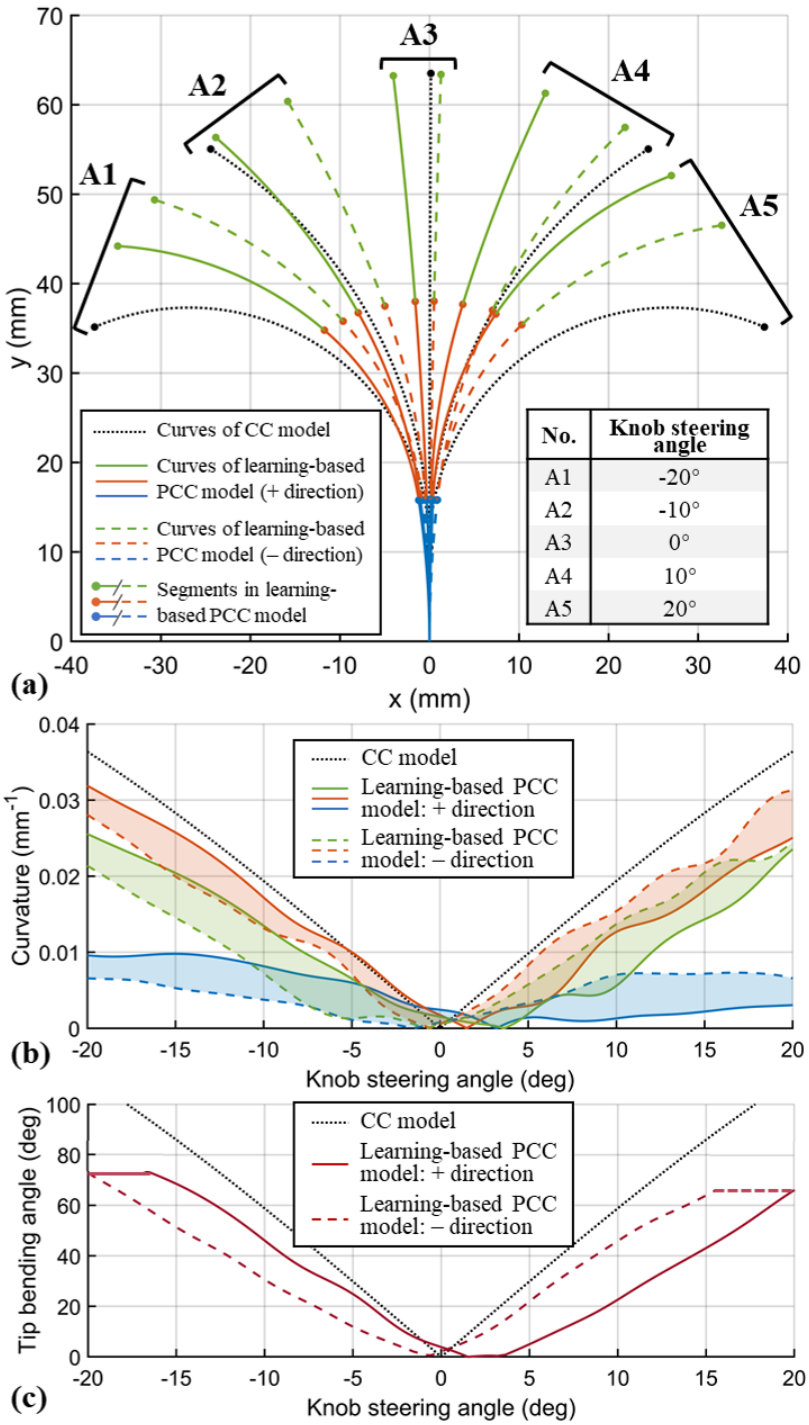


Fig. 5.19: **(a)** Simulated catheter bending curves by the CC model and the learning-based PCC model with five knob steering angles (I1-I5); **(b)-(c)** Diagrams showing the curvature and tip bending angle variation trends for CC model and learning-based PCC model. The learning-based PCC model can predict the actuation hysteresis by considering the actuation direction.

5.7.5 Autonomous Targeting

An autonomous targeting experiment was carried out by implementing the proposed shape tracking and learning-based modeling on the robotic platform. Five targets (T1 to T5 in **Fig. 5.20a**), which were in a volume of $78.4\times75.7\times46.3$ mm^3, were chosen within the workspace of catheter tip. The EP catheter passed through a Polytetrafluoroethylene (PTFE) pipeline with 0.8-m length and 4-mm inner diameter, which started at the robotic platform and ended before the catheter bendable section. The catheter's bending section could move freely in the workspace. The PTFE pipeline simulated a path along the vessel to heart chamber. During the task, the catheter was automatically manipulated by the robotic platform to reach the targets in a sequence from T1 to T5. The tip position and catheter shape were recorded and fed back by the proposed shape tracking method. The pre-trained catheter model introduced in **Section 5.7.4** was used to control the robot.

The footprint of catheter tip during autonomous targeting is shown in **Fig. 5.20a**, with the trajectories represented by different colors. The red curves with dots indicate the reconstructed catheter shape and position when its tip reached the targets. The deviations from the tip to targeting points during the five stages are illustrated in **Fig. 5.20b**. The average duration towards each target was 16.9 s, with the maximum duration of 25.8 s (T3). The longer duration towards T3 could be attributed to the larger targeting distance (85.8 mm), as the tip maximum speed was limited in the control algorithm to ensure a safe catheter manipulation. The results indicate a fast and efficient autonomous targeting performance of the robotic system incorporating with the shape tracking and control systems. These five targets were representatively chosen to illustrate different configurations of the reconstructed catheter, and to prove the volume that targets located. In our lab-based experiment, more randomly defined points had been tested, and their response time (duration from the initial position to the target) was mostly related to the distance between starting point and the target, as shown in **Fig. 5.20b**.

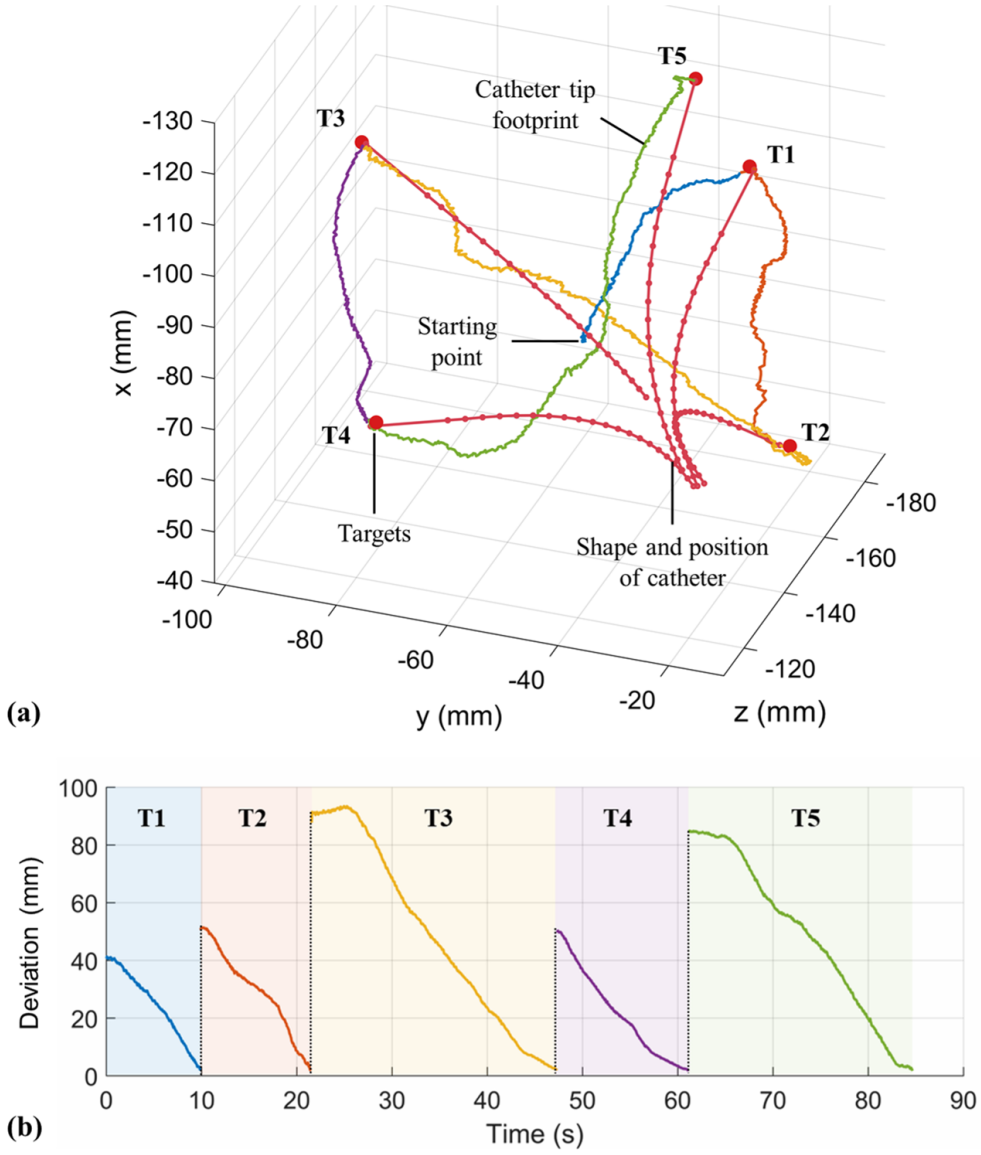


Fig. 5.20: **(a)** Diagrams showing the catheter tip trajectory towards the five targets during the autonomous targeting. The reconstructed catheter shape and position are plotted for showcasing the instances when the tip reached the targets; **(b)** Deviation from the tip to the targets during the five stages.

5.7.6 Path Following

To further investigate the overall feedback control performance of the robotic catheter system, a path following task was conducted. The static reference path was pre-defined in the shape of a sideways figure *eight* ("∞"), with a dimension of 60×30 mm. The learning-based PCC model was implemented on the robot. The catheter tip was autonomously controlled to trace the reference path periodically, with the duration of each loop set to 60 seconds.

The diagram in **Fig. 5.21a** depicts the tip footprint recorded by the shape tracking system over 5 cycles. The color gradient indicates the tip deviation from the desired path. **Fig. 5.21b** illustrates the corresponding tip deviation over the 5 tracing cycles. A mean deviation of 0.62 mm could be found during the trajectory tracking after the approaching stage. The controller maintained a smooth and close tracking along the desired path. Several sections of the footprint had larger deviations (max. 2.34 mm), which may mainly be due to the backlash and nonlinearity of the catheter manipulation and robotic actuation. The reconstructed curves of the catheter are also overlaid on the diagrams as red curves with dots, representing the instantaneous shape and position of the bending section at 12s, 24s, 36s, 48s and 50s during the 1st cycle. A close tracing between the tip footprint and desired path could be clearly observed in **Fig. 5.21c**, together with the reconstructed catheter shapes in various bending curvatures.

As a comparison, the CC model was implemented on the robot for a path flowing task as well. All the initial parameters of the CC model and controller remained the same as the proposed model, with the only difference in the modeling method and absence of training process. The footprint recorded by shape tracking system over 5 cycles is shown in **Fig. 5.22a**. The deviation from tip to desired path is illustrated in **Fig. 5.22b**, with an average value of 1.23 mm and a maximum value of 4.02 mm. It can be seen that the path controlled by CC model has a larger deviation, which could be attributed to the overestimation of bending by the model. In contrast, the learning-based PCC model could reflect the catheter configuration characteristics more precisely, thus offering a more accurate mapping between the robotic actuation and the tip motion.

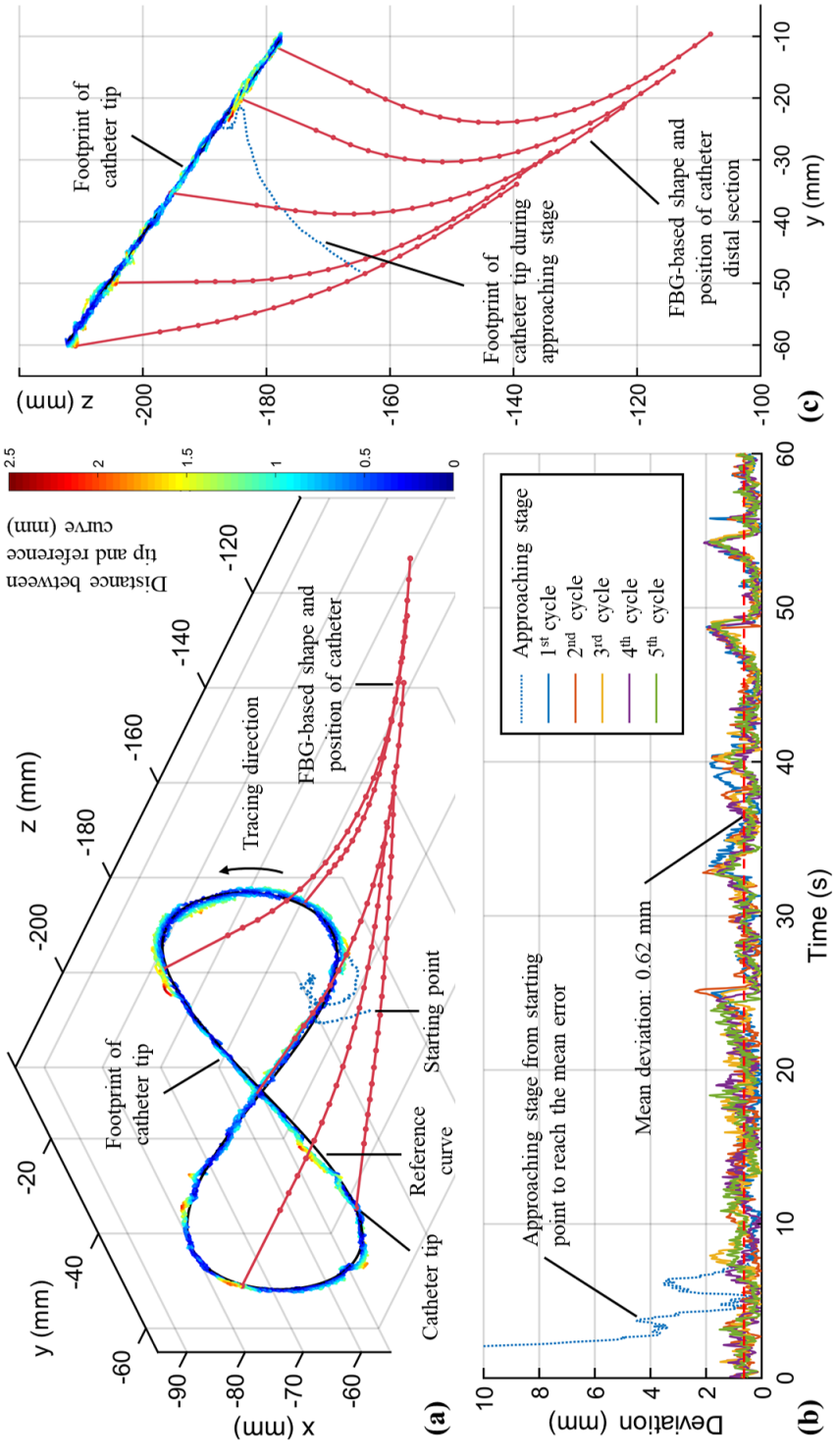


Fig. 5.21: **(a)** The rendering color (cool to warm) represents the tip deviation (small to large) from the desired path. The reconstructed catheter shapes and positions were obtained from the proposed shape tracking approach; **(b)** Deviation of the catheter tip from the desired path during 5 cycles. The catheter was controlled to quickly trace the path during the approaching stage; **(c)** Top view showing the catheter bending shapes at the five instances.

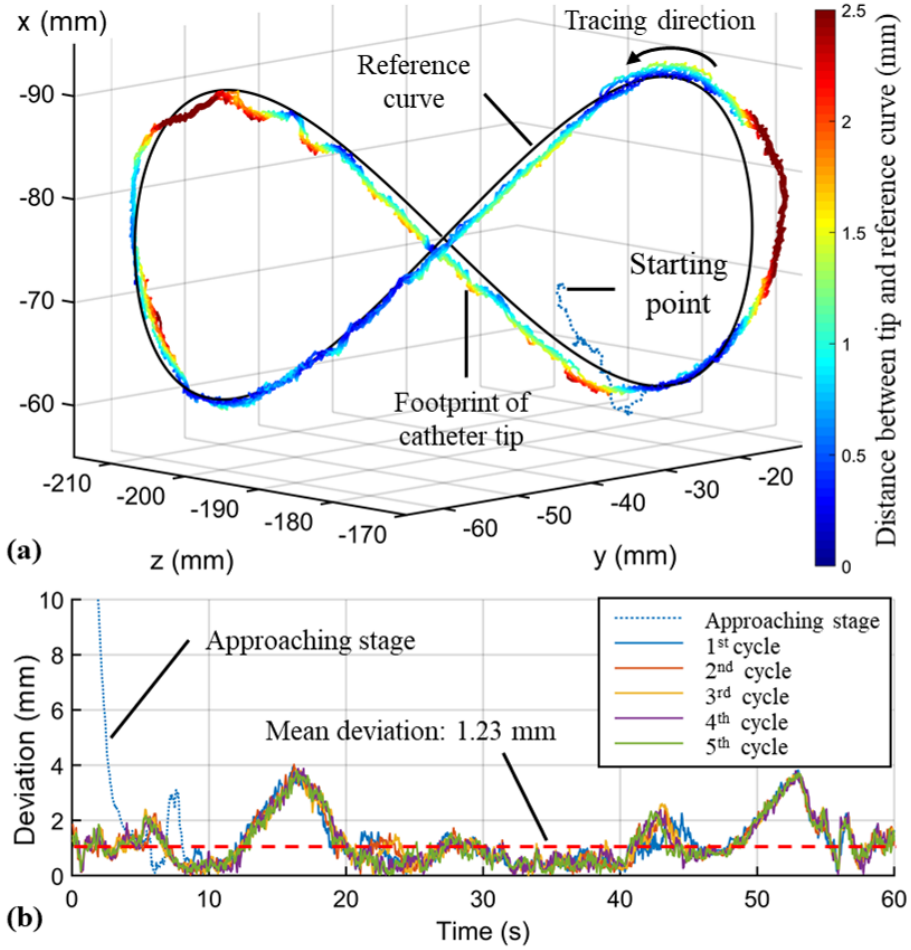


Fig. 5.22: **(a)** Catheter tip footprint under autonomous control with the CC model. Large deviation could be found at segments with red color; **(b)** Tip deviation from the desired path. Slow and delayed retracement could be observed at several segments due to the inaccurate estimation of kinematic mapping done by the CC model.

5.7.7 LA Simulator with Pulsatile-like Flow and *Ex-Vivo* Tissue

To perform a simulated PVI task, an LA simulator was designed based on patient-specific imaging data (**Fig. 5.23a**). The simulator was molded by silicone [187] in attempt to mimic the LA tissues consisting of myocardium and endocardium. A semi-rigid sheath was fixed at the puncture on simulator wall, simulating a path along the femoral vein, IVC, RA to LA. Simulated pulsatile liquid flow can be generated by a water pump through pipelines to the LA simulator, forming a water circulation system. The flow direction is indicated by yellow arrows in **Fig. 5.23a**, which follows the LA blood flow from pulmonary veins to left ventricle. The LA pressure and resultant motion during EP procedure could be simulated by controlling the magnitude and frequency of flow, which could be gated with the patient-specific electrocardiogram (ECG) signal. **Fig. 5.23b** shows the simulated liquid pressure compared to the human ECG and LA pressure [14, 188]. **Fig. 5.23c** shows the pressure curves at simulated fast and slow pulsatile rates. The rates could be adjusted within the range, covering the common patient heartbeat rate during EP procedure. The simulated LA pressure range was adjusted to 10$\sim$22 mmHg, referring to the record in the research for LA systolic function assessment [14].

To conduct the *ex-vivo* tissue ablation, a slice of swine tissue with 3-mm thickness was attached on the simulator inner surface around PVO. In EP procedure, a neutral electrode is usually attached on patient's body to form a close electric circuit containing catheter and RF generator, in order to monitor tissue resistance and conduct RF ablation. In this setup with LA simulator, the *ex-vivo* tissue was linked to a neutral electrode through electrical wires, which was connected to an RF generator (Biosense Webster Stockert 70). The control program was gated with the generator to automatically trigger the RF ablation when the catheter tip reached the target range and contacted the tissue.

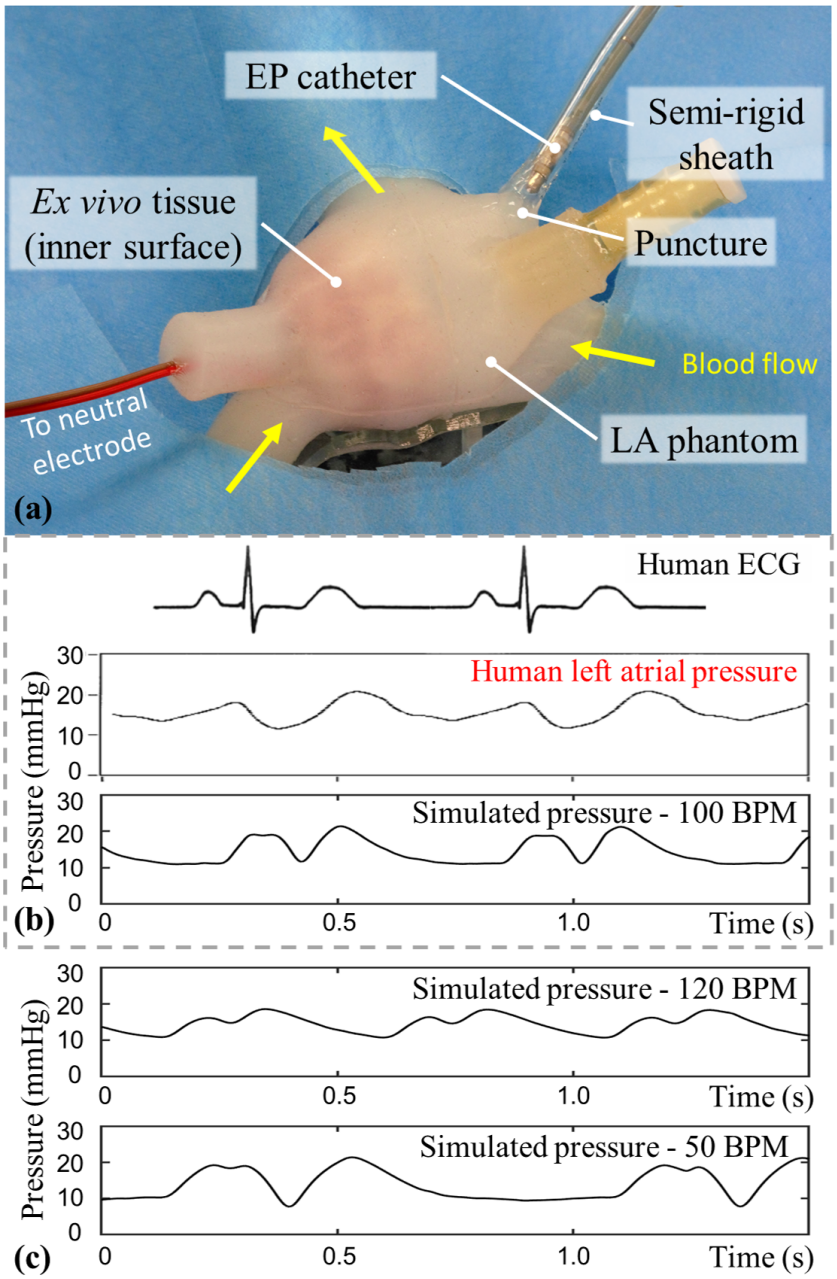


Fig. 5.23: **(a)** Left atrial (LA) simulator filled with liquid, of which the simulated pulsatile flow was generated by a hydraulic pump. A slice of swine tissue was attached on the inner surface at the target ablation area (around PVO); **(b)** Simulated liquid pressure compared to the human ECG and LA pressure; **(c)** Pressures in LA simulator with fast and slow rates. The simulated pulsatile rate could be adjusted within the range. **Image Source**: [14]

5.7.8 *Ex-vivo* Tissue Ablation with Simulated Pulsatile Flow

A simulated PVI task with *ex-vivo* tissue ablation was conducted to assess the overall performance of the shape tracking and learning-based modeling in control (**Section 5.5**-**5.6**). The experiment was performed inside the LA simulator with pulsatile-like liquid flow. During the task, the catheter was manipulated by the robot to reach the predefined ablation targets on the simulator (**Fig. 5.24a**). A 2-mm tolerance was set for each target. The ablation process was automatically triggered by the RF generator (Biosense Webster Stockert 70) when the catheter tip reached the target region and had tissue contact with impedance $<$200 Ω, as introduced in **Section 5.6.3**. After conducting ablation at each target point, the robot was reset to the initial position for another targeting process. This reset aimed to separate the catheter tip with the ablated tissue, as well as avoid the collision with the simulator wall during the next targeting.

As shown in **Fig. 5.24a**, the red spheres represent the predefined ablation targets, which have a total number of 15 and construct a volume of $20.9\times23.0\times11.1$ mm^3. These targets form a yellow line corresponding to the circumferential lesion targeted path nearby the PVO. The footprint (blue line) indicates the targeting motions towards the lesions. 12 out of the 15 (80.0%) lesion targets were successfully reached by the catheter tip to conduct RF ablation. For the unsuccessful cases, the catheter was blocked by the tissue on the targeting path. **Fig. 5.24b** shows the actual lesion points on the *ex-vivo* tissue. The average time taken to complete each targeting process was 7.76 s, with the average traveling distance of 43.4 mm. This ablation task was performed twice with two pieces of *ex-vivo* tissue, and the aforementioned results were analyzed from one of the experiments. All successfully-ablated points were repeatedly tested for at least two times, so as to guarantee the proper access to the target using the proposed control method. The results demonstrate the proposed sensing and feedback control methods enable autonomous RF ablation even in a more dynamic environment with pulsatile-like liquid flow. In **Fig. 5.24a**, three red curves with dots indicate examples of the reconstructed catheter shape and position when the tip reached corresponding target. Other than providing feedback control for the robotic system, the catheter shape tracking can also offer informative visual feedback of the current catheter shape and tip orientation. This additional information could help surgeons keep tracking on the targeting process and evaluate the ablation result, thus improving the treatment safety and effectiveness.

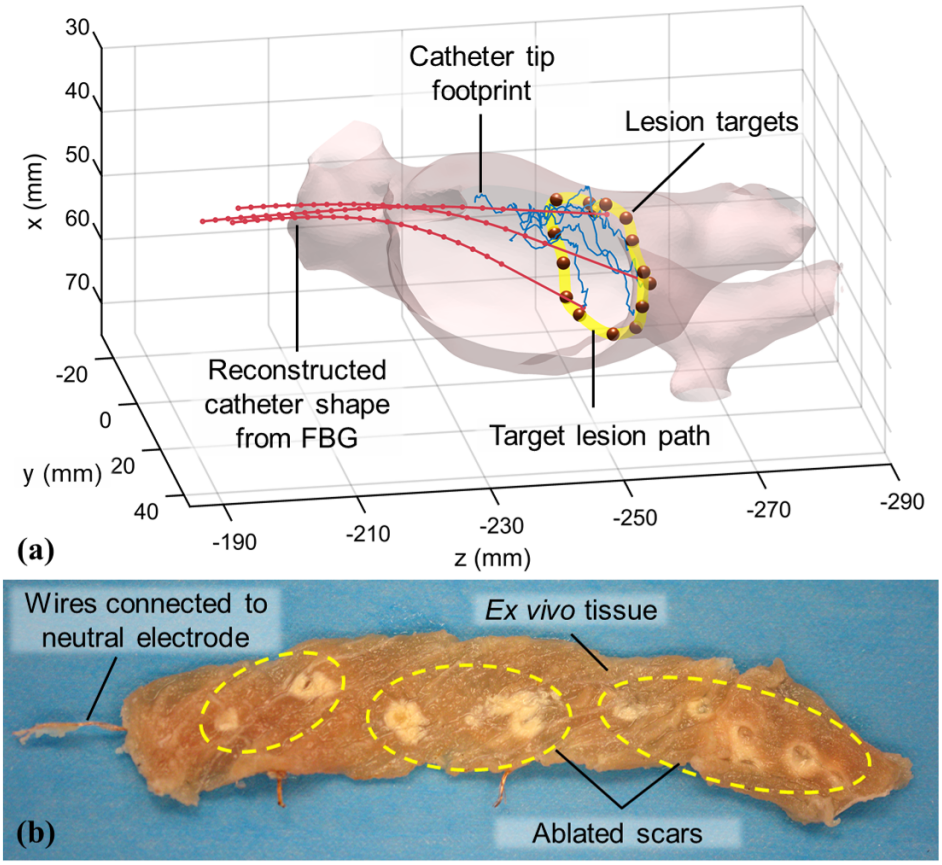


Fig. 5.24: Results of the PVI task conducted in the LA simulator. A standard EP catheter was autonomously controlled to reach a series of targets around the simulated PVO. **(a)** Front view showing the catheter tip footprint, the targeted ablation points (red sphere), and the desired lesion path (yellow). The footprint (blue line) indicates three selective tip trajectories towards the targets; **(b)** Tissue with ablated points.

5.8 Conclusion

In this chapter, a teleoperated catheterization platform is introduced and validated. For the simulated RF ablation task, the experimental setup comprised a customized catheterization robot and devices to close the loop teleoperated control. Virtual endoscopic view was designed to provide an intuitive visualization of the ablation procedure. Kinematic model-based and model-free methods were applied for comparison. Subject test task imitating the cardiac EP ablation was conducted for quantitative evaluation. Accuracy indices (e.g., mean of the closest distance between catheter tip and the closest target during ablation) and efficiency indices (e.g., proportion of missed lesion target) were adopted. It could be seen that for in a static LA simulator, the model-free control method had advantages in accuracy (19.1% improvement in tip-to-target ablation distance) and efficiency (35.8% reduction in the missed-target proportion and 46.2% reduction in the number of continuously missed targets). As kinematic model-based methods would theoretically embrace reliable convergence, a learning-based modeling method was investigated in a more complicated setup.

A shape tracking and feedback control method for slim continuum manipulators is also presented. Information from multi-core FBG fiber and positional tracking coils was integrated to enable catheter shape estimation under MRI, which was subsequently incorporated with an MR safe catheter robot system. A learning-based modeling method is proposed for continuum manipulators, especially instruments like cardiac catheter, with the FBG shape tracking used for system characterization. Comparing with conventional CC models for continuum robots, the proposed modeling method can resolve the modeling uncertainties from heuristic parameter tuning and tendon backlash. The shape sensing using multi-core FBG fiber could achieve 2.33° average error for each sensing segment, and 1.52 mm positional accuracy for the catheter tip. Autonomous targeting of five points within the robot workspace showed effective convergence rates (average 16.9 s). Performance of the proposed learning-based PCC and conventional CC models were compared in a path following task, where the average deviations are 0.62 mm (max. deviation 2.34 mm) and 1.23 mm (max. deviation 4.02 mm), respectively. A simulated PVI task with ex-vivo tissue ablation was conducted to demonstrate the overall performance of the shape tracking and controller.

In our future work, pre-clinical animal trials will be conducted to validate the robotic catheter system for MRI-guided EP procedure. RF ablation will be conducted on a live

porcine or ovine model with arrhythmia. The proposed shape tracking and controller will be adapted to commercial MR-conditional systems equipped with RF ablation system. The necrosis created by RF ablation, and its efficacy, will also be examined with post-mortem histology. The proposed catheter shape tracking method would provide sufficient feedback for visualization and robotic control under MRI, potentially reducing workload of the surgeon as well as post-procedural disease recurrence.

In addition, the proposed FBG-based shape tracking method could be implemented on other MR safe/conditional continuum manipulators for interventional procedures, such as urologic surgery, ophthalmic surgery and neurosurgery. The direct measurement and reconstruction of shape and position for continuum manipulators would reduce the computational cost of MRI, enabling high-performance feedback control and interactive instrument manipulation. It is believed that both safety and overall operational efficiency would be enhanced, with accurately and intra-operatively updated morphological information of interventional instruments.

Chapter 6

Optical-FBG-Enhanced Visual Servoing

6.1 Introduction

In the feature/object tracking of eye-in-hand visual servoing as conducted in **Section 4.2** of **Chapter 4**, the robot has enabled accurate path following in the camera view with LEGO®-constructed scenes. I used a typical template matching algorithm [103] which enables visual feature recognition for motion estimation. However, camera feedback is sometimes unreliable due to sudden and unpredicted motions which may deteriorate the image quality [60]. 2D motion estimation relying only on image plane feedback is also easily affected by vision occlusion, blurring, or poor lighting. This will inevitably reduce the motion estimation accuracy. For the commonly-used template matching method, tracking performance greatly depends on the image quality. As soft robots are inherently compliant and adaptive in scenarios with high safety requirements (e.g. endoscopy and laparoscopy [119, 120]), enhancing feature tracking and motion estimation is essential for maintaining precise control, and therefore, efficiency during interventional operations.

FBGs, a type of high-frequency flexible strain sensor, can be used as an assistant device for soft robot control. I propose a method to enhance motion estimation in soft robotic visual servoing (**Chapter 4**) by fusing the results from template matching and FBG wavelength shifts to achieve more accurate tracking in applications such as MIP. In this chapter, path following performance was validated in a simulated laparoscopic scene and LEGO®-constructed scene, demonstrating significant improvement to feature tracking and

robot motion, even under external forces. Similarly, in **Chapter 4**, the integrated camera enables 6D pose estimation of the robot end-effector. However, estimation methods only based on camera could not handle cases under deficient feature correspondences or sudden movement. Fusion of camera feedback and strain measurement can facilitate robust pose estimation of robotic endoscope end-effector.

Visual sensing part plays an important role in the robot performance. In this Chapter, I propose a 2D motion sensing modality to enhance the visual servo performance under feature-deficient and force-disturbed conditions. This work is the *first* to consider and solve the feature "drifting" problem in soft robot visual servoing. We equip a soft continuum robot with a monocular camera and a helically-wrapped FBG fiber along its body, thus fusing both visual and strain sensing data. The learning-based method is inherently advantageous in soft robot sensing using optical fibers [189], circumventing modeling uncertainties in soft materials. As for the 6D pose estimation, I also improved the SLAM sensing accuracy by learning-based fusion utilizing FBG fiber. The estimation result can be further used as the sensing feedback on soft robot position control. The major contributions of this chapter are:

1. Design of a prototype endoscopic soft robot which is integrated with a helically wrapped FBG fiber. The learning-based motion and pose estimation approaches alleviates the need for precise FBG placement on the soft robot;

2. Development of **a)** an enhanced eye-in-hand tracking algorithm by incorporating camera-based 2D motion estimation with sparse strain data collected by single-core optical FBG fiber to improve the real-time tracking performance in dim/feature-lacking environments; and **b)** online learning-based pose estimation using this FBG fiber and sensing fusion with mono-camera SLAM;

3. Hybrid control combining model-based and learning-based methods on tracking tasks of soft continuum robots, without the need for accurate parameter tuning or data collection;

4. Experimental validation of **a)** the enhanced motion estimation method by path tracking in laparoscopic and LEGO®-constructed scenes even under disturbance, **b)** the proposed sensing fusion modality under poor visual conditions and **c)** the robust hybrid controller via target tracking tasks.

6.2 FBG Enhanced 2D Visual Servoing Control

The camera enables 2D motion estimation in the image plane while FBG fiber wrapped on the robot could also reflect the robot configuration and be trained to estimate the motion (**Fig. 6.1**). Fusing of camera and FBG sensing data will enable enhanced motion estimation.

6.2.1 Task Space Definition

The endoscopic camera mounted at the robot end-effector allows eye-in-hand visual servoing. The actuator input at time step k (at equilibrium state) is represented as $\boldsymbol{u}(k) \in \mathbb{U}^m$, where m denotes the dimension of actuation space. Similar as in **Section 4.2.1**, the robot deformation/bending induces corresponding differences of camera coordinate frames at time step k and $(k+1)$ (**Fig. 6.1b**). The task space is defined in the 2D camera frame (**Fig. 6.1c**), with the incremental 2D displacement denoted as $\Delta \boldsymbol{z}(k) \in \mathbb{R}^2$. The frame is always perpendicular to the robot tip normal. The control objective is to generate an actuation command $\Delta \boldsymbol{u}(k)$, achieving the desired movement $\Delta \boldsymbol{z}^*(k)$.

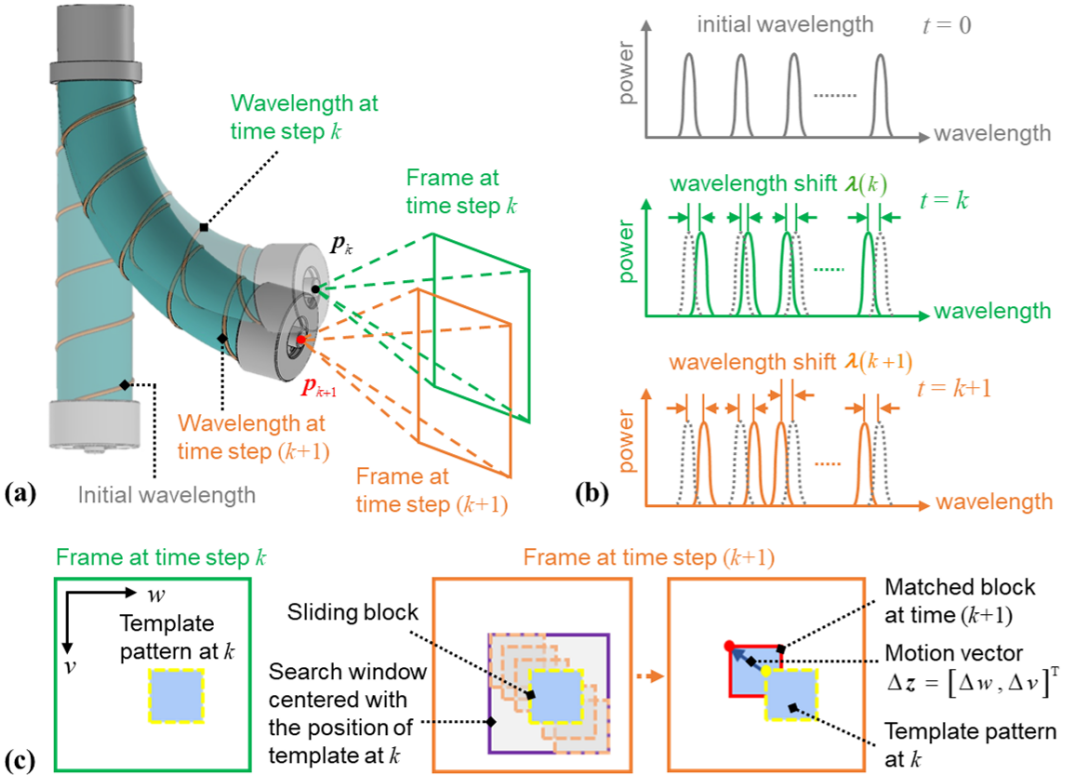


Fig. 6.1: **(a)** Example FBG wavelength shift changes from $\lambda(k)$ to $\lambda(k+1)$, **(b)** due to the corresponding robot deformation/bending, the difference of camera coordinate frames at k and $k+1$ can be seen; **(c)** Incremental motion in the image plane acquired based on the displacement from the template pattern (yellow dotted block) to the matched block (red) that is found by block sliding in the search window (purple block).

6.2.2 Learning-based Motion Estimation Combined with FBG

The image-based estimation method for 2D motion in image plane, that is template matching method, has been introduced in **Section 4.2.2** of **Chapter 4**. Although template matching has been widely used to estimate relative motions, the light and feature conditions will affect the estimation results to a great extent. The improvement on image-based algorithms could not overcome these intrinsic weaknesses. With the strain feedback of FBGs that reflects the change of robot configuration, a model combining the strain data and camera motion can be trained to assist the real-time camera-based motion estimation.

6.2.2.1 Data-driven Motion Estimation with FBG

FBGs in the optical fiber should reflect the robot configuration during robot motion. The bases for fiber placement are: **1)** The FBGs should be distributed along the places with large strain changes; [103]; **2)** The strain changes or wavelength shifts of FBGs should uniquely map with the robot configuration; **3)** The addition of fiber should not affect the mechanical properties (e.g. bending stiffness and elasticity) of soft robot; **4)** Surface contact on the cylindrical robot body should not dominate the FBG feedback.

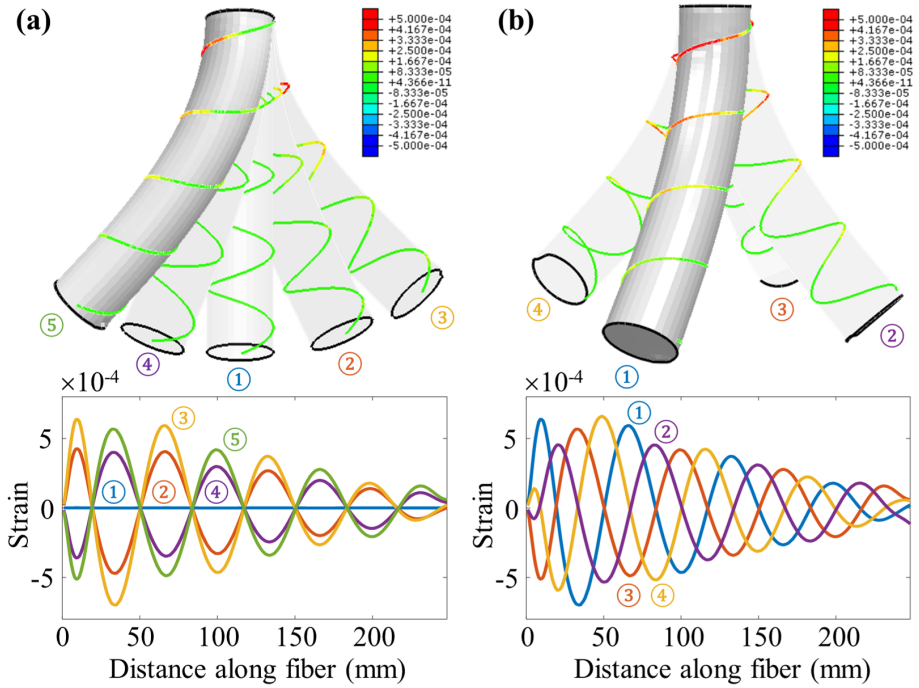


Fig. 6.2: Finite element modeling (FEM) of the strains helically distributed along an elastic continuum manipulator. **(a)** Strains varying in amplitude when the manipulator bends on the same plane/direction. **(b)** Strains under four different-bending directions distinguished by their phase differences.

Based on these considerations, I wound the fiber helically on the robot and sealed it using silicone adhesive (**Fig. 6.2**). The wrapped optical fiber with L FBGs provides L wavelength shifts which can be represented by $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}(1) & \boldsymbol{\lambda}(2) & \cdots & \boldsymbol{\lambda}(N) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^L$. Wavelength shift $\boldsymbol{\lambda}(k)$ at time step k is the difference between current wavelength vector and the original wavelength vector $\boldsymbol{\lambda}_0$ (corresponds to initial robot configuration) (**Fig. 6.1a**). For each two adjacent time steps k and $(k+1)$, the 2D motion vectors in the camera frame $\Delta \boldsymbol{z}(k+1) = \boldsymbol{z}(k+1) - \boldsymbol{z}(k) \in \mathbb{R}^2$ could be obtained (**Section 6.2.1**), as well as the wavelength shift $\boldsymbol{\lambda}(k)$ and $\boldsymbol{\lambda}(k+1)$. Represent their wavelength difference as $\Delta\boldsymbol{\lambda}(k+1) = \boldsymbol{\lambda}(k+1) - \boldsymbol{\lambda}(k)$ (**Fig. 6.1a**). In the camera view with abundant features, the estimation results from template matching during slow and smooth manipulation could be regarded as a ground truth, which is used to train the FBG-related estimation model.

Training: The pre-training procedure would collect the captured endoscopic image at each time step, as well as the wavelength shift vector $\boldsymbol{\lambda}$. The actuation sequence

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}(1) & \boldsymbol{u}(2) & \cdots & \boldsymbol{u}(N) \end{bmatrix}, \tag{6.1}$$

where N is the sampling number, to the robot for data exploration is predefined to cover the whole workspace. The wavelength difference could therefore be obtained as

$$\Delta\boldsymbol{\Lambda} = \begin{bmatrix} \Delta\boldsymbol{\lambda}(1) & \Delta\boldsymbol{\lambda}(2) & \cdots & \Delta\boldsymbol{\lambda}(N) \end{bmatrix}, \tag{6.2}$$

where $\Delta\boldsymbol{\lambda}(i) = \boldsymbol{\lambda}(i) - \boldsymbol{\lambda}(i-1)$, $i = 1,2,\ldots,N$. The images captured on each step are saved to calculate the 2D motion $\Delta\boldsymbol{z}(i) = \boldsymbol{z}(i) - \boldsymbol{z}(i-1)$ offline, forming the motion sequence

$$\Delta\boldsymbol{Z} = \begin{bmatrix} \Delta\boldsymbol{z}(1) & \Delta\boldsymbol{z}(2) & \cdots & \Delta\boldsymbol{z}(N) \end{bmatrix}. \tag{6.3}$$

Using the feedforward neural network in the deep learning toolbox of MATLAB, with $\begin{bmatrix} \Delta\boldsymbol{\Lambda} & \boldsymbol{\Lambda} \end{bmatrix}^{\mathrm{T}}$ as input and $\Delta\boldsymbol{Z}$ as output, a motion estimation model can be trained as

$$\Delta\boldsymbol{z}(i) = M(\Delta\boldsymbol{\lambda}(i), \boldsymbol{\lambda}(i)), \ i = 1,2,\ldots,N. \tag{6.4}$$

Prediction: To increase the processing speed, I convert the model generated in MATLAB to a dynamic link library (DLL) that could be called in the C++ environment. The wavelength receiving via User Datagram Protocol (UDP) and the motion estimation are both accomplished in Qt Creator, with the motion estimation at the k^{th} step obtained by

$$\begin{aligned}\Delta \boldsymbol{z}_w(k) &= M(\Delta\boldsymbol{\lambda}(k), \boldsymbol{\lambda}(k)) \\ &= M((\boldsymbol{\lambda}(k) - \boldsymbol{\lambda}(k-1)), \boldsymbol{\lambda}(k)),\ k = 1,2,\ldots.\end{aligned} \tag{6.5}$$

As this model is not related to real-time image processing once finished the training procedure, it could be regarded as an independent 2D motion estimator using FBG with a frequency of at least 30 Hz.

6.2.2.2 Camera-FBG-combined Motion Estimation

The template matching algorithm was used for camera-based motion estimation. For each update of the sliding block in time step k (**Fig. 6.1c**), there is a coherence variable β characterizing the similarity between the sliding block and the template pattern in the last step $(k-1)$. The new template position $\boldsymbol{z}(k)$ is finalized by finding the sliding block that possesses the maximum coherence with the template at $\boldsymbol{z}(k-1)$. Define the 2D motion estimation from step $(k-1)$ to (k) by image processing as $\boldsymbol{z}_c(k-1)$, with a coherence $\beta(k)$. Another motion estimation obtained from the FBG-related model in (**Eq. 6.5**) is $\boldsymbol{z}_w(k-1)$. For the camera-based estimation, the quality of features limits the accuracy. On the other hand, FBG is highly sensitive to temperature and tiny strains, sometimes may involve noise. To avoid their defects for more reliable estimations, I designed a weighted estimator, represented as

$$\Delta \boldsymbol{z}_p(k) = \alpha\beta(k) \cdot \Delta \boldsymbol{z}_c(k) + (1 - \alpha\beta(k)) \cdot \Delta \boldsymbol{z}_w(k), \tag{6.6}$$

where α is a constant parameter to adjust the ratio of coherence in the camera-estimated portion. It is a scaling factor to normalize the fusion results of two motion estimations. This parameter would be tuned by a heuristic procedure before robot execution. In the experiment, it was initialized as 0.7. Its value did not bring notable differences to the tracking accuracy within a small range of variation $(0.6 \sim 0.8)$.

6.2.3 Model-less Visual Servoing Based on 2D Motion Estimation

With the above enhanced estimation as feedback, I implement a model-free controller that is inspired by the optimal control method in [96]. The forward kinematic mapping from actuation space to task space is represented as

$$\dot{\boldsymbol{z}}_{\boldsymbol{p}} = \boldsymbol{J}\dot{\boldsymbol{u}}, \tag{6.7}$$

where $\boldsymbol{z}_p$ and $\boldsymbol{u}$ are the absolute 2D position of template to be tracked and the actuator input respectively, $\boldsymbol{J}$ is the Jacobian matrix. The inverse kinematics can be discretized to

$$\Delta\boldsymbol{u} = \boldsymbol{J}^{-1}\Delta\boldsymbol{z}_p. \tag{6.8}$$

The core of obtaining $\Delta\boldsymbol{u}$ in **Eq. 6.8** for accurate motion is to find a proper Jacobian matrix $\boldsymbol{J}$. For the robot, the 3 degrees of freedom (DoFs) of the actuator are in the same unit and independent of each other. The initialization of $\boldsymbol{J}$ could be achieved by actuating the 3 DoFs in turn with an incremental amount $\Delta u_i, i = 1,2,3$ (i.e. the actuation commands are $\begin{bmatrix} \Delta u_1 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & \Delta u_2 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 0 & \Delta u_3 \end{bmatrix}$ successively), and measuring the corresponding displacements $\Delta\boldsymbol{z}_{pi}$. The initial Jacobian matrix could be constructed as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{J}_2 & \boldsymbol{J}_3 \end{bmatrix}, \tag{6.9}$$

where $\boldsymbol{J}_i = \Delta\boldsymbol{z}_{pi}/\Delta u_i$. The obtained Jacobian matrix could also be updated online, relying on continuously solving a quadratic programming problem as follows

$$\begin{aligned} \text{minimize} \quad & \|\Delta\boldsymbol{J}(k+1)\| \\ \text{subject to} \quad & \Delta\boldsymbol{z}_p(k) = \boldsymbol{J}(k+1)\Delta\boldsymbol{u}(k) \\ & \boldsymbol{J}(k+1) = \boldsymbol{J}(k) + \Delta\boldsymbol{J}(k+1) \end{aligned}, \tag{6.10}$$

where $\Delta\boldsymbol{J}(k+1)$ is the variable to be optimized, $\boldsymbol{J}(k)$ is the Jacobian matrix at time k, $\boldsymbol{J}(k+1)$ is the estimation for Jacobian at time $(k+1)$. After obtaining the latest Jacobian, the command to actuators could be calculated by

$$\Delta\boldsymbol{u}(k+1) = \boldsymbol{J}(k+1)^{-1}\Delta\boldsymbol{z}_p^*(k+1), \tag{6.11}$$

where $\Delta\boldsymbol{z}_p^*(k+1)$ is the desired motion in camera view.

As illustrated in **Fig. 6.3**, the positional command is obtained by the difference of

the desired and current positions of the feature (**Section 6.2.1**). The robot is actuated accordingly with the inverse solution generated from the model-free feedback controller (**Section 6.2.3**). The change of robot configuration causes the object movement in the camera view as well as the change of FBG wavelength. The actual 2D motion will be measured by fusing the estimates of image processing (**Section 4.2.2**) and optical sensing (**Section 6.2.2**).

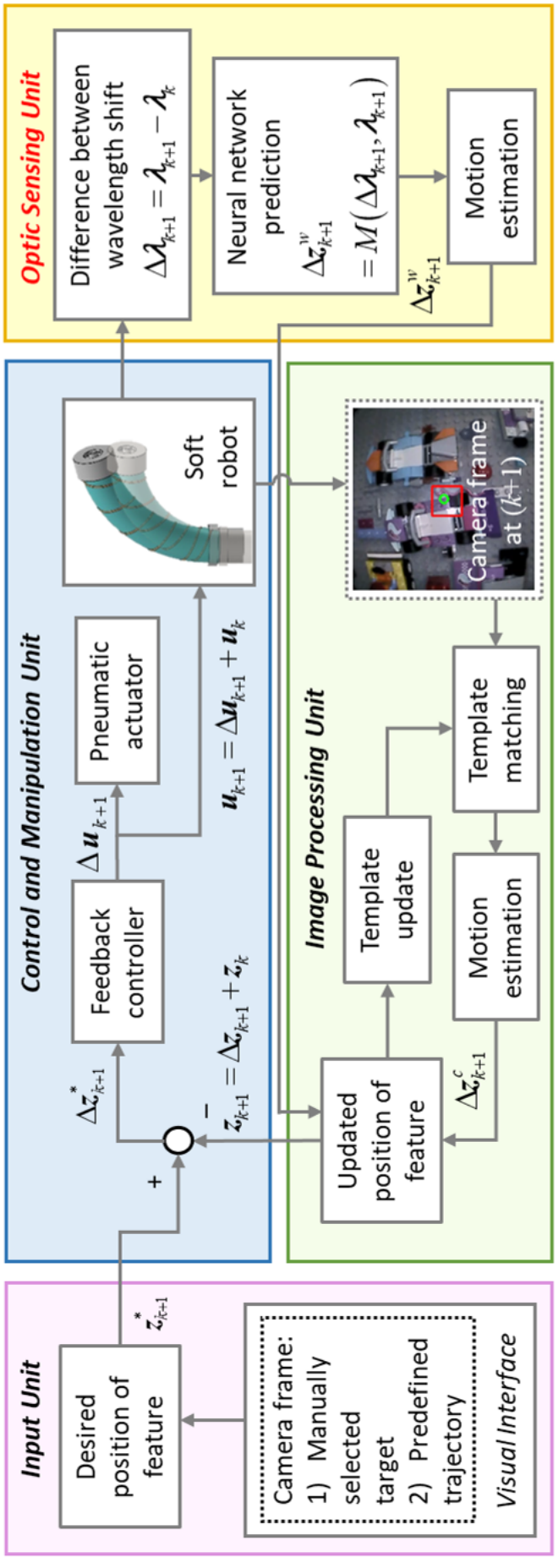


Fig. 6.3: Control architecture of the proposed control method. Parameters $\boldsymbol{u}$ and $\boldsymbol{z}$, respectively, denote the actuation command and the position of the captured feature in camera view. The input unit provides the positional command in the image view. A model-free feedback controller is responsible for generating the inverse solution to the actuator, causing the object movement in the camera view as well as the change of FBG wavelength. The actual displacement after the execution of actuation command is obtained by the weighted sum of FBG-based motion estimation and image processing.

6.3 Experiments of Enhanced 2D Visual Servoing

Experiments were conducted in two scenarios, separately with insufficient visual features and external disturbances. With the EM tracker measurement as ground truth, the enhanced tracking method was compared with the pure camera-based method, to validate the improvement.

6.3.1 Continuum Endoscopic Robot with FBG Sensor Device

The soft robot used in this study was constructed from a moulded silicone rubber (Ecoflex50, Smooth-on Inc.) segment, a monocular endoscopic camera, FBG fiber and 3D printed fixation components (**Fig. 6.4**). The continuum robot comprised of three cylindrical fluidic chambers spaced 120° apart from each other. Each chamber was constrained radially by a helically wrapped Kevlar strings in order to only allow expansions on axial direction. Different combinations of three inflation pressures capacitated the omnidirectional bending of the robot ($\sim$ 100°). An endoscopic camera (OV6930, Depth of view: 8 to 150 mm, Shanghai E-vision Optoelectronic Co., Ltd.) and a module of LED illumination were fixed on the tip cap. A central cavity inside the robot body was reserved to house the cables of camera and LEDs. With a 90° diagonal field of view, the camera captures images of 400 pixels, indicating that a pixel translates to 0.16° field of view.

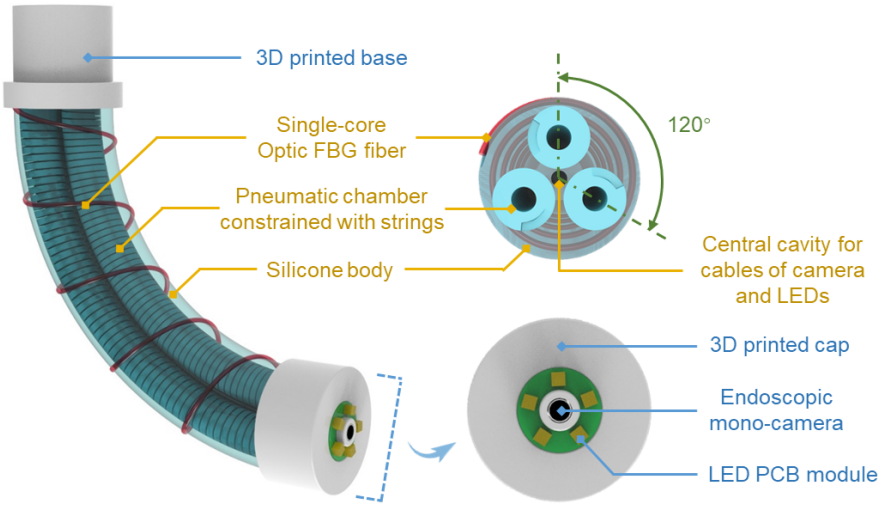


Fig. 6.4: Pneumatic-driven 3-chamber robot was used. An optical fiber with multiplexing 16 FBGs was helically wrapped around the manipulator for sensing feedback of robot configurations in real-time. A monocular endoscopic camera and a LED module were fixed on the tip cap of robot. Cross-section view of the robot could show the silicone chambers constrained individually with helical Kevlar strings.

The 3D printed (Stereolithographic) cap could be worn on the distal (relative to the base of robot) part of robot. A single-core optical fiber with 16 FBGs (8 mm length for each FBG) was helically wrapped and adhered (by Sil-Poxy, Smooth-on Inc.) along the silicone body. The bottom of the robot was fixed to a 3D printed base that remains stationary. The outer diameter of the robot was 19 mm and the total length was 95 mm (including the tip cap). With the optical fiber ($421.5) and the camera ($780), the price of soft manipulator was $\sim$ \$1,210. The non-consumable FBG interrogator cost $\sim$ \$20,000, while the actuation components cost $\sim$ \$400.

6.3.2 Experimental Setup

Two different scenarios were constructed, as illustrated in **Fig. 6.5** and **Fig. 6.9** respectively. The soft manipulator described in **Fig. 6.4** was fixed downward, viewing the workspace scene built from swine viscera to simulate the endoscopic scene in laparoscopy (**Fig. 6.5**), or LEGO® (**Fig. 6.9**). The distances between the endoscopic camera and the scene surface were 3 cm and 15 cm, respectively.

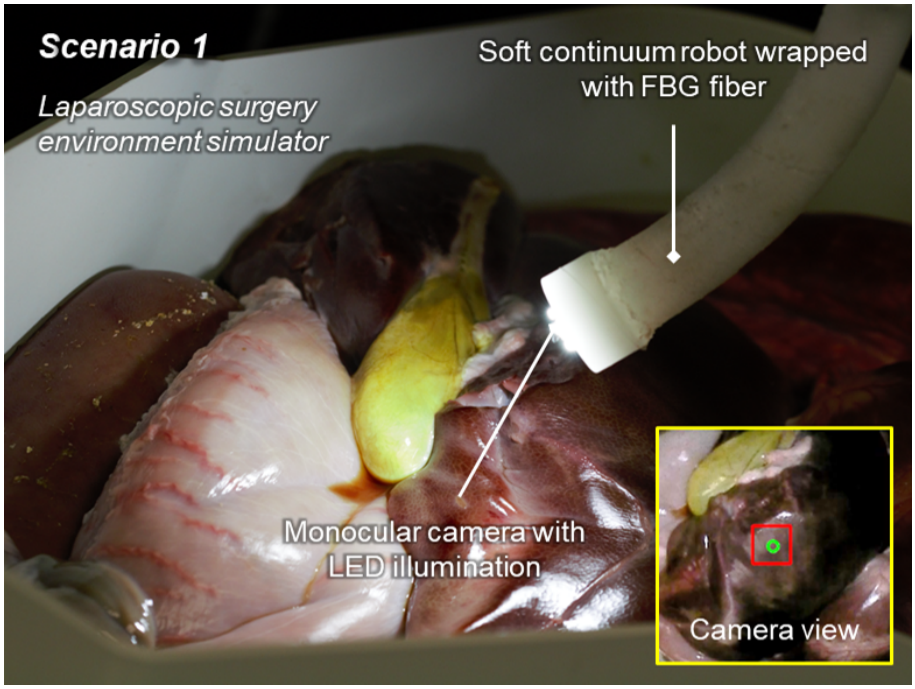


Fig. 6.5: Endoscopic scene in abdominal surgery simulated with swine viscera. The robot was pre-bended to approximate insertion angle of a laparoscope. EM tracking coils were attached on the tip as the ground truth of end-effector pose.

The precision of visual servo control depends on two main aspects, which are the camera-based motion estimation and the robot control strategy. The tracking errors could

also be considered from two aspects. The first kind ignores the drift of tracked template among different image frames, and the motion estimation is supposed to be reliable. It only reflects the controller performance (i.e. **Fig. 6.6a** and **Fig. 6.7a**). Another kind includes the accuracy of motion estimation (i.e. **Fig. 6.6c** and **Fig. 6.7c**) and should be evaluated by a more stable ground truth. As the main purpose of this section is to improve the motion estimation accuracy, the validation tasks are designed to compare the performances of camera-based motion estimation in [190] with the new FBG-enhanced method, using the same robot controller. During the tests, two 6D electromagnetic (EM) trackers (#610059, NDI Aurora) were laterally attached on the robot tip to record the pose of the end-effector, which was used as the evaluation ground truth. After each actuation step, the 2D motion vectors in the camera view could be calculated by projecting the 3D motion vectors on the current image plane, which would always be perpendicular to the end-effector normal direction. The actual path in the camera view could be reconstructed by accumulating these 2D motion vectors.

To obtain the accumulated actual path, define the 3D positions and 4D quaternions of two EM coils at time step k as $\boldsymbol{p}_i(k)$ and $\boldsymbol{h}_i(k)$, $i = 1,2$. The position of end-effector is

$$\boldsymbol{p}(k) = (\boldsymbol{p}_1(k) + \boldsymbol{p}_2(k))/2, \tag{6.12}$$

the rotation matrix $\boldsymbol{R}(k)$ can be calculated from $\boldsymbol{h}_1(k)$ or $\boldsymbol{h}_2(k)$. Suppose the orthogonal basis for original local end-effector coordinate is represented as $\boldsymbol{u}_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{v}_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{n}_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$, then the basis vectors at time step k will be $\boldsymbol{u}(k) = \boldsymbol{R}(k) \cdot \boldsymbol{u}_0$, $\boldsymbol{v}(k) = \boldsymbol{R}(k) \cdot \boldsymbol{v}_0$ and $\boldsymbol{n}(k) = \boldsymbol{R}(k) \cdot \boldsymbol{n}_0$. With the 3D motion vector and the current image plane obtained by $\Delta\boldsymbol{p}(k) = \boldsymbol{p}(k) - \boldsymbol{p}(k-1)$ and $\boldsymbol{U}(k)\boldsymbol{V}(k)$ respectively, the projected motion vector on the image plane is

$$\Delta\widehat{\boldsymbol{p}}(k) = \Delta\boldsymbol{p}(k) - \frac{\Delta\boldsymbol{p}^{\mathrm{T}}(k) \cdot \boldsymbol{n}(k)}{|\boldsymbol{n}(k)|^2} \cdot \boldsymbol{n}(k), \tag{6.13}$$

w.r.t. the generator coordinate. The 2D motion vector w.r.t. image plane $\boldsymbol{U}(k)\boldsymbol{V}(k)$ is calculated by

$$\Delta\boldsymbol{z}(k) = \begin{bmatrix} \dfrac{\Delta\widehat{\boldsymbol{p}}^{\mathrm{T}}(k) \cdot \boldsymbol{u}(k)}{|\boldsymbol{u}(k)|} & \dfrac{\Delta\widehat{\boldsymbol{p}}^{\mathrm{T}}(k) \cdot \boldsymbol{v}(k)}{|\boldsymbol{v}(k)|} \end{bmatrix}^{\mathrm{T}}. \tag{6.14}$$

There would be a constant bias angle θ between coil-obtained $\Delta\boldsymbol{z}(k)$ and camera-based motion estimation $\Delta\boldsymbol{z}_c(k)$, resulting from the mounting angle of EM trackers. According to this calculation method, θ could be obtained by calculating the mean angle difference using

a set of sampling data, similarly the multiple that reflects the displacement length could also be calculated. After compensating this bias θ by

$$\Delta \boldsymbol{z}_s(k) = \boldsymbol{T} \cdot \Delta \boldsymbol{z}(k) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \Delta \boldsymbol{z}(k). \tag{6.15}$$

The standard motion vector $\Delta \boldsymbol{z}_s$ is thus obtained to test the accuracy of image-processing-based estimation and FBG-enhanced estimation.

6.3.3 Visual Servo in Feature-deficient Scenes

The improvement of motion estimation was investigated via path following tasks. In **Section 4.3** [190], a desired path in the "∞" shape have been tested. The smooth path was defined by 72 points on the 400 px camera frame (with the mean interval of 15.68 pixels), generating a closed curve with continuous gradient. To increase the tracking difficulty, the path designed in this section was the outline of Batman logo with several acute angles, consisting of 127 points with the mean interval of 7.91 pixels. The robot was instructed to track a same template pattern in the laparoscopic scene (**Fig. 6.5**) along the Batman path for three cycles, using template matching alone (**Fig. 6.6**) and the enhanced sensing method (**Fig. 6.7**), respectively.

In the camera view, the tracking trajectory of the target block roughly accorded with the Batman path when the 2D motion estimation were achieved by both template matching and FBG-enhanced method (**Fig. 6.6a**, RMSE: 8.60 pixels, **Fig. 6.7a**, RMSE: 8.64 pixels), benefitting from the well-performed closed-loop controller. However, the accuracy of motion estimation could be reflected in the actual projected path of end-effector (**Fig. 6.6c**, **Fig. 6.7c**), which should also be identical with **Fig. 6.6a**, **Fig. 6.7a**. The 2D estimation error at each step was accumulated and deviate the EM tracker-reconstructed actual path from the desired path. The target template drifted and gradually missed the original target feature (**Fig. 6.6b**, **Fig. 6.7b**). In this task, the to-be-tracked template block was intentionally defined on the feature-deficient liver surface. **Fig. 6.6b** includes two camera views captured at the beginning of 1st cycle and the end of 3rd cycle, so does **Fig. 6.7b**. The drift of target template could be observed referring to the feature marked by black dotted block. The absolute tracking errors between the actual and desired trajectories are demonstrated in **Fig. 6.8**. The root-mean-square error (RMSE) for each cycle gradually increased after three-cycle execution, which were 108.56, 114.89 and 122.35 pixels when

using camera-based tracking, and 14.36, 17.76 and 27.25 pixels when using FBG-enhanced tracking. After the enhancement, the overall tracking error was reduced by 82.3%.

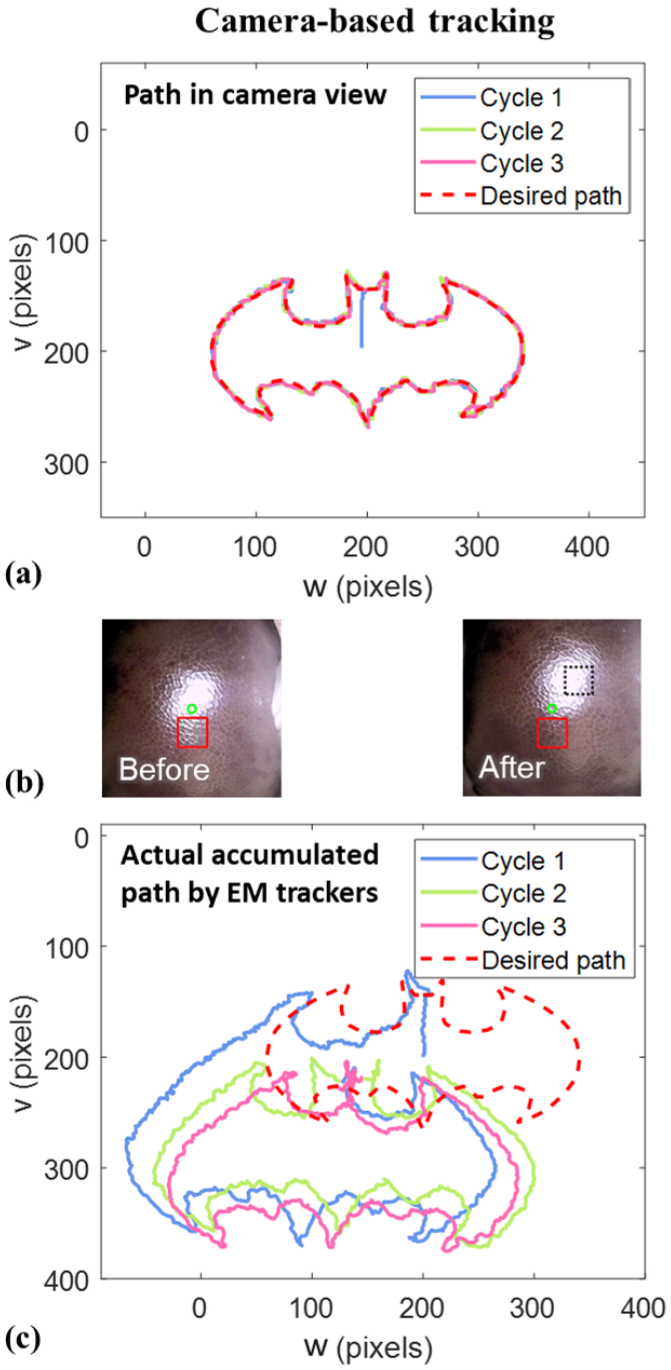


Fig. 6.6: Robot following of a predefined "Batman" path in the scene of **Fig. 6.5**. **(a)** The motion estimated by image processing of the endoscopic view *alone*; **(b)** Features in the red block were selected by the user *before* the motion displacement. *After*, such a block was expected to keep matching/tracking at the same square of features (black dotted block), acting as a static reference for robot to "draw" the path; **(c)** Actual path of end-effector recorded by EM tracking coils, which was projected on the same *u*-*v* coordinates.

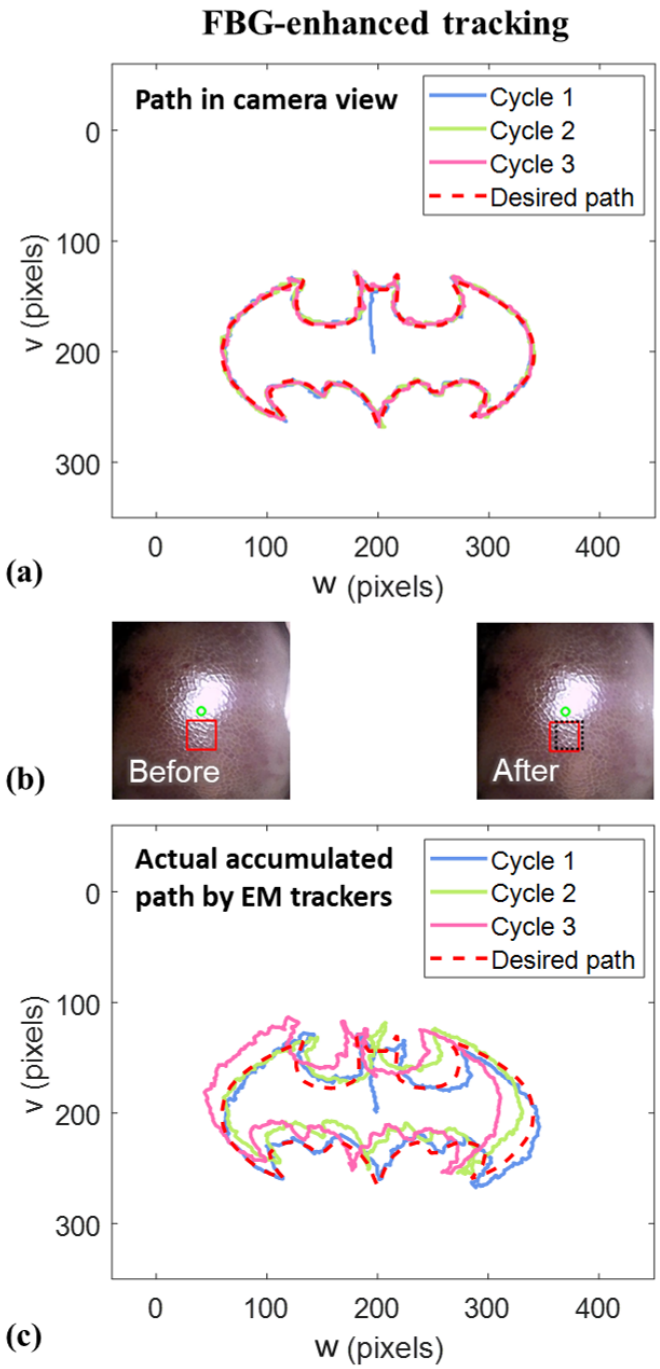


Fig. 6.7: FBG-enhanced tracking performance of the same "Batman" path. **(a)** The trajectory in the endoscopic view; **(b)** The deviation between the selected feature after three cycles. **(c)** Actual trajectory recorded by EM trackers.

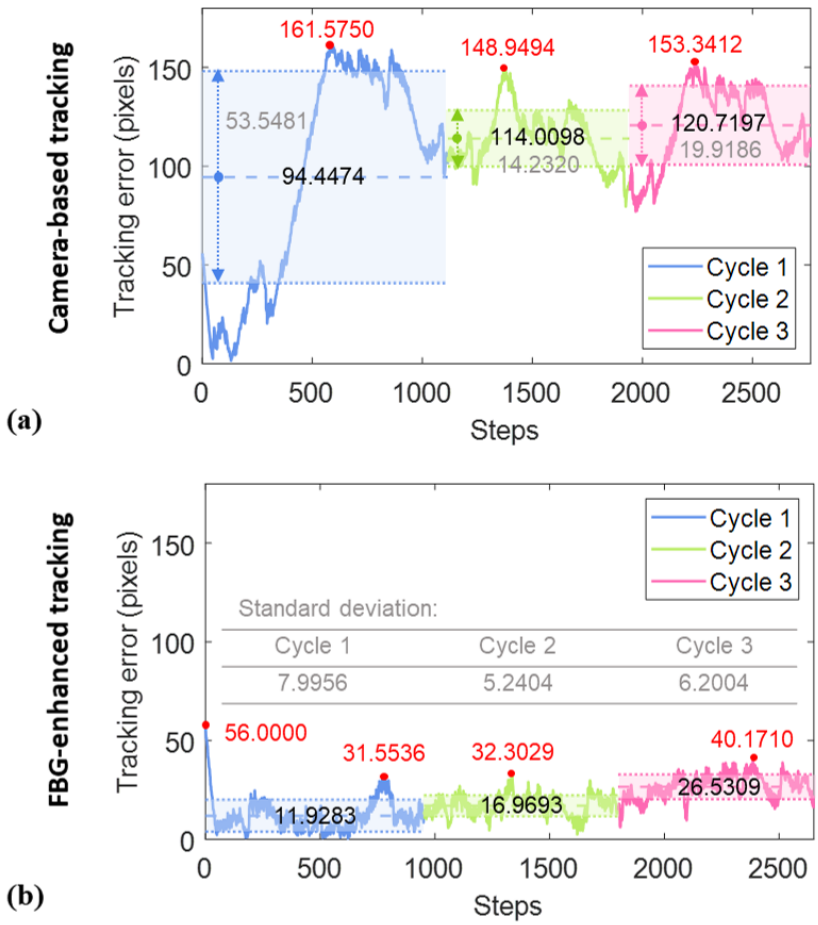


Fig. 6.8: Tracking error analysis of the "Batman" path in the laparoscopic scene (**Fig. 6.5**). **(a)** Tracking achieved by image processing, e.g. mean error: 94.45 (SD: 53.55) induced in Cycle 1; **(b)** Tracking achieved by FBG-enhanced estimation. Curves represent the absolute tracking error in **Fig. 6.6c**, and **Fig. 6.7c** and red points illustrate the maximum error in each cycle.

It could be seen that the tracking performance in visual servoing would be deteriorated under two conditions, one was in feature-deficient scenes, such as the liver surface in this task; another is the step length of desired motion. As template matching is purely based on the features captured in adjacent camera frames, the estimation would be inaccurate if there are few features to be tracked or the features are repeatable in the camera frame. When the tracking path is defined by dense points, the motion step size has to be small, aggravating the proportion of error in each step.

6.3.4 Visual Servo under External Disturbances

Besides the laparoscopic-simulation task, I also set up an environment using LEGO® (**Fig. 6.9**) to test another challenge in visual servo for soft robots, external forces. An air nozzle was fixed facing the tip of soft robot in a distance of 5 cm and was connected to an air compressor. The robot was actuated to follow the same Batman path in **Section 6.3.3** for 3 cycles. A gust of "wind" with 4 bar air pressure was exerted on the robot at the beginning of the 2nd cycle and increased to 6 bar in 0.5s when half of the 2nd cycle was finished (taking $\sim$ 38 s). The external pressure was released at the beginning of the 3rd cycle. The 6-bar pressure was almost the maximum at which the robot could sustain stable tracking. Although the air compressor output constant air pressure, the force direction and power applied to the robot varied unpredictably due to the robot motion. The tracking performances using two kinds of motion estimation methods are exhibited in **Fig. 6.10** and **Fig. 6.11**.

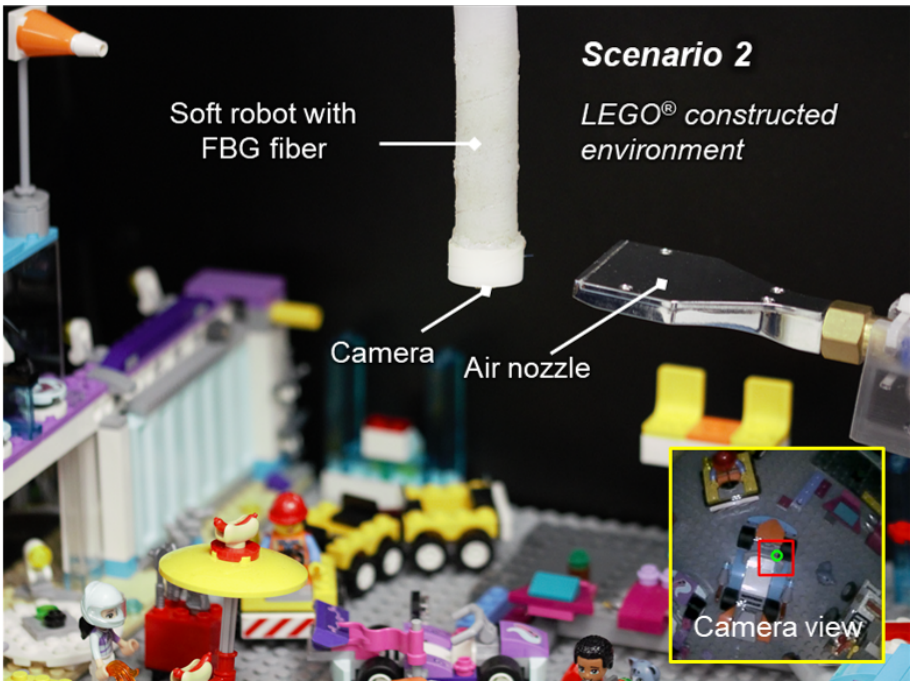


Fig. 6.9: Setup in a scene of LEGO®. Robot was fixed downward viewing the workspace scene. An air nozzle was fixed facing the robot to generate force disturbances. EM tracking coils were attached as the ground truth.

In the camera view, the actual trajectory when using enhanced motion estimation method was always consistent with the Batman path, although the path becomes unsmooth when the external force was applied (**Fig. 6.11a**, RMSE: 9.16 pixels. max. error: 31.71 pixels in Cycle 2). However, by the original camera-based tracking, the path could not return stable quickly when the air pressure was released (**Fig. 6.10a**), the max. tracking

error in Cycle 3 (32.36 pixels) was even larger than that of Cycle 2 (26.68 pixels). One of the reasons was that the step size used in this method was larger than the one in the enhanced estimation method (**Fig. 6.11**), to reduce the portion of error in each motion step. If the step size was as small as in the enhanced method, the actual following path (**Fig. 6.12a**, RMSE: 107.63 pixels, max. error: 147.42 pixels in Cycle 3) will deviate the original path position more seriously. When the force was applied (Cycle 2) and removed (Cycle 3), the controller using the proposed motion estimation enabled to maintain a stable tracking performance (**Fig. 6.12b**, RMSE: 24.75 pixels, Max. error: 44.79 pixels in Cycle 2; RMSE: 21.02 pixels, Max. error: 36.59 pixels in Cycle 3).

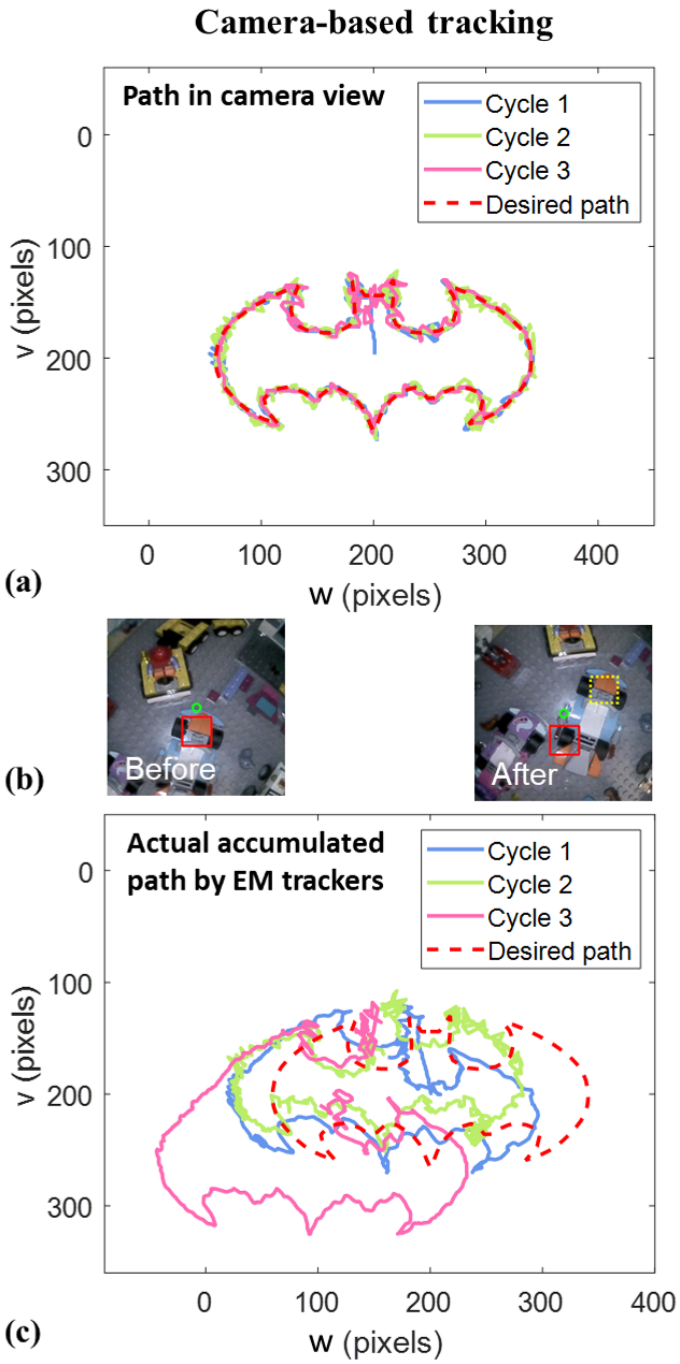


Fig. 6.10: Performance of robot following on the predefined "Batman" path in the scene of **Fig. 6.9**. Force disturbances are applied to the end-effector from the nozzle, with 4 bar to 6 bar pressure and released in the 3rd cycle. **(a)** The motion estimated by image processing *alone* vibrates during and after the disturbances applied; **(b)** Features in red block was selected by user *before* the motion displacement. *After* then, yellow dotted block indicates the position in *before*; **(c)** Actual path of end-effector deviates gradually after three cycles.

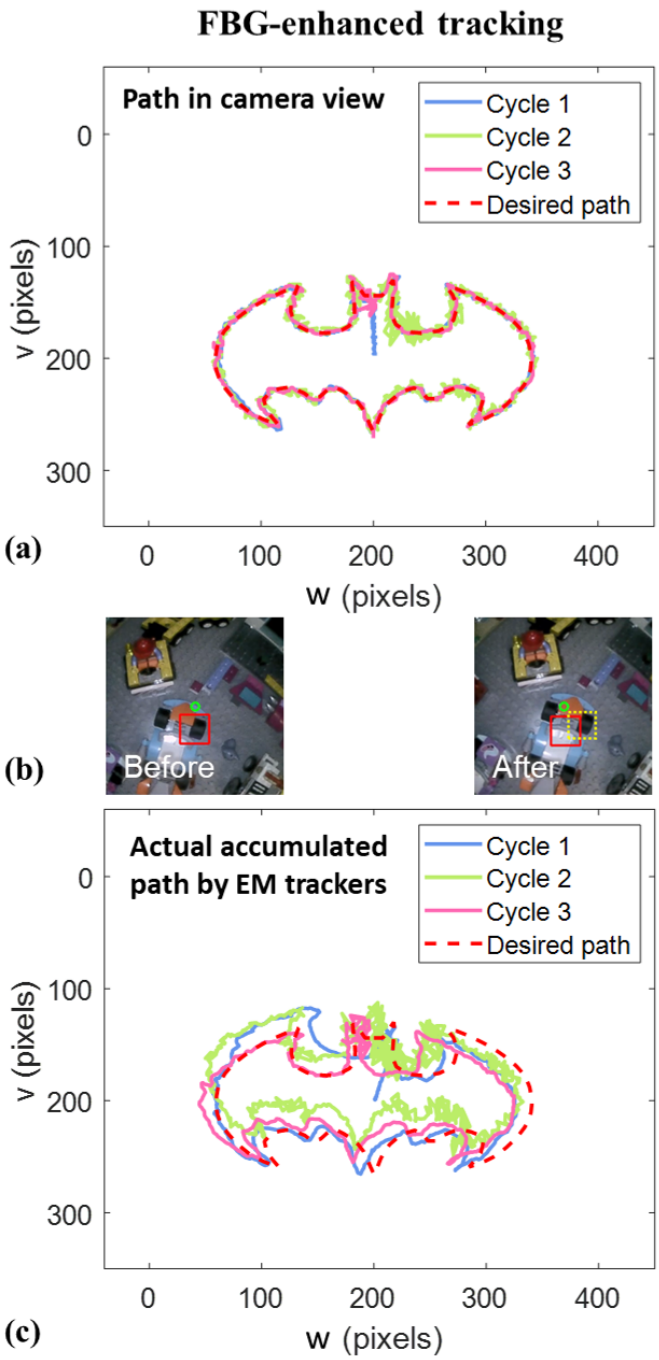


Fig. 6.11: Performance of robot following on the "Batman" path in the scene of **Fig. 6.9** with the motion estimation by enhanced method. **(a)** Robot motion in the endoscopic view maintains stable under and after external forces; **(b)** Feature deviation is well solved; the actual path in **(c)** demonstrates the improvement.

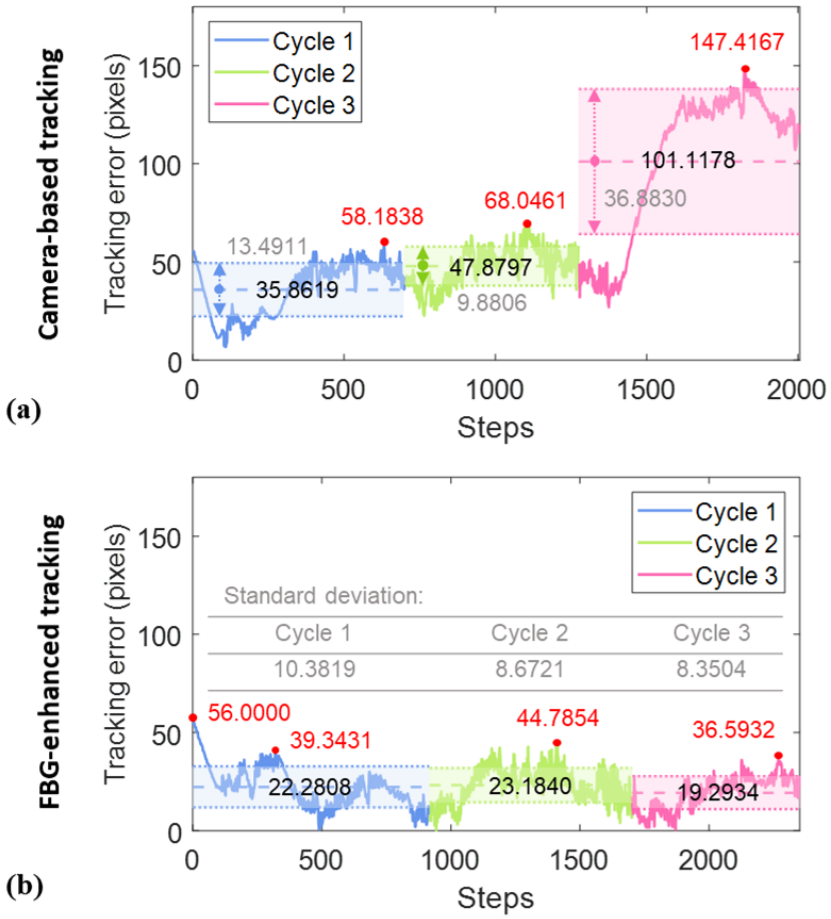


Fig. 6.12: Tracking error analysis in the LEGO® scene (**Fig. 6.9**). **(a)** Tracking achieved by image processing; **(b)** Tracking achieved by FBG-enhanced estimation. Curves represent the absolute error in (**Fig. 6.10c** and **Fig. 6.11c**).

6.4 FBG Enhanced SLAM Pose Estimation

Although the monocular camera has enabled eye-in-hand pose estimation as described in **Section 4.4.2**, the accuracy of estimation result will highly depend on the abundance of image features. In the cases particularly with repetitive structures, texture-less objects, and substantial viewpoint changes, the hand-crafted descriptors may have difficulty to find sufficient feature correspondences [105]. The motion blur, occlusions, and illumination changes will also affect the result [191]. To compensate for the error induces by the absence of features, fusing the camera with other sensors has been investigated.

The most prevalent fusion approaches would be integrating cameras with inertial sensors [106, 192], where the acceleration and angular velocity can be utilized employing filters like extended Kalman filter [106]. This kind of statistical filtering technique solves the sensor fusion problem with the requirement of complex system modeling and calibrations between the camera and the inertial sensors. Learning-based fusion methods, such as long short-term memory (LSTM) [191] and convolutional neural network (CNN) [193], were also investigated with the prevalence of model-less algorithms. In conventional rigid-link robots, such sensor fusion has been widely discussed, no matter in the aspects of device choices or fusing methods. However, considering soft robots built from elastic materials, the integration with inertial sensors and other kinds of rigid sensing devices requires special packing design.

As discussed in **Section 2.3**, multi-core FBGs can be used as a standalone shape sensor based on modeling [64, 67], while single-core FBGs mounted on soft manipulators could also achieve surface force or shape sensing using various approaches [10, 11]. **Section 6.2**-**6.3** have employed the combination of single-core optical FBG fiber and endoscopic camera on a continuum robot for accurate and robust 2D motion estimation in the camera view [194]. This sort of sensing modality can be further used on end-effector pose estimation of continuum robots.

Camera pose estimation can be achieved in SLAM. As for soft endoscopic robot, integrating a single-core FBG fiber could provide an additional measurement to reflect the robot configuration via wavelength shift (**Fig. 6.13a**). Training the FBG fiber as a real-time pose estimator using extreme learning machine and fusing it with the estimation from SLAM will significantly improve the sensing accuracy in soft continuum robot control, especially under extreme cases.

6.4.1 Task Space Definition

The single-core FBG fiber is helically wrapped on the continuum robot. The multiplexing l FBGs inscribed are independent of each other, providing l wavelength/strain measurement points. Wavelength shift vector $\boldsymbol{\lambda}(k) \in \mathbb{R}^l$ means the difference between wavelength vector at time step k and the original wavelength vector $\boldsymbol{\lambda}_0$ (corresponds to initial robot configuration) (**Fig. 6.13a**). Define the camera pose estimated by SLAM (**Section 4.4.2**) at time step k as $\boldsymbol{z}_c(k) = \begin{bmatrix} \boldsymbol{p}_c(k) & \boldsymbol{q}_c(k) \end{bmatrix} \in \mathbb{R}^7$, including the 3D position $\boldsymbol{p}_c \in \mathbb{R}^3$ and quaternion-represented orientation $\boldsymbol{q}_c \in \mathbb{R}^4$. The actual pose is $\boldsymbol{z}(k) = \begin{bmatrix} \boldsymbol{p}(k) & \boldsymbol{q}(k) \end{bmatrix}$. This SLAM-estimated pose in feature-abundant camera views under stable and smooth movement could be considered as ground truth of the robot end-effector pose, i.e., $\boldsymbol{z}_c \approx \boldsymbol{z}$. With a pose ground truth available, the training for FBG can be activated. However, in feature-deficient cases, the pose $\boldsymbol{z}_c$ will not be reliable. It is worth noticing that the SLAM-based pose estimation would not be the only option; other pose measurements will also be valid. Here the reason to use SLAM-based pose estimation is to utilize the integrated camera without the need for any external sensing devices. The control objective is to generate an actuation command $\Delta\boldsymbol{u}(k)$, that can meet the desired end-effector movement $\Delta\boldsymbol{p}^*(k)$ or $\Delta\boldsymbol{q}^*(k)$.

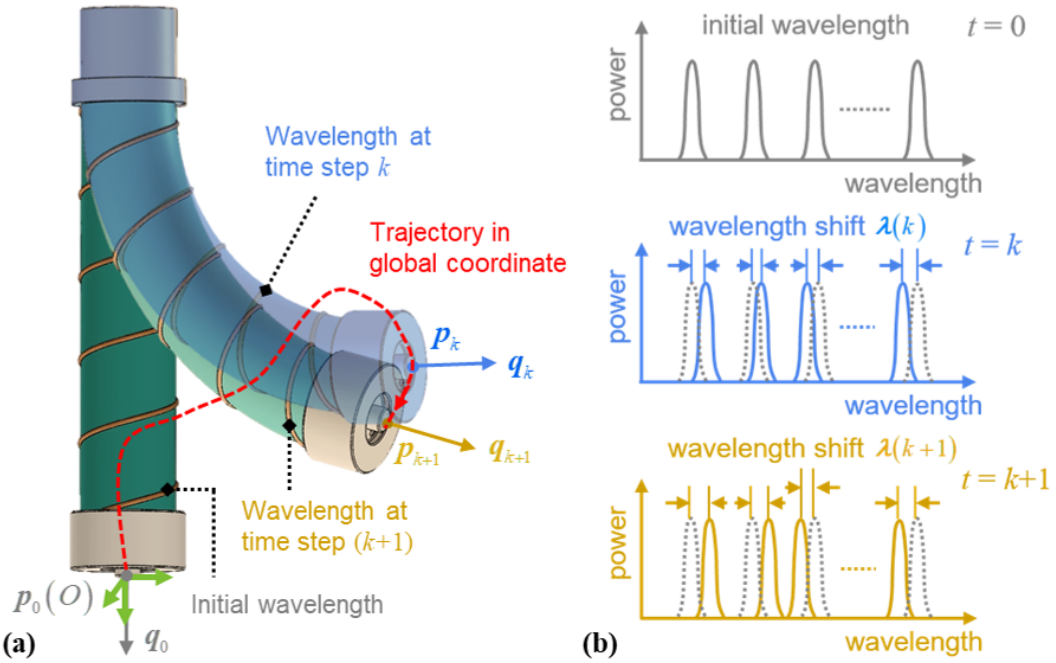


Fig. 6.13: Soft continuum robot wrapped with a helically-wound single-core FBG fiber. **(a)** Tip camera poses obtained at each time step, k, based on SLAM algorithm. **(b)** FBG wavelengths shifted correspondingly, i.e., from $\boldsymbol{\lambda}(k)$ to $\boldsymbol{\lambda}(k+1)$.

6.4.2 Learning-based Pose Estimation using FBGs

With an appropriate pose estimation frequency in SLAM algorithm (e.g., $20 \sim 50$ Hz, related to the camera and computer performance), both the wavelength shift $\boldsymbol{\lambda}(k)$ and the pose estimation $\boldsymbol{z}_c(k)$ (**Section 4.4.1** in **Chapter 4**) at each 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.

Extreme learning machine (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. 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

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda}(1) & \boldsymbol{\lambda}(2) & \cdots & \boldsymbol{\lambda}(N_0) \end{bmatrix} \in \mathbb{R}^{l \times N_0} \tag{6.16}$$

and

$$\boldsymbol{Z}_c = \begin{bmatrix} \boldsymbol{z}_c(1) & \boldsymbol{z}_c(2) & \cdots & \boldsymbol{z}_c(N_0) \end{bmatrix} \in \mathbb{R}^{7 \times N_0} \tag{6.17}$$

respectively. The mapping

$$\boldsymbol{z}_c(k) = f(\boldsymbol{\lambda}(k)) \tag{6.18}$$

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 a SLFN with N hidden nodes can be represented by [195–197]

$$\boldsymbol{o}_j = \sum_{i=1}^{N} \boldsymbol{\beta}_i \phi_i(\boldsymbol{\lambda}(j)) = \sum_{i=1}^{N} \boldsymbol{\beta}_i \phi(\boldsymbol{\lambda}(j), \boldsymbol{a}_i, b_i),\ j = 1, 2, \ldots N_0, \tag{6.19}$$

where for the i^{th} hidden node, $\boldsymbol{a}_i = \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{il} \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{\beta}_i = \begin{bmatrix} \beta_{i1} & \beta_{i2} & \cdots & \beta_{i7} \end{bmatrix}^{\mathrm{T}}$ are the weighting vector linking to the input nodes and output nodes, respectively. 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 function (RBF) here, i.e.,

$$\phi_i(\boldsymbol{\lambda}(j)) = \phi(\boldsymbol{\lambda}(j), \boldsymbol{a}_i, b_i) = \exp\left(\frac{\|\boldsymbol{\lambda}(j) - \boldsymbol{a}_i\|^2}{b_i}\right). \tag{6.20}$$

Represent the inner product of vectors $\boldsymbol{a}_i$ and $\boldsymbol{\lambda}(j)$ as $\boldsymbol{a}_i\boldsymbol{\lambda}(j)$. Equation **Eq. 6.19** can be written compactly as:

$$\Phi\ \beta = \boldsymbol{O}, \tag{6.21}$$

where

$$\begin{aligned}
\Phi &= \begin{bmatrix} \phi(\boldsymbol{a}_1\boldsymbol{\lambda}(1)+b_1) & \cdots & \phi(\boldsymbol{a}_N\boldsymbol{\lambda}(1)+b_N) \\ \vdots & \cdots & \vdots \\ \phi(\boldsymbol{a}_1\boldsymbol{\lambda}(N_0)+b_1) & \cdots & \phi(\boldsymbol{a}_N\boldsymbol{\lambda}(N_0)+b_N) \end{bmatrix} \in \mathbb{R}^{N_0\times N}, \\
\beta &= \begin{bmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} & \boldsymbol{\beta}_2^{\mathrm{T}} & \cdots & \boldsymbol{\beta}_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N\times 7}, \\
\boldsymbol{O} &= \begin{bmatrix} \boldsymbol{o}_1^{\mathrm{T}} & \boldsymbol{o}_2^{\mathrm{T}} & \cdots & \boldsymbol{o}_{N_0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N_0\times 7}.
\end{aligned} \tag{6.22}$$

Here, Φ is called the output matrix of the hidden layer. Obviously, the ideal to-be-learn parameters will result in:

$$\Phi\beta = \boldsymbol{Z}_c. \tag{6.23}$$

The goal during training is to minimize the network cost function $\|\boldsymbol{O} - \boldsymbol{Z}_c\|$, i.e., to find a solution vector which includes the to-be-tuned parameters $\boldsymbol{a}_i^{\mathrm{T}}$, $\boldsymbol{\beta}_i^{\mathrm{T}}$, and b_i^{T}, $i = 1, 2, \ldots N$. Several algorithms could be utilized to search this solution, e.g., gradient-based iteration, least-square solution. Here we use the latter method. An advantage of ELM is that the values of input weights $\boldsymbol{a}_i^{\mathrm{T}}$ and hidden layer threshold b_i could be assigned randomly without having to consider the input data, thus the output matrix Φ could then be obtained. Given an input set $\boldsymbol{\Lambda}$, the least square solution $\widehat{\beta}$ of linear system **Eq. 6.23** could thus be determined by:

$$\left\|\phi(\boldsymbol{A},\boldsymbol{b})\,\widehat{\beta} - \boldsymbol{Z}_c\right\| = \min_{\beta}\|\phi(\boldsymbol{A},\boldsymbol{b})\,\beta - \boldsymbol{Z}_c\|, \tag{6.24}$$

where $\boldsymbol{A} = \begin{Bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \end{Bmatrix}$ and $\boldsymbol{b} = \begin{Bmatrix} b_1 & b_2 & \cdots & b_N \end{Bmatrix}$. Finally, the output weights β can be analytically determined as follows:

$$\widehat{\beta} = \Phi^{\dagger}\boldsymbol{Z}_c, \tag{6.25}$$

where $\Phi^{\dagger}$ means the Moore–Penrose generalized inverse of Φ. After these steps, the global nonlinear mapping model **Eq. 6.18** generated by ELM is ready for prediction.

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

$$\boldsymbol{z}_w(k) = f(\boldsymbol{\lambda}(k)),\ k = 1, 2, \cdots. \tag{6.26}$$

Since during the prediction procedure, the model is independent of camera-based pose estimation $\boldsymbol{z}_c(k)$ (**Section 4.4.1**), thus it can be regarded as another pose estimator which could be further fused with $\boldsymbol{z}_c(k)$.

Updating: Suppose the existing prediction vector $\beta^{(0)}$ is obtained by initial training data set $\boldsymbol{D}^{(0)}$ composed of $\boldsymbol{\Lambda}$ and $\boldsymbol{Z}_c$ with N_0 distinct sample pairs. The expression of $\beta^{(0)}$ based on **Eq. 6.25** could be rewritten as

$$\beta^{(0)} = \left(\left(\Phi^{(0)}\right)^{\mathrm{T}}\Phi^{(0)}\right)^{-1}\left(\Phi^{(0)}\right)^{\mathrm{T}}\boldsymbol{Z}_c = \left(\boldsymbol{K}^{(0)}\right)^{-1}\left(\Phi^{(0)}\right)^{\mathrm{T}}\boldsymbol{Z}_c. \tag{6.27}$$

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

$$\beta^{(1)} = \begin{bmatrix} \Phi^{(0)} \\ \Phi^{(1)} \end{bmatrix}^{+} \begin{bmatrix} \boldsymbol{Z}_c^{(0)} \\ \boldsymbol{Z}_c^{(1)} \end{bmatrix} = \beta^{(0)} + \left(\mathrm{K}^{(1)}\right)^{-1}\left(\Phi^{(1)}\right)^{\mathrm{T}}\left(\boldsymbol{Z}_c^{(1)} - \Phi^{(1)}\beta^{(0)}\right), \tag{6.28}$$

where

$$\boldsymbol{K}^{(1)} = \begin{bmatrix} \Phi^{(0)} \\ \Phi^{(1)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \Phi^{(0)} \\ \Phi^{(1)} \end{bmatrix} = \boldsymbol{K}^{(0)} + \left(\Phi^{(1)}\right)^{\mathrm{T}}\Phi^{(1)}. \tag{6.29}$$

Following this iteration, the ELM model would be updated after the k^{th} training data set $\boldsymbol{D}^{(k)}$ as follows [198]:

$$\beta^{(k)} = \beta^{(k-1)} + \left(\mathrm{K}^{(k)}\right)^{-1}\left(\Phi^{(k)}\right)^{\mathrm{T}}\left(Z_c^{(k)} - \Phi^{(k)}\beta^{(k-1)}\right), \tag{6.30}$$

where

$$\boldsymbol{K}^{(k)} = \boldsymbol{K}^{(k-1)} + \left(\Phi^{(k)}\right)^{\mathrm{T}}\Phi^{(k)}. \tag{6.31}$$

The reduction of old data's effect in the update procedure of ELM model can be achieved by introducing and adjusting several weight parameters for the old measurements.

Eq. 6.30 can be expressed in the form of

$$\beta^{(k)} = \left(\begin{bmatrix} \Phi^{(k-1)} \\ \Phi^{(k)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \Phi^{(k-1)} \\ \Phi^{(k)} \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi^{(k-1)} \\ \Phi^{(k)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Z_c^{(k-1)} \\ Z_c^{(k)} \end{bmatrix} = \boldsymbol{H}^{(k)} \boldsymbol{M}^{(k)}, \quad (6.32)$$

where

$$\boldsymbol{H}^{(k)} = \left[\left(\Phi^{(k-1)} \right)^{\mathrm{T}} \Phi^{(k-1)} + \left(\Phi^{(k)} \right)^{\mathrm{T}} \Phi^{(k)} \right]^{-1}, \quad (6.33)$$

$$\boldsymbol{M}^{(k)} = \left(\Phi^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k-1)} + \left(\Phi^{(k)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k)}. \quad (6.34)$$

Weighting w is added to the variables that relate to old training samples, thus the two factors **Eq. 6.33** and **Eq. 6.34** will be

$$\widehat{\boldsymbol{H}}^{(k)} = w \left[\left(\Phi^{(k-1)} \right)^{\mathrm{T}} \Phi^{(k-1)} + \left(\Phi^{(k)} \right)^{\mathrm{T}} \Phi^{(k)} \right]^{-1}, \quad (6.35)$$

$$\widehat{\boldsymbol{M}}^{(k)} = w \left(\Phi^{(k-1)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k-1)} + \left(\Phi^{(k)} \right)^{\mathrm{T}} \boldsymbol{Z}_c^{(k)}. \quad (6.36)$$

The recursive expression of **Eq. 6.35** can be obtained by Sherman-Morrison formula as [199]

$$\widehat{\boldsymbol{M}}^{(k)} = \frac{\widehat{\boldsymbol{M}}^{(k-1)}}{w} - \frac{\boldsymbol{N}^{(k)} \left(\boldsymbol{N}^{(k)} \right)^{\mathrm{T}}}{w \left[w + \Phi^{(k)} \boldsymbol{N}^{(k)} \right]}, \quad (6.37)$$

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

As the training for FBG-based pose estimator is completed online, there is no prior data collection required. The number of samples for model initialization N_0 can be customized based on the prediction accuracy and computation time in offline training test. Details will be introduced in the experimental result **Section 6.6.2**.

6.4.3 Camera-FBG Sensing Fusion

In the ORB-SLAM2, the reprojection error e_S can reflect the motion estimation accuracy (introduced in **Section 4.4.2**). The fusion result can be regarded as a combination of SLAM- and FBG-based portions, with an adjustable weighting that characterize 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.

$$\boldsymbol{z} = \begin{cases} \boldsymbol{z}_c, & e_S \leq E_L \\ K_S\left(e_S - E_L\right)\boldsymbol{z}_w - \left[1 - K_S\left(e_S - E_L\right)\right]\boldsymbol{z}_c, & E_L < e_S < E_U \\ \boldsymbol{z}_w, & e_S \geq E_U \end{cases}, \tag{6.38}$$

where E_L and E_U are the error bounds distinguishing to entirely trust and discard SLAM estimation, respectively, K_S is an adjusting factor. This final weighted linear combination of two estimations is a loosely-coupled approach. However, complex fusion has already been implicitly considered during the online training and updating of the FBG-based model **Section 6.4.2**, since the ELM model utilizes camera-based pose estimation as the model input to refine the FBG-based one. It is not necessary to find a tightly-coupled approach aiming to increase the last-step fusion complexity, as the linear weight defined by projection error e_S has been sufficient to balance this combination. A tightly-coupled approach may increase the estimation accuracy partly, nevertheless, the increased computational burden should also be taken into account during this evaluation.

6.5 Hybrid Position Control of Continuum Robots

With real-time learning-based sensing feedback available, the design of continuum robot controller can also leverage the advantages of data-driven refinement. The work introduced in **Section 4.2** has utilized online-learning Gaussian process regression (GPR) for the visual servoing control. Such pure data-driven control has encouraging potential in soft robots, but is subject to time-consuming data-exploration procedures. Meanwhile, although the analytical kinematic 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 kinematic mapping. The combination of kinematic model-based and learning-based approaches could utilize 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, an 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.

6.5.1 Kinematics Initialization by CC Model

In consideration of the limited dimension of actuation space $\mathbb{U}^3$ comparing 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 $\boldsymbol{p}$. The robot kinematics can be initialized based on the CC assumption, which has been introduced in **Section 3.2**. The lengths of three fluidic chambers (**Fig. 3.5**) are denoted as $\boldsymbol{u} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^{\mathrm{T}}$, which could represent the configuration parameters r, θ, and ϕ as in **Eq. 3.19**. The configuration parameters can further determine the end-effector position by the forward kinematics analyzed in **Eq. 3.2**, thus the differential inverse kinematic function **Eq. 3.22** is available. 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 [200]:

To accommodate additional constraints into the motion generation for redundant robots, e.g., maintain the closest status to a preferred configuration, this scheme could be extended by an additional factor as [200]:

$$\Delta \boldsymbol{u}^{(k+1)} = \boldsymbol{J}^{\dagger} \Delta \boldsymbol{p}^{*(k+1)} + \left(1 - \boldsymbol{J}^{\dagger} \boldsymbol{J}\right) \cdot \beta \left(\boldsymbol{u}_0 - \boldsymbol{u}^{(k)}\right), \tag{6.39}$$

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

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

$$\boldsymbol{u}^{(k+1)} = K_p \Delta \boldsymbol{u}^{(k+1)} + \boldsymbol{u}^{(k)}, \tag{6.40}$$

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

$$\widehat{\boldsymbol{u}}^{(k+1)} = \boldsymbol{\alpha} \cdot \left(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}_0\right), \tag{6.41}$$

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 steady errors in the robot control. Thus a learning-based component to compensate tracking deviations is investigated in the following section.

6.5.2 Online Data-driven Error Compensator using GP

Online update of an additional error compensator enables the controller to compensate steady errors and even adapt with mechanical property changes, e.g., material fatigue. Once the actuation change $\Delta \boldsymbol{u}(k)$ is executed in the new step, the corresponding actual motion vector $\Delta \boldsymbol{p}(k)$ could be estimated as in **Section 4.4.1**. Thereby, a set of new sample pair including input $\boldsymbol{x} = \left[\boldsymbol{u}(k-1)^{\mathrm{T}}, \Delta \boldsymbol{z}(k)^{\mathrm{T}}\right]^{\mathrm{T}}$ and output $\boldsymbol{y} = \Delta \boldsymbol{p}_e(k)$ will be produced. Here the variable $\Delta \boldsymbol{p}_e(k)$ represents the difference between desired motion and actual motion, i.e.,

$$\Delta \boldsymbol{p}_e = \Delta \boldsymbol{p}^* - \Delta \boldsymbol{p}_r, \tag{6.42}$$

where $\Delta \boldsymbol{p}_r$ is the actual motion in task space corresponding to the desired $\Delta \boldsymbol{p}^*$. The purpose of the 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

$$\Delta \tilde{\boldsymbol{p}}^* = \Delta \boldsymbol{p}^* + \Delta \boldsymbol{p}_e \cdot |\Delta \boldsymbol{p}^*| . \tag{6.43}$$

This newly collected sample could reflect the latest robot mechanical status, and be added into the model training dataset, i.e., input matrix $\boldsymbol{X}$ and output $\boldsymbol{Y}$. GPR is utilized here for the model training. The working principle of GPR has been introduced in [190]. 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 $[\boldsymbol{x}_1, \boldsymbol{y}_1]$ 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. Actual incremental displacement $\Delta \boldsymbol{p}_r$ can be obtained at each step, thus the error to be compensated is thus calculated (**Eq. 6.42**). After several motion steps N_c ($\leq N_r$), the GPR compensator will be initialized and updated in the following steps. For time step $k > N_c$, an additional compensated component is added as in **Eq. 6.43**. The total control scheme of this project is depicted in **Fig. 6.14**, including the sensing and control units.

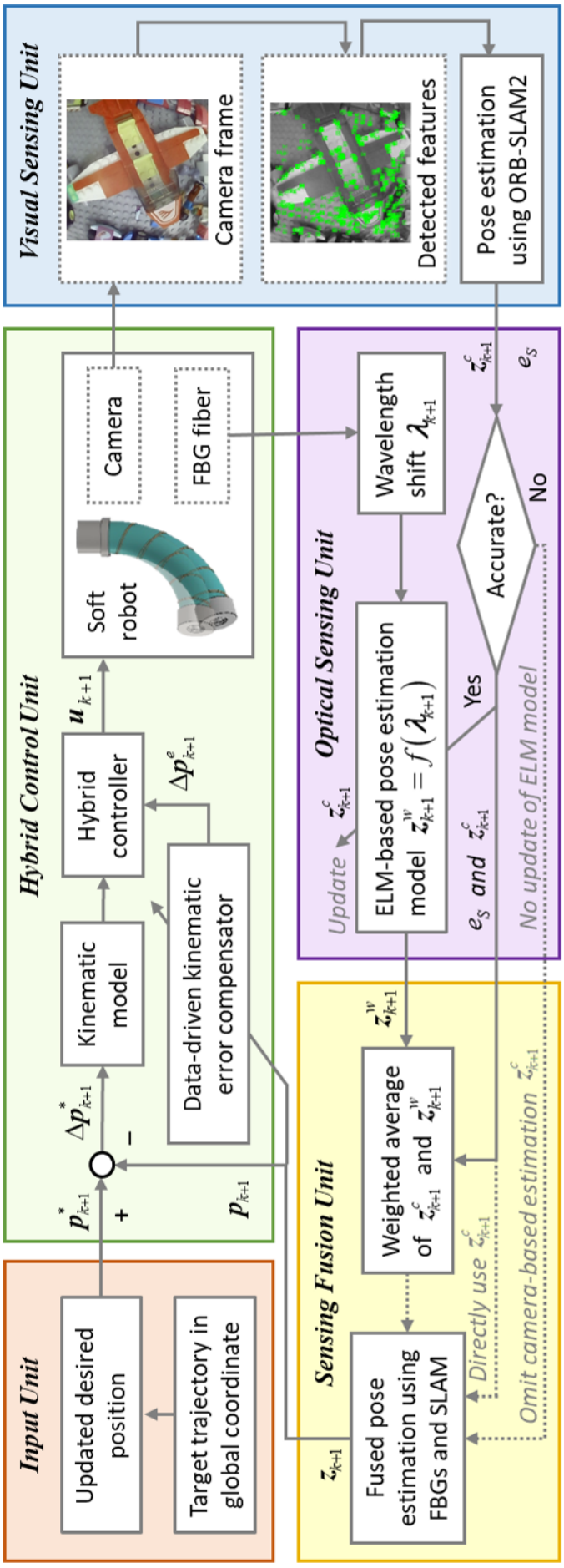


Fig. 6.14: Robot control architecture. Hybrid controller combining kinematic model and data-driven trained compensator is implemented, with the pose feedback obtained by the sensing fusion of strain measurement and visual feedback. Mono-camera ORB-SLAM2 is used, which serves as the ground truth to initialize and update the ELM-based model using FBG strain data. The FBG-estimated proportion in the sensing fusion will be more heavily weighted in visual feature-deficient scenarios.

6.6 Experiments of 6D Pose Estimation and Hybrid Control

This section demonstrates the results of optic-vision fused pose estimation, as well as hybrid control. 6D image stitching was also conducted, which could situate the ROS-captured images successively based on the pose fusion. The online learning-based compensator in the hybrid controller could adjust the untuned kinematic model distinctly in the tracking task. The continuum robot customized to validate the proposed control framework was moulded by silicone rubber (Ecoflex30, Smooth-on Inc.), softer than the robot used in **Section 6.3** (**Fig. 6.4**). The more "floppy" the robot is, the more difficult for stable control. Other structure parameters were as same as that of the robot in **Section 6.3**.

6.6.1 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 $\boldsymbol{U}$ (**Fig. 6.15**) 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.

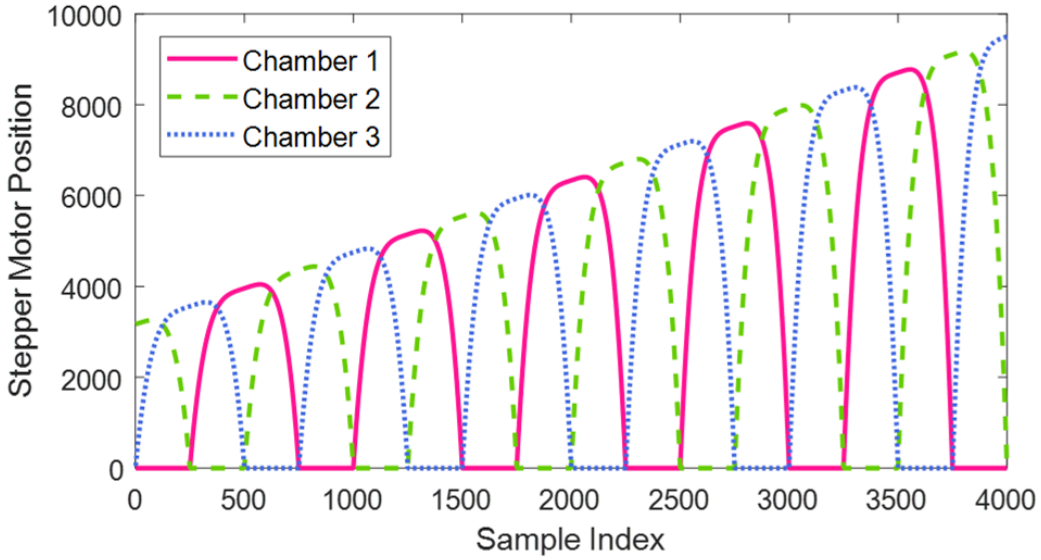


Fig. 6.15: Command sequence $\boldsymbol{U}$ to the actuation unit (three sets of stepper motors adjusting the volume of pneumatic chambers via cylinders) during the data-exploration.

A pair of EM trackers (as the ground truth) were attached on the robot tip to record the actual positions. Intrinsic calibration of the mono-camera 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 are ready. With the robot actuated by sequence $\boldsymbol{U}$ with N steps, the set of SLAM estimation $\widehat{\boldsymbol{Z}}_c = \begin{bmatrix} \widehat{\boldsymbol{z}}_c(1) & \widehat{\boldsymbol{z}}_c(2) & \cdots & \widehat{\boldsymbol{z}}_c(N) \end{bmatrix}$ and EM tracker measurement $\boldsymbol{Z}_g = \begin{bmatrix} \boldsymbol{z}_g(1) & \boldsymbol{z}_g(2) & \cdots & \boldsymbol{z}_g(N) \end{bmatrix}$ could be obtained.

Position recorded by the EM trackers was recorded and represented by $\boldsymbol{p}_g$, as the ground truth to calibrate and evaluate the measurement of SLAM $\boldsymbol{p}_c$. The affine transformation from $\boldsymbol{P}_g$ to $\widehat{\boldsymbol{P}}_c$ could be obtained, thus the SLAM position measurement is calibrated as

$$\boldsymbol{p}_c = \mathbf{R}\widehat{\boldsymbol{p}}_c \cdot \boldsymbol{k} + \widehat{\boldsymbol{p}}, \tag{6.44}$$

where $\mathbf{R}$, $\boldsymbol{k}$ 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, 0.596 and 0.385 mm, while the root mean square error (RMSE) was 0.998 mm (**Fig. 6.16a**). The trajectories constructed by $\boldsymbol{P}_c$ and $\boldsymbol{P}$ could be found in **Fig. 6.16b**. As can be seen, in conditions of abundant visual 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 6D image stitching of scenario is possible (**Fig. 6.16c**). 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.

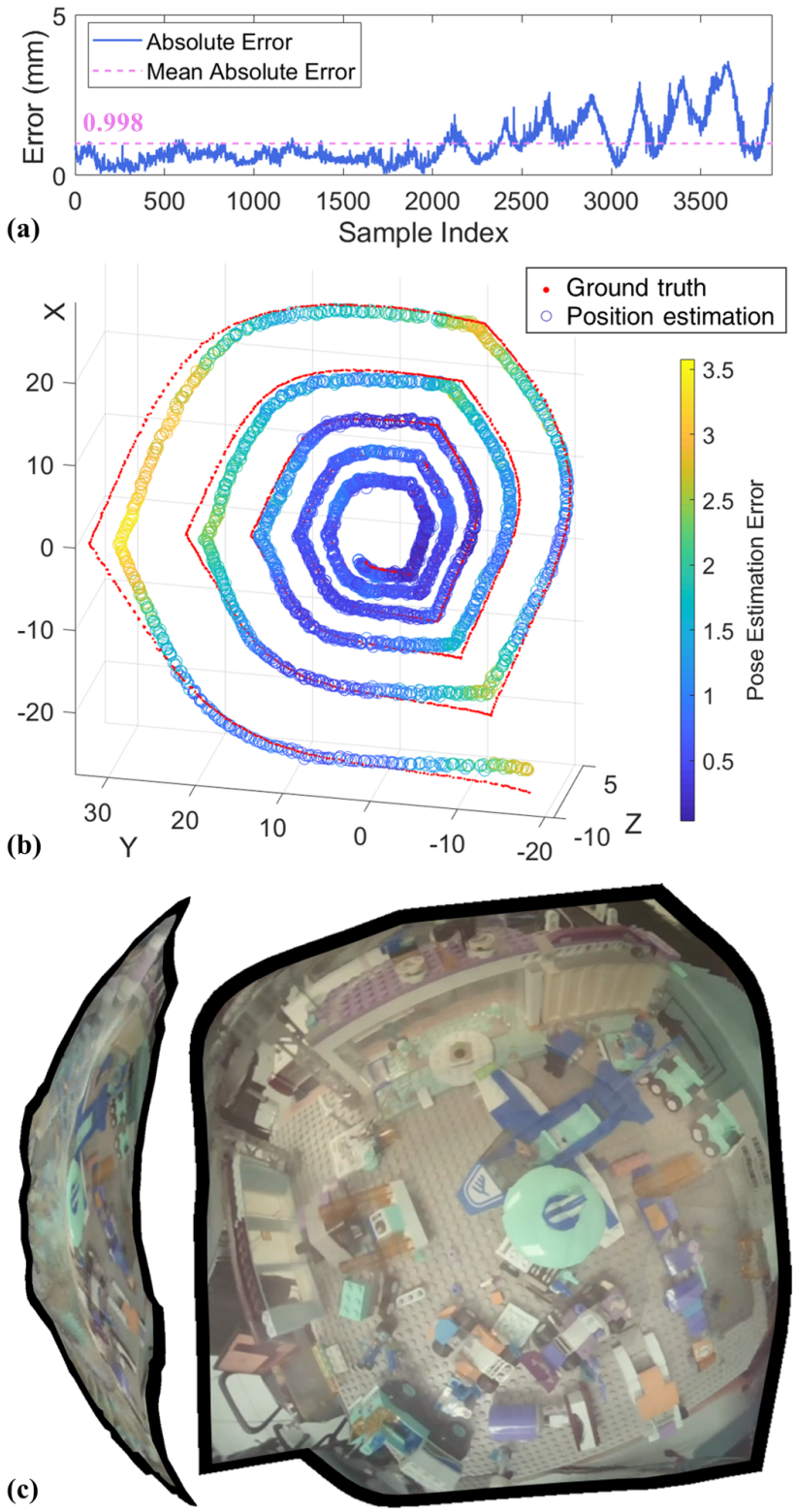


Fig. 6.16: Camera-based pose estimation results, where the SLAM-based estimation was compared with the EM-tracker measurement (ground truth). **(a)** Pose estimation errors as well as the mean value. **(b)** Trajectories recorded by EM trackers (red) and estimated by ORB-SLAM2 ('o'). **(c)** Front and side views of the stitched images in 3D, which are reconstructed using the SLAM pose estimation and image feedback.

6.6.2 Sensor Fusion Pose Estimation

ELM was chosen to train and update the FBG-based pose estimation model in real-time, i.e., the model would not need to be trained and tested offline. With the same actuation sequence (**Fig. 6.15**) as in **Section 6.6.1**, 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 offline. A specific aspect of this ELM model is that, after its initialization using N_0 samples, the model will be updated online with newly obtained samples. This updating mechanism indicates all samples could not be divided as training, validation and testing datasets as usual, but training and testing datasets. Samples that were not used in the model-initialization could be regarded as testing data. In our offline test to determine N and N_0, 2,000 samples in total were pre-collected using similar sequence to **Fig. 6.15**. In order to achieve fast initialization online, and guarantee the model convergence, $N_0 = 450$ was first confirmed. Since in our online use, all points should be predicted in order, no cross-validation was performed, and the first 450 points were used for training/initialization, while all remaining samples were predicted and fed into the model successively. 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 prediction results were compared with the measurement using ORB-SLAM2 which was the benchmark in training. The trajectory reconstructed by ELM estimation online approached that of the SLAM measurement closely. The mean estimation errors along x, y and z axes of ELM were 1.82×10^{-4}, 3.95×10^{-4} and 4.39×10^{-4} mm, respectively, while the mean spatial error was 8.28×10^{-4} mm. This result demonstrates that ELM is capable for 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, as denoted by the section headers.

It is worth to be noticed that replacing the ORB-SLAM2 algorithm with other kind of positional measurements (e.g. EM trackers measurements) will also enable successful training of this model. The reason to use ORB-SLAM2 here was to make the most of integrated components, and expand the range of application for this sensing modality. To test the effectiveness of fusion-based pose estimation when SLAM was unable to

achieve consistent and stable estimation, I validated the sensing fusion methodology in two conditions with visual disturbances: **1)** moving obstacles and **2)** varying lighting.

6.6.2.1 Pose Estimation under Moving Obstacles

The robot was actuated with the similar predefined spiral sequence as in **Fig. 6.16**. Moving or stable obstacles would disturb in the camera view, examples of camera view and features are illustrated in **Fig. 6.17a**. In the first 2,000 steps (marked with ① in **Fig. 6.17a**), no disturbances in the camera view was applied. For the testing scenario, this period would guarantee around 600 features at each frame, and the mean error of SLAM and fusion were 0.840 mm and 0.768 mm, respectively. After that, a hand appeared statically in front of the camera, or moving quickly to partly shield the field of view, but would not consistently cover the whole features (marked with ②-③). During these period, the number of features was less than or even reduced to 0 in rare frames, the mean SLAM error was 1.694 mm (max. 26.272 mm), and the fusion error was 1.132 mm (max. 2.573 mm). For the last case, the hand would statically cover the whole field of view (marked with ④) for $\sim$2 second each time. During this period, the visual features would consistently lose, thus the SLAM procedure would be stagnated (max. error 38.483 mm), but the fusion results could be maintained (max. error 4.747 mm). In the moving-obstacle period, there would also be moments (all the features in the camera view were blurred ③) that the SLAM estimation paused. However, in our sensing modality, the FBG-based 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. 6.17c-d**, the trajectory of fused pose would not be affected by the moving obstacles, while pure SLAM-based pose would be deteriorated as expected.

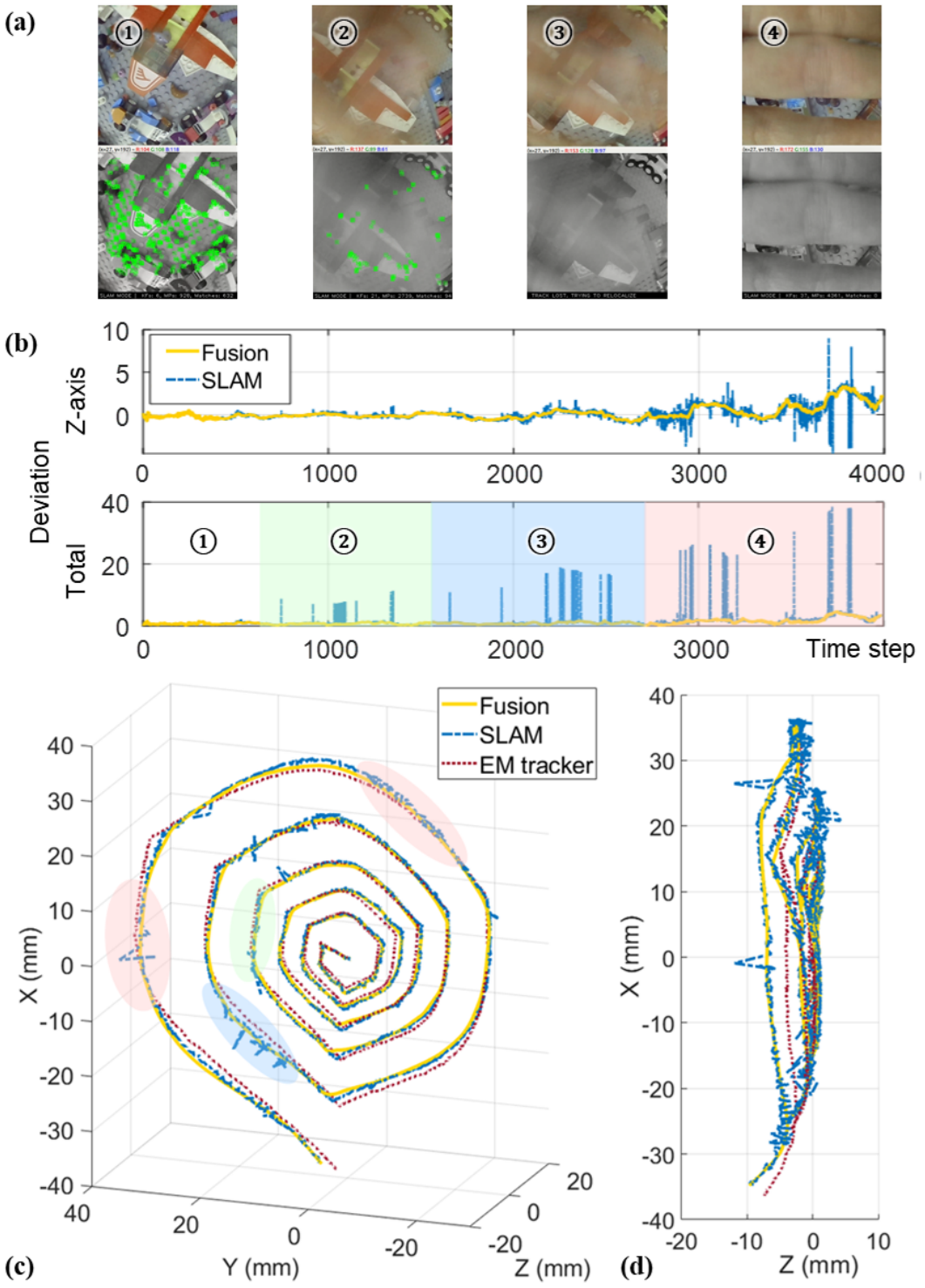


Fig. 6.17: Sensor fusion under moving obstacles. **(a)** Examples of camera view and corresponding visual features under ① LEGO®-constructed scenario with abundant visual features; ② moving/static hand where features were partly detected; ③ moving hand with no features detected; ④ static hand where all 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. **(c)** Trajectories of fusion-, SLAM- and tracker-obtained camera positions.

6.6.2.2 Pose Estimation under Varying Lighting Condition

Varying lighting conditions were also tested in the pose estimation experiment (**Fig. 6.18**). Under the lighting provided by incandescent lamp (marked with ① in **Fig. 6.18a**), the robot started the same series of movement as in the previous **Section 6.6.2.1**).

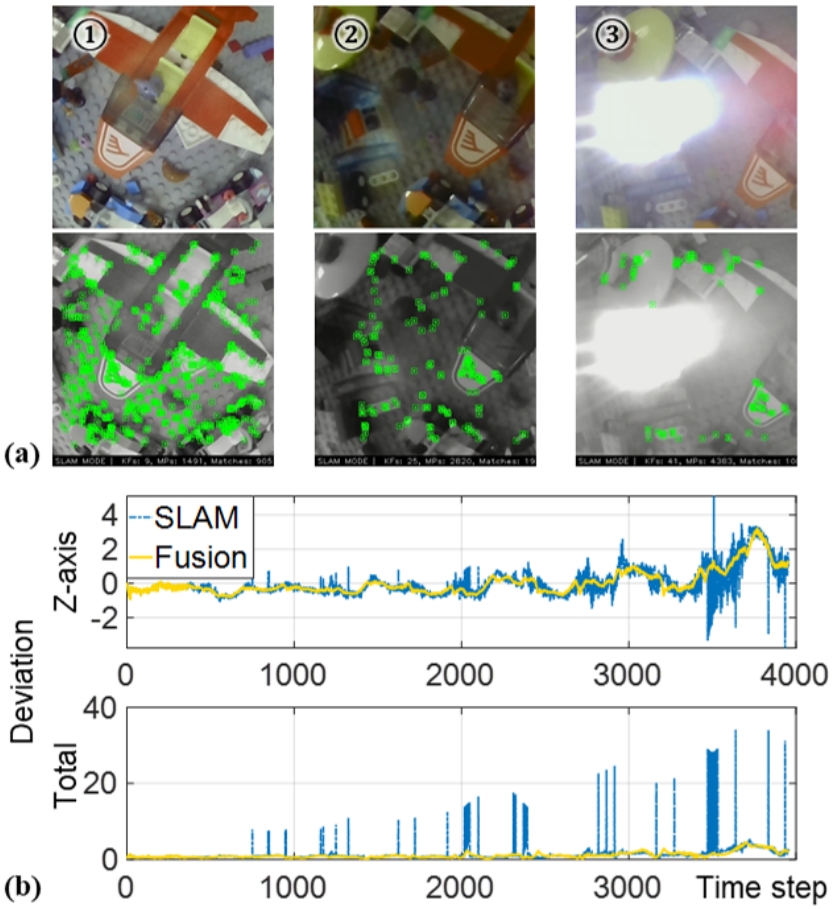


Fig. 6.18: Sensor fusion under varying lighting conditions. **(a)** Examples of camera view and corresponding visual features with ① ordinary lab lighting, ② low level lighting, and ③ moving additional LED lighting. **(b)** Deviations of SLAM-based and fusion-based pose estimation compared with the ground truth obtained by EM tracking.

During manipulation of the robot, the lighting condition was changed gradually or abruptly to weak lighting (marked with ②) or pure dark from time step 1,500 to 3,000, and then returning to the initial lighting. An additional moving LED source was tested from step 3,000 to 4,000, which was even oriented directly towards 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 noises in the SLAM-based pose estimation (**Fig. 6.18b**). However, once the lighting was removed in the camera view, 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. Fusion based estimation could keep a stable estimation level (RMSE: 1.324 mm), largely improving the SLAM estimation accuracy (RMSE: 3.116 mm).

Scenario reconstructions under conditions in section **Section 6.6.2.1** and **Section 6.6.2.2** were also conducted (**Fig. 6.19**). Poses for image stitching were provided by the sensing fusion result. The blurs caused by moving hand and varying lighting were reflected in the reconstruction, several of which are marked with 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.

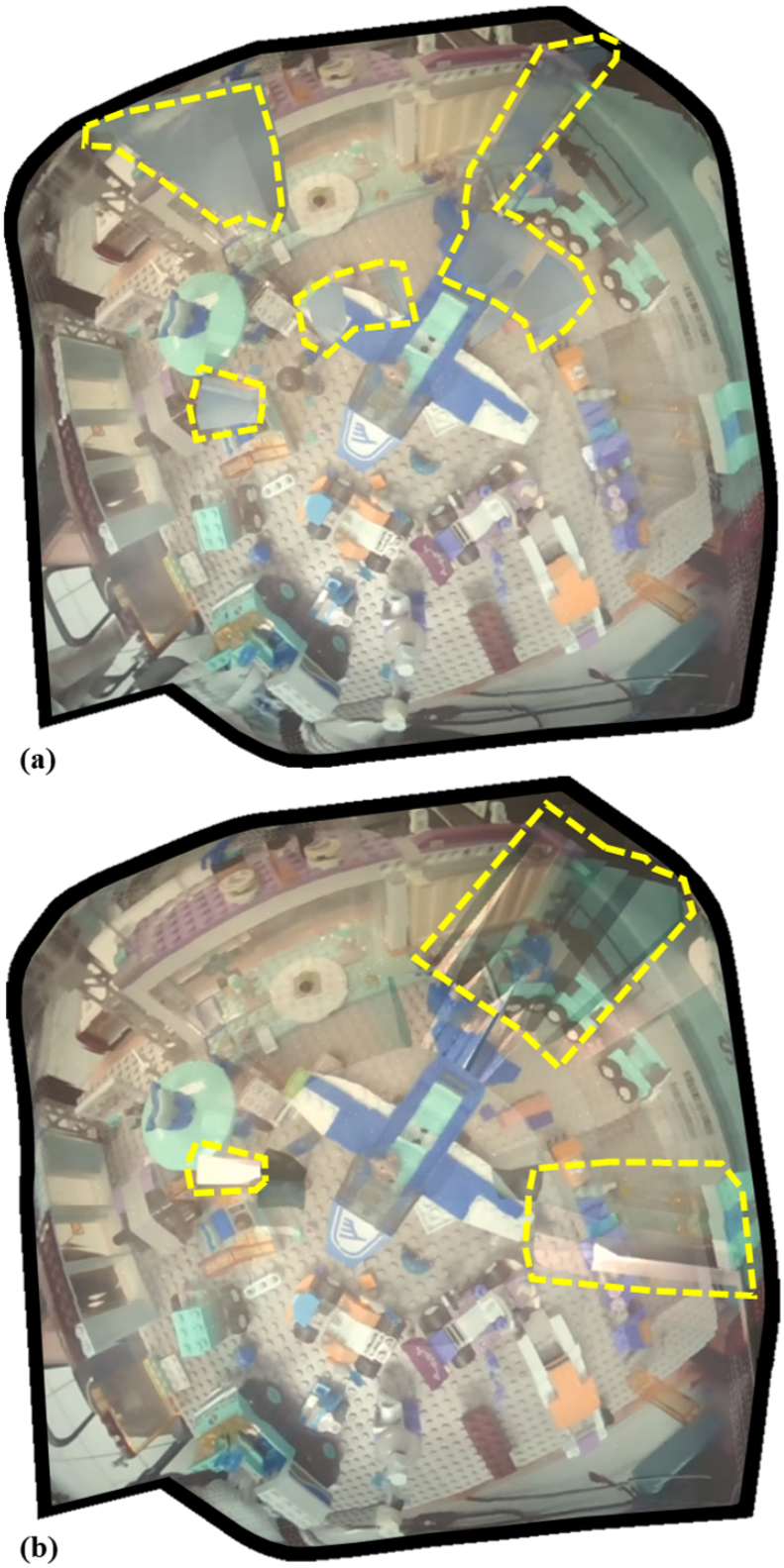


Fig. 6.19: Scenario reconstructions with disturbances of **(a)** moving obstacle (**Fig. 6.17**) and **(b)** varying lighting condition (**Fig. 6.18**). Several blurs due to the obstacle or varying lighting are indicated by dotted outlines.

6.6.3 Tracking: Hybrid Control vs Model-based Control

Control performances were tested by target tracking along two kinds of paths: *1)* pentagram trajectory combined with straight lines and sharp angles (**Fig. 6.20**); *2)* circle (**Fig. 6.21**). During this tracking task, the target was moving at a consistent speed. 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 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 **Eq. 6.44** 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 part could compensate for the tracking deviation under different values of K_p. The robot was actuated by model-based control at the beginning 2,000 or 1,500 steps (corresponding to 2 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 (step $1 \sim 2,000$ or $1 \sim 1,500$) and then hybrid control (step $2,001 \sim 4,000$ or $1,501 \sim 3,000$).

The density of points that construct the path is adjustable, and this further affects the step size of desired motions because the target switches at each actuation step successively. Targets were allocated dense, implying the step size could be set small. To reserve enough computation time, the control frequency was set as 20 Hz, i.e., the time step for motor actuation was 0.05 s. The distance that the end-effector could move during this period would be hampered by the stepper motor's output speed. In solely model-based control, if the tracking error is large, it should be resolved by coordinating choices of target density, motion step size, and motor output speed manually, as well as tuning parameters in the model. The proposed hybrid control aims to reduce the labor input in such circumvents, by compensating the tracking error automatically online.

1) Path with Sharp Angles

The performance of tracking the moving target along pentagram path is shown in **Fig. 6.20**. 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 actual end-effector position to the closest point on the desired path. When K_p was tuned as a proper value (**Fig. 6.20a**), the pure model-based controller's tracking performance was roughly acceptable. However, for the cases that K_p could not be tuned well (**Fig. 6.20b**), the model-based method was not capable of adapting to the speed of moving target, i.e., the "Error" obviously increased. Mean tracking errors and path deviations with different values of K_p are listed in **Table 6.1**. For these 4 cases, the hybrid controller could improve the performance ($68.9 \sim 86.4\%$ in tracking error, $65.8 \sim 80.9\%$ 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.

Table 6.1: Mean tracking errors and path deviations in *pentagram* tracking with four different gain K_p.

K_p	Tracking error			Path deviation		
	Model	Hybrid	**Improvement**	Model	Hybrid	**Improvement**
0.04	16.942	2.305	**86.4%**	3.646	0.696	**80.9%**
0.06	9.351	1.461	**84.4%**	2.252	0.492	**78.2%**
0.08	4.541	1.050	**76.9%**	1.461	0.384	**73.7%**
0.10	2.986	0.929	**68.9%**	1.057	0.361	**65.8%**

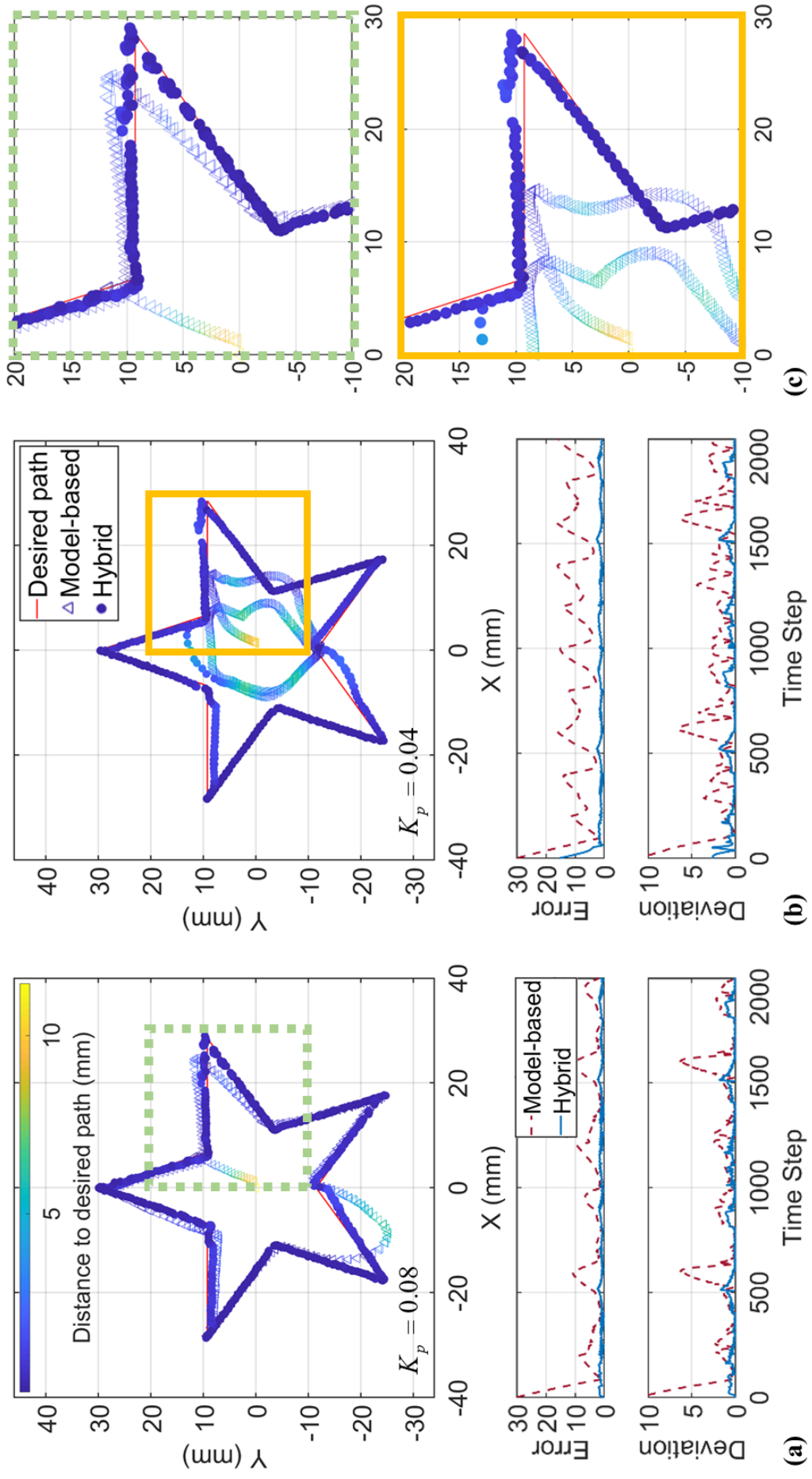


Fig. 6.20: Control performance of tracking a moving target along a pentagram path. Tracking trajectories and errors with gain factor K_p as **(a)** 0.08 and **(b)** 0.04 are plotted. "Error" represents the Euclidean distance between the current target and the actual end-effector position. "Deviation" 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 2,000 time steps (each 0.05s), afterwards another 2,000 steps run with the hybrid control. Two series of 2,000 time steps data are plotted and overlaid for ease of comparison. **(c)** Zoom-in of the square blocks in **(a)** and **(b)**.

2) Smooth Path

Similarly, as in the pentagram path tracking, the performance to follow a circle (**Fig. 6.21**) is shown according to the same arrangement of **Fig. 6.20**. The difference with *1)* is that this path was constructed with smooth curves. Three apparent deviations can be observed when K_p was set as both 0.1 and 0.06 (**Fig. 6.21a-b**). This is due to the highly nonlinear mapping from stepper motor positions to elastic chamber increments. The rough linearization **Eq. 6.41** could not meet the non-constant change of factor α. However, even with such an insufficiently tuned kinematics for control, the online learning-based error compensator still enables enhancement of the tracking performance (**Fig. 6.21c**). When K_p was not fine-tuned, the compensated part could steer the end-effector to approach the desired path quickly (**Fig. 6.21b**). Both the tracking error and the path deviation in tracing results can be largely reduced ($52.3 \sim 90.1\%$ in tracking error, $78.7 \sim 94.1\%$ in path deviation, **Table 6.2** under 4 values of K_p. Both experiments in **Section 6.6.3** demonstrated that the hybrid control scheme enables tracking convergence and gradually-increased tracking accuracy without fine modeling tuning and data-exploration.

Table 6.2: Mean tracking errors and path deviations in *circle* tracking with four different gain K_p.

K_p	Tracking error			Path deviation		
	Model	Hybrid	**Improvement**	Model	Hybrid	**Improvement**
0.04	28.910	2.864	**90.1%**	22.043	1.311	**94.1%**
0.06	24.979	6.132	**75.5%**	15.101	2.476	**83.7%**
0.08	15.868	4.655	**70.7%**	6.211	1.294	**79.2%**
0.10	5.874	2.803	**52.3%**	2.087	0.445	**78.7%**

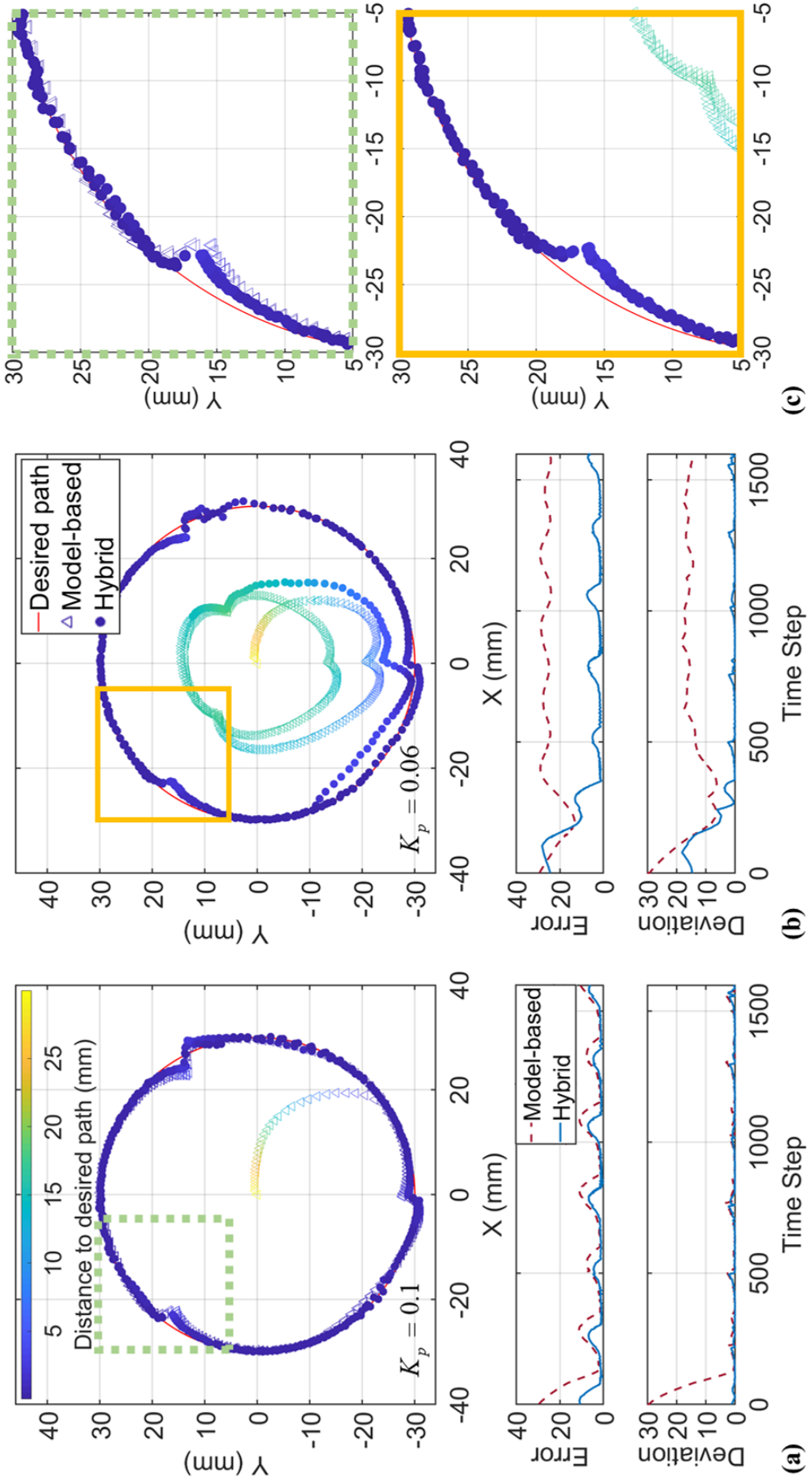


Fig. 6.21: Performance of tracking on a circle path, where K_p were set as **(a)** 0.1 and **(b)** 0.06. Subfigures are arranged as same as in **Fig. 6.20**.

It could be seen from these path tracking results that, the geometric complexity of predefined paths (**Fig. 6.20** and **Fig. 6.21**) did not affect the tracking accuracy directly. The main factors behind attribute to the proper selection of parameters (e.g., gain K_p). It also depends on whether the path segment would require frequent inflation switchover among chambers. The less switchover, the more robust tracking appears. As long as the targets are in a reachable range of robot actuation, the tracking accuracy will not be bounded by the actuation limit. This could also be found in the following experimental result (**Fig. 6.22**).

6.6.4 Hybrid Control with Sensing Fusion

An experiment integrating the visual-strain fusion-based pose estimation and hybrid controller was also conducted. The robot end-effector was instructed to track a complicated closed path in elephant contour (**Fig. 6.22**). The desired path involved 1,500 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 (step $1 \sim 17,00$). Considering the initial end-effector position $[0,0]$ was not on the path, 200 more time steps after the $1,500^{\text{th}}$ step was included in the validation of model-based control, to compensate for the initial approaching procedure. During this period, data collection and initialization of the GPR-based error compensator were also finished. Thus in the second cycle (step $1,701 \sim 3,200$), the robot was actuated utilizing hybrid control, which includes model-based control and the online-updated error compensator. In the final cycle (step 3: $201 \sim 4,700$), hybrid control continued being used, but moving obstacles were applied as visual disturbances in the camera view, similarly in **Section 6.6.2.1**. In this experiment, the robot-specific mapping **Eq. 6.41** was modified. Instead of a linear magnification, a nonlinear relation 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 (**Fig. 6.21a-b**) when switching inflated chambers could be eliminated. In the last cycle with visual disturbances, the tracking error (1.7506 mm) and path deviation (0.3365 mm) (**Fig. 6.22b-c**) could also maintain in a low level. This experiment demonstrates that the fusion-based pose estimation could provide valid feedback for the hybrid controller, or other controllers.

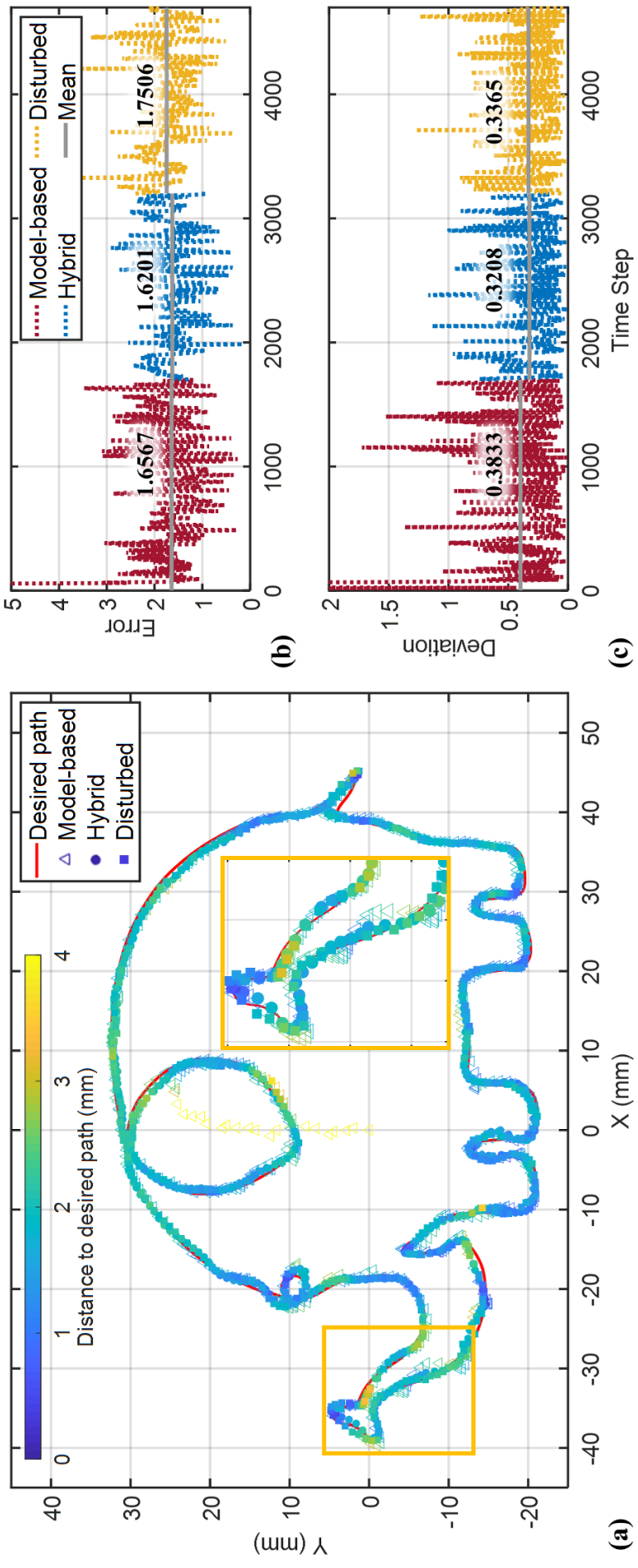


Fig. 6.22: Tracking performance along an elephant contour. The sensing feedback was provided by visual-strain pose fusion. CC model-based control was conducted for the 1st-cycle tracking, after which the learning-based error compensator was included. In the 3rd cycle, a moving obstacle applied visual disturbances in the camera view. Trajectories of the three cycles were plotted in **(a)**, as well as the tracking error **(b)** and the path deviation **(c)** with mean values indicated.

6.7 Conclusion

This chapter presents enhanced 2D motion and 6D pose estimation methods comparing with the soft robot visual servoing in **Chapter 4**. With the assistance of FBG sensors, the performance of the conventional template matching algorithm could be improved, especially under dim featured-lacking and force-disturbed conditions. The optical FBG fiber can be helically wrapped along the manipulator without requirements of assembly, providing an idea for increasing the capabilities of image processing in soft robotics by integrating with strain sensors. Sparse strain measurement is qualified to be an independent motion sensor, which can be combined with image processing to improve tracking accuracy. As for the 6D pose estimation, the FBG fiber can be trained as an online-updated pose estimator utilizing the SLAM-estimated pose measurement, and be further fused with SLAM for robust feedback of end-effector pose and 6D image stitching. Sensing accuracy and stability under extreme visual conditions such as moving obstacles and varying lighting conditions (even full shielding or absolute darkness) can be improved. A hybrid controller combining model-based kinematics and learning-based error compensators enables steady control in tracking tasks. Notice that feedback from the fiber can be observed as consistent. The fiber was wound along with the groove on robot surface, and completely adhered in its natural status. No significant axial strain should be introduced during this integration. Since the fiber-robot misalignment after multiple uses may destroy the measuring consistency, I took great notice of this matter during the fabrication. Moreover, in the FBG-based motion/pose estimation model, only relative wavelength shifts (the difference with initial wavelengths) are used as the input. Such wavelength shifts are correlated to the fiber characteristics and can be assumed consistent.

In the 2D visual servoing experiments, robot was actuated to follow a template in the endoscopic view along a “Batman” path, in a simulated laparoscopic scenario containing swine viscera. Another LEGO®-constructed scenario was also applied to validate the motion estimation, where the robot was tested with external forces. The enhanced estimation method increased tracking accuracy by 82.3% when capturing the dark and feature-deficient liver surface with a close distance (2-4 cm). Under external force, the enhanced method enabled to maintain the tracking in a stable state (RMSE: 24.75 pixels), while the tracking using image processing only would encounter an incremental tracking error (RMSE: 48.89 pixels in Cycle 2 and 107.63 pixels in Cycle 3) even after the force was released. The high sensing frequencies of FBG ($>$ 100 Hz) and camera ($\sim$ 30 Hz) will allow accurate

robot motion estimation even under high-frequency disturbances. Point-to-point tracking performance and the effect of reflective water in the endoscopic view are also tested.

In the 6D pose estimation and robot control, all the data-driven models used in the system can be conducted online, without prior data-collection. Sparse strain measurement along a single-core FBG fiber wrapped on the robot is qualified to be trained online as a camera pose sensor. SLAM estimation using the monocular camera on robot end-effector can be used for the FBG sensor training. The fusion result of SLAM and FBG can provide robust feedback of end-effector pose and accomplish 6D image stitching. Sensing accuracy and continuity under extreme visual conditions, such as moving obstacles and varying lighting conditions, are resolved, even when encountering full shielding or absolute darkness. The sensing fusion is immune to pauses in SLAM caused by poor feature quality on images. The mean estimation error could be increased and stabilized from RMSE 3.116 mm to 1.324 mm. In the aspect of control, the hybrid controller combining model-based kinematics and learning-based error compensator enables steady control in target tracking tasks. In the hybrid control, the learning-based compensator can reduce the tracking error by $> 80\%$. This controller can relax the requirement on modeling accuracy, and effectively accommodate un-modeled nonlinearity.

In our future work, the combination of vision and FBG strain sensing can be further explored. FBG fiber can be calibrated offline as a positional sensing device and integrated with a monocular camera to compose a visual-FBG soft-robot SLAM framework, similar to the VINS SLAM [201]. Well-calibrated FBGs can act as 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 re-localization, thus generating a new SLAM architecture for continuum robots. This visual-FBG SLAM system would have great potential to be used in endoscopic robot localization, navigation, and control.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

Data-driven approaches can facilitate the control and sensing of continuum robots, compensating for robot-specific modeling uncertainties and accommodating with external interactions. Model-based control algorithms can guarantee the convergence of inverse kinematic mapping, however, requiring time-consuming heuristic parameter tuning to increase the accuracy further. Learning-based approaches could train the desired relations by sensing data directly, which also brings forward the high demand for training data quality. The integration of model-based and data-driven methods will provide a solution for the balance between convergence and accuracy. Effective online update of kinematic models or error compensators can support the robot with fast adaption to the intrinsic and extrinsic uncertainties. The self-contained camera and flexible optical fiber also have much potential in the sensing modality for multi-application continuum robots.

The major achievements of the thesis are summarized as follows:

1. *Learning-based control for eye-in-hand visual servoing:* A non-parametric online learning control framework enables accurate visual servoing of a fluid-driven soft robot with very low stiffness. By estimating the inverse mapping solely from measured data, this controller alleviates the need for a kinematic model or camera calibration that may be challenging to acquire for soft manipulators. The locally weighted learning scheme supports the efficient online update of the inverse mapping, thus enabling precise robot manipulation even under external interactions. This work

first demonstrates a vision-based path following for a hyper-elastic robot with heavy variable loading (up to 105% of the robot weight). In the cycle with addition or removal of the payload, it maintained an acceptable accuracy (RMSE 14.4 pixels and maximum 54.1 pixels). Integrating image feedback with non-parametric learning can bring new opportunities to MIPs, such as continuum robotic endoscopy and laparoscopy, as they can take advantage of existing camera feedback.

2. *Experimental validation of a teleoperated robotic catheter platform*: The proposed experimental validation platform, comprising an MR Safe catheterization robot and a human-robot interface, could realize both kinematic model-based and model-free methods for catheter control. Subject test that emulated an ablation task was conducted to evaluate the performance of both control methods. Accuracy indices (e.g., mean of the closest distance between catheter tip and the closest target during ablation) and efficiency indices (e.g., proportion of missed lesion target) were adopted. The model-free control method had advantages compared to the model-based one in both aspects of accuracy (19.1% improvement in the tip-to-target ablation distance) and efficiency (35.8% reduction in the missed-target proportion and 46.2% reduction in the number of continuously missed targets). Additionally, the shape tracking and feedback control method for cardiac EP catheters was also validated using this platform, in a scenario that mimicked the clinical practice. A simulated PVI task with *ex-vivo* tissue ablation was conducted. Although in the LA simulator pumped with pulsatile-like liquid flow, the controller still enabled the catheter tip to target the desired position, and conduct RF ablation on the attached *ex-vivo* tissue.

3. *Shape tracking system and learning-based kinematic modeling method:* A shape tracking and feedback control method for slim continuum manipulators is presented. This sensing and control system was validated on a robotic catheterization platform integrated with a cardiac EP catheter. Information from multi-core FBG fiber and positional tracking coils were integrated to enable catheter shape estimation under MRI, which was subsequently incorporated with an MR safe catheter robot system. A learning-based modeling method is proposed for continuum manipulators, especially instruments like cardiac EP catheters, with the FBG shape tracking used for system characterization. Comparing with conventional CC models, the proposed modeling method can resolve the modeling uncertainties from heuristic parameter tuning and tendon backlash. Performances in a path following task demonstrated the increased

tracking accuracy of the proposed method. Autonomous targeting of five points within the robot workspace showed effective convergence rates (average 16.9 s).

4. *Enhanced eye-in-hand tracking incorporated optical FBG fiber*: The FBG-enhanced 2D motion estimation method in soft robot visual servo can improve the performance of conventional template matching algorithm under dim featured-lacking and force-disturbed conditions. The optical FBG fiber was helically wrapped along the manipulator without requirements of assembly, providing an idea for increasing the capabilities of image processing in soft robotics by integrating with strain sensors. Sparse strain measurement enables to be an independent motion sensor, which could be combined with image processing to improve tracking accuracy. In the experiments, the enhanced estimation method increased tracking accuracy by 82.3% when capturing the dark and feature-deficient liver surface with a close distance (2–4 cm). Under external force, the enhanced method enabled to maintain the tracking in a stable state (RMSE: 24.7509 pixels), while the tracking using image processing would encounter an incremental tracking error even after the force is released (RMSE: 48.8868 pixels to 107.6258 pixels).

5. *Learning-based visual-strain fusion for pose estimation and control*: FBG fiber can be trained as an online-updated positional sensor utilizing the SLAM-estimated 6D pose measurement, and be further fused with SLAM for robust feedback of end-effector pose and 6D image stitching. Sensing accuracy and stability under extreme visual conditions such as moving obstacles and varying lighting conditions (even full shielding or absolute darkness) can be improved. A hybrid controller combining model-based kinematics and learning-based error compensators enables steady control in tracking tasks. The training and update of the 6D pose estimator, as well as the error compensator, can be conducted online, without prior data-collection and parameter-tuning procedures. The fusion-based sensing system enables robust and continuous estimation feedback, immunity to the pause of SLAM due to poor feature quality. Estimation error could be increased and stabilized from RMSE 3.116 mm to 1.324 mm. In the hybrid control, the learning-based compensator can also reduce the tracking error by up to 80% (from 3.6540 mm to 0.6998 mm). This controller can reduce the requirement on modeling accuracy, and compensated for unmodeled nonlinearity distinctly.

7.2 Future Work

In our future work, the combination of vision and FBG strain sensing (**Chapter 6**) can be further explored. FBG fiber can be calibrated offline as a positional sensing device and integrated with a monocular camera to compose a visual-FBG soft-robot SLAM framework, similar to the VINS SLAM [201]. Well-calibrated FBGs can act as 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 re-localization, thus generating a new SLAM architecture for continuum robots. This visual-FBG SLAM system would have great potential to be used in endoscopic robot localization, navigation, and control.

For the robotic catheter platform in **Chapter 5**, pre-clinical animal trials will be conducted to validate the system for MRI-guided EP procedure. RF ablation will be conducted on a live porcine or ovine model with arrhythmia. The proposed shape tracking and the controller will be adapted to commercial MR-conditional systems equipped with an RF ablation system (e.g., ClearTrace^{TM}, MRI Interventions, Inc. or Imricor Medical System). The necrosis created by RF ablation, and its efficacy, will also be examined with post-mortem histology. The proposed catheter shape tracking method would provide sufficient feedback for visualization and robotic control under MRI, potentially reducing workload of the surgeon as well as post-procedural disease recurrence. In addition, the shape tracking method could be implemented on other MR safe/conditional continuum manipulators for interventional procedures, such as urologic surgery, ophthalmic surgery, and neurosurgery. The direct measurement and reconstruction of shape and position for continuum manipulators would reduce the computational cost of MRI, enabling high-performance feedback control and interactive instrument manipulation. It is believed that both safety and overall operational efficiency would be enhanced, with accurately and intra-operatively updated morphological information of interventional instruments.

References

[1] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics modeling of tendon-driven continuum manipulators," *IEEE transactions on robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.

[2] K.-H. Lee, D. K. Fu, M. C. Leong, M. Chow, H.-C. Fu, K. Althoefer, K. Y. Sze, C.-K. Yeung, and K.-W. Kwok, "Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation," *Soft robotics*, vol. 4, no. 4, pp. 324–337, 2017.

[3] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-octor," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2578–2585, 2005.

[4] A. J. Ishak and S. N. Mahmood, "Eye in hand robot arm based automated object grasping system," *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 555–566, 2019.

[5] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "Series pneumatic artificial muscles (spams) and application to a soft continuum robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5503–5510, 2017.

[6] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 57–62, 2013.

[7] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2016.

[8] F. Xu, H. Wang, W. Chen, and J. Wang, "Adaptive visual servoing control for an underwater soft robot," *Assembly Automation*, 2018.

[9] "Fiber bragg grating." Hittech, 2020. https://hittech.com/en/portfolio-posts/noria-the-fiber-bragg-grating-manufacturing-solution/, Accessed: 2020-07-09.

[10] 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 *2019 International Symposium on Medical Robotics (ISMR)*, pp. 1–6, 2019.

[11] R. Xu, A. Yurkewich, and R. V. Patel, "Curvature, torsion, and force sensing in continuum robots using helically wrapped fbg sensors," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1052–1059, 2016.

[12] "Atrial fibrillation corrected with ablation surgery." Nucleus Medical Media, 2020. https://catalog.nucleusmedicalmedia.com/atrial-fibrillation-corrected-with-ablation-surgery/view-item?ItemID=12714.

[13] E. Kholmovski, R. Ranjan, J. Silvernagel, J. Blauer, and N. Marrouche, "Assessment of acute cryo and rf ablation lesions by non-contrast and contrast enhanced mri techniques: Similarities and differences," *Circulation*, vol. 130, no. suppl_2, pp. A14649–A14649, 2014.

[14] S. Nakatani, M. J. Garcia, M. S. Firstenberg, L. Rodriguez, R. A. Grimm, N. L. Greenberg, P. M. McCarthy, P. M. Vandervoort, and J. D. Thomas, "Noninvasive assessment of left atrial maximum dp/dt by a combination of transmitral and pulmonary venous flow," *Journal of the American College of Cardiology*, vol. 34, no. 3, pp. 795–801, 1999.

[15] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[16] G. Robinson and J. B. C. Davies, "Continuum robots-a state of the art," in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 4, pp. 2849–2854, 1999.

[17] B. A. Jones and I. D. Walker, "Practical kinematics for real-time implementation of continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1087–1099, 2006.

[18] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[19] K. Cleary and T. M. Peters, "Image-guided interventions: technology review and clinical applications," *Annual review of biomedical engineering*, vol. 12, pp. 119–142, 2010.

[20] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006.

[21] Y. Bailly, Y. Amirat, and G. Fried, "Modeling and control of a continuum style microrobot for endovascular surgery," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 1024–1030, 2011.

[22] K. Ikuta, Y. Matsuda, D. Yajima, and Y. Ota, "Pressure pulse drive: A control method for the precise bending of hydraulic active catheters," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 5, pp. 876–883, 2011.

[23] J. Bishop-Moser and S. Kota, "Towards snake-like soft robots: Design of fluidic fiber-reinforced elastomeric helical manipulators," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5021–5026, 2013.

[24] G. Chen, M. T. Pham, and T. Redarce, "Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy," *Robotics and autonomous systems*, vol. 57, no. 6-7, pp. 712–722, 2009.

[25] G. Chen, M. T. Pham, and T. Redarce, "A guidance control strategy for semi-autonomous colonoscopy using a continuum robot," in *Recent Progress in Robotics: Viable Robotic Service to Human*, pp. 63–78, Springer, 2007.

[26] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, "A 3d steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm," *Bioinspiration & biomimetics*, vol. 7, no. 2, p. 025006, 2012.

[27] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 798–808, 2009.

[28] A. Shiva, A. Stilli, Y. Noh, A. Faragasso, I. De Falco, G. Gerboni, M. Cianchetti, A. Menciassi, K. Althoefer, and H. A. Wurdemann, "Tendon-based stiffening for a

pneumatically actuated soft manipulator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 632–637, 2016.

[29] J. B. C. Davies, D. Lane, G. Robinson, D. O'Brien, M. Pickett, M. Sfakiotakis, and B. Deacon, "Subsea applications of continuum robots," in *Proceedings of 1998 International Symposium on Underwater Technology*, pp. 363–369, 1998.

[30] T. Zheng, D. T. Branson, R. Kang, M. Cianchetti, E. Guglielmino, M. Follador, G. A. Medrano-Cerda, I. S. Godage, and D. G. Caldwell, "Dynamic continuum arm model for use with underwater robotic manipulators inspired by octopus vulgaris," in *2012 IEEE International Conference on Robotics and Automation*, pp. 5289–5294, 2012.

[31] T. Zheng, D. T. Branson, E. Guglielmino, R. Kang, G. A. Medrano Cerda, M. Cianchetti, M. Follador, I. S. Godage, and D. G. Caldwell, "Model validation of an octopus inspired continuum robotic arm for use in underwater environments," *Journal of Mechanisms and Robotics*, vol. 5, no. 2, 2013.

[32] G. Immega, "Tentacle-like manipulators with adjustable tension lines," June 7 1994. US Patent 5,317,952.

[33] O. Robotics, "Snake-arm robots access the inaccessible," *Nuclear Technology International*, vol. 1, pp. 92–94, 2008.

[34] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operation," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 3, pp. 2454–2459, 2001.

[35] T. Aoki, A. Ochiai, and S. Hirose, "Study on slime robot: development of the mobile robot prototype model using bridle bellows," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3, pp. 2808–2813, 2004.

[36] M. Shurrab, R. Schilling, E. Gang, E. M. Khan, and E. Crystal, "Robotics in invasive cardiac electrophysiology," *Expert review of medical devices*, vol. 11, no. 4, pp. 375–381, 2014.

[37] M. A. Tavallaei, Y. Thakur, S. Haider, and M. Drangova, "A magnetic-resonance-imaging-compatible remote catheter navigation system," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 899–905, 2012.

[38] M. A. Tavallaei, M. Lavdas, D. Gelman, and M. Drangova, "Magnetic resonance imaging compatible remote catheter navigation system with 3 degrees of freedom," *International journal of computer assisted radiology and surgery*, vol. 11, no. 8, pp. 1537–1545, 2016.

[39] D. T. Wallace, F. H. Moll, R. G. Younge, K. M. Martin, G. J. Stahler, D. F. Moore, D. T. Adams, M. R. Zinn, and G. D. Niemeyer, "Robotic catheter system," July 5 2011. US Patent 7,972,298.

[40] F. H. Moll, C. R. Carlson, F. Barbagli, D. T. Wallace, and D. Lundmark, "Robotic catheter system and methods," May 29 2012. US Patent 8,190,238.

[41] A. B. Slatkin, J. Burdick, and W. Grundfest, "The development of a robotic endoscope," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 2, pp. 162–171, 1995.

[42] H. Hoeg, A. B. Slatkin, J. W. Burdick, and W. S. Grundfest, "Biomechanical modeling of the small intestine as required for the design and operation of a robotic endoscope," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 1599–1606, 2000.

[43] T. Kato, I. Okumura, S.-E. Song, A. J. Golby, and N. Hata, "Tendon-driven continuum robot for endoscopic surgery: Preclinical development and validation of a tension propagation model," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2252–2263, 2014.

[44] Y. Chen, J. M. Oliveira, and I. W. Hunter, "Two-axis bend sensor design, kinematics and control for a continuum robotic endoscope," in *2013 IEEE International Conference on Robotics and Automation*, pp. 704–710, 2013.

[45] C.-K. Yeung, W.-L. Law, and K.-W. Kwok, "Endoscopic systems, devices, and methods for performing in vivo procedures," Mar. 30 2017. US Patent App. 14/985,587.

[46] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. F. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. J. Walsh, and G. M. Whitesides, "Pneumatic networks for soft robotics that actuate rapidly," *Advanced functional materials*, vol. 24, no. 15, pp. 2163–2170, 2014.

[47] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[48] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[49] H.-C. Fu, J. D. Ho, K.-H. Lee, Y. C. Hu, S. K. Au, K.-J. Cho, K. Y. Sze, and K.-W. Kwok, "Interfacing soft and hard: a spring reinforced actuator," *Soft Robotics*, vol. 7, no. 1, pp. 44–58, 2020.

[50] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[51] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, 2006.

[52] H. Wang, D. Guo, H. Xu, W. Chen, T. Liu, and K. K. Leang, "Eye-in-hand tracking control of a free-floating space manipulator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 1855–1865, 2017.

[53] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[54] F. Xu, H. Wang, Z. Liu, and W. Chen, "Adaptive visual servoing for an underwater soft robot considering refraction effects," *IEEE Transactions on Industrial Electronics*, 2019.

[55] F. Xu, H. Wang, J. Wang, K. W. S. Au, and W. Chen, "Underwater dynamic visual servoing for a soft robot arm with online distortion correction," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 979–989, 2019.

[56] Y. Lu, C. Zhang, S. Song, and M. Q.-H. Meng, "Precise motion control of concentric-tube robot based on visual servoing," in *2017 IEEE International Conference on Information and Automation (ICIA)*, pp. 299–304, 2017.

[57] 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 Transactions on Mechatronics*, vol. 24, no. 1, pp. 88–99, 2018.

[58] C. Laschi and M. Cianchetti, "Soft robotics: new perspectives for robot bodyware and control," *Frontiers in bioengineering and biotechnology*, vol. 2, p. 3, 2014.

[59] I. Boškoski and G. Costamagna, "Endoscopy robotics: current and future applications," *Digestive Endoscopy*, vol. 31, no. 2, pp. 119–124, 2019.

[60] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.

[61] M. M. Werneck, R. Allil, B. A. Ribeiro, and F. V. De Nazare, "A guide to fiber bragg grating sensors," *Current trends in short-and long-period fiber gratings*, pp. 1–24, 2013.

[62] A. Othonos, K. Kalli, and G. E. Kohnke, "Fiber bragg gratings: Fundamentals and applications in telecommunications and sensing," *PhT*, vol. 53, no. 5, p. 61, 2000.

[63] R. Kashyap, *Fiber bragg gratings*. Academic press, 2009.

[64] H. Liu, A. Farvardin, R. Grupp, R. J. Murphy, R. H. Taylor, I. Iordachita, and M. Armand, "Shape tracking of a dexterous continuum manipulator utilizing two large deflection shape sensors," *IEEE sensors journal*, vol. 15, no. 10, pp. 5494–5503, 2015.

[65] C. Shi, X. Luo, P. Qi, T. Li, S. Song, Z. Najdovski, T. Fukuda, and H. Ren, "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1665–1678, 2016.

[66] 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 transactions on mechatronics*, vol. 19, no. 4, pp. 1115–1126, 2013.

[67] 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.

[68] M. van der Heiden, K. Henken, L. Chen, B. van den Bosch, R. van den Braber, J. Dankelman, and J. van den Dobbelsteen, "Accurate and efficient fiber optical shape sensor for mri compatible minimally invasive instruments," in *Optical Systems Design 2012*, vol. 8550, p. 85500L, International Society for Optics and Photonics, 2012.

[69] Y.-L. Park, R. J. Black, B. Moslehi, M. R. Cutkosky, S. Elayaperumal, B. Daniel, A. Yeung, and V. Sotoudeh, "Steerable shape sensing biopsy needle and catheter," Feb. 11 2014. US Patent 8,649,847.

[70] K. Henken, D. Van Gerwen, J. Dankelman, and J. Van Den Dobbelsteen, "Accuracy of needle position measurements using fiber bragg gratings," *Minimally Invasive Therapy & Allied Technologies*, vol. 21, no. 6, pp. 408–414, 2012.

[71] P. Saccomandi, C. M. Oddo, L. Zollo, D. Formica, R. A. Romeo, C. Massaroni, M. A. Caponero, N. Vitiello, E. Guglielmelli, S. Silvestri, *et al.*, "Feedforward neural network for force coding of an mri-compatible tactile sensor array based on fiber bragg grating," *Journal of Sensors*, vol. 2015, 2015.

[72] J. P. Moore and M. D. Rogge, "Shape sensing using multi-core fiber optic cable and parametric curve solutions," *Optics express*, vol. 20, no. 3, pp. 2967–2973, 2012.

[73] P. S. Westbrook, T. Kremp, K. S. Feder, W. Ko, E. M. Monberg, H. Wu, D. A. Simoff, T. F. Taunay, and R. M. Ortiz, "Continuous multicore optical fiber grating arrays for distributed sensing applications," *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1248–1252, 2017.

[74] F. Alambeigi, S. A. Pedram, J. L. Speyer, J. Rosen, I. Iordachita, R. H. Taylor, and M. Armand, "Scade: Simultaneous sensor calibration and deformation estimation of fbg-equipped unmodeled continuum manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 222–239, 2019.

[75] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea, "Soft robotic grippers," *Advanced Materials*, vol. 30, no. 29, p. 1707035, 2018.

[76] H. In, B. B. Kang, M. Sin, and K.-J. Cho, "Exo-glove: A wearable robot for the hand with a soft tendon routing system," *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 97–105, 2015.

[77] T. Ranzani, G. Gerboni, M. Cianchetti, and A. Menciassi, "A bioinspired soft manipulator for minimally invasive surgery," *Bioinspiration & biomimetics*, vol. 10, no. 3, p. 035008, 2015.

[78] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[79] R. S. Penning, J. Jung, N. J. Ferrier, and M. R. Zinn, "An evaluation of closed-loop control options for continuum manipulators," in *2012 IEEE International Conference on Robotics and Automation*, pp. 5392–5397, 2012.

[80] M. Luo, Y. Pan, E. H. Skorina, W. Tao, F. Chen, S. Ozel, and C. D. Onal, "Slithering towards autonomy: a self-contained soft robotic snake platform with integrated curvature sensing," *Bioinspiration & biomimetics*, vol. 10, no. 5, p. 055001, 2015.

[81] E. L. White, J. C. Case, and R. Kramer-Bottiglio, "A soft parallel kinematic mechanism," *Soft robotics*, vol. 5, no. 1, pp. 36–53, 2018.

[82] D. Cao, D. Liu, and C. H.-T. Wang, "Three-dimensional nonlinear dynamics of slender structures: Cosserat rod element approach," *International Journal of Solids and Structures*, vol. 43, no. 3-4, pp. 760–783, 2006.

[83] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[84] D. C. Rucker and R. J. Webster, "Mechanics of continuum robots with external loading and general tendon routing," in *Experimental Robotics*, pp. 645–654, 2014.

[85] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5067–5074, 2015.

[86] F. Renda, V. Cacucciolo, J. Dias, and L. Seneviratne, "Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5495–5502, 2016.

[87] G.-Y. Gu, U. Gupta, J. Zhu, L.-M. Zhu, and X.-Y. Zhu, "Feedforward deformation control of a dielectric elastomer actuator based on a nonlinear dynamic model," *Applied Physics Letters*, vol. 107, no. 4, p. 042907, 2015.

[88] G.-Y. Gu, J. Zhu, L.-M. Zhu, and X. Zhu, "A survey on dielectric elastomer actuators for soft robots," *Bioinspiration & biomimetics*, vol. 12, no. 1, p. 011003, 2017.

[89] K.-H. Lee, M. C. Leong, M. C. Chow, H.-C. Fu, W. Luk, K.-Y. Sze, C.-K. Yeung, and K.-W. Kwok, "Fem-based soft robotic control framework for intracavitary

navigation," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 11–16, 2017.

[90] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2550–2555, 2015.

[91] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.

[92] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017.

[93] T. Li, K. Nakajima, and R. Pfeifer, "Online learning for behavior switching in a soft robotic arm," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1296–1302, 2013.

[94] T. G. Thuruthel, E. Falotico, M. Cianchetti, and C. Laschi, "Learning global inverse kinematics solutions for a continuum robot," in *Symposium on Robot Design, Dynamics and Control*, pp. 47–54, Springer, 2016.

[95] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 6, pp. 1147–1160, 2013.

[96] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[97] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Computed torque control with nonparametric regression models," in *2008 American Control Conference*, pp. 212–217, 2008.

[98] Y. Ansari, E. Falotico, M. Cianchetti, and C. Laschi, "Point-to-point motion controller for soft robotic manipulators," in *2016 IEEE International Conference*

on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), pp. 49–54, 2016.

[99] J. S. Kim and G. S. Chirikjian, "Conformational analysis of stiff chiral polymers with end-constraints," *Molecular Simulation*, vol. 32, no. 14, pp. 1139–1154, 2006.

[100] N. Simaan, K. Xu, W. Wei, A. Kapoor, P. Kazanzides, R. Taylor, and P. Flint, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *The International journal of robotics research*, vol. 28, no. 9, pp. 1134–1153, 2009.

[101] S. Hasanzadeh and F. Janabi-Sharifi, "Model-based force estimation for intracardiac catheters," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 154–162, 2015.

[102] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

[103] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[104] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[105] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 675–687, Springer, 2017.

[106] M. Maidi, F. Ababsa, and M. Mallem, "Vision-inertial tracking system for robust fiducials registration in augmented reality," in *2009 IEEE Symposium on Computational Intelligence for Multimedia Signal and Vision Processing*, pp. 83–90, 2009.

[107] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *European conference on computer vision*, pp. 15–29, Springer, 2012.

[108] B. Xing, Q. Zhu, F. Pan, and X. Feng, "Marker-based multi-sensor fusion indoor localization system for micro air vehicles," *Sensors*, vol. 18, no. 6, p. 1706, 2018.

[109] C. Feng, V. R. Kamat, and C. C. Menassa, "Marker-assisted structure from motion for 3d environment modeling and object pose estimation," in *Construction Research Congress 2016*, pp. 2604–2613, 2016.

[110] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation-an overview," in *Visualization of Large and Unstructured Data Sets-Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[111] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[112] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[113] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 815–830, 2009.

[114] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[115] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

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

[117] X. Du, T. Kurmann, P.-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated multi-instrument 2-d pose estimation using fully convolutional networks," *IEEE transactions on medical imaging*, vol. 37, no. 5, pp. 1276–1287, 2018.

[118] M. Runciman, A. Darzi, and G. P. Mylonas, "Soft robotics in minimally invasive surgery," *Soft robotics*, vol. 6, no. 4, pp. 423–443, 2019.

[119] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, p. e1875, 2018.

[120] F. Cosentino, E. Tumino, G. R. Passoni, E. Morandi, and A. Capria, "Functional evaluation of the endotics system, a new disposable self-propelled robotic colonoscope: in vitro tests and clinical trial," *The International journal of artificial organs*, vol. 32, no. 8, pp. 517–527, 2009.

[121] B. A. Childers, M. E. Froggatt, S. G. Allison, T. C. Moore Sr, D. A. Hare, C. F. Batten, and D. C. Jegley, "Use of 3000 bragg grating strain sensors distributed on four 8-m optical fibers during static load tests of a composite structure," in *Smart Structures and Materials 2001: Industrial and Commercial Applications of Smart Structures Technologies*, vol. 4332, pp. 133–142, International Society for Optics and Photonics, 2001.

[122] G. Flockhart, W. MacPherson, J. Barton, J. Jones, L. Zhang, and I. Bennion, "Two-axis bend measurement with bragg gratings in multicore optical fiber," *Optics letters*, vol. 28, no. 6, pp. 387–389, 2003.

[123] K. Mandal, F. Parent, S. Martel, R. Kashyap, and S. Kadoury, "Vessel-based registration of an optical shape sensing catheter for mr navigation," *International journal of computer assisted radiology and surgery*, vol. 11, no. 6, pp. 1025–1034, 2016.

[124] S. C. Ryu and P. E. Dupont, "Fbg-based shape sensing tubes for continuum robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3531–3537, 2014.

[125] S. C. Ryu, Z. F. Quek, J.-S. Koh, P. Renaud, R. J. Black, B. Moslehi, B. L. Daniel, K.-J. Cho, and M. R. Cutkosky, "Design of an optically controlled mr-compatible active needle," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 1–11, 2014.

[126] N. J. van de Berg, J. Dankelman, and J. J. van den Dobbelsteen, "Design of an actively controlled steerable needle with tendon actuation and fbg-based shape sensing," *Medical engineering & physics*, vol. 37, no. 6, pp. 617–622, 2015.

[127] T. Li, C. Shi, and H. Ren, "Three-dimensional catheter distal force sensing for cardiac ablation based on fiber bragg grating," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2316–2327, 2018.

[128] K. O. Hill and G. Meltz, "Fiber bragg grating technology fundamentals and overview," *Journal of lightwave technology*, vol. 15, no. 8, pp. 1263–1276, 1997.

[129] Y.-L. Park, S. Elayaperumal, B. Daniel, S. C. Ryu, M. Shin, J. Savall, R. J. Black, B. Moslehi, and M. R. Cutkosky, "Real-time estimation of 3-d needle shape and deflection for mri-guided interventions," *IEEE/ASME Transactions On Mechatronics*, vol. 15, no. 6, pp. 906–915, 2010.

[130] 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 *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 469–476, 2013.

[131] R. J. Roesthuis, S. Janssen, and S. Misra, "On using an array of fiber bragg grating sensors for closed-loop control of flexible minimally invasive surgical instruments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2545–2551, 2013.

[132] R. J. Roesthuis and S. Misra, "Steering of multisegment continuum manipulators using rigid-link modeling and fbg-based shape sensing," *IEEE transactions on robotics*, vol. 32, no. 2, pp. 372–382, 2016.

[133] H. Rafii-Tari, C. J. Payne, and G.-Z. Yang, "Current and emerging robot-assisted endovascular catheterization technologies: a review," *Annals of biomedical engineering*, vol. 42, no. 4, pp. 697–715, 2014.

[134] S. B. Kesner and R. D. Howe, "Position control of motion compensation cardiac catheters," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1045–1055, 2011.

[135] R. T. Blanco, R. Ojala, J. Kariniemi, J. Perälä, J. Niinimäki, and O. Tervonen, "Interventional and intraoperative mri at low field scanner–a review," *European journal of radiology*, vol. 56, no. 2, pp. 130–142, 2005.

[136] R. Blanco Sequeiros, R. Ojala, J. Kariniemi, J. Perälä, J. Niinimäki, H. Reinikainen, and O. Tervonen, "Mr-guided interventional procedures: a review," *Acta Radiologica*, vol. 46, no. 6, pp. 576–586, 2005.

[137] N. Abi-Jaoudeh, J. Kruecker, S. Kadoury, H. Kobeiter, A. M. Venkatesan, E. Levy, and B. J. Wood, "Multimodality image fusion–guided procedures: technique, accuracy, and applications," *Cardiovascular and interventional radiology*, vol. 35, no. 5, pp. 986–998, 2012.

[138] S. G. Hushek, A. J. Martin, M. Steckner, E. Bosak, J. Debbins, and W. Kucharzyk, "Mr systems for mri-guided interventions," *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 27, no. 2, pp. 253–266, 2008.

[139] G. A. Tung and L. M. Davis, "The role of magnetic resonance imaging in the evaluation of the soft tissue mass.," *Critical reviews in diagnostic imaging*, vol. 34, no. 5, p. 239, 1993.

[140] L. Jäger and M. Reiser, "Ct and mr imaging of the normal and pathologic conditions of the facial nerve," *European journal of radiology*, vol. 40, no. 2, pp. 133–146, 2001.

[141] J. Y. Cheng, T. Zhang, M. T. Alley, M. Uecker, M. Lustig, J. M. Pauly, and S. S. Vasanawala, "Comprehensive multi-dimensional mri for the simultaneous assessment of cardiopulmonary anatomy and physiology," *Scientific reports*, vol. 7, no. 1, pp. 1–15, 2017.

[142] E. M. Pedersen, E. V. Stenbøg, T. Fründ, K. Houlind, O. Kromann, K. E. Sørensen, K. Emmertsen, and V. E. Hjortdal, "Flow during exercise in the total cavopulmonary connection measured by magnetic resonance velocity mapping," *Heart*, vol. 87, no. 6, pp. 554–558, 2002.

[143] B. Quesson, J. A. de Zwart, and C. T. Moonen, "Magnetic resonance temperature imaging for guidance of thermotherapy," *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 12, no. 4, pp. 525–533, 2000.

[144] F. G. Shellock, "Metallic surgical instruments for interventional mri procedures: evaluation of mr safety," *Journal of Magnetic Resonance Imaging*, vol. 13, no. 1, pp. 152–157, 2001.

[145] Y. Feng, Z. Guo, Z. Dong, X.-Y. Zhou, K.-W. Kwok, S. Ernst, and S.-L. Lee, "An efficient cardiac mapping strategy for radiofrequency catheter ablation with active learning," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 7, pp. 1199–1207, 2017.

[146] R. K. Mukherjee, S. Roujol, H. Chubb, J. Harrison, S. Williams, J. Whitaker, L. O'Neill, J. Silberbauer, R. Neji, R. Schneider, *et al.*, "Epicardial electroanatomical mapping, radiofrequency ablation, and lesion imaging in the porcine left ventricle

under real-time magnetic resonance imaging guidance—an in vivo feasibility study," *Ep Europace*, vol. 20, no. FI2, pp. f254–f262, 2018.

[147] E. J. Schmidt, R. P. Mallozzi, A. Thiagalingam, G. Holmvang, A. d'Avila, R. Guhde, R. Darrow, G. S. Slavin, M. M. Fung, J. Dando, *et al.*, "Electroanatomic mapping and radiofrequency ablation of porcine left atria and atrioventricular nodes using magnetic resonance catheter tracking," *Circulation: Arrhythmia and Electrophysiology*, vol. 2, no. 6, pp. 695–704, 2009.

[148] M. E. Josephson, *Clinical cardiac electrophysiology: techniques and interpretations*. Lippincott Williams & Wilkins, 2008.

[149] F. Morady, "Radio-frequency ablation as treatment for cardiac arrhythmias," *New England Journal of Medicine*, vol. 340, no. 7, pp. 534–544, 1999.

[150] M. Haissaguerre, P. Jaïs, D. C. Shah, A. Takahashi, M. Hocini, G. Quiniou, S. Garrigue, A. Le Mouroux, P. Le Métayer, and J. Clémenty, "Spontaneous initiation of atrial fibrillation by ectopic beats originating in the pulmonary veins," *New England Journal of Medicine*, vol. 339, no. 10, pp. 659–666, 1998.

[151] A. Al-Ahmad, J. D. Grossman, and P. J. Wang, "Early experience with a computerized robotically controlled catheter system," *Journal of Interventional Cardiac Electrophysiology*, vol. 12, no. 3, pp. 199–202, 2005.

[152] W. Saliba, V. Y. Reddy, O. Wazni, J. E. Cummings, J. D. Burkhardt, M. Haissaguerre, J. Kautzner, P. Peichl, P. Neuzil, V. Schibgilla, *et al.*, "Atrial fibrillation ablation using a robotic catheter remote control system: initial human experience and long-term follow-up results," *Journal of the American College of Cardiology*, vol. 51, no. 25, pp. 2407–2411, 2008.

[153] W. Saliba, J. E. Cummings, S. Oh, Y. Zhang, T. N. Mazgalev, R. A. Schweikert, J. D. Burkhardt, and A. Natale, "Novel robotic catheter remote control system: feasibility and safety of transseptal puncture and endocardial catheter navigation," *Journal of cardiovascular electrophysiology*, vol. 17, no. 10, pp. 1102–1105, 2006.

[154] W. Wongcharoen, H.-M. TSAO, M.-H. WU, C.-T. TAI, S.-L. CHANG, Y.-J. LIN, L.-W. LO, Y.-J. CHEN, M.-H. SHEU, C.-Y. CHANG, *et al.*, "Morphologic characteristics of the left atrial appendage, roof, and septum: implications for the ablation of atrial fibrillation," *Journal of cardiovascular electrophysiology*, vol. 17, no. 9, pp. 951–956, 2006.

[155] A. C. Lardo, E. R. McVeigh, P. Jumrussirikul, R. D. Berger, H. Calkins, J. Lima, and H. R. Halperin, "Visualization and temporal/spatial characterization of cardiac radiofrequency ablation lesions using magnetic resonance imaging," *Circulation*, vol. 102, no. 6, pp. 698–705, 2000.

[156] A. N. Raval, P. V. Karmarkar, M. A. Guttman, C. Ozturk, R. DeSilva, R. J. Aviles, V. J. Wright, W. H. Schenke, E. Atalar, E. R. McVeigh, *et al.*, "Real-time mri guided atrial septal puncture and balloon septostomy in swine," *Catheterization and cardiovascular interventions*, vol. 67, no. 4, pp. 637–643, 2006.

[157] R. Razavi, D. L. Hill, S. F. Keevil, M. E. Miquel, V. Muthurangu, S. Hegde, K. Rhode, M. Barnett, J. van Vaals, D. J. Hawkes, *et al.*, "Cardiac catheterisation guided by mri in children and adults with congenital heart disease," *The Lancet*, vol. 362, no. 9399, pp. 1877–1882, 2003.

[158] R. C. Susil, C. J. Yeung, H. R. Halperin, A. C. Lardo, and E. Atalar, "Multifunctional interventional devices for mri: a combined electrophysiology/mri catheter," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 47, no. 3, pp. 594–600, 2002.

[159] K.-H. Lee, K. C. D. Fu, Z. Guo, Z. Dong, M. C. Leong, C.-L. Cheung, A. P.-W. Lee, W. Luk, and K.-W. Kwok, "Mr safe robotic manipulator for mri-guided intracardiac catheterization," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 586–595, 2018.

[160] M. Bock, S. Müller, S. Zuehlsdorff, P. Speier, C. Fink, P. Hallscheidt, R. Umathum, and W. Semmler, "Active catheter tracking using parallel mri and real-time image reconstruction," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 55, no. 6, pp. 1454–1459, 2006.

[161] W. Wang, C. L. Dumoulin, A. N. Viswanathan, Z. T. Tse, A. Mehrtash, W. Loew, I. Norton, J. Tokuda, R. T. Seethamraju, T. Kapur, *et al.*, "Real-time active mr-tracking of metallic stylets in mr-guided radiation therapy," *Magnetic resonance in medicine*, vol. 73, no. 5, pp. 1803–1811, 2015.

[162] K.-W. Kwok, K.-H. Lee, Y. Chen, W. Wang, Y. Hu, G. C. Chow, H. S. Zhang, W. G. Stevenson, R. Y. Kwong, W. Luk, *et al.*, "Interfacing fast multi-phase cardiac image

registration with mri-based catheter tracking for mri-guided electrophysiological ablative procedures," *Circulation*, vol. 130, no. suppl_2, pp. A18568–A18568, 2014.

[163] K.-W. Kwok, Y. Chen, T. C. Chau, W. Luk, K. R. Nilsson, E. J. Schmidt, and T. T. Zion, "Mri-based visual and haptic catheter feedback: simulating a novel system's contribution to efficient and safe mri-guided cardiac electrophysiology procedures," *Journal of Cardiovascular Magnetic Resonance*, vol. 16, no. 1, pp. 1–3, 2014.

[164] Y. Ganji, F. Janabi-Sharifi, *et al.*, "Catheter kinematics for intracardiac navigation," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 621–632, 2009.

[165] Y. Ganji and F. Janabi-Sharifi, "Kinematic characterization of a cardiac ablation catheter," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1876–1881, 2007.

[166] T. Liu and M. C. Çavuşoğlu, "Three dimensional modeling of an mri actuated steerable catheter system," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 4393–4398, 2014.

[167] T. Greigarn and M. C. Çavuşoğlu, "Task-space motion planning of mri-actuated catheters for catheter ablation of atrial fibrillation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3476–3482, 2014.

[168] R. H. Hoyle, *Statistical strategies for small sample research*. Sage, 1999.

[169] T. L. T. Lun, K. Wang, J. D. 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 Robotics and Automation Letters*, vol. 4, no. 2, pp. 1454–1461, 2019.

[170] S. Sefati, F. Alambeigi, I. Iordachita, M. Armand, and R. J. Murphy, "Fbg-based large deflection shape sensing of a continuum manipulator: Manufacturing optimization," in *2016 IEEE SENSORS*, pp. 1–3, 2016.

[171] B. Childlers, D. Gifford, R. Duncan, M. Raum, and M. Vercellino, "Fiber optic position and shape sensing device and method relating thereto," Jan. 19 2006. US Patent App. 11/180,389.

[172] C. Shi, S. Giannarou, S.-L. Lee, and G.-Z. Yang, "Simultaneous catheter and environment modeling for trans-catheter aortic valve implantation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2024–2029, 2014.

[173] S. Jäckle, T. Eixmann, H. Schulz-Hildebrandt, G. Hüttmann, and T. Pätz, "Fiber optical shape sensing of flexible instruments for endovascular navigation," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 12, pp. 2137–2145, 2019.

[174] C. L. Dumoulin, S. Souza, and R. Darrow, "Real-time position monitoring of invasive devices using magnetic resonance," *Magnetic resonance in medicine*, vol. 29, no. 3, pp. 411–415, 1993.

[175] Y. Chen, W. Wang, E. J. Schmidt, K.-W. Kwok, A. N. Viswanathan, R. Cormack, and Z. T. H. Tse, "Design and fabrication of mr-tracked metallic stylet for gynecologic brachytherapy," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 956–962, 2015.

[176] S. Hilbert, P. Sommer, M. Gutberlet, T. Gaspar, B. Foldyna, C. Piorkowski, S. Weiss, T. Lloyd, B. Schnackenburg, S. Krueger, *et al.*, "Real-time magnetic resonance-guided ablation of typical right atrial flutter using a combination of active catheter tracking and passive catheter visualization in man: initial results from a consecutive patient series," *Europace*, vol. 18, no. 4, pp. 572–577, 2016.

[177] C.-L. Cheung, J. D.-L. Ho, V. Vardhanabhuti, H.-C. Chang, and K.-W. Kwok, "Design and fabrication of wireless multilayer tracking marker for intraoperative mri-guided interventions," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 1016–1025, 2020.

[178] Z. Guo, Z. Dong, K.-H. Lee, C. L. Cheung, H.-C. Fu, J. D. Ho, H. He, W.-S. Poon, D. T.-M. Chan, and K.-W. Kwok, "Compact design of a hydraulic driving robot for intraoperative mri-guided bilateral stereotactic neurosurgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2515–2522, 2018.

[179] Z. He, Z. Dong, G. Fang, J. D.-L. Ho, C.-L. Cheung, H.-C. Chang, C. C.-N. Chong, J. Y.-K. Chan, D. T. M. Chan, and K.-W. Kwok, "Design of a percutaneous mri-guided needle robot with soft fluid-driven actuator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2100–2107, 2020.

[180] M. A. Rube, A. B. Holbrook, B. F. Cox, J. G. Houston, and A. Melzer, "Wireless mr tracking of interventional devices using phase-field dithering and projection reconstruction," *Magnetic resonance imaging*, vol. 32, no. 6, pp. 693–701, 2014.

[181] X. Wang, K.-H. Lee, D. K. Fu, Z. Dong, K. Wang, G. Fang, S.-L. Lee, A. P. Lee, and K.-W. Kwok, "Experimental validation of robot-assisted cardiovascular catheterization: model-based versus model-free control," *International journal of computer assisted radiology and surgery*, vol. 13, no. 6, pp. 797–804, 2018.

[182] M. B. Rubin, *Cosserat theories: shells, rods and points*, vol. 79. Springer Science & Business Media, 2013.

[183] M. Barkagan, M. Rottmann, E. Leshem, C. Shen, A. E. Buxton, and E. Anter, "Effect of baseline impedance on ablation lesion dimensions: A multimodality concept validation from physics to clinical experience," *Circulation: Arrhythmia and Electrophysiology*, vol. 11, no. 10, p. e006690, 2018.

[184] K. W. Kwok, Z. Dong, Z. Guo, K. C. D. Fu, K. H. Lee, and C. L. Cheung, "Robotic catheter system for mri-guided cardiovascular interventions," Dec. 28 2017. US Patent App. 15/630,406.

[185] Z. Dong, Z. Guo, K.-H. Lee, G. Fang, W. L. Tang, H.-C. Chang, D. T. M. Chan, and K.-W. Kwok, "High-performance continuous hydraulic motor for mr safe robotic teleoperation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1964–1971, 2019.

[186] C. G. Robinson, P. P. Samson, K. M. Moore, G. D. Hugo, N. Knutson, S. Mutic, S. M. Goddu, A. Lang, D. H. Cooper, M. Faddis, *et al.*, "Phase i/ii trial of electrophysiology-guided noninvasive cardiac radioablation for ventricular tachycardia," *Circulation*, vol. 139, no. 3, pp. 313–321, 2019.

[187] Y. Fan, K.-W. Kwok, Y. Zhang, G. S.-H. Cheung, A. K.-Y. Chan, and A. P.-W. Lee, "Three-dimensional printing for planning occlusion procedure for a double-lobed left atrial appendage," *Circulation: Cardiovascular Interventions*, vol. 9, no. 3, p. e003561, 2016.

[188] B. K. Foucha, "The abcs of a to v: Right atrial/ left atrial (pcw) pressures," *CathLab Digest*, vol. 17, no. 5, 2009.

[189] I. Van Meerbeek, C. De Sa, and R. Shepherd, "Soft optoelectronic sensory foams with proprioception," *Science Robotics*, vol. 3, no. 24, 2018.

[190] G. Fang, X. Wang, K. Wang, K.-H. Lee, J. D. Ho, H.-C. Fu, D. K. C. Fu, and K.-W. Kwok, "Vision-based online learning kinematic control for soft robots using local

gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1194–1201, 2019.

[191] J. R. Rambach, A. Tewari, A. Pagani, and D. Stricker, "Learning to fuse: A deep learning approach to visual-inertial camera pose estimation," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 71–76, 2016.

[192] P. Gemeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 591–605, 2007.

[193] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

[194] X. Wang, G. Fang, K. Wang, X. Xie, K.-H. Lee, J. D. Ho, W. L. Tang, J. Lam, and K.-W. Kwok, "Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2161–2168, 2020.

[195] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2, pp. 985–990, 2004.

[196] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[197] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[198] G.-B. Huang, N.-Y. Liang, H.-J. Rong, P. Saratchandran, and N. Sundararajan, "On-line sequential extreme learning machine.," *Computational Intelligence*, vol. 2005, pp. 232–237, 2005.

[199] T. Li, *Research on Soft-sensing Methods for the Size of Melt Pool in MgO Single Crystal Furnace*. PhD thesis, Dalian University of Technology, 12 2012.

[200] 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, p. 311, 2017.

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