Identification for a Class of Soft Pneumatic Actuators

Xiaochen Wang

Principle Supervisor:
Dr Chakravarthini Saaj
Surrey Space Center

Secondary Supervisor:
Dr Constantina Lekakou
Department of Mechanical Engineering Sciences

2016
Abstract

There is an increasing interest to advance the state-of-the-art of soft robots due to its capability to extend the limits of traditional rigid robot manipulators. Many novel soft robots have been designed and proposed in the recent years of which, inflatable soft robots attract considerable research efforts, as the actuation mechanism is readily compatible with bio-inspired designs. However, the research on inflatable soft robots is currently at an early stage. Most prototypes lack a proper mathematical model and control design for real-time application. This study proposes a unified model identification framework for the pneumatic bending soft actuator and similar flexible systems. This method is one of the first attempts to deal with the complex nonlinear dynamics of soft robots that is challenging to model. The outcomes of this research could improve the current modelling and control design of inflatable soft robots for more advanced medical applications.

Firstly, a statistics-based model identification approach is investigated. It introduces a newly developed model identification model structure, namely the Difference Input Outputs PieceWise Linear Orthonormal Basis Function (DIO-PWL-OBF). This method collects local dynamic responses and linearizes local subsystems that are approximated by a set of optimally selected OBFs. With the DIO setting, the switching between different subsystems is smooth especially in the input direction-dependent situations. The advantage is that this method can capture the local dynamics accurately and the local models can be used to construct global dynamics without adding offset terms or nonlinear weighting terms. The identification approach by the DIO-PWL-OBF model structure has been experimentally validated for common bending profiles using a three-chambered soft pneumatic actuator made of silicone.

Regarding an identified DIO-PWL-OBF model structure, a solution that further improves its local partition and local linear approximations is presented. The analysis shows that, under the DIO setting, the switching between two subsystems is coupling invariant whether in the same input direction or not. Utilising this unique feature, an iterative refinement algorithm is developed. The algorithm consists of two main steps, adjusting the offsets on individuals DIO sequences and redoing the linear approximate under eigenvalue constraints. The refinement algorithm has been experimentally validated by the soft actuator in a typical bending case.
4.1 Introduction ............................................. 44
4.2 Observed Nonlinearities .......................... 46
4.3 Kinematic Modelling ................................. 48
  4.3.1 The current kinematic setting .................. 48
  4.3.2 The auxiliary kinematic setting ................. 50
4.4 Model Identification Setting ...................... 53
  4.4.1 The experimental design ....................... 53
  4.4.2 The DIO-PWL-OBF model structure ............. 53
4.5 Implementation and Results ....................... 59
  4.5.1 Platform setup .................................. 59
  4.5.2 Single chamber identification ................. 59
  4.5.3 Double chamber identification ................. 65
4.6 Summary .............................................. 72

5 A Direction-Dependent Multisegment Piecewise-Linear Identification Method for Hysteresis-like Nonlinearities 73
  5.1 Introduction ........................................ 73
  5.2 Direction-Dependent Multisegement Piecewise-linear Switched Systems 74
  5.3 Coupling Invariant Switching .................... 75
  5.4 Refining the Reference Partitions and the Local Models ......................... 78
    5.4.1 The equivalent MISO expression ............... 78
    5.4.2 The recursive algorithm for refinement ....... 79
  5.5 Structural Approximation by Orthonormal Basis Functions ................... 86
  5.6 Implementation and Results ..................... 89
  5.7 Summary ............................................ 93

6 Conclusions .......................................... 94
  6.1 Summary ........................................... 94
  6.2 Future Work ....................................... 94
# List of Figures

1.1 The Stiff-Flop robot arm is operating with phantom organs .......................... 1

2.1 Robots classification based on materials and degree of freedoms [4].
Hatched area represents empty set .......................................................... 6

2.2 Comparison between rigid robot and soft robot mechatronic systems.
Rigid robots make up through integration while soft robots through
fusion [5] ................................................................. 7

2.3 Deformation types from selected inflatable actuators: (a) contracting [6], (b) expanding [10], (c) elongating [12], (d) bending [12], (e) twisting [21] .............................................. 9

2.4 Soft inflatable robots used in rigid robots and wearable devices: (a) a robotic leg [27], (b) a robotic arm [25], (c) a robotic hand [31], (d) an orthotic device for ankle foot pathologies [33], and (e) a glove for hand rehabilitation [[37]] ............................................. 10

2.5 Recent soft robots: (a) a soft hand [46], (b) soft fingers [47], (c) soft gripper [49], and (d) soft legs [50] ................................................................. 11

2.6 Some selected bio-inspired soft inflatable robots: (a) a robotic manta [51], (b) a locomotive soft robot [52], (c) a soft robotic arm mimicking elephant trunks made by Festo company, (d) a soft robotic arm mimicking octopus [41], (e) a swallowing robot [54], and (f) a soft robot mimicking cardiac motion [53] ................................................................. 11

3.1 Configuration of the preliminary platform .................................................. 23

3.2 Photo of the preliminary platform setup ..................................................... 23

3.3 The Simulink model that governs experimental procedures .......................... 24

3.4 The global and the local reference frames attached to the transmitter and the micro sensor .................................................. 25

3.5 The bending actuator was keeping increased constant curvatures during bending .................................................. 26

3.6 Bending measurement comparison: curvature $k$ versus bending angle $\theta$ 27

3.7 (a) An oblique view of the tip surface. (b) A side view of the soft bending actuator. (c) The ideal installation of micro sensor on the soft bending actuator .......................... 27
3.8 A setup of the transmitter and the micro sensor. 28
3.9 Measurement setting at the tip end of a bending actuator. The local reference frame is based on the trackSTAR™ magnetic sensor. 28
3.10 Bending angle $\theta$ in two cases. (a) $\theta$ is shown in the bending plane. (b) $\theta$ is shown in the 3D reference frame, where the frame $O'-X'Y'Z'$ is parallel to the transmitter global frame $O-XYZ$. 29
3.11 Bending angles that are calculated from experimentally measured task space coordinates ($X, Y, Z, \phi, \theta, \psi$). Three different cases are showed. 30
3.12 An experimental comparison between the calculated plane angles according to Method 1 and Method 2 respectively. The tube is within a very small bending angle $\theta$, i.e. less than $15^\circ$. 31
3.13 The plane angle $\phi$ is the intersection angle between the projected line of x-axis on the $X'Y'Z'$ plane and the X-axis. The $O'-X'Y'Z'$ is as the same definition in Figure 3.8. 32
3.14 The pyramid that structures the geometric relation between $\phi, B_e_x, B_e_y$ . 32
3.15 Experimental determination of plan angle $\phi$. 34
3.16 A comparison between the original plane angle calculation algorithm and its modified algorithm. 34
3.17 Configuration of the two valves system 35
3.18 Pressure regulation (above) under the tracking signal (below). The measurement was not calibrated yet. 36
3.19 Raw data of Test Case 1 corresponding to independent step responses. 37
3.20 A standard NARX model 39
3.21 The best fitting result with use of the NARX model 39
3.22 A standard HW model. 40
3.23 In HW model, the input nonlinearity block had 5 piecewise linear components. This configuration led to a 92.3% fitting result. 40
3.24 Raw data of Test Case 2 corresponding to dependent step responses. 42
4.1 Featured PLSA prototypes: (a) Flexible Micro-Actuator by [88]. (b) Colobot by [23]. (c) An early design of the STIFF-FLOP actuator. 44
4.2 (a) A simplified structural drawing for the PLSAs at inactive state. (b) The illustration of the bending forced by actuation. 46
4.3 Globally conducted local step responses. 47
4.4 The extracted local step responses from Figure 4.3. 48
4.5 Kinematic parameters for a single-section PLSA. 49
4.6 The kinematic framework for the PLSA actuators. 49
4.7 The illustration for the three configuration subspace 1, 2 and 3. They are sectioned by the three chambers’ individual bending directions. 50
4.8 The bending angle $\theta^b$ is equivalent to the deflection angle $\theta^d$ in the constant curvature condition. ........................................ 51
4.9 The modified auxiliary kinematic framework for the PLSA actuators. 52
4.10 Illustration of the DIO-PWL-OBF model structure. .................... 58
4.11 Schematic drawing for the platform setup. The platform was revised from Figure 3.1. ................................................ 59
4.12 A photo of the platform. The platform was revised from Figure 3.2. 60
4.13 The local partition for the single chamber identification ............... 60
4.14 The experimental data for the single chamber identification .......... 61
4.15 The decay rate of the identified pole based on the selected pole structure in Table 4.5. ................................................ 63
4.16 The global test and validation for the single chamber identification. 64
4.17 The local partition for the double chamber identification .......... 66
4.18 The experimental data for the double chamber identification ....... 66
4.19 The decay rate of the identified pole based on the selected pole structure in Table 4.9 ........................................... 68
4.20 The Chamber 1 single actuation test and validation. ............... 70
4.21 The Chamber 2 single actuation test and validation ............... 70
4.22 The joystick-controlled test and validation. .......................... 71

5.1 A typical static hysteresis loop approximated by the direction-dependent multisegment piecewise-linear method. Each local region is represented by a linear model $\sigma_j \pm$. .................................... 74
5.2 The structural plot of the equivalent non-switched MISO expression for the switched SISO linear system (5.3) under the coupling invariant switching setting. .......................... 79
5.3 Signals in the first-round identification in Example 1: (a) the staircase input, (b) the assigned input parts for individual subsystems, (c) the resultant output. .......................... 84
5.4 The approximation plots after each iteration during the refinement in Example 1. .................................................. 85
5.5 The validation test under a random input in Example 1. ............. 85
5.6 Schematic drawing of the DIO-OBF structure. .......................... 87
5.7 The validation test under a sinusoid signal in Example 2. The system in Example 1 is approximated by the DIO-OBF model structure. .... 88
5.8 The bending demonstration of a low-pressure pneumatic soft actuator. 89
5.9 The identification signals in the pneumatic actuator experiment: (a) the response under the staircase input, (b) the staircase input signal. 90
5.10 The approximation plots after each iteration during the refinement in the pneumatic actuator experiment. .......................... 91
5.11 The validation test for the DIO-OBF model structure in the pneumatic actuator experiment. (a) the measured output and its comparison with the two simulated responses. (b) the measured input during the test.
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Characteristics of different robot types [4]. Hard robots appear on the first three columns and soft robots on the last column.</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Step responses at different pressures.</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Time intervals of each sample in Test Case 2</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Approximated transfer function for each sample in Table 3.2 from the raw data in Figure 3.24. The fitting results in the forms of Fit-To-Estimation (FTE), Final Prediction Error, and Mean-Squares Error (MSE) were used.</td>
<td>42</td>
</tr>
<tr>
<td>4.1</td>
<td>The unified identification procedure.</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Linear identification for the local data sets</td>
<td>61</td>
</tr>
<tr>
<td>4.3</td>
<td>Poles of the local data sets.</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>The unique pole set calculated by FKcM clustering</td>
<td>62</td>
</tr>
<tr>
<td>4.5</td>
<td>The pole set {\xi_1 \ldots \xi_{10}} selected for OBFs construction</td>
<td>62</td>
</tr>
<tr>
<td>4.6</td>
<td>The OBF approximation for the original local data sets</td>
<td>63</td>
</tr>
<tr>
<td>4.7</td>
<td>The fitting results for the test in Figure 4.16.</td>
<td>64</td>
</tr>
<tr>
<td>4.8</td>
<td>The linearisation result for the local data sets in the double chamber case.</td>
<td>67</td>
</tr>
<tr>
<td>4.9</td>
<td>The pole set {\xi_{ij}^1, \ldots, \xi_{ij}^{5}}_{i=1,2, j=1,2} for the OBFs construction.</td>
<td>68</td>
</tr>
<tr>
<td>4.10</td>
<td>The OBF approximation for the original local data sets</td>
<td>69</td>
</tr>
<tr>
<td>4.11</td>
<td>The validation results for the test in Figure 4.20, 4.21, and 4.22.</td>
<td>71</td>
</tr>
<tr>
<td>5.1</td>
<td>Structural comparison between identified subsystems in the pneumatic actuator.</td>
<td>91</td>
</tr>
</tbody>
</table>
Publications


Acknowledgement

I would like to express my sincere gratitude to my advisor Dr Chakravarthini Saaj and Dr Constantina Lekakou for the continuous support of my study and related research, for their patience, motivation, and immense knowledge. Your guidance helped me in all the time of research and writing of this thesis.

My sincere thanks to Karen Collar, who always helped me with great effort. And thanks to those colleagues and friends I met in Surrey Space Centre, Tao Geng, Seri Mustaza, Halidu Ibrahim, Wei Zhao, Ailian Cai, Fuguo Zhou, and Haitao Liu.

I would like to thank my family for supporting me spiritually throughout writing this thesis and my life in general.
Chapter 1

Introduction

1.1 Background

Robots that incorporate soft materials bring many advantages than their rigid counterparts [1]. The induced compliance permits adaptation to flexible interaction with unpredictable environments. In a robot design, the use of soft structure can potentially reduce the mechanical and algorithmic complexity. And robots developed in soft materials are relatively safe when working with human. Last but not least, soft technology can be combined with tissue engineering to create hybrid systems for medical applications. Soft robotics is an emerging field; its ongoing study will lead to a new wave of robotic applications.

Figure 1.1: The Stiff-Flop robot arm is operating with phantom organs

This research is inspired by a type of soft pneumatic bending actuators developed in the Stiff-Flop project. A prototype from the project is shown in Figure 1.1. It is based on a three-chamber integral design proposed by [2]. The Stiff-Flop project is investigating an octopus-like robot arm for minimally invasive surgery. The robot arm has three sections. Each section serves as a structure as well as an actuator.
1.2 Motivation

For the Stiff-Flop actuators and many other ones designed in different shapes but using similar pneumatic mechanisms, an improved modelling approach is still under exploration. Currently, the most applicable solution is to simplify a soft robotic body to constant curvature arcs and then to process the geometric relation statically. In this way, the nonlinear dynamics between actuation and shape deformation is not clearly known and therefore it becomes unemployed. The lack of such information degrades the control performance. Investigating the nonlinear dynamics in the pneumatic mechanism and utilising it in the control design can provide a more viable solution compared to the state-of-the-art ones.

1.3 Aims and Objectives

1.3.1 Aims

This thesis will propose a unified identification approach for the class of Stiff-Flop and the similar actuators. The key idea behind the approach is to utilise linearised local dynamics.

The identification approach is based on a new direction-dependent multisegment piecewise-linear model identification structure. The new structure is modified from classic multisegment piecewise-linear systems by introducing the concept of different input and output. The resultant locally linearised models use independent state variables; no state transformation is required. This configuration is practical in identifying the studied soft pneumatic actuators, as the global nonlinear dynamics can be captured accurately without explicitly seeking the physical meaning of local systems’ state variables.

1.3.2 Objectives

The objectives of the research are:

1 to explore a feasible identification approach for the Stiff-Flop actuators;

2 to further improve the obtained identification approach for general pneumatic soft actuators.
1.4 Contributions of this research

This research is among one of the first that explore dynamic models in soft inflatable robots. The proposed statistics-based model identification approach can be used to account for general nonlinear pressure-deformation relation effectively. This method has been highlighted by a recent study [3]. Besides, based on the model identification structure proposed for soft inflatable robotics, the refinement algorithm can broaden applications towards more general hysteresis-like nonlinearities.

1.5 Thesis outline

Regarding mechanical and control designs, the first half of Chapter 2 presents the literature survey of the state-of-the-art soft inflatable robots. Then the second half of this chapter introduces the background knowledge of the system identification method used in Chapter 4 and 5.

Chapter 3 firstly shows the system implementation, from which a close-loop real-time platform is built for identification experiments of general soft bending actuators. Then with using this platform, preliminary system identification experiments are demonstrated.

Chapter 4 proposes a unified identification approach for a class of pneumatic soft actuators. For such a single segment actuator, it can practically give reliable approximations for the nonlinear dynamical relation between the actuating pressures and the two principal Degrees Of Freedom (DOF), the bending and the steering. Section 4.1 introduces the background of the studied soft actuators, regarding which the nonlinearities are discussed in Section 4.2. Then Section 4.3 studies the auxiliary kinematic setting. Some aspects in the mainstream kinematic configuration are discussed in Subsection 4.3.1 firstly, and then the new setting is introduced in Subsection 4.3.2. After theoretical foundations and the new kinematic setting are ready, the unified identification approach is developed in Section 4.4. Especially, Subsection 4.4.2 studies the new proposed identification model structure in detail and summarises the unified identification procedure at the end. Section 4.5 shows the implementation and results. Section 4.6 summarises the work.

Following the study of Chapter 4, Chapter 5 develops a refinement algorithm to improve the identification accuracy by adjusting local partitions and tuning local linear parametrisation. Firstly, Section 5.1 and 5.2 classify the identification model structure as a type of direction-dependent multisection piecewise-linear systems. Section 5.3 discusses the coupling invariant switching concept that is configured in the studied switched systems. Based upon this, the refinement algorithm is developed in Sec-
tion 5.4. With the use of orthonormal basis function briefly re-introduced in Section 5.5, the implementation and results for the refinement algorithm are demonstrated in Section 5.6. Section 5.7 summarises this chapter.

Chapter 6 summarises the thesis and highlights the future work.
Chapter 2

Literature Review

2.1 Introduction

Regarding the identification objective addressed in this thesis, this chapter is going to provide related background information. As the identification approach aims at the application on a class of soft inflatable actuators, the state of the art of soft inflatable robotics will be introduced. Specifically, the survey will go through the research of both the mechanical designs and manipulation designs so that the identification problem can be formulated explicitly. On the other hand, the Orthonormal Basis Functions (OBF) system identification techniques will be introduced; it will be used for developing the solution in the following chapters. The rest of this section will introduce briefly the classification of the soft inflatable robots in the overall robot family; in this way, one can quickly build the sense of how soft robots, an emerging area, relate to other types of robots.

Figure 2.1 generalises the classification of a broad range of robots. Hard robots, in general, consist of joints in connection with rigid links. Those robots are designed to be stiff so that vibration and deformation of the structure and drivetrain can be neglected in dynamic analysis. Hard nonredundant robots are commonly used in well-defined environments, e.g. manufacture shops, where they repetitively perform a prescribed motion with great precision. Hard redundant robots, especially hyper-redundant ones, are capable of high dexterity when work in unstructured environments.
Compared with hard robots, soft robots employ different mechanisms to reach dexterous mobility. Distributed deformation with a theoretically infinite number of degree-of-freedoms (DOF) characterise soft robots. Therefore they can achieve hyper-redundancy in the configuration space. In other words, the robot tip can go to every point in the three-dimensional workspace with a countless number of shape configurations. Another advantage of soft robots over hard ones is the lower resulting resistance when the robot compresses a contact surface. The possibility of causing injuries can be reduced significantly, which makes soft robots a promising choice for surgical applications. However, the price for the hyper-redundancy and the comfort contact is the increased difficulty in robot modelling and control design.

Table 2.1: Characteristics of different robot types [4]. Hard robots appear on the first three columns and soft robots on the last column.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Rigid</th>
<th>Discrete hyperredundant</th>
<th>Hard continuum</th>
<th>Soft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuators</td>
<td>Few</td>
<td>Large</td>
<td>Infinite</td>
<td>Continuous</td>
</tr>
<tr>
<td>Material strain</td>
<td>None</td>
<td>Metals, plastics</td>
<td>Small</td>
<td>Rubber, electroactive polymer</td>
</tr>
<tr>
<td>Capabilities</td>
<td>Very high</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Accuracy</td>
<td>High</td>
<td>Lower</td>
<td>Lower</td>
<td>Lowest</td>
</tr>
<tr>
<td>Load capacity</td>
<td>Dangerous</td>
<td>Dangerous</td>
<td>Dangerous</td>
<td>Safe</td>
</tr>
<tr>
<td>Safety</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Dexterity</td>
<td>Structured only</td>
<td>Structured and unstructured</td>
<td>Structured and unstructured</td>
<td>Structured and unstructured</td>
</tr>
<tr>
<td>Working environment</td>
<td>Fixed sized</td>
<td>Variable size</td>
<td>Variable size</td>
<td>Variablesize</td>
</tr>
<tr>
<td>Conformability to obstacles</td>
<td>None</td>
<td>Good</td>
<td>Fair</td>
<td>Highest</td>
</tr>
<tr>
<td>Design</td>
<td>Easy</td>
<td>Medium</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>Controllability</td>
<td>Easy</td>
<td>Harder</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>Path planning</td>
<td>Easy</td>
<td>Harder</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>Position Sensing</td>
<td>Easy</td>
<td>Harder</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>Inspiration</td>
<td>Mammalian limbs</td>
<td>Snakes, fish</td>
<td>Muscular hydrostats</td>
<td></td>
</tr>
</tbody>
</table>

The trade-offs among the variety of robots are indicated in Table 2.1. Several reasons
are responsible for the difficulty in control design. The soft robot model is nonlinear and coupled. Their models have a higher portion of assumption and approximation, which increase the extent of errors. Plus, soft robots are under-actuated, as not every DOF corresponds to an actuator. These characteristics together increases the difficulty in their modelling and control design towards robotic applications.

Furthermore, the mechatronic components of a soft robot are closely dependent on or merged to each other. To exhibit a soft robot’s full potential, it is necessary to consider all of them components systematically at the same time. Otherwise, one component that functions individually may not work together with others. As there is no clear physical boundary between individual components of a soft robot, the term *fusion* is used as an analogue for the *integration* to a rigid robot mechatronic system. Similar to a rigid robot, from a functional point of view, a soft robot have components like mechanisms, actuators, sensors, power, and structure. But these different functional components of the soft robot may come from one common physical structure. Figure 2.2 illustrates the difference between rigid and soft robots to make up mechatronic systems.

![Figure 2.2: Comparison between rigid robot and soft robot mechatronic systems. Rigid robots make up through integration while soft robots through fusion [5].](image)

Many soft inflatable robots can be further classified as intrinsic type, because their actuators not only provide force or torque but also serve as structures to transmit them. Soft inflatable robots can be driven pneumatically or hydraulically. The two mechanisms work in a similar way, and a lot of designs allow both of them. The pneumatic solutions results in less gravitational effects imposing on soft robots than the hydraulic ones, meanwhile it is easier to maintain the former than the latter. Therefore pneumatic actuations are more favourably chosen in the majority of the related studies.
2.2 Soft Inflatable Robots: the State of the Art

2.2.1 Mechanical designs

Soft inflatable robots have largely variant designs. They are researched as actuators firstly. The earliest study on soft inflatable actuators can be traced back to the late 1950s when Pneumatic Artificial Muscles (PAM) were developed to obtain a high force/torque-to-weight actuation solution for rigid manipulators. In the early 1990s, a different bending actuator was proposed and since then a wide range of novel soft inflatable actuator have been explored to utilise their flexibility feature. Especially due to the advances on hyper-elastic materials and the maturity of their fabrication since about 2010, a significantly increasing number of soft inflatable robots or robotic devices based on soft inflatable actuators have been researched and developed. This section tries to collect the most typical soft inflatable robot to date from literature. The survey scope will mainly concentrate on the pneumatic type as the similarity to the hydraulic one and the former is more preferred by research due to its subjection to less gravitational effects than the latter. As the basic components in soft robots or other related robotic applications, soft inflatable actuators will be introduced firstly. Then the survey will move to robotic applications or devices where soft inflatable actuators are purely or mainly used for actuation. Next, soft robots built up by or merged from soft inflatable actuators will be introduced; in these designs, the actuators provide structural function as well as actuation. Moreover, the designs of soft inflatable robots are not confined to the vision of traditional robots; it is more advantageous for the former to realise diverse bio-inspired designs. The most outstanding studies will be shown briefly here. The literature survey here will primarily highlight studies that present mechanical designs of soft inflatable robots. Theoretical developments regarding their modelling and control designs will be discussed in the next section.

From a functional perspective, the deformation effects of soft inflatable actuators include contracting, expanding, elongating, bending, and twisting. PMAs are a major type of contracting actuators [6, 7, 8, 9]. The membrane of a PMA is braided by a cylindrical netting that restrains its volume under high pressures. Thus when PMAs are inflated, a contraction force is generated as the radial swelling shortens the longitudinal length. Expanding actuators directly use the inflation effect [10, 11]. Such actuators can be designed in many convex shapes like balls, cubes, or cylinders, and usually they do not wear a constraint suit on the outer surface. Compared with other soft inflatable actuators, operating an expanding actuator is more similar to blowing a balloon. The contact force exerted by the inflated surfaces can be used for levitation or pushing. Elongating actuators includes those like [12, 13]. They are
commonly designed in cylinders. Their lateral surface wears a bellow suit so that the radial expansion is restricted while the longitudinal extension is permitted; a contact force is generated along the elongation direction. Studies of bending actuators can be found in [14, 15, 16, 17, 18, 19, 20]. Those actuator are usually beam-like elements where by inflation one side stretches in the longitudinal direction more than the other, thus generating a bending moment. Twisting actuators are a relatively rare type [21, 22]. The twisting structure is created by installing the internal chambers in a crossing way, as a result of that these cambers do not bending towards the same direction. Figure 2.3 shows the five deformation types from selected soft inflatable actuators in both a relaxed state and an inflated state. Beside those designed to perform only one deformation type, many soft inflatable actuators are capable of acting more than one types, or one deformation type is passively coupled with another. For example, for PMAs, its longitudinal contracting is coupled with radial expanding; for bending actuators the expanding on one side. Or, in a three-chamber design like [23, 24], the actuators can perform bending and steering, while both the two deformations are driven by the elongations of their individual chambers.

Figure 2.3: Deformation types from selected inflatable actuators: (a) contracting [6], (b) expanding [10], (c) elongating [12], (d) bending [12], (e) twisting [21].

In the first three decades since the unveiling, PAMs, the solely type of soft inflatable actuators at that time, were designed to complement rigid manipulators. PAMs have a higher force-to-weight ratio than their electric counterparts. Using PAMs greatly reduces a manipulator’s inertial and thus eases control efforts for stabilization. This makes PMAs apt for highly dynamic applications that are weight-critical. Also, robots using PAMs are inherent compliant so that their robotic tasks involving
human interaction are greatly safe. One can refer to studies and products to see PAMs used in the developments of robotic arms [8, 25, 26], robotic legs [27, 28, 29, 30], and robotic hands [31, 32]. Besides, in the past few years, PAMs began to appear in wearable robots [33, 34, 35, 36]. These intelligent devices were developed for rehabilitation or assistance purposes. In the meantime, bending type soft inflatable actuators were proposed for wearable device as well [37, 38, 39, 40]. Figure 2.4 shows selected robotic applications that mainly use soft inflatable actuators’ actuation function apart from the structural function.

![Figure 2.4: Soft inflatable robots used in rigid robots and wearable devices: (a) a robotic leg [27], (b) a robotic arm [25], (c) a robotic hand [31], (d) an orthotic device for ankle foot pathologies [33], and (e) a glove for hand rehabilitation ([37]).](image)

When soft inflatable actuators serve structure function as well as produce force or moment, they make up soft robots; this is a growingly popular research topic in the recent years. The research direction is to design soft inflatable robots that extend the technical capability of their rigid counterparts. Although soft inflatable robots are underactuated, they can perform uncompromisingly or even over-perform certain tasks robustly. And they are mechanically simple and low-cost. As a great deal of research attention has been paid to this area, soft inflatable robots are beginning to realise more and more applications that used to be implemented by expansive traditional robots with complex and multi-joint structures. Up to the present, the studies of soft inflatable robot have explored soft arms [41, 42, 43, 44, 45], soft hands [46], soft fingers [47, 48], soft grippers [19, 45, 49], and soft legs [45, 50]. Figure 2.5 presents some latest novel designs of soft inflatable robots.
Figure 2.5: Recent soft robots: (a) a soft hand [46], (b) soft fingers [47], (c) soft gripper [49], and (d) soft legs [50].

Figure 2.6: Some selected bio-inspired soft inflatable robots: (a) a robotic manta [51], (b) a locomotive soft robot [52], (c) a soft robotic arm mimicking elephant trunks made by Festo company, (d) a soft robotic arm mimicking octopus [41], (e) a swallowing robot [54], and (f) a soft robot mimicking cardiac motion [53].

On the other hand, the flexible structure allows soft inflatable robots to have more design freedoms beyond traditional robots. Research teams use soft inflatable robots to mimic biological systems. For the soft robot arms referred previously, a lot of the designs were inspired from their animal counterparts such as [43] from elephant trunks and [41] from octopuses. Besides, [51] proposes an interesting design that mimics manta swimming. Inspired by squid, starfish, and worms, [52] propose a unique locomotive robot. Meanwhile, some groups considered to use soft inflatable robots to simulate normal physiological and pathological motion. [53] uses contracting type soft inflatable actuators to mimic human heart motion. [54] uses expanding deformation to simulate human human esophageal peristalsis. Figure 2.6 shows some resent bio-inspired designs made by soft inflatable robots.
2.2.2 Manipulation designs

Similar to the hard robotics approach, a manipulation task performed by soft robots can be decomposed into three parts, motion planing, motion control, and force control; they will be discussed separately below.

Motion Planning

For soft continuum manipulators, the state-of-the-art motion planning is mainly based on the constant-curvature geometry configuration generalized by [55]. Such an approach reduces the kinematic parameter number and simplifies the shape configuration of continuum mechanics into discrete. Motion planning based on constant-curvature kinematics includes a recent study by [56], which was further extended to including dynamics [57]. Meanwhile, the study by [58] introduced a learning-based approach to let the shape of a soft manipulator more adaptive to complex environments during end-effector operation. In overall, the motion planning research for soft continuum manipulator are relatively blank and it receive little attention. This is mainly because the current mechanical designs are still not standardised. Also, as the behaviors of soft manipulators are largely constrained by the current mechanical designs, it is unnecessary to develop a particular knowledge and techniques; one can refer to hard continuum and joint-link manipulators where the similar solutions are already there.

The motion planning for soft continuum manipulators will becomes more interesting and challenging when a more octopus-like manipulator is developed. Such a soft manipulator is expected to have a much higher dimension in the actuation space, i.e. probably more than 30 inputs (estimated according to [59]). A system like this will be fundamental different from other manipulators'. By that time, like [60] and [61], exploring and extracting motion primitives from the system behaviors can be an advantageous solution, as the dimension for control inputs and induced computational load [62] will be reduced greatly.

Motion Control

Motion control concerns the mapping between the joint space and the configuration space. Trajectory tracking is a typical problem in motion control. For a given trajectory in the configuration space, the designed controller needs to find the its counterpart in the joint space inversely. This can be done by either model-based or data-driven approaches.

For the model-based control designs, the methodologies are largely inspired by hard
continuum manipulators. The early control designs for soft robots were mainly based on the generalised kinematic framework using constant curvature arcs [55]. The control designs of the studies [23, 63] were developed based on the kinematic description, where the soft continuum manipulators were regarded as one or a few sections of constant curvature arc. In those studies, whether set-point tracking or force compensation, the key calculation is find individual chamber elongation through inverse kinematics. Such an approach is effective when the mechanical design is close enough to a constant curvature arc or when the payload or disturbances are relatively small so that the constant arc shape is less affected. Compared with the kinematic models, the dynamic models for soft robots were less researched. Early study by [64] derived a dynamical model of a soft continuum manipulator using the Lagrangian method; the potential energy included the gravitational energy and the elastic energy from bending and elongation. This modelling approach was further developed by a shape function through the study by [65] and extended to general multi-section cases [57, 66]. The control design that first used such a dynamical model is the study by [56], where the soft continuum manipulator demonstrate successfully the conforming to the goal shape-configurations. Regardless of the recent advances in dynamic modelling and control designs, it is necessary to point out constant curvature is still the underlying constant curvature assumption and the design and implementation were arranged in a way that maintain this condition; either the stiffness of a soft manipulators was increased significantly so that it behaved like a hard continuum manipulator [41, 66], or it was operated in a vertical configuration with the base end on topic, in which way, the transverse disturbance caused by gravity and payloads can be reduced [56].

Model-free is another approach for control design. A previous study by [67] explored a neural network feed-forward method to compensate for dynamic uncertainties. Then the study by [68] introduced supervised learning to train such a feed-forward numeral network. Other than those two studies, there are not a complete soft manipulator control demonstration that used a model-free approach. The lack of DOFs in actuation and sensing is the main reason accounting for this research gap. Unlike rigid robots with finite DOFs, soft robots have infinite DOFs. Thus, for the soft robots, the whole body dynamics are difficult to actuate and the kinematics are difficult to measure. From the survey of the mechanical designs in the previous subsection, it can be seen clearly that the state-of-the-art soft robots are too under-actuated to model-free control design. Nevertheless, in the long term after the soft robots have mechanical design with improved actuation and sensing, the model-free control design will have advantage over the model-based control design. This is because softness increases the complexity in dynamics and environmental interactions. The classic modelling approach for rigid manipulator cannot describe complex behaviours of soft manipulators effectively, as the information contained in the latter
is much higher than the one of the former. With sufficiently collected data, the model-free approach can handle the nonlinear dynamics in complex environments in a capable way.

**Force Control**

In recent years, there appeared many novel prototypes demonstrating how soft robots interacted with human/objects and environments [69, 70, 71], but those studies are mainly qualitatively not quantitatively. The passive compliance inherited in soft manipulators offers a simple and cheap solution for safe interaction, as force/torque sensors are not required. Without a close loop, soft end-effector or manipulators can adapt to the shapes of various objects in faster and more compliant way than their hard counterparts. Nevertheless, for more challenging manipulation applications such as surgery and home service, it is necessary to coordinate force control together with motion control properly, to ensure reliable robotic interaction with complex and unstructured environments.

For soft robotics, the force control research can largely refer to the counterpart in hard robotics. Nevertheless, the state-of-the-art soft robotics have a few extra problems to address, Firstly, the deformation behaviors of a soft robot are still not described sufficiently. The simplified constant-curvature approaches are mainly suitable when (a) the soft manipulator is as stiff as a hard continuum manipulator [72], (b) the soft manipulator is operating at a free space [56], and (c) the object to grasp is very light [57]. When encountering heavy loads or kinematic constraints, a soft robot may easily deform to a shape beyond the pre-defined domain of inverse kinematics or dynamics, without which the force control will provide erroneous feed-forward actuation commands. In the meantime, the study by [73] presented a quasi-static analytic model for bending in free space and force generation. Their model for force generation can only account for a specific no-bending case at a single contact point; the model becomes invalid immediately when the interaction with an object begins. Secondly, regarding the shape of a soft robot, the current sensing setup can not provide sufficient measurements, especially during interaction. Installing a number markers is the mainstream solution, but this approach may fail easily if un-modeled deformation like buckling occurs. A skin-like sensor that provides high-dimension information is desired; if such a sensing system is available, the force and motion control can be designed even without using a mathematical model. Last but not least, the current soft manipulators can not generate very large forces for more realistic applications; therefore it is very difficult or too early to carry on a very targeted force control study. For example, the force control of the current Stiff-Flop manipulator is indirect; as the output force is generally low, the operation is fairly safe, and thus active force control has not been researched particularly yet.
2.3 Fundamental Theories for System Identification

System identification is a subject of building mathematical models of dynamics systems from observed input-output data. Beside the capturing the behaviours of a system, an identified model is usually used for control design later. In order to capture the nonlinearities observed in soft bending actuators, a new model structure will be developed later in Section 4.4.2. In this model structure, the backbone is Orthonormal Basis Function, regarding which, the essential concepts will be firstly introduced here in Section 2.3.1. Besides, finding an optimal underlying pole structure for a set of the OBFs is crucially important in practice. The study chooses the Fuzzy-Kolmogorov c-Max (FKcM) clustering algorithm to solve the pole section problem, which is introduced in Section 2.3.2.

2.3.1 The OBF model structure in the linear system identification

In the field of the LTI system identification, the orthonormal basis function model structure has been firstly generalised and proposed by [74] and [75]. The method was developed under the classic predication error identification framework of [76]. Continued by the work of [77] and [78], theoretical foundation for the OBF approach became well structured. Meanwhile many applied issues, such as the pole selection, the transformation and the realization, were explored and studied. This made the OBF-based identification approach towards sophistication. It also serves as an analysis and design tool for many areas, e.g. signal, system and control theory. A comprehensive monograph for this method can be found in [79].

The OBF model structure holds both the properties, (a) the linear-in-the-parameter and (b) the independent process and noise model parametrizations [79]. The former (a) is also retained in the model structures of AutoRegresive model with eXternal input (ARX) and Finite Impulse Response (FIR). It indicates the linear relationship between the coefficients set and the one-step ahead predictor. Mature linear regression type techniques for parameter estimation, such as convex optimization, can be used. The latter (b) is also possessed by the counterparts FIR, Output Error (OE) and Box-Jenkin (BJ). It implies the process model estimate is robust against the noise property. The underlying advantage is the non-asymptotic bias and variance bounds. The FIR model structure holds the two properties too. In the rigours classification, FIR is a special case of the unified OBF model structure where all basis functions are formed in an power sequence of increase order of the time backward
shift operator [77]. But the FIR model structure usually results in a large number of the parameter set especially when a physical system has a stable pole close to the unit cycle, i.e. slow impulse responses. In contrast, using the OBF model structure can significantly decrease the number of the parameter set and preserve the two attractive properties.

A stable and strictly proper transfer function $G(z)$ can be decomposed into a series expansion of OBFs

$$G(z) = \sum_{i=1}^{\infty} c_i B_i(z)$$

(2.1)

where $c_i$ is the coefficient for each OBF $B_i(z)$. The set of OBFs $\{B_i(z)\}_{i=1}^{\infty}$ provides bases for the function space $H_2(E)$, i.e. Hilbert space of complex functions that are squared integrable on the unit circle. The span $\{B_i\}$ is dense in $H_2(E)$ if and only if $\sum_{i=0}^{\infty} (1 - |\xi_i|) = \infty$, where $\{\xi_i\}_{i=0}^{\infty}$ is the underlying pole set for the OBF set $\{B_i(z)\}_{i=1}^{\infty}$ [77].

By an appropriate truncation on the series of $\{B_i(z)\}_{i=1}^{\infty}$, the resulted finite sum of $\{B_i(z)\}_{i=1}^{n}$ can be sufficient to approximate the original transfer function with a relatively small number of OBFs $n$. Take this idea to the time domain and substitute the forward time-shift operator $q$ for $z$, the approximation for the general linearisable systems in the Input-Output (IO) partition form can be expressed as

$$y_k \approx \sum_{i=1}^{n} c_i B_i(q) u_k$$

(2.2)

where $u_k$ and $y_k$ are a pair of observed input and output. This simple expression is the core idea through out the OBF-based LTI system identification.

According to the standard inner product on $H_2(E)$, any two OBFs $B_{i_1}(q)$ and $B_{i_2}(q)$ have

$$\langle B_{i_1}(q), B_{i_2}(q) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} B_{i_1}(e^{j\omega}) \overline{B_{i_2}(e^{j\omega})} d\omega$$

(2.3)

The orthonormal property can thereby be examined and it will show

$$\langle B_{i_1}(q), B_{i_2}(q) \rangle = \begin{cases} 1, & \text{if } i_1 = i_2 \\ 0, & \text{if } i_1 \neq i_2 \end{cases}$$

(2.4)

The construction for a set of OBFs is very dependent on the configuration of the underlying pole set. The selected poles become optimal when they get closed to the actual system poles [77]. In this research the general repetitive pole structure is adopted [79]. It is in the form of

$$\{\xi_1, \ldots, \xi_{n_b}, \xi_{n_b+1}, \ldots, \xi_{n_b+n_b}, \ldots\}$$

(2.5)
where complex pole arranged in the complex-conjugated pair. Regarding the poles and their order in (2.5), individual OBF can be constructed in a unified approach [77] as

$$B_i(q) = \sqrt{1 - |\xi_i|^2} \prod_{k=0}^{i-1} \frac{1 - \xi_k q}{q - \xi_k}$$  \hspace{1cm} (2.6)$$

The related inner function is

$$G_b(q) = \prod_{j=1}^{n_b} \frac{1 - \xi_k q}{q - \xi_k}.$$  \hspace{1cm} (2.7)$$

The convergence rate of the series expansion (2.1) is bounded by the decay rate $\rho = \max_k |G_b(\xi_k^{-1})|$. A small decay rate implies a less approximation error. In the best case, $G$ and $G$ have the same same poles.

Except those who only take the real poles, the complex poles in (2.5) will generally cause the OBFs (2.6) to be complex-valued. This is physically unreasonable for the system identification except some certain problems in communications. A few more steps of manipulation on the formula (2.6) is needed to be taken for the acquisition of the real-valued responses [77].

Suppose a pair of the complex-conjugated poles firstly appear at orders of $m$ and $m+1$ in the pole structure (2.5). Correspondingly the two OBFs are $B_m(q)$ and $B_{m+1}(q)$. Two new OBFs $B'_m$ and $B'_{m+1}$ can be formed by linear combinations of the original $B_m$ and $B_{m+1}$ as

$$\begin{bmatrix} B'_m \\ B'_{m+1} \end{bmatrix} = \begin{bmatrix} c_{1a} & c_{1b} \\ c_{2a} & c_{2b} \end{bmatrix} \begin{bmatrix} B_m \\ B_{m+1} \end{bmatrix}$$  \hspace{1cm} (2.8)$$

where $c_{1a}, c_{1b}, c_{2a}, c_{2b} \in \mathbb{C}$. The solutions set should inherently obey the orthogonality and the normality of the OBFs, which leads to the constraints respectively

$$\begin{cases} c_{1a}c_{2a} + c_{1b}c_{2b} = 0 \\
|c_{1a}|^2 + |c_{1b}|^2 = 1 \\
|c_{2a}|^2 + |c_{2b}|^2 = 1 \end{cases}$$  \hspace{1cm} (2.9)$$

Based on the unified structure (2.6), one can derive a recursive form of $B_m$ and $B_{m+1}$ from $B_{m-1}$. Compare this alternative form with the linear combination one in (2.8), the relation can be obtained as

$$\begin{cases} c_{1a} = \frac{\beta_1 + \xi_m \mu_1}{1 - \xi_m} , \\ c_{1b} = \frac{\xi_m^2 \beta_1 + \mu_1}{1 - \xi_m^2} \\ c_{2a} = \frac{\xi_m \mu_1 + \beta_1}{\xi_m - 1} , \\ c_{2b} = \frac{\mu_1 + \xi_m \beta_1}{\xi_m - 1} \end{cases}$$  \hspace{1cm} (2.10)$$

17
regarding the later new $B'_m$ and $B'_{m+1}$. One can treat (2.8) and (2.10) in matrix quadratic form. After some matrix manipulations, it can be found then that both 
$\begin{bmatrix} \beta_1 & \mu_1 \end{bmatrix}^T$ and 
$\begin{bmatrix} \beta_2 & \mu_2 \end{bmatrix}^T$ lie on an ellipse, while the two vectors are orthogonal to each other. Any solution set \{\(\beta_1, \mu_1, \beta_2, \mu_2\)\} that fulfils the constraints (2.9) of the linear combination (2.8) can be represented as

$$
\begin{bmatrix}
B'_m \\
B'_{m+1}
\end{bmatrix} = \frac{1}{\sqrt{1 - \alpha^2}} \begin{bmatrix} \alpha & 1 \\ -1 & -\alpha \end{bmatrix} \begin{bmatrix} B_m \\
B_{m+1} \end{bmatrix},
$$

(2.11)

where $\alpha$ is

$$
\alpha = \frac{\xi_m + \xi_{m+1}}{1 + \xi_m^2}
$$

(2.12)

As a result, the new $B'_m$ and $B'_{m+1}$ based on a pair of the complex-conjugated poles $\xi_m = \xi_{m+1}$ can be constructed as

$$
B'_m(q) = \sqrt{1 - |\xi_m|^2(\beta_1 q + \mu_1)} \prod_{k=0}^{m-1} \frac{1 - \xi_k q}{1 - \xi_k}
$$

(2.13)

$$
B'_{m+1}(q) = \sqrt{1 - |\xi_m|^2(\beta_2 q + \mu_2)} \prod_{k=0}^{m-1} \frac{1 - \xi_k q}{1 - \xi_k}
$$

(2.14)

This subsection gives a quick introduction to the OBF-based model identification approach. It emphasises that the OBF model structure provides an effective way to represent the LTI systems. Essential formulas and key steps that serve for the OBF construction and implementation are highlighted. They are both theoretically and practically usefully. They will also play an important role in Section 4.4.2 when expanding the OBF model structure to an adaptive form for the nonlinear system identification.

### 2.3.2 The FKcM algorithm for the OBF pole selection

It is crucially important to select an appropriate pole set when building up the OBF model structure (2.2), because the underlying pole set determines the convergence rate of the truncated series expansion. In the linear system identification, the optimal pole selection primarily relates to minimizing a suitable norm of the output error signal [79]. However in this research the pole selection problem goes to a different situation. The task here is to find a common pole set for many linearised models. Those linearised models and their poles are to be identified from the local data sets through the proposed identification experiment later discussed in Section 4.4.1. As it has been discussed in Section 4.2 that the local data sets can be linearised accurately, the identified poles are therefore assumed to be the actual local poles. Hence
formally, the pole section problem in our research is to find a set of poles that can optimally approximate all the poles identified from the local data sets.

The FKcM approach developed by [80] is employed to solve our pole selection problem. The method is originally studied for the OBF-based Linear Parameter-Varying (LPV) system identification where an optimal pole set is to be selected for all of the sampled frozen linear time-invariant (LTI) systems [81]. Differently we seek an optimal pole set for all of the identified local LTI systems. However the two applications share the same pole selection problem. Therefore the FKcM approach can be adopted seamlessly in this research. The FKcM approach is developed based on the Kolmogorov n-width (KnW) theory and the Fuzzy c-Mean (FcM) clustering technique.

The fundamental concept is the optimal subspace in the Kolmogorov n-width (KnW) sense which is defined as

$$\pi_n(\mathcal{T}, \mathcal{H}_2 - (E)) \triangleq \inf_{\psi_n \in \mathcal{M}_n} \sup_{G \in \mathcal{T}} d_{\mathcal{H}_2 - (G, \Psi_n)}$$  \hspace{1cm} (2.15)

In the above expression, \( \mathcal{T} \) denote a set of transfer functions \( \{G\} \subseteq \mathcal{H}_2 - (E) \); \( \Psi_n = \text{span}(\Phi_n) \) where \( \Phi_n = \{\phi_i\}_{i=1}^n \) represents a sequence of \( n \) linearly independent elements of \( \mathcal{H}_2 - (E) \); \( \mathcal{M}_n \) is the collection of all \( n \)-dimensional subspace of \( \mathcal{H}_2 - (E) \); and the distance between \( F \in \mathcal{H}_2 - (E) \) and \( \Psi_n \) is defined as

$$d_{\mathcal{H}_2 - (G, \Psi_n)} \triangleq \inf_{F \in \psi_n} \|G - F\|_{\mathcal{H}_2}.$$  \hspace{1cm} (2.16)

The outcome of (2.15) defines the smallest possible approximation error for the worst-cast \( G \in \mathcal{T} \). The augment \( \hat{\Psi}_n \in \mathcal{M}_n \) is called the optimal subspace in the KnW sense. The KnW concept was studied in the \( n \)-width approximation theory [82] and it has been applied to the area of the LTI system identification like [83]. In a different work by [84], this approach is specifically studied for cases when the linear dependent elements are OBFs. The result there shows that, by the OBFs with a pole set (2.5) and a structure (2.6), the spanned subspace

$$\Phi_n = \text{span}(\{B_i\}_{i=1}^n) = \text{span}(\{U_l(z)G^j_b(z)\}_{l=1,...,b_n}^{j=0,...,n_e}),$$  \hspace{1cm} (2.17)

is optimal in the KnW sense for the set of systems with the transfer functions analytic in the complement of the region

$$\Omega(\Xi, \rho) = \{z \in \mathbb{C}, |G_b(z^{-1}) \leq \rho\},$$  \hspace{1cm} (2.18)

and squared integrable on its boundary. In (2.18), \( \rho \) is the decay rate, and the worst-cast approximation error is proportional to \( \rho^{n_e+1} \).
In the pole selection scenario, the problem become inversely addressed. Given a non-analytical region \( \Omega_p \subset \mathbb{D} \), the objective is to find an inner function \( G_b \) to approximate this region in the form (2.18) with \( \rho \) as small as possible. Regarding the non-repeated poles to be selected for (2.5) and the non-analytic region induced by sampled poles \( z \), a min-max problem can therefore be formulated as [80]

\[
    \min_{\xi_1, \ldots, \xi_{n_b}} \max_{z \in \Omega_p} \left| \frac{z - \xi_j}{1 - z \xi_j} \right|.
\]  

(2.19)

A repetitive pole set as (2.5) based on the non-repeated part \( \{\xi_1, \ldots, \xi_{n_b}\} \) can make the inner function \( G_b \) best fit the region \( \Omega(\Xi_{n_b}, \rho) \), and the resulted OBFs \( \{B_i\}_{i=1}^n \) are optimal in the KnW sense with \( n = (n_e + 1)n_b \).

The solution of (2.19) was sought by [81] in the scope of Fuzzy c-Mean (FcM) clustering, where the pole selection problem is transformed to a pole clustering problem as formulated below.

**Problem 1** For a set of sampled pole locations \( Z \) and for a given number of clusters \( c \), find a set of cluster centres \( \{v_1\}_{i=1}^c \), a set of membership functions \( \{\mu_i\}_{i=1}^c \), and the maximum of \( \varepsilon \), such that

- \( \Omega_\varepsilon \) contains \( Z \)
- With respect to \( \Omega_\varepsilon \), the OBFs with pole \( \Xi_\varepsilon \) in the cluster centres \( \{v_i\}_{i=1}^c \) are optimal in the KnW sense, where \( n = c \).

The pole clustering problem above is asymptotically related to the KnW sense (2.19) by defining the Kolmogorov metric (KM) of \( \mathbb{D} \) as

\[
    \kappa(x, y) := \frac{|x - y|}{1 - xy^*} : \mathbb{D} \times \mathbb{D} \to \mathbb{R}_0^+,
\]  

(2.20)

which is seen as the 1-width version of (2.19). The KM metric is introduced between \( v_i \) and \( z_k \) to measure the dissimilarity, denoted for \( Z \) with respect to each candidate cluster. Take the \( d_{ik} = \kappa(v_i, z_k) \) into the FcM functionals, Problem 1 is formulated as

\[
    J_m(U, V) = \max_{\kappa \in \Gamma_N} \sum_{i=1}^{c} \mu_{ik}^m d_{ik}
\]  

(2.21)

The FKcM algorithm is yielded as

**Algorithm 1** Fuzzy-Kolmogorov c-Max

1. **Initialization**
   - Fix \( c \) and \( m \); and initialize \( V_0 \in \mathbb{D}^c, l = 0 \)

2. **Membership update**
With (16), solve \( U_{i+1} = \arg\min_{U \in U} J_m(U, V_i) \)

3. Cluster center update

With (17), solve \( V_{i+1} = \arg\min_{V \in \mathbb{R}^c} J_m(U_{i+1}, V) \)

4. Check of convergence

If \( J_m(U_{i+1}, V_{i+1}) \) has converged, then stop, else \( l = l + 1 \) and go to Step 2.

To begin with this section, we emphasized the importance of an appropriately selected pole set for the OBFs, and explained the objective of the pole selection problem. Then the \( n \)-width theory is introduced and the pole selection problem is firstly formulated in the KnW sense. Relating the KnW concept to KM, the problem is transformed into the fuzzy clustering context where it can be solved by the FKcM approach. In the later work, the FKcM algorithm will be used to calculate the optimal pole set. Particularly in the two-chamber steering and bending case presented in Section 4.5.3, the clustering algorithm will be individually taken on each of the SISO LTI parts after the decomposition.

2.4 Summary

In summary, from mechanical design side, soft inflatable robots are at an expectorating stage. Preliminary prototypes have demonstrated many promising applications such as those in gripping tasks. Yet, large research efforts are still needed to build soft robots for wider ranges of tasks. Meanwhile, in the robotic manipulation side, the current solutions largely referred to the methodologies of hard continuum robots. In the long term, a more octopus-like robots is expected. By that time, model-based approach may be insufficient, as the dimension of system behaviours will be much higher than existing rigid-link robots. Hence to be inline with improved mechanical designs in the near future, statistics-based system identification approaches are concerned. Then the rest of the chapter introduced an OBF-based system identification approach selected in this study.
Chapter 3
System Implementation and Preliminary Experiments

3.1 Introduction

This chapter will introduce the platform setup for operating general soft inflatable actuators. The platform developed provides a real-time environment so that actuation and sensing signals can be sent and received in a rigorous schedule; and the data collected in this way is reliable and easy to process. The fluidic transmission system is controlled by a unit of two solenoid valves and one pressure sensor. The measurement by a 6-axis sensor is used to estimate the shape of soft actuator during deformation. Besides, in order to tune the platform setup, a few preliminary experiments are conducted and the corresponding results are presented.

3.2 Platform Overview

To implement a system capable of identification and control design, a close-loop real-time platform was constructed initially using Matlab Real Time Windows environment. A trackSTAR magnetic sensor was used; it provided 6-axis measurement. The pressure control was implemented by a unit of two solenoid valves and a pressure sensor. The developed system platform is easy to setup, and can provide a reliable testing experiment. This approach can be used for general soft fluidic actuators.

Figure 3.1 illustrates the configurations of the experimental platform. Its real picture is shown in Figure 3.2. This platform has both a close-loop and a real-time environment. The system identification and control design in the further will both rely on this platform. In the current setting, every component in the platform operates at its pre-specified level. And a compressor provides the overall pneumatic pressure.
Figure 3.1: Configuration of the preliminary platform.

Figure 3.2: Photo of the preliminary platform setup.

Figure 3.3 shows the Simulink model governing experimental procedures. As divided by the three red dash-line rectangular blocks, the Simulink model was developed to implement three functions: actuation, sensing, and generating task-dependent actuation trajectories. The sensing part reads the position and orientation of the magnetic sensor in a pre-defined coordinate. The block for task-dependent actuation trajectories is used to realise various identification input pressure signals, e.g. step responses. In the future, with introducing controller and feedback loop, this part can be used to implement control design. The actuation part is realised by a regulation scheme. It receives a reference pressure from the previous trajectory generation part,
and then used feedback control to remove the measured run-time pressure error.

3.3 The Measurement

The trackSTAR magnetic sensor contains a transmitter set and a micro sensor. For installation, the micro sensor is inversely and perpendicularly placed on the tip centre of the single section bending soft tube; while the transmitter is fixed at the test table. During tests, the transmitter and the micro sensor give the global and local reference frame, i.e. $G(OXYZ)$ and $B(oxyz)$ respectively, as shown in Figure 3.4. One measured sample includes translational position $(X,Y,Z)$ and angular position $(\phi, \theta, \psi)$. The former shows the offset vector between $G$ and $B$. The latter represents a local yaw-pitch-roll rotation.

The constant curvature condition was verified in Figure 3.5. The observation shows the shape of a bending actuator can be simplified as a constant curvature arc in kinematic and dynamic analysis. So the original goal here was to derive the configuration space parameters $(\kappa, l, \psi)$ from the measured translation positions $(X, Y, Z)$ and angular position $(\phi, \theta, \psi)$. Then it was realised that the calculation of constant curvature $\kappa$ encounters singularity when the bending was in a small degree, i.e. nearly straight. A solution to this problem is to set a threshold for small bending angle; when the bending angle is small and below the threshold, the curvature $\kappa$ is set to a constant large value instead of being obtained via calculation. This solution is not recommended as it introduces discontinuity. A better solution is to release the constant curvature assumption in configuration space. In this way, the bending
angle $\theta$ is used in place of the curvature $\kappa$; the other two parameters, plan angle $\phi$ and the arc length $l$ are unaltered. $(\theta, \phi, l)$ are the configuration space parameters now. The new configured triplet is more convenient to perceive the bending dynamics. The disturbance introduced by elongation can be diminished. The term curvature radius $r$, which is highly coupled with the length of the tube, is not used in calculation. The comparison between using curvature $k$ and bending angle $\theta$ is shown in Figure 3.6

Although the constant curvature assumption is relaxed in analysing bending $\theta$, measuring length $l$ still needs this condition for approximation. Moreover, it would be helpful to mention here that the axial direction of the micro sensor goes with the $x$-axis of the frame $B$ (illustrated Figure 3.7). This axis will play an important role in deriving the bending angle $\theta$ and the plane angle $\phi$. The reference frames of the transmitter and the micro sensor are shown in Figure 3.8
Figure 3.5: The bending actuator was keeping increased constant curvatures during bending.
Figure 3.6: Bending measurement comparison: curvature $k$ versus bending angle $\theta$

Figure 3.7: (a) An oblique view of the tip surface. (b) A side view of the soft bending actuator. (c) The ideal installation of micro sensor on the soft bending actuator.
Figure 3.8: A setup of the transmitter and the micro sensor.

Figure 3.9 shows a more general situation of measuring the bending and the plane angles. The two parameters are extracted from the orientation of the tip surface of a bending actuator. The following two subsections will present the calculations for obtaining the two angles.

Figure 3.9: Measurement setting at the tip end of a bending actuator. The local reference frame is based on the trackSTAR™ magnetic sensor.
3.3.1 Bending angle

Figure 3.10: Bending angle $\theta$ in two cases. (a) $\theta$ is shown in the bending plane. (b) $\theta$ is shown in the 3D reference frame, where the frame $O' - X'Y'Z'$ is parallel to the transmitter global frame $O - XYZ$.

In ideal cases, the axial direction of the micro sensor is always perpendicular to the centre of the tip surface (Figure 3.7). It can be found that the bending angle $\theta$ is equivalent to the intersection angle between the $x$-axis of the local frame $B$ and the $Z$-axis of the global frame $G$. This relation is illustrated in Figure 3.10. The formula of $\theta$ is

$$\theta = \arccos(\langle B_{eZ}, B_{ex} \rangle)$$

(3.1)

where $B_{eZ}$ is a unit vector along $Z$-axis in $G$ transformed to $B$, $B_{ex}$ is a unit vector along $x$-axis in $B$. By using the formula,

$$B_{AG} = A_{x,\psi}A_{\theta,\theta}A_{z,\phi}$$

(3.2)

the yaw-pitch-roll rotation matrix is obtained as,

$$
\begin{bmatrix}
\cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\
-\cos \psi + \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \theta \sin \psi \\
\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\cos \phi \sin \psi + \sin \theta \cos \psi \sin \phi & \cos \theta \cos \psi
\end{bmatrix}.
$$

(3.3)

The $B_{eZ}$ can be got from the $B_{ex}$, i.e. the unite vector in $G$. So the equation (3.1) becomes

$$\theta = \arccos(\langle B_{AG}G_{eZ}, B_{ex} \rangle), \quad G_{eZ} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad B_{ex} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

(3.4)
The equation derived above has been experimentally validated. Figure 3.11 shows the calculated bending angles measured from different experiments.

![Figure 3.11: Bending angles that are calculated from experimentally measured task space coordinates \((X, Y, Z, \phi, \theta, \psi)\). Three different cases are showed.](image)

However, some practical issues are necessary to address. The micro sensor is not strictly perpendicular to the tip surface, which causes an offsetted deflection angle. This angle can be further decomposed to two orthogonal parts, one within the bending plane and one perpendicular to the bending plane, i.e. similar to \(\theta\) and \(\psi\) in Figure 3.10. The former, i.e. the in-bending plane deflection, can be seen as a constant offset value. For example, the offset value is about 11 degree in the experiment of Figure 3.11. On the other hand, the off-bending plane deflection, does not disturb the bending measure here. Instead, it introduces an offset to the measurement of plane rotation angle \(\phi\).

### 3.3.2 Plane angle

After knowing the task space coordinates \((X, Y, Z, \phi, \theta, \psi)\), there are two possible approaches to derive the plane angle \(\phi\). The first approach uses the simple formula

\[
\phi = \arctan \left( \frac{\rho' e_y'}{\rho' e_x'} \right) \tag{3.5}
\]

where \(\rho' e_y'\) and \(\rho' e_x'\) are coordinates with respect to the reference frame \(O' - X'{Y}'{Z}'\), i.e. Figure 3.7 (b). However, the calculation of this method becomes in-
feasible when the bending angle $\theta$ is very small, i.e. within $20^\circ$. The main reason behind it is the limited resolution of the magnetic sensor; the orders of $\Omega Y$ and $\Omega X$ are frequently varying when the bending actuator is in a small bending angle. This leads to highly oscillated arctangent results. In contrast, the second approach (discuss later) performs fairly reliable. Figure 3.12 shows an experimental comparison between the first and the second approaches. The tube stayed statically at a very small bending angle $\theta$, i.e. less than $15^\circ$. The measurement using the first approach was fluctuating over a very large scale, i.e. $\pm 150^\circ$; this was apparently against the actual situation. In contrast, the second approach achieved a stable and accurate measurement. So the second one is adopted, which will be explained next.

![Figure 3.12: An experimental comparison between the calculated plane angles according to Method 1 and Method 2 respectively. The tube is within a very small bending angle $\theta$, i.e. less than $15^\circ$.](image)

The idea of the second approach is to find the intersection angle $\phi$ between the projected line of $x$-axis on the plane $O'X'Y'$ and the $X'$-axis, as shown in Figure 3.13. This intersection angle $\phi$ is the plane (rotation) angle.

Based on the known conditions, a geometry question is formulated. It is illustrated in Figure 3.14 with the reasoning beside.
Figure 3.13: The plane angle $\phi$ is the intersection angle between the projected line of x-axis on the $X'Y'Z'$ plane and the X-axis. The $O'-X'Y'Z'$ is as the same definition in Figure 3.8.

The geometric relation between $\phi$, $^B_a e_x$, $^B_G e_X$ and $^B_G e_Y$ is equivalently presented by a pyramid in Figure 3.14. From the relation, it can be immediately obtained

$$OB \perp ABC, OB' \perp AB'C.$$ 

This leads to

$$\frac{OB}{AB} = \frac{OC}{OB} \quad \frac{OB'}{AB} = \frac{OC}{OB'} \quad \frac{OB'}{OC} = \frac{AB'}{OC}$$
and correspondingly

\[
\cos \angle AOB = \cos \angle COB \cos \angle AOC,
\]
\[
\cos \angle AOB' = \cos \angle COB' \cos \angle AOC.
\]

As \( \angle COB + \angle COB' = 90^\circ \), we get

\[
\frac{\cos \angle AOB'}{\cos \angle AOB} = \frac{\sin \angle COB}{\cos \angle COB} = \tan \angle COB
\]

Hence, the steering angle can be expressed as

\[
\phi = \arctan \left( \frac{\langle B e_x, B e_y \rangle}{\langle B e_x, G e_X \rangle} \right).
\] (3.6)

Taking the transformation of (3.3) into (3.6), the steering angle can be calculated as

\[
\phi = \arctan \left( \frac{\langle B e_x, B A G e_Y \rangle}{\langle B e_x, B A G e_X \rangle} \right)
\] (3.7)

where \( B e_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \), \( G e_X = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \) and \( G e_Y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \).

Compared with the variable chamber length way of calculating the plane angle [55],

\[
\phi = \arctan \left( \frac{\sqrt{3}(l_5^c + l_5^c - 2l_1^c)}{3(l_3^c - l_2^c)} \right),
\] (3.8)

(3.7) does not use the term \( l_3^c - l_2^c \), i.e. the difference between two chambers’ lengths. The latter is sensitive to disturbed sensor reading for chamber lengths, while the former has no such a problem.

The experimental results obtained using the equation (3.7) is shown in Figure 3.15. The dynamic response of the plane angle is presented with relatively high quality. However, there are some problems to consider. One problem, as mentioned in the previous section, is the offset issue. Different from the in-bending plane case, after a certain time of testing, a new offset could appear as shown in Figure 3.15. This is possibly due to some nonlinear dynamics, e.g. relay with hysteresis. This will be further investigated.

Another problem relates to the boundary issue. The range of atan2 function is \((-180^\circ, +180^\circ]\). When the plane angle \( \phi \) changes from \(+180^\circ\) to \(-180^\circ\), a discontinued jump happens (the blue line in Figure 3.16). This is undesired, because the kinematics and dynamics are physically continuous. A modified algorithm can be used to partially solve the problem (the green line in Figure 3.16). The extra function of the new algorithm is to give extra margin when \( \phi \) advances beyond the defined boundary. For example, an trajectory \(+170^\circ, -170^\circ, +170^\circ\), can be shifted
Figure 3.15: Experimental determination of plan angle $\phi$.

to $+170^\circ$, $+190^\circ$, $+170^\circ$. However, this only works when $\phi$ inversely comes back, i.e. along its previous path. For a $360^\circ$ rotation that the sign of the derivative of $\phi$ is unaltered, the modified algorithm does not work (the second half of the green line in Figure 3.16). Further investigation on this problem is needed.

Figure 3.16: A comparison between the original plane angle calculation algorithm and its modified algorithm.
3.3.3 Arc length

After knowing the bending angle $\theta$ and assuming the bending actuator shaped as a constant curvature arc, the arc length $l$ can be calculated as

$$l = \frac{\theta d}{1 - \cos \theta}$$

(3.9)

where $d$ is the deflection distance, i.e. length of the vertical line from the micro sensor origin $O$ to the original axial line $Z$-axis; $\theta$ is the calculated bending angle obtained in previous step. Use of this equation can encounter the same singularity problem as in the curvature issue (Figure 3.6). Measuring the arc length is still an open issue. Using a specific bending actuator sensor will be a good solution.

3.4 Actuation

A solenoid valve that was controlled by Pulse-Width Modulation (PWM) signal was previously used to regulate one chamber’s pressure. This method was dropped mainly due to the vibration issue. The current configuration for one chamber pressure regulation is shown in Figure 3.17.

![Figure 3.17: Configuration of the two valves system](image)

Two valves are used for one chamber’s pressure regulation. Each valve is two port and normally closed. The valve A is in charge of injecting air, while the valve B is used to drain out air. When both the valves are open, a closed environment can be formed to maintain the in-chamber pressure. This is the third state regardless of
increasing and decreasing the pressure, i.e. a state of no action. It allows a dead zone for the tracking error, within which no control action need to take. By doing this, vibration caused by regulating small error can be eliminated. However, enlarging the dead zone range will reduce the regulation accuracy. Conversely, if the dead zone is too small, vibration will be caused during regulation. In the current tuning, the vibration can still happen some time. The problem is being further studied. A test pressure regulation against the tracking signal is shown in Figure 3.18

![Figure 3.18: Pressure regulation (above) under the tracking signal (below). The measurement was not calibrated yet.](image)

The vibrations in the PWM case and the current one are due to different reasons. In the PWM case, the vibration is caused by the periodical pulse. Especially in low duty factor, the vibration become more intensified. On the other hand, the vibration in the current configuration is due to the regulation: the tracking error is overly compensated after every control loop.

### 3.5 Preliminary Identification 1: Independent Step Responses

The model identification starts with a simple case, one directional bending by only one chamber. So the system is single-input-single-output (SISO). To capture the character of the bending dynamics, the pressure of the feed-in air flow is used as the input; while the deflection (bending) angle is the output. The identification experiment was to collect data of step responses under different values of input.

In the experiment, the input pressure was recorded in the measure of voltage. The reason of not converting it to the physical pressure unit was because the reference voltage of the pressure sensor had not been calibrated to its physical quantity. From
previous quick tests, it had been observed that the sensor was very sensitive to the pressure dynamic change within the operation range, i.e. 0 ~ 0.8 bar. Therefore, it was safely assumed that the measured voltages linearly mapped the in-chamber pressure. The output was measured in degree (°). The range was between 0° and 180°, as it was single-direction, half plane bending. Figure 3.19 shows the raw data from the identification test. The test can be further divided to 20 individual tests. Each small case recorded a course of the deflection angle variation from initial steady state, response subject to air flow, and steady state again after exhaustion of the air flow. The magnitude of the feed-in pressure was the variable, which was set from 0.080 to 0.120V with 0.002V as the increment throughout the entire test. From the raw data of Figure 3.19, it can be seen that a constant offset exists in both the input and the output channels. The former input offset was due to the sensor electric character. For the latter offset in the output channel, it was mainly due to the sensor mounting issue, in which case, the axis for orientation measurement was not initially perpendicular to single tube tip surface. Another reason was a small resulting intersection angle between the soft tube tip surface and the horizontal plane. This was due to the tube deformation subject to gravity or some design issues. From the analysis above, it is safely to regard the offsets in input and output channels as constants. Thereby they are removed in the following identification process.

From observation, the time response of each sample behaved relatively in low order,
Table 3.1: Step responses at different pressures.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Input Pressure (V)</th>
<th>90% Setting Time (ms)</th>
<th>Steady State Bending Angle (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0154</td>
<td>212.5</td>
<td>1.6</td>
</tr>
<tr>
<td>2</td>
<td>0.0175</td>
<td>225.0</td>
<td>2.3</td>
</tr>
<tr>
<td>3</td>
<td>0.0195</td>
<td>237.5</td>
<td>3.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0215</td>
<td>275.0</td>
<td>4.1</td>
</tr>
<tr>
<td>5</td>
<td>0.0233</td>
<td>275.0</td>
<td>5.5</td>
</tr>
<tr>
<td>6</td>
<td>0.0253</td>
<td>337.5</td>
<td>7.5</td>
</tr>
<tr>
<td>7</td>
<td>0.0300</td>
<td>587.5</td>
<td>10.8</td>
</tr>
<tr>
<td>8</td>
<td>0.0320</td>
<td>625.0</td>
<td>14.7</td>
</tr>
<tr>
<td>9</td>
<td>0.0335</td>
<td>625.0</td>
<td>18.9</td>
</tr>
<tr>
<td>10</td>
<td>0.0340</td>
<td>637.5</td>
<td>21.2</td>
</tr>
<tr>
<td>11</td>
<td>0.0353</td>
<td>675.0</td>
<td>24.0</td>
</tr>
<tr>
<td>12</td>
<td>0.0355</td>
<td>700.0</td>
<td>27.5</td>
</tr>
<tr>
<td>13</td>
<td>0.0373</td>
<td>700.0</td>
<td>32.8</td>
</tr>
<tr>
<td>14</td>
<td>0.0382</td>
<td>n/a</td>
<td>52.2</td>
</tr>
<tr>
<td>15</td>
<td>0.0402</td>
<td>n/a</td>
<td>59.3</td>
</tr>
<tr>
<td>16</td>
<td>0.0427</td>
<td>n/a</td>
<td>81.2</td>
</tr>
<tr>
<td>17</td>
<td>0.0445</td>
<td>n/a</td>
<td>95.3</td>
</tr>
<tr>
<td>18</td>
<td>0.0461</td>
<td>n/a</td>
<td>111.3</td>
</tr>
<tr>
<td>19</td>
<td>0.0485</td>
<td>n/a</td>
<td>124.4</td>
</tr>
<tr>
<td>20</td>
<td>0.0501</td>
<td>n/a</td>
<td>134.8</td>
</tr>
</tbody>
</table>

i.e. ≤ 3rd. So firstly, the analysis takes a quick glance at two explicit characters from the data. They are 90% setting time and steady state magnitude, which are listed in Table 3.1.

In Table 3.1, it is seen that the input pressure was linearly increased. Meanwhile, the 90% setting time was also gained and its tendency is very likely in the same order with the pressure rate. The setting time data from Sample 14 to Sample 20 were not available due to the pressure regulation issue. On the other hand, the relation between the input pressure and the static bending angle was polynomial. This reflects the same conclusion from the work [23] in the past.

The second step was to further confirm the orders of the input and the output channels. The nonlinear auto regressive exogenous (NARX) model was used in this practice. A standard NARX model is shown in Figure 3.20. The test employed Matlab System Identification Toolbox.

After a number of trials, the best fitting was achieved by selecting $u(t - 0)$, $u(t - 1)$ from input channel and $y(t - 1)$, $y(t - 2)$ from the output channel as the regressors.
The time delay between the input and the output channels didn’t play a critical role in the fitting test. The fitting error dramatically surged when the estimating delay was over 3 samples. Another particular point is, the nonlinear block only appeared strong dependency to the input channel regressors. Without customising the regressors for the nonlinear block, 92.3% fitting is currently the best result. Its comparison with the raw data is shown in Figure 3.21. The outcome from the NARX fitting and the straightforward observation in the previous steps have several implications on the dynamics of the single bending actuator. Primarily, according to the best fitting model, the input side was 1st order and the output side was 2nd order. Meanwhile from the observation of Figure 3.19 and Table 3.1, it can be pointed out that the step responses in different cases behaved similar dynamic pattern, i.e the response curves. So it is possible to lead to the conclusion that the variation of the input pressure was the major source of the underlying nonlinearity; the order of the dynamics are maintained throughout the full pressure range; the mechanical structure was changed. Besides, the steady state bending angle was apparently nonlinear with respect to the linearly increasing levels of the input pressure. The
reason lies in the pressure itself. For the case of increasing the pressures, it led to increased damping ratio, decreased natural frequency and increased steady state value.

Apart from the conclusion derived from the previous NARX model analysis, the identification process also included the Hammerstein-Wiener (HW) model. Its standard form is illustrated in Figure 3.22.

After many trials of different nonlinear configurations, it was turned out that results of confident fittings could be achieved by only setting the input nonlinearity as a piecewise linear function while the linear block had two poles and one zero. Over 90% fitting could be guaranteed by using three linear units. Overly increasing the number of the linear unit would not further improve the closeness to the actual data. Instead, the fitting results converged to about 94%. Figure 3.23 shows one of an input function with 5 piecewise linear units, which led to a 92.3% fitting.

Comparing the HW fitting outcome with the previous NARX one, a common point is that each one’s linear part has two poles and one zero. This can be seen as a concrete condition for the bending model being identified. However, a critical discrepancy between them is: the coefficients in the NARX linear part is variable, which means
the placements of poles and zeros are not fixed; in contrast, the linear transfer function in the HW one has a fixed pair of poles and zeros. This fact will result in different understanding on the underlying physics although they achieved closed fitting results. Models identified based on the two different nonlinear mechanisms are fundamentally different, which will affect the control design in the later stage. Therefore, it is necessary to continue researching on this issue: the source and representation of the underlying nonlinearity. Text Case 2 discussed next is for this purpose.

3.6 Preliminary Identification 2: Dependent Step Responses

In Test Case 2, the platform remained same as in Test Case 1. The difference was the planning of the step response commands. In Test Case 1, each step response in each sample was independent. The initial pressure always started from zero, and it always returned to zero too. In Test Case 2, a step response started from the steady state of its previous one, and it would be maintained at its own steady state for the beginning of the next step response. In other words, the starting point of each step response in Test Case 2 was initially offsetted by its previous steady value.

The raw data of Test Case 2 is displayed in Figure 3.24, which has been pre-globally offsetted. It comprises 9 samples, each of which is an individual step response. The 9th sample is ignored, as the starting bending angle had already been nearly 180°, further increasing the input pressure would not further bend the tube over 180°. The time intervals of first 8 samples are listed Table 3.2 below.

Although there are 8 individual samples in Figure 3.24, the actual number of the step shapes appear more than this. Each of the 3rd, 4th and 5th intervals has two steps, which was due to the pressure regulation problem in Section 3.4.

A linear analysis was carried out for each sample. The offset of each sample that was due to its previous sample’s steady state value was pre-removed. The transfer function model used in fitting took the same order structure as the suitable NARX and HW models in Test Case 1, i.e. 2 poles and 1 zero. There were about a

<table>
<thead>
<tr>
<th>Sample</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 ~ 13s</td>
<td>13 ~ 23s</td>
<td>23 ~ 32s</td>
<td>32 ~ 43s</td>
</tr>
<tr>
<td>Sample</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>43 ~ 53s</td>
<td>53 ~ 63s</td>
<td>63 ~ 73s</td>
<td>73 ~ 83s</td>
</tr>
</tbody>
</table>

Table 3.2: Time intervals of each sample in Test Case 2
constant two-sample length time delay between the input and output channels. It was neglected in the linearisation analysis. The fitting results of each sample are listed in Table 3.3.

Table 3.3: Approximated transfer function for each sample in Table 3.2 from the raw data in Figure 3.24. The fitting results in the forms of Fit-To-Estimation (FTE), Final Prediction Error, and Mean-Squares Error (MSE) were used.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Approximated Transfer Function</th>
<th>FTE</th>
<th>FPE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-\frac{(1777\pm738.6)s+(31020\pm6320)}{s^2+(13.3\pm2.373)s+(48.96\pm9.963)})</td>
<td>97.44%</td>
<td>0.1399</td>
<td>0.1384</td>
</tr>
<tr>
<td>2</td>
<td>(-\frac{(25920\pm1169)s+(17480\pm1516)}{s^2+(9.404\pm0.4484)s+(4.258\pm0.3773)})</td>
<td>93.79%</td>
<td>0.07083</td>
<td>0.06987</td>
</tr>
<tr>
<td>3</td>
<td>(-\frac{(22140\pm1300)s+(5071\pm774.7)}{s^2+(6.792\pm0.3916)s+(0.966\pm0.1822)})</td>
<td>93.55%</td>
<td>0.2258</td>
<td>0.2224</td>
</tr>
<tr>
<td>4</td>
<td>(-\frac{(26500\pm1532)s+(15360\pm1580)}{s^2+(9.128\pm0.6057)s+(3.319\pm0.3531)})</td>
<td>96.01%</td>
<td>0.1136</td>
<td>0.1122</td>
</tr>
<tr>
<td>5</td>
<td>(-\frac{(27360\pm1540)s+(25560\pm2238)}{s^2+(12.42\pm0.7407)s+(7.977\pm0.7068)})</td>
<td>94.42%</td>
<td>0.05203</td>
<td>0.05132</td>
</tr>
<tr>
<td>6</td>
<td>(-\frac{(18830\pm838.8)s+(9114\pm754.4)}{s^2+(9.039\pm0.4023)s+(3.785\pm0.3221)})</td>
<td>93.07%</td>
<td>0.02591</td>
<td>0.02556</td>
</tr>
<tr>
<td>7</td>
<td>(-\frac{(19580\pm1573)s+(11110\pm1663)}{s^2+(13.08\pm1.093)s+(4.952\pm0.7491)})</td>
<td>93.15%</td>
<td>0.019006</td>
<td>0.0188</td>
</tr>
</tbody>
</table>

Table 3.3 indicates in a certain range of the input pressure the bending dynamics behaved linearly even though each sample took a transfer function with different coefficients. This outcome supports the conclusion of the NARX analysis rather than the HW one. Furthermore, if the sample interval can be narrowed, possibly to a point, the system might completely become linearly at this point. This idea is not physically feasible as every motion of the bending actuator is driven by the pressure so that the pressure can not be fixed at any one point. However this idea still
trigger the motivation to generalise the bending dynamics over different operating pressures, which leads to the consideration of identifying a linear parameter varying system.

3.7 Summary

This chapter has presented preliminary platform setup for the later identification experiments. The platform arranged all the functions, sensing, actuation, and task-specified actuation trajectories, in real-time. Such a setup could make the following identification experiments less difficult. Also, some initial identification tests with using on-shelf model identification structures showed the non-linear behaviours of a bending actuator can be captured by modified linear systems. This initial identification outcome greatly encouraged to use an improved modification on linear models to capture general non-linear dynamic behaviours of soft bending actuators.
Chapter 4

A Unified Identification Approach for a Class of Pneumatic Soft Actuators

4.1 Introduction

In the fields of surgery and therapy, soft robotic instruments begin to draw researchers’ attention in the past 20 years [85, 86]. Compared with traditional rigid ones, the soft robots can potentially give a more viable solution when the operation is taken in human bodies. The inherent compliance offers human friendly interaction; the kinematic redundancy provides the manoeuvre competency in highly unstructured in-body organ environments. One promising choice is the class of Pneumatically-driven Low-pressure Soft Actuators (PLSA). They are designed compactly in small sizes and operated safely at low pressures. An immediately foreseeable application is endoscopes [87]. It can overperform the current rigid endoscopes by giving the doctors increased adroitness meanwhile reducing damage on the patients. Figure 4.1 shows three typical PLSA examples. In this thesis, we specifically use the term PLSA to distinguish it from the Pneumatic Muscle Actuators (PMA) like in [72]. The latter commonly have much larger sizes and higher operation pressures. But both the PLSAs and the PMAs are in the class of the pneumatically-driven soft continuum robots, and they share many similar properties.

Figure 4.1: Featured PLSA prototypes: (a) Flexible Micro-Actuator by [88]. (b) Colobot by [23]. (c) An early design of the STIFF-FLOP actuator.
For the general pneumatic soft continuum robots, the kinematic modelling is still the backbone in the state-of-the-art control designs [23, 42, 43, 67, 89], which is based on a purely geometrical description of the actuators’ shapes. Completely validated dynamic modelling for a single section of the pneumatic soft robot arms is lagging behind. The potential applicability is therefore restricted. The modelling has been analytically studied in [64, 65]. The Lagrange formation under the constant curvature condition was used to derive the dynamic relation between the individual chamber lengths and the shape. The worked models are ineffective in handling the coupling effect between each chamber. Unlike the PMAs, this internal effect can not be neglected in the PLSA cases, because the latter are structured in one integral rubber material. The radial expansion of each individual chamber is not constrained as the way by the individual bellow suits used in a PMA. Besides, the above modellings do not include the complex nonlinear dynamic relation between the actuating pressures and the chamber lengths, which is an independent open problem too. A recent work by [90] started to tackle the issue of lacking the actuator dynamics based on the previous work in [65]. The proposed model correctly accounts for the identified hysteretic behaviour using the Bouc-Wen restoring force method. This research is on the right track, but it is mainly investigated for the PMAs. Due to the structural difference, the involved intermediate step of the separate pressure-elongation identification for a PMA’s individual bladders can not be applied to the internal chambers of a PLSA. And the internal coupling issue has not been formally studied. Besides, another work by [91] directly used a second-order linear transfer function to approximate the pressures-shape relation. But only the single chamber bending situation was partially studied. The model is only valid for a small degree of bending. The deformation was measured based on the deflection distance projected onto the horizontal base plane. This kinematic variable, however, does not naturally represent the bending deformation in a linear manner.

Commonly, the PLSAs have the 1-, 2-, 3- and 4-chamber designs. Among them, the 3-chamber design is the most versatile type for robotic applications. It has all three DOFs compared with the 1- and the 2-chamber ones, and at the same time it uses one less chamber in contrast to the 4-chamber one. Our work will focus on the general 3-chamber PLSA design as sketched in Figure 4.2-(a). It is designed optimally in a very symmetric way. Three identical pneumatic chambers are regularly disposed at 120° apart, and they are together parallel to the central longitudinal axis of the actuator.

Figure 4.2-(b) gives a brief explanation for the ideal bending. The outer covering of a PLSA usually wear a kind of bellow suits so that the radial expansion induced by the fed-in pressure is largely reduced and can be neglected. The individual chamber elongations are the main type of the deformations structured by the bellow. Thereby they deterministically drive the deformation of the entire actuator. The related
DOFs are classified as bending, steering, and elongating. As the case in Figure 4.2-(b), Chamber 2 takes the longest incremental length $\Delta l_2$ as it is subjected to a higher pressure than Chamber 1 and 3, which makes Chamber 2 lead the bending direction.

4.2 Observed Nonlinearities

Figure 4.3 shows a typical example of the single chamber deflection\(^1\) - actuation relation from our experiments. The test consists of 28 local step responses, each of which starts the initial condition from its predecessor’s final condition. And both the pressure increase and the decrease situations are taken.

The first nonlinearity type that can be spotted is the dead-zone effect at the initial pressure range. It is due to the gas fill-in phase during the initial actuation. Secondly, in the pressure increase cases, it can be seen that the steady state values of each local step response are not in a linear relations with their input ones. The similar things happen in the pressure decrease cases. By further comparing both the increase and

---

\(^1\)The deflection can be seen as the complementary motion of the bending. And they are equivalent under the constant curvature condition. In the rest of the paper, we will use the two words interchangeably when no confusion is caused.
the decrease cases, it is not hard to figure out an underlying hysteresis loop. The cause of the static nonlinearity roots in the nonlinear elongation-pressure relation of a chamber or bladder. Especially the hysteresis effect is due to the friction with the braid [86, 92, 93]. At last, we extracted the data sets of the increase cases, removed their offsets on both the input and output channels, and then put them together aligned to a same trigger time as shown in Figure 4.4. The post-analysis showed that all those data sets can be linearised very well, but each linearised model is formed in an inconsistent model structure with others and they behave in different transient responses. They have very similar increments at the input channel. But as indicated from the output channel, the slopes under the local responses became increasingly steep as the operating pressure rose step by step. A similar result was also obtained in the decrease cases. In run time, varying the fed-in pressure alters the actuator’s stiffness, which is the main reason that causes the dynamical nonlinearity here. On the other hand, the analysed static and the dynamical nonlinearities also inherit in the cases when operating more than one chambers. Additionally, new types of nonlinearities are introduced due to the coupling effect between the active chambers.

The state-of-the-art nonlinear identification model structures might appear to be deficient to describe the combined static and dynamical nonlinearities observed in Figure 4.3 and 4.4. We tested the Hammerstein-Wiener and the nonlinear Auto-Regression with eXogeneous variables identification approaches. And by them, an Input-Output sequence sampled from a PLSA experiment could be well fitted. But the identified models could not be fully validated when applying them to different scenarios. The nonlinear estimators in the HW and the nonlinear ARX model structures incorporate with a fixed linear part [94, 95]. The structure settings of this kind might not be capable of closely capturing the varied dynamical proprieties observed particularly in Figure 4.4. The intelligent methods such as the artificial
neural network are adoptable, but the implementations are usually computationally costly. And their further extensions to the control design is needed to be differently reconsidered.

As the fact that the local data sets have already shown strong linearity, we considered to use them to reconstruct the global nonlinear dynamical relation directly. In this way, the nonlinear estimators with the nonlinear weighting on the regressors are not used, and modelling error caused by the inconsistent nonlinear mapping between the input and the output channels will therefore be avoided. Alternatively, the local linearised dynamics is preserved and transferred to the global sense. Besides, the local linearity idea is potentially compatible with the mature linear robust control techniques. The two concerns stated above led to the motivation behind the later proposed new model identification structure for the PLSAs in Section 4.4.2.

4.3 Kinematic Modelling

4.3.1 The current kinematic setting

In the mainstream approaches, the kinematic modelling of the PLSAs is studied under the constant curvature condition where the actuators or robots are approximated to one or many pieces of constant curvature arc. Figure 4.5 illustrates such kinematic configuration for a single section PLSA. The constant curvature condition has also been generally held in various types of the continuum robots. Regarding this, a comprehensive review is presented in [55]. In their work, a unified constant curvature kinematic setting has been generalised. It considers three spaces, actuator, configuration and task, with the connections by two mappings, robot-independent and robot-dependent. The robot-independent mapping is set between the configu-
ration space by the curvature $\kappa = \frac{1}{r}$, the plane angle $\phi$ and the arc length $l^a$, and the task space by the 3-dimension position, $x$, $y$ and $z$. It can be applied to a wide range of continuum robots. The transformation between them can be calculated via the conventional Denavit-Hartenberg approach which is commonly used in the cases of the rigid manipulators. The robot-dependent mapping bridges the actuator space to the configuration space. It is specified by the actuation mechanism and the designed structure. As the PLSAs configuration shown in Figure 4.5, the actuator space parameters are lengths of the three chambers $l^c_{1,2,3}$. The robot-specific mapping with respect to Figure 4.5 is described as

$$\kappa = \frac{2\sqrt{l^c_1^2 + l^c_2^2 + l^c_3^2 - l^c_1l^c_2 - l^c_2l^c_3 - l^c_1l^c_3}}{d},$$  \hspace{1cm} (4.1)$$

$$\phi = \arctan \left( \frac{\sqrt{3(l^c_2 + l^c_3 - 2l^c_1)}}{3(l^c_3 - l^c_2)} \right),$$  \hspace{1cm} (4.2)$$

$$l^a = \frac{l^c_1 + l^c_2 + l^c_3}{3},$$  \hspace{1cm} (4.3)$$

where $d$ is the average distant between the actuator coaxial centre and each of the three chambers coaxial centres. The overall constant curvature kinematic setting for the single segment three-chamber PLSAs is illustrated in Figure 4.6.
Using the robot-dependent mapping, (4.1), (4.2) and (4.3), this setting is adopted by nearly all of the similar PLSA work [23, 64, 65, 88, 89]. In this way, the DOFs of the bending and the steering do not come from a straightforward measurement, but estimated from the actuator space or from the task space. From the experimental exercises, we found that the bending and the steering estimations in the configuration space might be improved by adding an auxiliary setting which can bring a direct measurement to these two important types of the shape information. The new setting is based on the equivalence between the tip surface orientation and the shape of a single segment PLSA. It will be introduced in the next two subsections.

4.3.2 The auxiliary kinematic setting

The auxiliary setting is based on the tip surface orientation of a PLSA. It is currently implemented by the trackSTAR™ magnetic sensor in our research, but this method can be applied to general gyroscopic sensors. This paper focuses on the one-chamber and the two-chamber scenarios. Three-chamber actuation will lead to kinematic redundancy and variable stuffiness of a PLSA. This is an open and active area of research. The extension to the three-chamber case will be studied together with a ready stiffening mechanism in the future.

Figure 4.7: The illustration for the three configuration subspace 1, 2 and 3. They are sectioned by the three chambers’ individual bending directions.

An overall view of the new auxiliary setting is illustrated in Figure 4.7. The modification takes place at the robot-dependent mapping part; the part of the robot-independent mapping is kept unchanged. The actuator space is replaced by three
configuration subspaces as shown in Figure 4.7. In this setting, the scenario of the actuation by any two certain chambers is uniquely confined to each of the subspaces. The three configuration subspaces are defined on the same polar plane that is for the configuration space; the latter is divided by the former. Each of the three configuration subspaces is a fan-shaped area, and its two radial boundaries are defined where the two chambers’ singly-driven deflections project onto the polar plane. The three sectors are symmetric by the conceptual definition. But due to many practical factors such as mechanical designs or fabrication, they are likely to be different. The three configuration subspaces can be calibrated according to the actual situations. Therefore this reality gap won’t affect the later system identification work.

\[ \theta_d \]

\[ \theta_b \]

\[ Z \]

\[ x \]

Figure 4.8: The bending angle $\theta_b$ is equivalent to the deflection angle $\theta_d$ in the constant curvature condition.

Every individual configuration subspace consists of a deflection angle $\theta_i$ and a steering angle $\phi_{i\text{sub}}$ to represent the deflection and the steering. As the equivalence to the tip surface orientation, the term arc length $l^e$ is not used to estimate the bending and the steering. Compared with the configuration space, the parameters for representing the DOFs of the bending are modified in the configuration subspaces. The deflection angle is directly used. But as shown in Figure 4.8, the deflection angle $\theta_d$ is equal to the bending angle $\theta_b$ under the constant curvature condition. Hence the former can be used to approximate the latter. For the DOF of the steering, the plane angle $\phi$ is divided and separately brought to each of the configuration subspaces as $\phi_{1\text{sub}}$, $\phi_{2\text{sub}}$ and $\phi_{3\text{sub}}$. The only change made to them is the reduced range which now is the intersection angle of two projected beams by the single chamber deflection. Taking the three subspace plane angles $\phi_{1\text{sub}}$, $\phi_{2\text{sub}}$ and $\phi_{3\text{sub}}$ to the configuration space simply needs to add different offsets to each of them. In run time, switching between the subspaces can be implemented under the state machine framework. The problem due to the physically adjoined upper and lower mathematical bounds in the mainstream method [55] is thereby avoided. As to the new setting, Figure 4.9 shows the modified kinematic framework updated from Figure 4.6.
The simple setting in Figure 4.9 is effective to capture the DOFs of the bending and the steering. Each subspace individually links to a deformation scenario played by two certain chambers. In this way, the identification work can be taken separately according to the three subspaces. It is a practical approach when the deformations by every two chamber become greatly unsymmetrical due to different reasons such as designs, fabrication or usage. And it is supported by the current sensor technologies.

On the other hand, the new auxiliary setting can incorporate well with the current elongation measurement to estimate the task space parameters. A PLSA’s length can be accurately measured by the optical fibre sensors as in [23], or be estimated by the polynomially pre-identified pressure-length relation. With an acquired tube length \( l^a \) and the bending and the steering angles \( \theta^b \) and \( \phi \) obtained from this proposed auxiliary setting, one can use the summarised constant curvature approach in [55] to easily estimate the task space parameters \( x, y \) and \( z \).

The calculation for the deflection angle and the steering angle is presented in this subsection. The configuration is based on the trackSTAR™ magnetic sensor which has a sensing base and a sensing tip. The base provides the global reference frame \( G \), and it is fixed at a place during operation. The sensing tip is attached with the local reference frame \( B \), and it is installed perpendicular to the tip surface of a PLSA. The measurement setting is illustrated in Figure ??, which can be adopted by the gyroscopic sensors with minor adjustments. One can refer to Section 3.3 for the detailed configuration and calculation.
4.4 Model Identification Setting

4.4.1 The experimental design

The system identification approach is proposed in the bottom-up fashion. The local data sets are identified at first and they will be combined together to reconstruct the global responses. The motivation comes from the fact that each local data set can be fairly linearised. Using them would lead to a globally accurate dynamical description.

One can refer to Figure 4.13 and 4.14 for the single chamber bending identification test design, and Figure 4.17 and 4.18 for the bending and the steering one. Before the test, a number of pressure levels on the input channel are selected. They are set as the reference points for partitioning the local data sets. For various designs of the PLSAs, densely partitioned local data sets are usually unnecessary, as they react in slow dynamical responses. The bandwidth is low for the PMAs [96], and specifically for the PLSAs, it is normally below 20 Hz.

During the identification test, two local step responses in an increase and decrease pair between each two neighbour reference points are separately arranged. The increase and the decrease mean that in a local interval the step response will be further individually identified regarding the mutually inverted value-changing directions between the two reference points. This can give an effective solution to the hysteresis effect as observed in Figure 4.3.

In the two-chamber bending and steering cases, the system has two inputs and two outputs. The identification test is arranged in a decoupled way, but the main idea is still similar to the single chamber cases. Each time only one of the two input channels is allowed to excite a local step response, meanwhile the responses on the two output channel are being sampled. In this way, the Single-Input-Multiple-Output (SIMO) models can be separately identified with respect to each input channel. They will be used to approximate the complete Multiple-Input-Multiple-Output (MIMO) model in a local interval.

4.4.2 The DIO-PWL-OBF model structure

In the light of the local approach with the designed experimental procedure, a new nonlinear model structure specified for the PLSA identification is going to be developed. It is named as Delta Input Output PieceWise Linear Orthonormal Basis Function (DIO-PWL-OBF) model structure as the indication of the three involved theoretical constituents. Along the line of reasoning, firstly the general modification
on the LPV-OBF model structure will be discussed, and it will then be brought to the context of the PieceWise Linear (PWL) systems; secondly the new ingredient Delta Input Output (DIO) will be introduced, and eventually the specific development by taking the DIO to arrive at the DIO-PWL-OBF form will be presented.

The first OBF-based adaptive model structure is the LPV-OBF one which has been profoundly developed for the LPV system identification in [81]. Based on the OBF-based LTI system setting of (2.2), the LPV-OBF model structure can be expressed as

\[ y_k \approx \sum_{i=1}^{n} (c_i \circ p) B_i(q) u_k \]  

where \( p \) is the scheduling signal with respect to the frozen LTI systems and \( \circ \) is the scheduling operator. In the behavioural perspective [97], the term \((c_i \circ p)\) can be seen as the scheduling part which on-line updates the coefficient \( c_i \) of \( B_i(q) \) according to \( p \). The linearisation type in the LPV identification is based on a floating reference point, whereas our linearisation method relies on a number of fixed equilibrium points. The discrepancy lies in the different conceptual formulation of the scheduling part, but it does not affect the role of the OBF part. The general idea of the coefficient-adaptive OBF structure is preserved. We take (4.4) to the category of the piecewise linear systems which is generally expressed as

\[ y_k \approx G^l(q) u_k, \ l \in N_{Loc}. \]  

Thus the model structure is modified to

\[ y_k \approx \sum_{i=1}^{n} c_i^l B_i(q) u_k, \ l \in N_{Loc}, \]  

which is temporarily named as the PWL-OBF structure.

In the PWL context, the OBF selection objective is redirected for suiting our application: to optimally approximate the partitioned local data sets. The underlying pole selection problem for the related LTI systems remains the same. The OBF coefficients are varied based on the localised regions. The region that the system is currently staying at decides the related OBF coefficients currently being used. There are two issues regarding applying the OBF structure to the PWL setting. They are termed as switching effect and offset effect. The brief discussions about them below are based on the ordinary difference equations (ODE) in the standard form as

\[ \sum_{i=0}^{n_a} a_i q^{-i} y_k = \sum_{j=1}^{n_b} b_j q^{-j} u_k \]  

The switching effect happens during the transition from one local region to another.
If the post local model structure hardly follows the previous one in a consistent manner like in a same order or having similar frequency characters, a relatively large modelling error will be caused. As the fixed dominator structure in each individual OBFs, the synthesised LTI $\sum_{i=1}^{n_c} c_i B_i(q) u_k$ results in a fixed transience part $\sum_{i}^{n_a} a_i q^{-i}$. Due to this, the switching effect should have been small. But because of the different local dynamical characters, the embodied Finite Impulse Response (FIR) parts $\sum_{j=0}^{n_b} b_j q^{-j}$ on the input side commonly result in inconsistent parametrization. This can easily enlarge the modelling error during the transition. Especially, when the transience part has a pole near the unit circle, the convergence rate of the transition modelling error will be relatively slow if the switching action on the FIR part causes a relatively large deviation.

On the other hand, the offset effect is caused by the initial and final conditions in each sampled local data set. In the designed experiment, the system starts and ends at two equilibrium points in each of the partitioned local regions. This will lead to two steady-state simultaneous equations in the linear regression process as

$$\begin{align*}
\sum_{i=0}^{n_a} a_i y_{ss1} &= \sum_{j=1}^{n_b} b_j u_{ss1} \\
\sum_{i=0}^{n_a} a_i y_{ss2} &= \sum_{j=1}^{n_b} b_j u_{ss2}
\end{align*}$$

(4.8)

where $u_{ss1}$ and $y_{ss1}$ are the input and output at the initial condition; $u_{ss2}$ and $y_{ss2}$ are at the final condition. The global static nonlinearity has been observed in Figure 4.3. As a result, the inequality relation by

$$\frac{y_{ss1}}{y_{ss2}} \neq \frac{u_{ss1}}{u_{ss2}}$$

(4.9)

takes over the equality relation, where the latter is only held in a globally linear sense. This fact will let the coefficient sets $\{a_i\}$ and $\{b_j\}$ extend to a relatively long size to fulfil the condition made by (4.8). Consequently the linearised model results in a relatively high-order model structure.

In order to overcome the two issues discussed in the temporary PWL-OBF model structure, a specific but simple development is further conducted. Firstly, the offsets on both input and output channels are removed so that (4.8) can be combined to one equation as

$$\sum_{i=0}^{n_a} a_i (y_{ss2} - y_{ss1}) = \sum_{j=0}^{n_b} b_j (u_{ss2} - u_{ss1}).$$

(4.10)

In this way, the problem caused by the global static nonlinearity are eliminated. The local data sets can be accurately identified by a relatively low-order configuration. But after this step, the model only works in the local sense. This means such a
model cannot be directly applied to the PWL setting as in (4.5). In parallel, our consideration reverts to the superposition principle and the convolution sum which are the foundations to evaluate the dynamics of an ODE (4.7).

Based on the input and output time series, \( \{u_k\} \) and \( \{y_k\} \), the Delta Input and Output (DIO) time series are defined as

\[
\delta y_k \triangleq y_k - y_{k-1}, \quad \delta y_1 \triangleq y_1, \\
\delta u_k \triangleq u_k - u_{k-1}, \quad \delta u_1 \triangleq u_1. 
\] (4.11)

It can be seen that \( \{\delta u_k\} \) and \( \{\delta y_k\} \) represent the immediate change between every two neighbour sampling points in \( \{u_k\} \) and \( \{y_k\} \). The relation between the former and the latter is

\[
u_k = \sum_{n=1}^{k} \delta u_n, \quad y_k = \sum_{m=1}^{k} \delta y_m. \tag{4.12}
\]

By substituting (4.12) into the standard ODE form (4.7), it can directly achieve that

\[
\sum_{i=0}^{n_a} a_i q^{-i} \sum_{m=1}^{k} \delta y_m = \sum_{j=1}^{n_b} b_i q^{-j} \sum_{n=1}^{k} \delta u_n. \tag{4.13}
\]

It is the ODE in the DIO sense. Simplify the synthesised operators \( q^{-i} \sum_{m=1}^{k} \delta y_m \) and \( q^{-j} \sum_{n=1}^{k} \delta u_n \) on both sides of (4.13) to \( \sum_{m=1}^{k-i} \delta y_m \) and \( \sum_{n=1}^{k-j} \delta u_n \) respectively, and expand them. (4.13) becomes

\[
a_0 \sum_{1}^{k} \delta y_m + \ldots + a_1 \sum_{1}^{k-i} \delta y_m = b_1 \sum_{1}^{k-1} \delta u_n + \ldots + b_j \sum_{1}^{k-j} \delta u_n, \tag{4.14}
\]

which can be further decomposed into a set of simultaneous equations as

\[
\begin{align*}
a_0 \delta y_k + a_1 \delta y_{k-1} + \ldots + a_i \delta y_{k-i} &= b_1 \delta u_{k-1} + \ldots + b_j \delta u_{k-j} \\
& \vdots \\
a_0 \delta y_{i+1} + a_1 \delta y_i + \ldots + a_i \delta y_1 &= b_1 \delta u_j + \ldots + b_j \delta u_1 \\
& \vdots \\
a_0 \delta y_2 + a_1 \delta y_1 + \ldots + 0 &= b_1 \delta u_1 + \ldots + 0 
\end{align*} \tag{4.15}
\]

Compared with the orthodox expression (4.7), the derived (4.15) could give us a different insight into the dynamics of a system. Instead of \( \{u_k\} \) and \( \{y_k\} \), the input and output channels are now viewed in the definition of \( \{\delta u_k\} \) and \( \{\delta y_k\} \). The entire dynamics of a system can be interoperated as the composition of all the individual DIO impulse responses by per sampling point forward along the time line. The superposition principle and the convolution sum are the unaltered foundations for the evaluation of the dynamic responses. However based on the new perspective induced by (4.15), the two calculation rules can be applied in a separate and independent
way. That is, the dynamic responses of each $\delta u_k$ can be individually evaluated. In the LTI systems, this feature is redundant, since it makes no difference as putting $\delta u_k$ and $\delta y_k$ back into $u_k$ and $y_k$ by (4.12) and then doing the same evaluation according to (4.7). But in the nonlinear systems, this feature is very appreciated because it gives an extra degree of freedom to manipulate the dynamics.

Inspired by the reasoning above, the outcome (4.15) is going to be taken back to the temporary PWL-OBF model structure (4.5). In (4.15), preserve the equation in the first line for the latest sample $\delta u_k$ and $\delta y_k$, and use (4.7) to replace the rest of the equations for the time series $\{\delta u_k-1\}$ and $\{\delta y_k-1\}$, a reduced expression can obtained as

$$
\begin{align*}
\left\{ a_0\delta y_k + a_1\delta y_{k-1} + \ldots + a_i\delta y_{k-i} = b_1\delta u_{k-1} + \ldots + b_j\delta u_{k-j} \\
a_0y_{k-1} + a_1y_{k-2} + \ldots + a_iy_{(k-1)-i} = b_1u_{k-1} + \ldots + b_ju_{(k-1)-j}
\right.,
\end{align*}
$$

(4.16)

which is equivalently in the transfer-function form

$$
y_k = G(q)q^{-1}u_k + G(q)\delta u_k
$$

(4.17)

in the term of the LTI transfer function $G(q)$. Moreover, (4.17) can be more generally adopted to the nonlinear systems by simply replacing $G(q)q^{-1}u_k$ with $y_{k-1}$, which then becomes

$$
y_k = y_{k-1} + G(q)\delta u_k.
$$

(4.18)

In run time, the one-step-before output sample $y_{k-1}$ can be directly acquired from the feedback or be estimated by iterations from the previous input samples $\{u_{k-1}\}$. The derived (4.18) is particularly suitable for integrating the PWL model structure (4.5) with our designed local identification approach as

$$
y_k \approx y_{k-1} + G^l(q)\delta u_k, \ l \in N_{Loc},
$$

(4.19)

because the offset issue of the locally identified models $G^l(q)$ is solved. Each identified local model $G^l(q)$ can be used in (4.19) without any modification in evaluating a global dynamic response. Besides, (4.19) also indicates an alternative approach to handle the hysteresis effect. Between two neighbour equilibrium points, the local models are separately identified based on the increase and decrease directions. Compared with the HW and the nonlinear ARX approaches, this manner would be more advantageous for the PLSA identification, because all the local properties of the dynamics are preserved.

Finally the OBF ingredient is added to (4.19). It now becomes

$$
y_k \approx y_{k-1} + \sum_{i=1}^{n} c_iB_i(q)\delta u_k, \ l \in N_{Loc},
$$

(4.20)
which is illustrated in Figure 4.10. By adding the OBF part, the switching can be smooth and the side effect can be diminished significantly. Unlike in the PWL-OBF model where the switching acts on \{\{u_k\} and \{y_k\}, here it only manipulates the transience of the latest sample of \(\delta u_k\) and \(\delta y_k\). Only the related portions of \(\delta u_k\) and \(\delta y_k\) in \(u_k\) and \(y_k\) are processed by the adaptive mechanism at a sample point. The rest are not affected. The \(n\)-step-ahead predictor based on (4.20) is formulated as

\[
\hat{y}(k + n) = y(k) + \sum_{m=k+1}^{k+n} \sum_{i=1}^{n_l} c_i^l B_i(q) \delta u(m), l \in N_{\text{Loc}}, \tag{4.21}
\]

Similarly, for the two-input-two-output cases, (4.20) is extended to

\[
y_{1,k} \approx y_{1,k-1} + \sum_{i=1}^{n_{11}} c_{11,i}^l B_{11,i}(q) \delta u_{1,k} + \sum_{i=1}^{n_{12}} c_{12,i}^l B_{12,i}(q) \delta u_{2,k}, \quad l \in N_{\text{Loc}}, \tag{4.22}
\]

\[
y_{2,k} \approx y_{2,k-1} + \sum_{i=1}^{n_{21}} c_{21,i}^l B_{21,i}(q) \delta u_{1,k} + \sum_{i=1}^{n_{22}} c_{22,i}^l B_{22,i}(q) \delta u_{2,k}, \quad l \in N_{\text{Loc}}.
\]

Figure 4.10: Illustration of the DIO-PWL-OBF model structure.

Table 4.1: The unified identification procedure.

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Setup the platform. The measurement and the calculation for the bending angle and the steering are based on the auxiliary kinematic setting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Select a number of reference pressures on each chamber.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Take identification experiment according to the local partitions.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Extract the local data sets. Deoffset and linearise them.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Apply the FKcM pole clustering algorithm for the locally identified poles.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Construct the OBF set based on the optimally selected poles.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Arrange the constructed OBFs in the DIO-PWL-OBF setting. Referred to Subsection</td>
</tr>
</tbody>
</table>
The development of the new DIO-PWL-OBF model structure (4.20) and (4.22) has been presented in detail. It is capable of handling the nonlinearities discussed in Section 4.2 and suitable for the identification test designed in Section 4.4.1. With the use of the auxiliary kinematic setting in Section 4.3, the unified identification procedure for the class of the PLSAs are concluded and presented in Table 4.1.

4.5 Implementation and Results

4.5.1 Platform setup

The schematic drawing in Figure 4.11 illustrates the implementation setup of Figure 4.12. A desktop using Matlab Real-Time Windows Target™ serves as the central platform to interface the other parts. The system works at a sample rate of 80 Hz, and it is constantly supplied with stable 1 bar compressed air. Every two two-port solenoid valves together with one pressure sensor operate as one unit, which is used to regulate one chamber’s pressure. A test PLSA actuator used in the following identification work is made in the silicon rubber and it is 9 cm in length and 3 cm in diameter. The operating pressure of a single chamber is below 0.4 bar. The constant curvature condition is held satisfactorily for the test PLSA. The deflection angle is accurately approximated to the bending angle. Thus the elongation measurement is neglected.

Figure 4.11: Schematic drawing for the platform setup. The platform was revised from Figure 3.1.

4.5.2 Single chamber identification

The single chamber identification follows the instruction summarised in Section 4.1. The chamber 1 of the PLSA is used. As shown in Figure 4.13, the local data sets are
partitioned based on different levels of pressures. The partition interval is chosen to be 0.05 bar.

Figure 4.14 demonstrates the experimental data acquired according to the partition setting in Figure 4.13. The entire data can be seen as a collection of individual step responses acting at different pressure levels. By deoffsetting the initial condition on both the input and output channels, these step responses are brought to the local sense.

The standard Newton-Guess method is employed to take a linear analysis on the extracted local data sets. The Normalised Root Mean Squire (NRMSE) is used as the measurement of the goodness of the fit. The Mean Square Error (MSE) is also used, and particularly the Akaikes Final Prediction Error (FPE) is employed to assess the qualities of the identified LTI models. Table 4.2 highlights the results such as the pole zero structure and the linear approximation. It can be seen that each local data set can be well linearised with an over 90% NRMSE fit, except for
those below 0.10 bar pressure level. In the latter cases, the approximation error was due to that the very small bending angle driven by a low pressure was interfered by the sensor noise. $loc_{icr}^1$ and $loc_{dcr}^1$ are neglected as their bending angle ranges are below $0.5^\circ$. The linear approximation for $loc_{icr}^2$ and $loc_{dcr}^2$ are acceptable because their FPE and MSE are both relatively small. Table 4.2 also reveals that the model structures from each linearised data sets are not kept in a consistent way. Compared with the low pressure levels, the high pressure levels always result in a higher-order model structure. Overall the linearised models in the pressure increase cases have a higher order than their counterparts in the pressure decrease cases. Table 4.3 lists

<table>
<thead>
<tr>
<th>Local data set</th>
<th>Pole Num.</th>
<th>Zero Num.</th>
<th>NRMSE</th>
<th>FPE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$loc_{icr}^1$</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$loc_{icr}^2$</td>
<td>2</td>
<td>1</td>
<td>78.98%</td>
<td>0.0037</td>
<td>0.0037</td>
</tr>
<tr>
<td>$loc_{icr}^3$</td>
<td>2</td>
<td>1</td>
<td>90.00%</td>
<td>0.0095</td>
<td>0.0094</td>
</tr>
<tr>
<td>$loc_{icr}^4$</td>
<td>2</td>
<td>1</td>
<td>91.70%</td>
<td>0.0647</td>
<td>0.0642</td>
</tr>
<tr>
<td>$loc_{icr}^5$</td>
<td>3</td>
<td>2</td>
<td>96.42%</td>
<td>0.0883</td>
<td>0.0869</td>
</tr>
<tr>
<td>$loc_{icr}^6$</td>
<td>3</td>
<td>2</td>
<td>95.90%</td>
<td>1.0341</td>
<td>1.0220</td>
</tr>
<tr>
<td>$loc_{icr}^7$</td>
<td>3</td>
<td>2</td>
<td>96.68%</td>
<td>0.2653</td>
<td>0.2626</td>
</tr>
<tr>
<td>$loc_{dcr}^1$</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$loc_{dcr}^2$</td>
<td>2</td>
<td>1</td>
<td>63.21%</td>
<td>0.0069</td>
<td>0.0069</td>
</tr>
<tr>
<td>$loc_{dcr}^3$</td>
<td>2</td>
<td>1</td>
<td>89.37%</td>
<td>0.0433</td>
<td>0.0429</td>
</tr>
<tr>
<td>$loc_{dcr}^4$</td>
<td>1</td>
<td>0</td>
<td>94.38%</td>
<td>0.5167</td>
<td>0.5135</td>
</tr>
<tr>
<td>$loc_{dcr}^5$</td>
<td>1</td>
<td>0</td>
<td>96.06%</td>
<td>0.6418</td>
<td>0.6378</td>
</tr>
<tr>
<td>$loc_{dcr}^6$</td>
<td>1</td>
<td>0</td>
<td>95.67%</td>
<td>0.2727</td>
<td>0.2710</td>
</tr>
<tr>
<td>$loc_{dcr}^7$</td>
<td>1</td>
<td>0</td>
<td>92.62%</td>
<td>0.0547</td>
<td>0.0548</td>
</tr>
</tbody>
</table>
the identified poles for each local data sets. The results of Table 4.2 and 4.3 confirm
the nonlinearities observed early in Figure 4.3 and 4.4.

Table 4.3: Poles of the local data sets.

<table>
<thead>
<tr>
<th>Local data set</th>
<th>Identified poles</th>
<th>Local data set</th>
<th>Identified poles</th>
</tr>
</thead>
<tbody>
<tr>
<td>loc\text{icr}^2</td>
<td>0.9288±0.0828i</td>
<td>loc\text{dcr}^2</td>
<td>0.9689±0.0812i</td>
</tr>
<tr>
<td>loc\text{icr}^3</td>
<td>0.8878±0.0775i</td>
<td>loc\text{dcr}^3</td>
<td>0.9786±0.1072i</td>
</tr>
<tr>
<td>loc\text{icr}^4</td>
<td>0.8717±0.0154i</td>
<td>loc\text{dcr}^4</td>
<td>0.8930</td>
</tr>
<tr>
<td>loc\text{icr}^5</td>
<td>0.9883, 0.9125±0.0716i</td>
<td>loc\text{dcr}^5</td>
<td>0.9699</td>
</tr>
<tr>
<td>loc\text{icr}^6</td>
<td>0.9908, 0.9179±0.0363i</td>
<td>loc\text{dcr}^6</td>
<td>0.9690</td>
</tr>
<tr>
<td>loc\text{icr}^7</td>
<td>0.9774, 0.9174±0.0793i</td>
<td>loc\text{dcr}^7</td>
<td>0.9665</td>
</tr>
</tbody>
</table>

The FKcM clustering is taken on the identified poles. By setting the fuzziness
\(m=3\), the optimal pole set is calculated as presented in Table 4.4.

Table 4.4: The unique pole set calculated by FKcM clustering

| 0.9910 | 0.9783±0.1068i | 0.9700 | 0.8930 | 0.8770 |

To further improve the approximation accuracy, the real poles are repeated once.
Table 4.5 lists the pole structure for the OBF construction.

Table 4.5: The pole set \(\{\xi_1 \ldots \xi_{10}\}\) selected for OBFs construction

<table>
<thead>
<tr>
<th>(\xi_{1,2})</th>
<th>(\xi_{3,7})</th>
<th>(\xi_{4,8})</th>
<th>(\xi_{5,9})</th>
<th>(\xi_{6,10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9783±0.1068i</td>
<td>0.8770</td>
<td>0.8930</td>
<td>0.9700</td>
<td>0.9910</td>
</tr>
</tbody>
</table>

The contour plot in Figure 4.15 shows the decay rates of each identified poles based
on the selected pole structure in Table 4.5. The overall decay rates are very low.
Most of the identified poles are below the decay rate of 0.04, while a few are about
0.9. The result means the OBF set using the selected pole structure is sufficient to
approximate all the local data sets.

Table 4.6 shows the fitting result by taking the OBFs to the original local data
sets. Its closeness to Table 4.2 shows that the constructed OBFs set is justified for
approximating all the local data sets.

Finally, the test in the global sense is carried out. It is a full-range step response.
The test and its validation results are demonstrated in Figure 4.16. The numerical
fitting result is presented in Table 4.7. Especially in Figure 4.16, the switching
state is plotted in parallel with the input and output channels. The 14 states are
numbered in the order: \(\text{loc}_{1}^{\text{icr}}, \ldots, \text{loc}_{7}^{\text{icr}}, \text{loc}_{7}^{\text{dcr}}, \ldots, \text{loc}_{1}^{\text{dcr}}\). Note that \(\text{loc}_{1}^{\text{icr}}\) and \(\text{loc}_{1}^{\text{dcr}}\)
are virtually included, but their OBF coefficients are all set to zeros.
Figure 4.15: The decay rate of the identified pole based on the selected pole structure in Table 4.5.

Table 4.6: The OBF approximation for the original local data sets

<table>
<thead>
<tr>
<th>Local data set</th>
<th>NRMSE</th>
<th>MSE</th>
<th>Local data set</th>
<th>NRMSE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>loc_{4}^{cr}</td>
<td>81.04%</td>
<td>0.0030</td>
<td>loc_{2}^{dcr}</td>
<td>52.25%</td>
<td>0.0116</td>
</tr>
<tr>
<td>loc_{3}^{cr}</td>
<td>91.38%</td>
<td>0.0070</td>
<td>loc_{3}^{dcr}</td>
<td>91.92%</td>
<td>0.0248</td>
</tr>
<tr>
<td>loc_{4}^{cr}</td>
<td>93.95%</td>
<td>0.0340</td>
<td>loc_{4}^{dcr}</td>
<td>94.99%</td>
<td>0.4075</td>
</tr>
<tr>
<td>loc_{5}^{cr}</td>
<td>96.76%</td>
<td>0.0712</td>
<td>loc_{5}^{dcr}</td>
<td>96.58%</td>
<td>0.4810</td>
</tr>
<tr>
<td>loc_{6}^{cr}</td>
<td>93.95%</td>
<td>0.4498</td>
<td>loc_{6}^{dcr}</td>
<td>96.13%</td>
<td>0.2167</td>
</tr>
<tr>
<td>loc_{7}^{cr}</td>
<td>97.28%</td>
<td>0.2355</td>
<td>loc_{7}^{dcr}</td>
<td>93.54%</td>
<td>0.0416</td>
</tr>
</tbody>
</table>

The transience of the global step response lasts about 1.5 second. Hence the 40- and 80-step-ahead predications which respectively allows a 0.5 and 1 second forerunner are proper settings for validation or other applications. Both the predication and the simulation results show over 90% NRMSE fits. Particularly the goodness measures for the individual transience parts are given. It can be seen that the proposed identification approach is applicable in the single chamber cases.
Figure 4.16: The global test and validation for the single chamber identification.

Table 4.7: The fitting results for the test in Figure 4.16.

<table>
<thead>
<tr>
<th>Results</th>
<th>NRMSE</th>
<th></th>
<th></th>
<th>MSE</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Full</td>
<td>Icr.</td>
<td>Dcr.</td>
<td>Full</td>
<td>Icr.</td>
</tr>
<tr>
<td>40-step-ahead</td>
<td>97.90%</td>
<td>96.94%</td>
<td>97.17%</td>
<td>1.6585</td>
<td>2.7656</td>
<td>2.2352</td>
</tr>
<tr>
<td>80-step-ahead</td>
<td>96.56%</td>
<td>95.17%</td>
<td>95.21%</td>
<td>4.4438</td>
<td>6.8719</td>
<td>6.4195</td>
</tr>
<tr>
<td>Simulation</td>
<td>95.28%</td>
<td>94.34%</td>
<td>93.09%</td>
<td>8.3832</td>
<td>9.4654</td>
<td>13.3304</td>
</tr>
</tbody>
</table>
4.5.3 Double chamber identification

The double chamber steering and bending identification is conducted by following the instructions in Section 4.1. Chamber 1 and Chamber 2 of the PLSA are used. As illustrated in Figure 4.17, the local data sets are partitioned based on different pressure levels of both the chambers. The pressure range for each chamber is set between 0.20 bar and 0.35 bar. This is because the bending angle is relatively small when the pressure is below 0.20 bar, and this makes the steering motion difficult to measure by our current magnetic sensor.

Figure 4.18 shows the experimental data obtained for the identification. It is based on the partition in Figure 4.17. Similar to the single chamber situation, each local step response depends on the previous steady state. Each local data set takes the step response driven only by one chamber, while the other chamber’s pressure is kept at its initial value throughout the time range. It is a decoupled approach to identify the SIMO relation between one chamber’s pressure and the resultant two DOF movements.

After removing the offsets on both the input and output channels, linear analysis is taken individually. Similar to the single chamber situation, the local sets can be accurately linearised. However, in the double chamber cases, the overall orders of the identified models become lower. The 1-pole-0-zero structure can be properly used for every decoupled SISO dynamical relation. The reason is partly because the stiffness of the PLSA is higher in the double chamber situation than in the single one. The increased internal pressure acts on stabilising the PLSA’s deformation. The linearisation result is shown in Table 4.8.
Figure 4.17: The local partition for the double chamber identification.

Figure 4.18: The experimental data for the double chamber identification.
Table 4.8: The linearisation result for the local data sets in the double chamber case.

<table>
<thead>
<tr>
<th>Local data set</th>
<th>NRMSE $y_1$ (%)</th>
<th>NRMSE $y_2$ (%)</th>
<th>FDE</th>
<th>MSE</th>
<th>Local data set</th>
<th>NRMSE $y_1$ (%)</th>
<th>NRMSE $y_2$ (%)</th>
<th>FDE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>locchr1,1,1</td>
<td>94.68</td>
<td>91.42</td>
<td>.0005</td>
<td>.0336</td>
<td>locchr2,1,1</td>
<td>91.03</td>
<td>93.81</td>
<td>.0167</td>
<td>.1519</td>
</tr>
<tr>
<td>locchr1,1,2</td>
<td>95.65</td>
<td>93.36</td>
<td>.0037</td>
<td>.1346</td>
<td>locchr1,2,1</td>
<td>94.92</td>
<td>95.39</td>
<td>.1284</td>
<td>.4667</td>
</tr>
<tr>
<td>locchr1,1,3</td>
<td>94.73</td>
<td>84.07</td>
<td>.0360</td>
<td>.2061</td>
<td>locchr1,3,1</td>
<td>96.06</td>
<td>94.55</td>
<td>.0143</td>
<td>.2046</td>
</tr>
<tr>
<td>locchr1,1,3</td>
<td>93.47</td>
<td>91.07</td>
<td>.0002</td>
<td>.0154</td>
<td>locchr1,3,2</td>
<td>96.99</td>
<td>85.08</td>
<td>.0005</td>
<td>.0036</td>
</tr>
<tr>
<td>locchr1,1,2</td>
<td>97.25</td>
<td>82.60</td>
<td>.0013</td>
<td>.0413</td>
<td>locchr1,2,2</td>
<td>98.11</td>
<td>92.26</td>
<td>.0002</td>
<td>.0207</td>
</tr>
<tr>
<td>locchr1,1,1</td>
<td>97.90</td>
<td>94.76</td>
<td>.0023</td>
<td>.0658</td>
<td>locchr1,1,1</td>
<td>98.21</td>
<td>94.61</td>
<td>.0028</td>
<td>.0599</td>
</tr>
<tr>
<td>locchr1,2,1</td>
<td>91.76</td>
<td>92.23</td>
<td>.0010</td>
<td>.0672</td>
<td>locchr1,2,2</td>
<td>91.81</td>
<td>89.86</td>
<td>.0019</td>
<td>.0795</td>
</tr>
<tr>
<td>locchr1,2,2</td>
<td>95.66</td>
<td>94.31</td>
<td>.0074</td>
<td>.1294</td>
<td>locchr1,2,2</td>
<td>95.14</td>
<td>95.04</td>
<td>.0092</td>
<td>.1549</td>
</tr>
<tr>
<td>locchr1,2,3</td>
<td>96.20</td>
<td>59.15</td>
<td>.0047</td>
<td>.0082</td>
<td>locchr1,2,3</td>
<td>95.34</td>
<td>92.98</td>
<td>.0092</td>
<td>.1471</td>
</tr>
<tr>
<td>locchr1,2,3</td>
<td>94.40</td>
<td>62.04</td>
<td>.0002</td>
<td>.0167</td>
<td>locchr1,2,3</td>
<td>96.91</td>
<td>91.42</td>
<td>.0000</td>
<td>.0037</td>
</tr>
<tr>
<td>locchr1,2,2</td>
<td>97.34</td>
<td>84.05</td>
<td>.0000</td>
<td>.0338</td>
<td>locchr1,2,2</td>
<td>97.81</td>
<td>92.95</td>
<td>.0013</td>
<td>.0370</td>
</tr>
<tr>
<td>locchr1,2,1</td>
<td>97.38</td>
<td>94.60</td>
<td>.0136</td>
<td>.1164</td>
<td>locchr1,2,1</td>
<td>97.62</td>
<td>93.34</td>
<td>.0265</td>
<td>.1782</td>
</tr>
<tr>
<td>locchr1,3,1</td>
<td>89.36</td>
<td>94.32</td>
<td>.0006</td>
<td>.0311</td>
<td>locchr1,3,1</td>
<td>91.31</td>
<td>93.22</td>
<td>.0000</td>
<td>.0144</td>
</tr>
<tr>
<td>locchr1,3,2</td>
<td>92.43</td>
<td>91.63</td>
<td>.0198</td>
<td>.1650</td>
<td>locchr1,3,2</td>
<td>94.23</td>
<td>95.19</td>
<td>.0002</td>
<td>.0145</td>
</tr>
<tr>
<td>locchr1,3,3</td>
<td>96.68</td>
<td>95.77</td>
<td>.0122</td>
<td>.1147</td>
<td>locchr1,3,3</td>
<td>95.69</td>
<td>97.20</td>
<td>.0037</td>
<td>.0613</td>
</tr>
<tr>
<td>locchr1,3,3</td>
<td>96.59</td>
<td>90.53</td>
<td>.0000</td>
<td>.0084</td>
<td>locchr1,3,3</td>
<td>96.92</td>
<td>91.88</td>
<td>.0000</td>
<td>.0031</td>
</tr>
<tr>
<td>locchr1,3,2</td>
<td>96.76</td>
<td>95.61</td>
<td>.0045</td>
<td>.0681</td>
<td>locchr1,3,2</td>
<td>94.88</td>
<td>96.35</td>
<td>.0034</td>
<td>.0581</td>
</tr>
<tr>
<td>locchr1,3,1</td>
<td>79.08</td>
<td>96.27</td>
<td>.0888</td>
<td>.3466</td>
<td>locchr1,3,1</td>
<td>63.93</td>
<td>94.41</td>
<td>.0161</td>
<td>.1271</td>
</tr>
<tr>
<td>locchr1,4,1</td>
<td>88.28</td>
<td>89.70</td>
<td>.0000</td>
<td>.0059</td>
<td>locchr1,4,1</td>
<td>80.70</td>
<td>82.11</td>
<td>.0000</td>
<td>.0072</td>
</tr>
<tr>
<td>locchr1,4,2</td>
<td>88.88</td>
<td>95.67</td>
<td>.0000</td>
<td>.0068</td>
<td>locchr1,4,2</td>
<td>82.33</td>
<td>89.62</td>
<td>.0002</td>
<td>.0078</td>
</tr>
<tr>
<td>locchr1,4,3</td>
<td>83.93</td>
<td>94.70</td>
<td>.0004</td>
<td>.0168</td>
<td>locchr1,4,3</td>
<td>74.51</td>
<td>90.79</td>
<td>.0006</td>
<td>.0150</td>
</tr>
<tr>
<td>locchr1,4,3</td>
<td>71.21</td>
<td>95.39</td>
<td>.0001</td>
<td>.0024</td>
<td>locchr1,4,3</td>
<td>61.94</td>
<td>88.39</td>
<td>.0000</td>
<td>.0148</td>
</tr>
<tr>
<td>locchr1,4,2</td>
<td>86.42</td>
<td>96.33</td>
<td>.0000</td>
<td>.0100</td>
<td>locchr1,4,2</td>
<td>86.67</td>
<td>90.10</td>
<td>.0002</td>
<td>.0148</td>
</tr>
<tr>
<td>locchr1,4,1</td>
<td>90.93</td>
<td>96.76</td>
<td>.0000</td>
<td>.0033</td>
<td>locchr1,4,1</td>
<td>83.31</td>
<td>83.80</td>
<td>.0000</td>
<td>.0080</td>
</tr>
</tbody>
</table>
The FKcM clustering is taken for the identified poles of each decoupled SISO pair. The calculated poles are listed in Table 4.9, Without repetition, they are directly used for the OBF construction.

Table 4.9: The pole set \( \{\xi_{i1}^{ij}, \ldots, \xi_{i5}^{ij}\}_{i=1,2,j=1,2} \) for the OBFs construction.

<table>
<thead>
<tr>
<th>Decoupled IO pair ((u_i, y_j))</th>
<th>Pole structure ( {\xi_{i1}^{ij}, \ldots, \xi_{i5}^{ij}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_1 \rightarrow y_1)</td>
<td>0.981, 0.969, 0.964, 0.957, 0.945</td>
</tr>
<tr>
<td>(u_2 \rightarrow y_1)</td>
<td>0.986, 0.975, 0.960, 0.950, 0.942</td>
</tr>
<tr>
<td>(u_1 \rightarrow y_2)</td>
<td>0.998, 0.979, 0.949, 0.935, 0.884</td>
</tr>
<tr>
<td>(u_2 \rightarrow y_2)</td>
<td>0.983, 0.973, 0.961, 0.946, 0.939</td>
</tr>
</tbody>
</table>

Figure 4.19 shows the decay rates of the identified poles in each decoupled SISO system. All the pole are lower than the level of 0.02, which indicates the selected poles are very suitable for the OBF construction.

Figure 4.19: The decay rate of the identified pole based on the selected pole structure in Table 4.9
Table 4.10 shows the OBF approximation result for the decoupled local data sets. Its closeness to Table 4.8 shows that the constructed OBF set is justified for all the local data sets.

### Table 4.10: The OBF approximation for the original local data sets

<table>
<thead>
<tr>
<th>Local data set</th>
<th>NRMSE $y_1$ (%)</th>
<th>NRMSE $y_1$ (%)</th>
<th>MSE</th>
<th>Local data set</th>
<th>NRMSE $y_1$ (%)</th>
<th>NRMSE $y_1$ (%)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$loc_{ch1,1.1}$</td>
<td>91.97</td>
<td>89.95</td>
<td>.0726</td>
<td>$loc_{ch2,1.1}$</td>
<td>90.11</td>
<td>90.36</td>
<td>.2278</td>
</tr>
<tr>
<td>$loc_{ch1,1.2}$</td>
<td>93.37</td>
<td>90.91</td>
<td>.3089</td>
<td>$loc_{ch2,1.2}$</td>
<td>92.58</td>
<td>92.11</td>
<td>.7625</td>
</tr>
<tr>
<td>$loc_{ch1,1.3}$</td>
<td>94.77</td>
<td>80.03</td>
<td>.2391</td>
<td>$loc_{ch2,1.3}$</td>
<td>94.02</td>
<td>91.46</td>
<td>.4758</td>
</tr>
<tr>
<td>$loc_{ch1,1.4}$</td>
<td>93.48</td>
<td>91.07</td>
<td>.0154</td>
<td>$loc_{ch2,1.4}$</td>
<td>93.33</td>
<td>85.02</td>
<td>.0141</td>
</tr>
<tr>
<td>$loc_{ch1,2.1}$</td>
<td>96.22</td>
<td>66.82</td>
<td>.2494</td>
<td>$loc_{ch2,1.2}$</td>
<td>96.66</td>
<td>90.30</td>
<td>.0595</td>
</tr>
<tr>
<td>$loc_{ch1,2.2}$</td>
<td>96.06</td>
<td>94.46</td>
<td>.2063</td>
<td>$loc_{ch2,1.1}$</td>
<td>92.03</td>
<td>92.17</td>
<td>.2276</td>
</tr>
<tr>
<td>$loc_{ch1,2.3}$</td>
<td>93.51</td>
<td>94.01</td>
<td>.0710</td>
<td>$loc_{ch2,2.1}$</td>
<td>94.06</td>
<td>91.45</td>
<td>.2365</td>
</tr>
<tr>
<td>$loc_{ch1,2.4}$</td>
<td>93.37</td>
<td>82.18</td>
<td>.0191</td>
<td>$loc_{ch2,2.2}$</td>
<td>95.56</td>
<td>90.75</td>
<td>.0072</td>
</tr>
<tr>
<td>$loc_{ch1,2.5}$</td>
<td>97.18</td>
<td>89.11</td>
<td>.0549</td>
<td>$loc_{ch2,2.2}$</td>
<td>96.69</td>
<td>91.06</td>
<td>.0763</td>
</tr>
<tr>
<td>$loc_{ch1,2.6}$</td>
<td>97.33</td>
<td>94.55</td>
<td>.1203</td>
<td>$loc_{ch2,2.1}$</td>
<td>95.98</td>
<td>92.95</td>
<td>.2902</td>
</tr>
<tr>
<td>$loc_{ch1,3.1}$</td>
<td>91.24</td>
<td>90.71</td>
<td>.0340</td>
<td>$loc_{ch2,3.1}$</td>
<td>91.06</td>
<td>91.61</td>
<td>.0176</td>
</tr>
<tr>
<td>$loc_{ch1,3.2}$</td>
<td>89.61</td>
<td>88.78</td>
<td>.3004</td>
<td>$loc_{ch2,3.2}$</td>
<td>90.08</td>
<td>91.80</td>
<td>.0427</td>
</tr>
<tr>
<td>$loc_{ch1,3.3}$</td>
<td>93.78</td>
<td>89.57</td>
<td>.5935</td>
<td>$loc_{ch2,3.3}$</td>
<td>93.75</td>
<td>95.11</td>
<td>.1630</td>
</tr>
<tr>
<td>$loc_{ch1,3.4}$</td>
<td>93.44</td>
<td>88.27</td>
<td>.0220</td>
<td>$loc_{ch2,3.3}$</td>
<td>95.62</td>
<td>91.95</td>
<td>.0048</td>
</tr>
<tr>
<td>$loc_{ch1,3.5}$</td>
<td>96.23</td>
<td>94.05</td>
<td>.1116</td>
<td>$loc_{ch2,3.2}$</td>
<td>94.16</td>
<td>95.21</td>
<td>.0874</td>
</tr>
<tr>
<td>$loc_{ch1,3.6}$</td>
<td>94.54</td>
<td>94.09</td>
<td>.4476</td>
<td>$loc_{ch2,3.1}$</td>
<td>63.86</td>
<td>93.00</td>
<td>.4423</td>
</tr>
<tr>
<td>$loc_{ch1,4.1}$</td>
<td>86.49</td>
<td>89.83</td>
<td>.0060</td>
<td>$loc_{ch2,4.1}$</td>
<td>76.75</td>
<td>81.83</td>
<td>.0080</td>
</tr>
<tr>
<td>$loc_{ch1,4.2}$</td>
<td>83.65</td>
<td>91.00</td>
<td>.0283</td>
<td>$loc_{ch2,4.2}$</td>
<td>80.95</td>
<td>87.86</td>
<td>.0105</td>
</tr>
<tr>
<td>$loc_{ch1,4.3}$</td>
<td>88.75</td>
<td>91.39</td>
<td>.0438</td>
<td>$loc_{ch2,4.3}$</td>
<td>69.73</td>
<td>88.47</td>
<td>.0234</td>
</tr>
<tr>
<td>$loc_{ch1,4.4}$</td>
<td>80.52</td>
<td>93.52</td>
<td>.0047</td>
<td>$loc_{ch2,4.4}$</td>
<td>82.92</td>
<td>87.60</td>
<td>.0058</td>
</tr>
<tr>
<td>$loc_{ch1,4.5}$</td>
<td>86.67</td>
<td>92.25</td>
<td>.0253</td>
<td>$loc_{ch2,4.5}$</td>
<td>81.89</td>
<td>90.90</td>
<td>.0197</td>
</tr>
<tr>
<td>$loc_{ch1,4.6}$</td>
<td>89.77</td>
<td>92.02</td>
<td>.0183</td>
<td>$loc_{ch2,4.6}$</td>
<td>82.94</td>
<td>83.34</td>
<td>.0085</td>
</tr>
</tbody>
</table>
The single chamber actuation is firstly arranged for the validation. Figure 4.20 and 4.21 separately demonstrate the chamber 1’s and the chamber 2’s results. Secondly, a joystick-based pressure control was applied to the platform, by which, both of the chambers can be actuated. Its validation result is plotted in Figure 4.22. Table 4.11 provides the numeric results for the figures above. The local state indexing is similar to the way used in the single chamber, or one can refer to the local partition in Figure 4.17. On average, the NRMSE fit results are above 80%. The system’s dynamical responses are tracked within tolerable errors. It can be seen the proposed identification approach is applicable in the double chamber cases.

![Figure 4.20: The Chamber 1 single actuation test and validation.](image1)

![Figure 4.21: The Chamber 2 single actuation test and validation.](image2)
Figure 4.22: The joystick-controlled test and validation.

Table 4.11: The validation results for the test in Figure 4.20, 4.21, and 4.22.

<table>
<thead>
<tr>
<th>Cases</th>
<th>Results</th>
<th>NRMSE $y_1$</th>
<th>NRMSE $y_2$</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMO</td>
<td>40-step-ahead prediction</td>
<td>95.60%</td>
<td>87.61%</td>
<td>1.7701</td>
</tr>
<tr>
<td>Ch. 1</td>
<td>80-step-ahead prediction</td>
<td>93.78%</td>
<td>80.30%</td>
<td>4.9386</td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>90.72%</td>
<td>73.61%</td>
<td>6.4509</td>
</tr>
<tr>
<td>SIMO</td>
<td>40-step-ahead prediction</td>
<td>96.11%</td>
<td>94.23%</td>
<td>1.8768</td>
</tr>
<tr>
<td>Ch. 2</td>
<td>80-step-ahead prediction</td>
<td>93.92%</td>
<td>90.57%</td>
<td>4.6520</td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>91.06%</td>
<td>88.67%</td>
<td>4.7713</td>
</tr>
<tr>
<td>MIMO</td>
<td>40-step-ahead prediction</td>
<td>91.83%</td>
<td>90.98%</td>
<td>3.6655</td>
</tr>
<tr>
<td>Joystick</td>
<td>80-step-ahead prediction</td>
<td>88.29%</td>
<td>87.97%</td>
<td>5.2224</td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>87.69%</td>
<td>77.29%</td>
<td>8.0041</td>
</tr>
</tbody>
</table>
4.6 Summary

In this chapter, a unified identification approach for the general PLSAs has been presented and summarised. And its implementations for the single chamber bending and the double chamber bending and steering are demonstrated separately. The identification approach is based on the new developed DIO-PWL-OBF models structure which is capable of effectively capturing the nonlinear pressure-shape dynamical relation of the general PLSAs. The DIO-PWL-OBF model structure is consistent with the local identification manner, and it is efficient in implementation. Besides, in the kinematic setting, the new auxiliary configuration subspaces are used. We have shown that by utilising this setting the DOFs of the bending and the steering can be captured in a parameter-free way, and the auxiliary setting can be incorporated with the mainstream kinematic setting. Using the acquired identification model, one can estimate deformation under payloads; this will facilitate the control design in the future study.
Chapter 5

A Direction-Dependent Multisegment Piecewise-Linear Identification Method for Hysteresis-like Nonlinearities

5.1 Introduction

Direction-dependent characteristics is manifested in many physical systems such as examples presented in the survey paper [98]. Those systems operate differently according to whether the input or the output direction is increasing or decreasing. Multisegment piecewise-linear (affine) identification has been studied profoundly in [99]. Those resultant switched linear systems operate differently according to the partitioned input or state space. Note that under this context both direction-dependent and multisegment piecewise-linear are referred to input only. Combining the two operating configurations mentioned above, a so-called direction-dependent multisegment piecewise-linear system can be formulated. A typical static input-output relation generated by this type of systems is illustrated in Figure 5.1. There are five linear pieces in each of the directions. The plot is similar to limiting input-output maps in systems with hysteresis [100, 101]. This observation motivates us to study such model structure for identification of hysteresis-like nonlinearities.

The use of a direction-dependent multisegement piecewise-linear system can bring two major benefits: (a) accuracy in local dynamics and (b) convenience for control design. The linearisation here is taken piecewisely, and the identified local models are not imposed by any weighting term. In contrast, current Hammerstein approaches like those in [102, 103] use one linear model globally with static weighting added on the input end. The former gives a better solution to preserving a system’s dynamic properties than the latter, because each dynamic partition can be characterized individually by an independent set of eigenvalues and eigenvectors. On the other hand, the proposed model structure is a subclass of switched linear systems. Therefore many well-developed control techniques like those in [104, 105] for systems of the parent class can be employed.

In the rest of the chapter, a direction-dependent multisegement piecewise-linear iden-
Figure 5.1: A typical static hysteresis loop approximated by the direction-dependent multisegment piecewise-linear method. Each local region is represented by a linear model $\sigma_{j\pm}$.

tification method is proposed. Firstly, the switching analysis is conducted. Based on it, a coupling invariant principle is defined and chosen. The new setting can avoid a state transformation issue between two subsystems. Secondly based on the new switching setting, a refinement scheme is developed. It can adjust subsystems to more suitable operating regions and improve dynamic approximation quality by tuning local models’ parameters. Thirdly, the original model structure is recast in a Difference Input Output Orthonormal Basis Functions form to make the method efficiently executable in real time. In the end, implementation of the method is demonstrated by bending identification of a low-pressure pneumatic soft actuator system.

5.2 Direction-Dependent Multisegment Piecewise-linear Switched Systems

This study considers discrete-time systems. Given an input-output data set $\{u_{1:k}, y_{1:k}\}$, the difference input output sequences are defined as

$$\delta u_k := u_k - u_{k-1}, \delta u_1 := u_1,$$  \hspace{1cm} (5.1)

and

$$\delta y_k := y_k - y_{k-1}, \delta y_1 := y_1.$$  \hspace{1cm} (5.2)
Based on (5.1) and (5.2), the changing direction of each sample pair \( \{u_k, y_k\} \) can be judged. A direction-dependent multisegement piecewise-linear single-input-single-out (SISO) switched system is formulated as

\[
x_{k+1} = \begin{cases} 
A_{j+} x_k + f_{j+} + B_{j+} u_k, & \delta u_k \geq 0, j_+ \in \mathbb{N}_{\sigma+} \\
A_{j-} x_k + f_{j-} + B_{j-} u_k, & \delta u_k < 0, j_- \in \mathbb{N}_{\sigma-} 
\end{cases}
\]

\[
y_k = \begin{cases} 
C_{j+} x_k + g_{j+} + D_{j+} u_k, & \delta u_k \geq 0, j_+ \in \mathbb{N}_{\sigma+} \\
C_{j-} x_k + g_{j-} + D_{j-} u_k, & \delta u_k < 0, j_- \in \mathbb{N}_{\sigma-} 
\end{cases}
\]

(5.3)

where all the parameters have compatible dimensions. \( \{A_{j+}, B_{j+}, C_{j+}, D_{j+}, f_{j+}, g_{j+}\}_{\sigma+} \) and \( \{A_{j-}, B_{j-}, C_{j-}, D_{j-}, f_{j-}, g_{j-}\}_{\sigma-} \) are subsystems’ parameters of the input-increasing and the input-decreasing directions respectively. In each of the directions, the subsystems are partitioned based on the input level. The two reference partitions along the two directions can be chosen asymmetrically. For convenience, \( \{A_j, B_j, C_j, D_j, f_j, g_j\}_\sigma \) is used to refer to any subsystem and its parameters when the specification for direction is not needed. Besides it is assumed that all the subsystems’ outputs do not have direct feedthrough from the inputs, i.e. \( D_j = 0 \).

### 5.3 Coupling Invariant Switching

The related switched linear subsystems and the way to set their operating regions have been briefly discussed in the previous section. This section now focuses on the switching configuration and will introduce the use of coupling invariant principle to implement (5.3).

In the current setting, a switched linear system uses a common state space for all the subsystems, and the switching is realised by an on-off function like

\[
s(\sigma_j) = \begin{cases} 
1, & \text{if } \sigma_j \text{ is operating} \\
0, & \text{if } \sigma_j \text{ is not operating} 
\end{cases}
\]

(5.4)

attached before the parameters of a corresponding subsystem \( \sigma_j \) [106]. Under the context of analytic nonlinear models [107], this fashion is naturally led to. However in an identified piecewise-linear system, the local subsystems are likely to have different orders. This leads to no solution when switching from a subsystem with a lower order to another with a higher order. Even all the subsystems are identified in the same order, the state transformation between two neighbouring local models is needed to be calibrated, which has be preliminarily studied in [108].

Now the difference between the current switching setting and the newly-defined coupling invariant one that is proposed to use is compared. The main idea is to
segment an input sample and analyse responses of its individual parts. The DIO sequences are a preferred choice because their segmentation contains the information of both value level and direction.

A SISO linear time-invariant system in state-space form is

\[
x_{k+1} = Ax_k + Bu_k \\
y_k =Cx_k.
\] (5.5)

The output response from the time instant \(t_0\) to \(t_k\) is

\[
y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^{k-i-1}Bu(i).
\] (5.6)

The impulse response of the system is given by

\[
h(k) = \begin{cases} 
0, & k = \cdots, -1, 0 \\
CA^{k-1}B, & k = 1, 2, \cdots 
\end{cases}
\] (5.7)

which are called the Markov parameters. Take \(y_k - y_{k-1}\) with the expansion of (5.6), one can get

\[
\delta y_k = C(A^k - A^{k-1})x_0 + \sum_{i=0}^{k-1} CA^{k-i-1}B\delta u_i.
\] (5.8)

It can be interpreted as the response of an individual difference output term \(\delta y_k\) evaluated by all past difference input terms \(\delta u_{0:k}\). In light of difference terms, an original sample pair \(\{u_k, y_k\}\) can be decomposed as

\[
u_k \rightarrow \delta u_{0:k} = [\delta u_0, \cdots, \delta u_k]^T,
\]

\[
y_k \rightarrow \delta y_{0:k} = [\delta y_0, \cdots, \delta y_k]^T,
\]

where in each row the term on the left side and the ones on the right side have the relation

\[
u_k = \sum_{i=0}^k \delta u_i, \ \ \ \ y_k = \sum_{i=0}^k \delta y_i.
\] (5.10)

Using (5.7), (5.8), (5.9) and (5.10), the relation between \(\delta u_{0:k}\) and \(\delta y_{0:k}\) are derived as

\[
\begin{bmatrix}
\delta y_0 \\
\delta y_1 \\
\vdots \\
\delta y_k
\end{bmatrix}
= \begin{bmatrix}
C \\
CA - C \\
\vdots \\
CA^k - CA^{k-1}
\end{bmatrix}
\begin{bmatrix}
\delta x_0 \\
\delta x_0 \\
\vdots \\
\delta x_0
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
h_1 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
h_k & h_{k-1} & \cdots & h_1 & 0
\end{bmatrix}
\begin{bmatrix}
\delta u_0 \\
\delta u_1 \\
\vdots \\
\delta u_k
\end{bmatrix}.
\] (5.11)

Equation (5.11) gives a segmented expression for dynamics of a system. The in-
individual parts are independent of each other. Using the difference terms on both input and output channels brings the convenience in indicating input direction. Besides, the input-level segmentation is flexible; it does not depend on a set of fixed reference points. Symbolically, (5.11) is simplified as

$$\delta y_{0:k} = O^\Delta_k x_0 + T^\Delta_k \delta x_{0:k}. \tag{5.12}$$

For a response from the time instant $t_0$ to $t_{k+m}$, a virtual separation are placed between $t_k$ and $t_{k+1}$. $O^\Delta_{k+m}$ and $T^\Delta_{k+m}$ become

$$\begin{bmatrix}
    C \\
    CA - C \\
    CA^2 - CA \\
    \vdots \\
    CA^k - CA^{k-1} \\
    CA^{k+1} - CA^k \\
    \vdots \\
    CA^{k+m} - CA^{k+m-1}
\end{bmatrix},
\begin{bmatrix}
    0 & 0 & 0 & \cdots & \cdots & 0 \\
    h_1 & 0 & 0 & \cdots & \cdots & 0 \\
    h_2 & h_1 & 0 & \cdots & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\
    h_k & h_{k-1} & \cdots & h_1 & 0 & \cdots & 0 \\
    h_{k+1} & h_k & \cdots & h_1 & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
    h_{k+m} & h_{k+m-1} & \cdots & \cdots & h_1 & 0
\end{bmatrix}. \tag{5.13}$$

which are denoted as

$$O^\Delta_{k+m} = \begin{bmatrix}
    O^\Delta_k \\
    O^\Delta_{k+1,m}
\end{bmatrix}, T^\Delta_{k+m} = \begin{bmatrix}
    T^\Delta_k \\
    T^\Delta_{k+1,m} \\
    T^\Delta_m
\end{bmatrix}. \tag{5.14}$$

$O^\Delta_{k+m}$ and $T^\Delta_{k+m}$ structurally show how the initial state $x_0$ and the past input changes $\delta u_{0:k+m}$ are weighted and added inside one output sample $y_k$. Particularly, $T^\Delta_{k+1,m}$ is the coupling term that indicates how the changes of the past input samples $\delta u_{0:k}$ act on the changes of the future output samples $\delta y_{k+1:m}$.

Next, the switched linear system is introduced. The virtual separation considered in (5.13) and (5.14) is re-defined naturally as the switching moment $t_{k:k+1}$ from subsystem $\sigma_1$ to $\sigma_2$. Structuring the weighting of input changes before the switching when the system is operating under $\sigma_1$, $T^\Delta_k$ is parametrised by $\sigma_1$. $T^\Delta_m$ is parametrised by $\sigma_2$ for the same reason. $T^\Delta_{k+1,m}$ is re-interpreted as the coupling effect that $\sigma_1$ acts on $\sigma_2$. There are, however, two ways to parametrize it. The first one uses $\sigma_2$’s parameters $\{A_{\sigma_2}, B_{\sigma_2}, C_{\sigma_2}\}$. In other words, the state variables are iterated by the new operating subsystem immediately after the switching. This is the current switching setting mentioned in the beginning of this section. On the contrary, the second one takes $\sigma_1$’s parameters $\{A_{\sigma_1}, B_{\sigma_1}, C_{\sigma_1}\}$. The state variables are still iterated by the previous operating subsystem even after the switching. In this way, the dynamic responses induced by the post-switching input changes $\delta u_{0:k}$ are invariant after arbitrary times of switching. The weighting matrix to construct a post-switching output
sample $y_{k+m}$ is unaltered under any ongoing switching event. The second switching setting is the coupling invariant one.

The state transformation issue in the coupling invariant switching is avoided since each subsystem evaluates its state variables independently. This is the primary reason to choose it. Furthermore, there are several other features listed below that make the coupling invariant switching more suitable than the current one for the direction-dependent multisegement piecewise-linear identification method.

1. The switching is numerically stable, as no state jump is caused.
2. For each subsystem $\sigma_j$, the offsets $f_j$ and $g_j$ on both input and output channels are not used. Locally identified models can be used directly in constructing global dynamics.
3. Each subsystem is implemented in the time-invariant way. By doing this, the frequency components of individual subsystems are preserved. Sophisticated frequency-based stability theories can be used in the following control design.

5.4 Refining the Reference Partitions and the Local Models

Overall, the identification procedures are designed in two rounds. In the first round, the local models are to be identified by a staircase input signal. Standard least-square techniques such as subspace methods can be used there. The second round is to refine the used reference points and the estimated local models’ parameters. Generally, an initial reference partition setup is not an optimal choice. The resultant local models’ eigenvalues and eigenvectors probably deviate from actual ones, which can cause poor fitting performance in later applications.

A recursive algorithm for the second-round refinement will be developed in Section 5.4.2. Prior to that, one requisite will be discussed briefly in Section 5.4.1. An equivalent multi-input-single-output (MISO) state space expression will be formulated under the coupling invariant switching setting. This form manages to bring the refinement problem to the subspace identification framework.

5.4.1 The equivalent MISO expression

Under the coupling invariant switching setting, the responses of the subsystems are independent. Figure 5.2 illustrates that in this way a switched SISO linear system can be expressed equivalently as a non-switched MISO one. All the subsystems are
in parallel connection. On the input end, $u_{1:k}$ is separated into individual parts $u_{1,k}^{\sigma_j}$ as
\[
  u_{1:k} = u_{1,k}^{\sigma_1} + \cdots + u_{1,k}^{\sigma_{N_{\sigma}}}
\]  
(5.15)
with respect to their target subsystems. Therefore given $u_{1,k}^{\sigma_j}$s for all subsystems, (5.3) can be seen as a non-switched MISO system as
\[
  x_{k+1} = \bar{A}x_k + \bar{B}[u_{k}^{\sigma_1} \cdots u_{k}^{\sigma_{N_{\sigma}}}]^T
\]
(5.16)
\[
y_k = \bar{C}x_k
\]
where
\[
  \bar{A} = \text{diag}(A_{\sigma_1}, \cdots, A_{\sigma_{N_{\sigma}}}), \quad \bar{B} = \text{diag}(B_{\sigma_1}, \cdots, B_{\sigma_{N_{\sigma}}}),
\]
and
\[
  \bar{C} = [C_{\sigma_1} \cdots C_{\sigma_{N_{\sigma}}}].
\]

$u_{1,k}^{\sigma_j}$s are obtained in the order of

\[
u_{1:k} \rightarrow \delta u_{1:k} \rightarrow \delta u_{1,k}^{\sigma_1}, \cdots, \delta u_{1,k}^{\sigma_{N_{\sigma}}} \rightarrow u_{1,k}^{\sigma_1}, \cdots, u_{1,k}^{\sigma_{N_{\sigma}}}
\]  
(5.17)

For an input data sequence $u_{1:k}$, it is firstly converted to the difference form $\delta u_{1:k}$. Then at each time instant, according to the current input sample value $u_k$ and the direction from $\delta u_k$, $\delta u_k$ is divided and assigned to a number of target subsystems. Finally by applying (5.10), $u_{1,k}^{\sigma_j}$s are obtained.

### 5.4.2 The recursive algorithm for refinement

The subspace identification for the MISO system (5.16) is to solve the optimisation problem
\[
  \min_{\bar{A}, \bar{B}, \bar{C}, x_0} \|y_{1:k} - \mathcal{O}x_0 - \mathcal{T}u_{1:k}\|_2
\]  
(5.18)
where $O$ is the extended observability matrix, and $T$ is the Hankel matrix formed by a system’s Markov parameters. In identification, it is often manageable to set the experiment starting at a steady state. By doing this, the initial state $x_0$ is zero and the term $Ox_0$ can be removed; (5.18) is reduced to

$$
\min_{A,B,C} \|y_{1:k} - Tu_{1:k}\|_2
$$

which is the central expression that will be discussed in detail below.

In the view of the individual subsystems $\{\sigma_j\}_{j \in N_{\sigma}}$, (5.19) can be split to

$$
\min_{\{A_{\sigma_j}, B_{\sigma_j}, C_{\sigma_j}\}_{j \in N_{\sigma}}} \|y_{1:k} - [T_{\sigma_1} \cdots T_{\sigma_{N_{\sigma}}}] [u_{1:k}^{\sigma_1} \cdots u_{1:k}^{\sigma_{N_{\sigma}}}]^T \|_2
$$

(5.20)

In the recursive algorithm, the first step is to adjust the offset values on individual subsystems’ input sequences. As the reference partitions not only depend on the value level of an input but also its direction, it is erroneous to directly vary the lower and upper bounds of each local region. It will cause a subsystem receiving a mismatched offset that is used for a different one. This issue can be tackled by considering the offset adjustment at viewpoint of difference input sequences (5.1), (5.17) and the decomposed structures (5.11), (5.13).

For a subsystem’s difference input sequence $\delta u_{1:k}^{\sigma_j}$, a corresponding offset vector $d_{1:k}^{\sigma_j}$ will be defined and added for adjustment as

$$
\delta u_{1:k}^{\sigma_j} + d_{1:k}^{\sigma_j}.
$$

(5.21)

$d_{1:k}^{\sigma_j}$ comes from the offsets on the lower and the upper bounds of a local region

$$
d_{1:k}^{\sigma_j} = d_{1:k}^{\sigma_j,j-1} + d_{1:k}^{\sigma_j,j+1},
$$

(5.22)

and the two bounds are adjacent to the neighbouring subsystems $\sigma_{j-1}$ and $\sigma_{j+1}$ respectively. For a bound shared by two subsystems $\sigma_{j_1}$ and $\sigma_{j_2}$, $d_{1:k}^{\sigma_{j_1,j_2}}$ is formed as

$$
d_{1:k}^{\sigma_{j_1,j_2}} = c_{\sigma_{j_1,j_2}} \delta u_{1:k}^{\sigma_{j_1,j_2}}, \quad v_{k}^{\sigma_{j_1,j_2}} = \begin{cases} 0, & \delta u_k \text{ does not cross the bound} \\ 1, & \delta u_k \text{ crosses the bound} \end{cases}
$$

(5.23)

where $c_{\sigma_{j_1,j_2}}$ is a scalar that reflects the degree of the offset, and $v_{k}^{\sigma_{j_1,j_2}}$ is a vector that indicates the timing when such offset occurs.
Meanwhile, it is equivalent to transfer (5.19) in the terms of DIO to

$$\min_{\bar{A}, \bar{B}, \bar{C}} \| \delta y_{1:k} - T \delta u_{1:k} \|_2. \quad (5.24)$$

Similarly, (5.20) becomes

$$\min_{\{A_{\sigma_j}, B_{\sigma_j}, C_{\sigma_j}\}_{j \in N_{\sigma}}} \left\| \delta y_{1:k} - \left[ T_{\sigma_1} \cdots T_{\sigma_{N_{\sigma}}} \right] \left[ \delta u_{1:k}^{\sigma_1} \cdots \delta u_{1:k}^{\sigma_{N_{\sigma}}} \right]^T \right\|_2 \quad (5.25)$$

Substituting (5.21) into (5.25) and setting system matrices fixed result in

$$\min_{\{c_{\sigma_j,j-1}, c_{\sigma_j,j+1}\}_{j \in N_{\sigma}}} \left\| \delta y_{1:k} - \left[ T_{\sigma_j} \cdots T_{\sigma_{N_{\sigma}}} \right] \left[ \delta u_{1:k}^{\sigma_1} \cdots \delta u_{1:k}^{\sigma_{N_{\sigma}}} \right]^T \right\|_2 \quad (5.26)$$

It is a linear optimisation problem with variables $c_{\sigma_j,j-1}$ and $c_{\sigma_j,j+1}$. The optimal point by simultaneously searching all the offset variables, however, may probably be far from actual ones. This is because that any two subsystems can be very close in the pole structure; their inner products by the Markov parameters can be near 1. It means that the two subsystems’ responses characterised by poles are not completely distinguished from each other. Consequently, the searching may very likely go to a wrong direction and continue using a mismatched subsystem to reduce estimated output error. Hence, it is prudent to deal with only one offset regarding one local region bound each time. Under this constraint, the lower and upper bounds of a local region are reduced from (5.26) to

$$\min_{c_{\sigma_j,j-1}} \left\| \delta y_{1:k} - \left[ T_{\sigma_1} \cdots T_{\sigma_{N_{\sigma}}} \right] \left[ \delta u_{1:k}^{\sigma_1} \cdots \delta u_{1:k}^{\sigma_{N_{\sigma}}} \right] - T_{\sigma_j} d_{1:k}^{\sigma_{j-1}} + T_{\sigma_{j-1}} d_{1:k}^{\sigma_{j-1}} \right\|_2 \quad (5.27)$$

and

$$\min_{c_{\sigma_j,j+1}} \left\| \delta y_{1:k} - \left[ T_{\sigma_1} \cdots T_{\sigma_{N_{\sigma}}} \right] \left[ \delta u_{1:k}^{\sigma_1} \cdots \delta u_{1:k}^{\sigma_{N_{\sigma}}} \right] + T_{\sigma_j} d_{1:k}^{\sigma_{j+1}} - T_{\sigma_{j+1}} d_{1:k}^{\sigma_{j+1}} \right\|_2 \quad (5.28)$$

separately. The arguments $c_{\sigma_j,j-1}$ and $c_{\sigma_j,j+1}$ indicate that the reference points $r_{\sigma_j,j-1}$ and $r_{\sigma_j,j+1}$ for a subsystem $\sigma_j$’s region will be modified to $r_{\sigma_j,j-1} - c_{\sigma_j,j-1}$ and $r_{\sigma_j,j+1} + c_{\sigma_j,j+1}$. All the regions’ bounds can be revised in this way. After they all have been updated, the inputs to each subsystems will be re-allocated through (5.17).

In the second step, re-identification via subspace method is taken for the subsystems with the updated reference partitions and the re-allocated individual input sequences. The related searching for state matrices $A_j$ is subjected to less-one maximum singular value. This is to ensure the existence of a limiting input-output map [109] and can be implemented by a recent Linear Matrix Inequality (LMI) eigenvalue constraints technique [110].
Procedures regarding both first-round identification and second-round refinement are summarised through pseudocode in Algorithm 2.

**Algorithm 2: 1st-round identification and 2nd-round refinement**

1: Set a number of initial reference partitions $r_{initial}^+$, $r_{initial}^-$.  
2: Conduct a staircase identification according to $r_{initial}^+$, $r_{initial}^-$.  
3: Acquire the identification data set $\{u^{idn}_{1:k}, y^{idn}_{1:k}\}$.  
4: Extract local data sets by removing offsets on each stair part.  
5: Take subspace identification under the LMI eigenvalue constraints for the local data sets.  
6: Get initial system parameters of local models $\{A_j, B_j, C_j\}_{j \in \mathbb{N}_\sigma}^{initial}$.  
7: Take a second random test.  
8: Acquire the reference data set $\{u^{idn}_{1:k}, y^{idn}_{1:k}\}$ for refinement.  
9: repeat  
10: For each region, calculate the adjusting scalers $c_{\sigma_{j-1}}$ and $c_{\sigma_{j+1}}$ by (5.27) and (5.28) for its lower and upper bounds that coincide with neighbouring regions.  
11: Update the reference partitions to $r_{i+1}^+$, $r_{i+1}^-$ according to $\{c_{\sigma_{j-1}}, c_{\sigma_{j+1}}\}_{j \in \mathbb{N}_\sigma}$.  
12: Re-allocate the individual subsystems’ inputs based on $r_{i+1}^+$, $r_{i+1}^-$ by (5.17).  
13: Take subspace identification under the LMI eigenvalue constraints with the updated local data sets.  
14: Get updated parametrisation of local models $\{A_j, B_j, C_j\}_{j \in \mathbb{N}_\sigma}^{i+1}$.  
15: until the changing rate of the updated fitting result is below a designed threshold.  
16: Obtain the refined reference partitions $r_{refined}^+$, $r_{refined}^-$ and local models $\{A_j, B_j, C_j\}_{j \in \mathbb{N}_\sigma}^{refined}$.

**Example 1:** Consider a coupling invariant switched linear system. Its subsystems are parametrised as below:

$$\sigma_{1+} = \left\{ \begin{bmatrix} 0.8261 & 0.3351 & -0.2479 \\ 0.1086 & 0.8086 & -1.3053 \\ -0.0357 & 0.1414 & 0.8522 \end{bmatrix}, \begin{bmatrix} 4.3033 \\ 6.4259 \\ 5.0222 \end{bmatrix}, \begin{bmatrix} -0.1231 \\ 0.0426 \\ 0.0420 \end{bmatrix}^T \right\}$$

$$\sigma_{2+} = \left\{ \begin{bmatrix} 0.4943 & -0.1398 & 0.6757 & -0.8276 \\ -0.0491 & 0.2236 & 0.7035 & -0.8332 \\ 0.2882 & -1.1148 & 1.7640 & -1.3738 \\ -0.0268 & -0.0113 & 0.2798 & 0.2410 \end{bmatrix}, \begin{bmatrix} 36.7939 \\ 30.2478 \\ 31.0030 \\ 16.9798 \end{bmatrix}, \begin{bmatrix} -0.6720 \\ 0.7013 \\ -0.1942 \\ 0.5274 \end{bmatrix}^T \right\}$$
\( \sigma_1^- = \{0.9292, -2.1089, -0.1253\} \)

\[
\sigma_2^- = \begin{bmatrix}
0.8978 & 0.8758 & 1.7093 & -0.1345 \\
0.0102 & 0.3294 & -4.1889 & 0.1735
\end{bmatrix}^T.
\]

The input value ranges from 0 to 20, and the reference partitions are

\[
r_{\text{actual}}^+ = [0 \ 8 \ 20], \quad r_{\text{actual}}^- = [0 \ 13 \ 20]
\]

for the input-increasing and input-decreasing directions respectively. The output channel is added with a zero-mean white noise. The signal-to-noise ratio is 100. In the first-round identification, the partitions of reference are estimated as

\[
r_{\text{initial}}^+ = [0 \ 11 \ 20], \quad r_{\text{initial}}^- = [0 \ 10 \ 20].
\]

According to \( r_{\text{initial}}^+ \) and \( r_{\text{initial}}^- \), a staircase identification is conducted as shown in Figure 5.3 (a). Figure 5.3 (b) reveals the assigned input to each subsystem by the relation (5.17), and Figure 5.3 (c) is the output under the identification signal. Through the MOESP method [111], we acquire an initial parameter setting for the subsystems. In Figure 5.4, it is observed that the approximation by the initial setting under a new white-noise input ends up with a large deviation from the actual response. Then the initial parametrization and the white-noise input is used to refine the reference partitions and the identified subsystems by Algorithm 2. The stop condition is triggered when the difference of the normalised mean square error (NMSE) fitting result between two iteration steps is below 0.3%. Figure 5.4 shows the fitting plots at iteration; it converges to 95% NMSE fitting with respect to reference response after 7 steps. The refined reference partitions are

\[
r_{\text{refined}}^+ = [0 \ 8.34 \ 20], \quad r_{\text{refined}}^- = [0 \ 12.67 \ 20].
\]

At last, another different white-noise input is used to validate the refined outcomes. The result is shown in Figure 5.5. On average, the NMSE fitting of different validation results are above 90%.
Figure 5.3: Signals in the first-round identification in Example 1: (a) the stair-case input, (b) the assigned input parts for individual subsystems, (c) the resultant output.
Figure 5.4: The approximation plots after each iteration during the refinement in Example 1.

Figure 5.5: The validation test under a random input in Example 1.
5.5 Structural Approximation by Orthonormal Basis Functions

The implementation of (5.3) and (5.16) is not efficient, as it uses a state space with a dimension up to \( \sum_{j=1}^{N_r} \text{dim}(A_{r_j}) \). More critically, it has not yet given a real-time switching solution based on both input level and input direction. The method by (5.17) is only suitable for post processing because the direction-dependency requires all the past input samples when segmenting the current one. Such processing cannot be realised by finite-length registers in computers. The following will show the use of orthonormal basis functions and the DIO setting in tackling the two implementation issues.

A set of orthonormal basis functions \( \mathbf{B}_{1:n} \) span an \( n \)-dimensional subspace. A stable and strictly proper transfer function \( G \) can be approximated in the subspace as

\[
G \approx \sum_{i=1}^{N_B} \theta_i \mathbf{B}_i, \quad (5.29)
\]

where \( \theta_i \) is the coefficient. \( \mathbf{B}_i \) can be constructed recursively as

\[
\mathbf{B}_i = \sqrt{1 - |\xi_i^2|} \prod_{j=0}^{i-1} \frac{1 - \xi_j z}{z - \xi_j}, \quad (5.30)
\]

through a repetitive pole structure

\[
\{\xi_1, \cdots, \xi_{n_b}, \xi_1, \cdots, \xi_{n_b}, \cdots\}, \quad (5.31)
\]

(5.31) results in an optimal selection when the poles coincide with the actual system’s ones [77]. In this study, the pole set is determined from the identified subsystems. The pole selection problem is formulated as in [79]. The solution is provided by a recent work in [80] where a clustering algorithm named Fuzzy-Kolmogorov c-Max is developed.

When a proper pole set is chosen and \( \mathbf{B}_{1:n} \) are constructed, for each subsystem the parameters \( \theta_{1:n} \) can be calculated by either time-domain inner product through Markov parameters or frequency-domain inner product through transfer function.

The preliminaries for orthonormal basis functions are ready. Next, the DIO setting is incorporated into them to realise a real-time executable form of (5.3) and (5.16).

From (5.6) and (5.8), one can get

\[
y_k = y_{k-1} + C A^k x_0 + \sum_{i=0}^{k-1} C A^{k-i-1} B \delta u_i. \quad (5.32)
\]
The last two terms on the right side can be formed in the transfer function as

\[ y_k = y_{k-1} + C(zI - A)^{-1}B\delta u_k. \]  

(5.33)

To generate \( y_k \) from \( u_k \), an input sample firstly subtracts its previous sample and then the resultant difference \( \delta u_k \) is delivered to the system of \( C(zI - A)^{-1}B \). In this way, \( C(zI - A)^{-1}B \) produces \( \delta y_k \), and by continuously accumulating the output samples the normal \( y_k \) is obtained.

Applying (5.16) to (5.33), we get

\[ y_k = y_{k-1} + \bar{C}(zI - \bar{A})^{-1}\bar{B}[\delta u_k^{\sigma_1}, \ldots, \delta u_k^{\sigma_j}]^T. \]  

(5.34)

The DIO setting provides a solution for the input value segmentation in run time. In each iteration step, the segmentation is taken by using the latest difference part \( \delta u_k \). Therefore the judgement only needs the current and the previous samples, \( u_k \) and \( u_{k-1} \).

Finally, after approximating all the subsystems by a set of orthonormal basis functions, (5.34) becomes

\[ y_k \approx y_{k-1} + \sum_{i=1}^{N_B} B_i \sum_{\sigma_j} \theta_{i\sigma_j} \delta u_k^{\sigma_j}, \]  

(5.35)

which is illustrated schematically in Figure 5.6. The \( n \)-step-ahead predictor is formulated as

\[ y_{k+n} \approx y_k + \sum_{m=k+1}^{m+n} \sum_{i=1}^{N_B} B_i \sum_{\sigma_j} \theta_{i\sigma_j} \delta u_m^{\sigma_j}, \]  

(5.36)

![Figure 5.6: Schematic drawing of the DIO-OBF structure.](image)

The pseudocode of the on-line input segmentation and switching is listed in Algorithm 3. Note \( \epsilon \) is a tuning parameter. It sets a dead zone on the range of difference input samples, which can reduce the reaction sensitivity in the switching.

**Example 2:** With the same system parameters set in Example 1, the switched system
Algorithm 3: Segmentation and switching in DIO-OBF

1: Obtain $u_k$ and $u_{k-1}$.
2: Get $\delta u_k = u_k - u_{k-1}$.
3: Set $\epsilon \geq 0$.
4: if $\delta u_k > \epsilon$ then
5: Divide $\delta u_k$ to $\delta u^\sigma_{1^+}, \ldots, \delta u^\sigma_{N^+}$ based on $r_+$. 
6: Calculate the input to each $B_i$ as $u^{B_i}_k = \sum_{j=1}^{N^+} \theta^{B_i}_{j^+} \delta u^\sigma_{j^+}$
7: Feed $u^{B_i}_k$ to $B_i$, $\forall i$.
8: else if $\delta u_k < -\epsilon$ then
9: Divide $\delta u_k$ to $\delta u^\sigma_{1^-}, \ldots, \delta u^\sigma_{N^-}$ based on $r_-$. 
10: Calculate the input to each $B_i$ as $u^{B_i}_k = \sum_{j=1}^{N^-} \theta^{B_i}_{j^-} \delta u^\sigma_{j^-}$
11: Feed $u^{B_i}_k$ to $B_i$, $\forall i$.
12: else
13: No value is further delivered, $u^{B_i}_k = 0$, $\forall i$.
14: end if

is going to be approximated by the DIO-OBF model structure. The pole set for all the subsystems is

$$\Omega_p = \left\{ 0.9292, 0.9132, 0.9024, 0.7922 \pm 0.3780i, 0.7991, 0.6644 \pm 0.3753i, 0.5950, 0.3140 \right\}$$

By the FKcM clustering algorithm, an optimal pole set is calculated to be

$$\xi_{1:n_b} = \left\{ 0.3141, 0.5952, 0.7991, 0.9135 \right\}$$

Figure 5.7: The validation test under a sinusoid signal in Example 2. The system in Example 1 is approximated by the DIO-OBF model structure.

Then it is used in the OBF construction. On average, the infinite impulse responses
(IIR) approximated by the OBFS are over 95% NMSE for each subsystem. The global approximation by the DIO-OBF structure is over 90% NMSE fitting. Figure 5.7 shows the validation test by a sinusoid wave.

5.6 Implementation and Results

The implementation is demonstrated by bending identification of a low-pressure pneumatic soft actuator. A bending demonstration of such actuators is illustrated in Figure 5.8. The related background has been discussed in [112]. Especially, the static hysteresis of the actuator has been observed there. In the following, the proposed method will be applied to identify the nonlinear relation between input pressure and induced bending angle under single chamber actuation.

![Figure 5.8: The bending demonstration of a low-pressure pneumatic soft actuator.](image)

The sampling rate is 80 Hz. The operating pressure ranges from 0 to 0.35 bar, which drives the test actuator from the vertically standing state to an arc shape with nearly 150° bending angle. Both the input-increasing and the input-decreasing directions have the same initial reference partition

\[
\begin{bmatrix}
0 & 0.05 & 0.10 & 0.15 & 0.20 & 0.25 & 0.30 & 0.35
\end{bmatrix}.
\]

In each direction, the local subsystems are labelled in an ascending order. For example, \( \sigma_{3+} \) is in the input-increasing direction and it operates when the input is between 0.10 bar and 0.15 bar. The first-round staircase identification is conducted as shown in Figure 5.9.

The local data sets are obtained by extracting responses of each stair and removing
Figure 5.9: The identification signals in the pneumatic actuator experiment: (a) the response under the staircase input, (b) the staircase input signal.

their offsets on both input and output channels. Then the MOSEP subspace method are used to identify the local sets. All the NMSE fitting results are above 90%. Particularly in Table 5.1 we compares the poles, the pole and the zero numbers to emphasize the structural difference among the subsystems. Due to the gas fill-in phase of the actuator, $\sigma_{1+}$ and $\sigma_{1-}$ are regarded as dead zones.

A different test is then conducted and the obtained data is used in the second-round refinement. The default reference partitions and the identified local models are initial parameters in the recursive algorithm. The fitting is improved to over 95% after 5 iterations. Figure 5.10 shows the approximation plots regarding the reference response at each iteration step.

The refined reference partitions are

\[
r_+^{\text{refined}} = \begin{bmatrix} 0 & 0.0539 & 0.0947 & 0.1400 & 0.1927 & 0.2206 & 0.2959 & 0.3500 \end{bmatrix},
\]

and

\[
r_-^{\text{refined}} = \begin{bmatrix} 0 & 0.0475 & 0.1042 & 0.1477 & 0.1856 & 0.2496 & 0.2889 & 0.3500 \end{bmatrix}.
\]

In the local re-identification, the pole-zero structure for each subsystem is preserved.
Table 5.1: Structural comparison between identified subsystems in the pneumatic actuator.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Pole Num.</th>
<th>Zero Num.</th>
<th>Local poles</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ₁⁺</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>σ₂⁺</td>
<td>2</td>
<td>1</td>
<td>0.9288 ± 0.0828i</td>
</tr>
<tr>
<td>σ₃⁺</td>
<td>2</td>
<td>1</td>
<td>0.8878 ± 0.0775i</td>
</tr>
<tr>
<td>σ₄⁺</td>
<td>2</td>
<td>1</td>
<td>0.8717 ± 0.0154i</td>
</tr>
<tr>
<td>σ₅⁺</td>
<td>3</td>
<td>2</td>
<td>0.9883, 0.9125 ± 0.0716i</td>
</tr>
<tr>
<td>σ₆⁺</td>
<td>3</td>
<td>2</td>
<td>0.9908, 0.9179 ± 0.0363i</td>
</tr>
<tr>
<td>σ₇⁺</td>
<td>3</td>
<td>2</td>
<td>0.9774, 0.9174 ± 0.0793i</td>
</tr>
<tr>
<td>σ₁⁻</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>σ₂⁻</td>
<td>2</td>
<td>1</td>
<td>0.9689 ± 0.0812i</td>
</tr>
<tr>
<td>σ₃⁻</td>
<td>2</td>
<td>1</td>
<td>0.9786 ± 0.1072i</td>
</tr>
<tr>
<td>σ₄⁻</td>
<td>1</td>
<td>0</td>
<td>0.8930</td>
</tr>
<tr>
<td>σ₅⁻</td>
<td>1</td>
<td>0</td>
<td>0.9699</td>
</tr>
<tr>
<td>σ₆⁻</td>
<td>1</td>
<td>0</td>
<td>0.9690</td>
</tr>
<tr>
<td>σ₇⁻</td>
<td>1</td>
<td>0</td>
<td>0.9665</td>
</tr>
</tbody>
</table>

Figure 5.10: The approximation plots after each iteration during the refinement in the pneumatic actuator experiment.
After obtaining the refined poles for each subsystems, the optimal pole set

\[ \xi_{1:n_b} = \begin{bmatrix} 0.878 & 0.901 & 0.970 & 0.991 \end{bmatrix} \]

is calculated through the FKcM algorithm. By a twice repetition, the OBFs are constructed and the resultant IIR approximations for each subsystem are over 95% NMSE on average.

![Figure 5.11: The validation test for the DIO-OBF model structure in the pneumatic actuator experiment. (a) the measured output and its comparison with the two simulated responses. (b) the measured input during the test.](image)

Finally, another random test is taken to validate the identified DIO-OBF model structure. The result is shown in Figure 5.11. Besides the approximation by the equivalent MISO through post processing is also presented. All the NMSE fitting results are above 90%. Therefore the proposed identification method is validated.
5.7 Summary

This chapter has presented a direction-dependent multisegement piecewise-linear identification method for hysteresis-like nonlinearities. The method is based on the newly-defined coupling invariant principle. The refinement algorithm and the DIO-OBF structure have been developed to improve fitting quality and on-line implementation efficiency. The method has been validated experimentally through the bending identification of the pneumatic actuator. In the future study, a weighting term will be added after the difference input part in the developed DIO-OBF structure. In this way, rate-dependency can be addressed from extending the method discussed in this note. This will further enlarge the method’s application scope for a range of physical systems.
Chapter 6

Conclusions

6.1 Summary

This thesis proposed a unified identification approach for a class of soft pneumatic actuators, and then it conducted the theoretical study that complements the use of the underlying model identification structure. The new model identification structure was developed based on multisegment piecewise-linear systems. The major nonlinearity observed from typical low-pressure soft pneumatic actuators was approximated by a number of local linear models. In this way, standard linear identification techniques could be employed locally. Then with the newly introduced DIO concept, the switching between different subsystems became smooth so that the global nonlinearity can be accurately approximated. An initial identification result can be further improved by the developed refinement algorithm. Starting from the initially identified model, the algorithm searches an optimal set regarding local partitions and local models. In order to make real-time execution efficient, the direction-dependent multisegment piecewise-linear model structure was approximated and realised by an optimally selected set of OBFs. The OBF together with the DIO setting reduces the size of the system as individually responses for the subsystem are optimally merged to a common fixed dominator structure. The proposed identification framework can be applied to a wide range of soft pneumatic actuators, which will benefit control designs for soft robots potentially.

6.2 Future Work

The research outcomes suggest two immediate future studies. The first one is to use the free-space identified model to estimate contacts force when an soft bending actuator is interacting with objects. As the identification model can provide an effective prediction, the run-time discrepancy between the measured responses and calculated responses can be used to estimate the contact forces. On the other hand,
the identification model structure preserve the features of individual local dynamics, i.e. pole structures and time-invariance; this suggests the use of well-developed frequency-domain control methods, and thus the control design can be simplified.
Bibliography


