A DIGITAL TWIN FOR CONTROLLING THERMO-FLUIDIC PROCESSES

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus prof.dr.ir. F.P.T. Baaijen,
voor een commissie aangewezen door het College voor Promoties,
in het openbaar te verdedigen op

maandag 2 november 2020 om 16:00 uur

door

Amritam DAS

geboren te Shrirampur, India

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommisie is als volgt:

voorzitter:	prof. dr. ir. A. M. J. Koonen
1^e promotor:	prof. dr. S. Weiland
copromotor(en):	dr. M. M. Peet (Arizona State University)
leden:	prof. dr. ir. W. P. M .H. Heemels
	prof. dr. ir. H. M. A Wijshoff
	prof. dr. ir. T. Keviczky (Technische Universiteit Delft)
	prof. dr. A. Papachristodoulou (University of Oxford)
	dr. ir. S. Koekebakker (Cannon Production Printing)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

A Digital Twin for Controlling Thermo-Fluidic Processes

Amritam Das

This dissertation has been completed in fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

The work presented in this thesis has been supported by the programme *IMPULSE-II*, funded by Canon Production Printing B.V.

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-5140-8

Copyright © 2020 by Amritam Das.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Printed by Ridderprint | www.ridderprint.nl

Dedicated to my family

"A method is a trick I use twice"

– G. Pólya

Summary

A Digital Twin for Controlling Thermo-Fluidic Processes

Amritam Das

Digital twin is a computer-aided platform that virtually represents an industrial asset for emulating its conception, mechanism, usage and life cycle. As a digital testbed, its functionalities encompass visualizing the operation of the industrial asset, predicting its life cycle, optimizing its design, verification of its performance, detection of faults, and synthesis of estimators or controllers to improve its performance. This thesis presents a generic framework to develop a digital twin that describes thermo-fluidic processes and synthesizes controllers for them to achieve desirable performance. The thermo-fluidic processes are defined by the interaction of solids and fluids under the influence of external thermal energy. These processes involve physical quantities that dynamically vary over time, as well as over space. Moreover, due to the lack of sensors and actuators, these physical quantities are often not measured or cannot be influenced by external inputs directly.

At the core of a digital twin, there is a mathematical model that represents, describes and predicts the spatio-temporal dynamics of thermo-fluidic processes. Such a model is governed by Partial Differential Equations (PDEs) and Ordinary Differential Equations (ODEs) that are mutually interconnected over the boundary in a connected graph. The physical model of thermo-fluidic processes explains the energy exchange among various components as a result of the heterogeneous interconnection among solids, fluids and their dynamic interaction. The graph-theoretic structure makes the modeling framework modular and generic with decisive advantages in terms of flexibility, scalability, computability and versatility to build the digital twin. From a computational point of view, the very nature of spatio-temporal dynamics allows developing any functionality of the thermo-fluidic digital twin by either lumping the PDEs or numerically approximating the PDEs or else directly utilizing the coupled PDE-ODE model. This thesis provides these three approaches and introduces computational tools to establish any functionality on the digital twin involving optimization, estimation, control, virtual testing, prediction, and maintenance of the thermo-fluidic processes. The criteria, based on which these approaches are evaluated, are computational scalability, a guarantee of closed-loop performance and suitability of respective approaches to control the thermo-fluidic processes.

Among the three approaches, the computational method of lumping neglects the spatial dependency of the physical quantities by replacing the PDEs with lumped models that are governed by ODEs. Hence, the well-established theories of finite-dimensional systems can be used for quantitative analysis, control, monitoring, and fault diagnosis. This research exploits the graph-theoretic modeling framework to modularize and upscale this lumping method without compromising the model accuracy. A model predictive control strategy is implemented for controlling the thermo-fluidic process that demonstrates the practical functionality of the digital twin.

On the other hand, the computational method of approximation discretizes the spatial domain and approximate the PDEs using numerical methods. The numerical discretization of PDEs leads to a large-scale coupled ODEs that are subsequently used for developing model-based estimators or controllers. Here, the main contribution is a computational technique that can modularize the numerical discretization of thermo-fluidic processes under arbitrary spatial interconnection.

The third approach directly uses the spatially interconnected coupled PDE-ODE models for quantitative analysis, control, monitoring, and fault diagnosis. Here, a computational framework is developed for a class of Partial Integral (PI) operators that are used to derive a behaviorally equivalent representation of coupled PDE-ODE models, known as Partial Integral Equations (PIEs). It is shown that the analysis and designing estimator based optimal controllers of PIEs amounts to solving a set of Linear Matrix Inequalities (LMIs) without any need of discretization.

The developed methodologies are applied to build a digital twin for a commercial inkjet printer. In particular, the process of printing is viewed as a physical integration of liquid ink and solid medium in two stages: a) jetting where droplets of liquid are deposited on a solid medium based on the user's demand, and b) fixation where printed medium is dried to evaporate excess liquid. The digital twin serves as a virtual framework to design and test control strategies on the jetting and fixation stages for delivering the printed products with a desirable print-quality. To this end, the digital twin synthesizes a controller that minimizes the spatially varying inconsistency in liquid temperature during jetting without any additional sensors or actuators. In the fixation stage, the digital twin synthesizes an algorithm to estimate the spatial heterogeneity of moisture and temperature content in a printed medium by solely using a set of point-specific temperature sensors. Owing to the modular framework, the digital twin of an inkjet printer remains flexible, versatile and integrable towards the future designs of commercial printers.

Contents

Summary **vii**

I Prologue **1**

1 Introduction **3**

1.1 Digitization of Industrial Processes 4
1.2 Functionalities and Features of a Digital Twin 5
1.3 Application: Inkjet Printing 7
1.4 Building a Digital Twin for DoD Inkjet Printer 10
1.5 Research Objective 13
1.6 Subproblems and Approaches to Solve Them 13
1.7 Organization of the thesis 19

2 Digital Twin's Framework to Model Thermo-Fluidic Processes **21**

2.1 Introduction 22
2.2 Abstract Definition of Spatially Interconnected Thermo-Fluidic Processes 22
2.3 Building Thermo-Fluidic Model for the Digital Twin 24
2.3.1 Specifying Physical Asset's Topology 24
2.3.2 Specifying Spatial Configuration 25
2.3.3 Specifying Signals 26
2.3.4 Specifying Edges 28
2.3.5 Specifying Time-Varying Parametric Relation 30
2.3.6 Specifying Node Dynamics 30
2.4 Equivalent Representations of Lumped Nodes 36
2.5 Equivalent Representation of Spatially Distributed Nodes 40
2.5.1 Normalization of Spatial Domain 40

2.6 Combined Representation of Spatially Distributed Nodes and Lumped Nodes 42
2.7 Closing Remarks 46

II Computation with Lumping 47

3 Lumping Based Estimation and Control of Thermo-Fluidic Processes 49
3.1 Introduction 50
3.2 Specifications of Lumped Model 51
3.3 Designing a Soft Sensor for Monitoring Thermo-Fluidic Processes . 53
3.4 Synthesizing a Model Predictive Controller for Thermo-Fluidic Processes 57
3.5 Academic Case Study 63
3.5.1 Implementing the Digital Twin 64
3.5.2 Soft Sensing Based Model Predictive Control 68
3.6 Closing Remarks 72

Appendices 73
3.A Equivalence Between Thermo-Fluidic and Electrical Domain 73
3.B Proof of Algorithm 3.1 74
3.C Proof of Theorem 3.2 76

III Computation with Approximation 79

4 Approximating Diffusive Thermo-Fluidic Processes 81
4.1 Introduction 82
4.2 Specifications for Diffusive Thermo-Fluidic Processes 83
4.3 Finite-Dimensional Approximation of Diffusive Thermo-Fluidic Processes 85
4.3.1 Step 1: Separation of Solution and Homogenization 86
4.3.2 Step 2: Spectral Decomposition of $\mathbf{z}(s,t)$ 87
4.3.3 Step 3: Approximation of Solution and Finding $\theta_n(t)$ 92
4.4 Academic Case Study: Heat Diffusion in Composite 93
4.5 Closing Remarks 97

5 Estimation of Unknown Physical Parameters in Diffusive Thermo-Fluidic Processes 99
5.1 Introduction . . . 100
5.2 Setting Up the Parameter Estimation Problem . . . 101
5.3 Grey-Box Identification in Frequency Domain . . . 102
5.3.1 Parametrization of the unknown functions . . . 103
5.3.2 Discretization procedure . . . 104
5.3.3 Specification of the cost functional . . . 106
5.3.4 Optimization Procedure . . . 106
5.4 Academic Case Study . . . 108
5.4.1 Data Generation for the Case Study . . . 108
5.4.2 Estimation Results . . . 109
5.5 Closing Remarks . . . 112

Appendices 113
5.A Proof of Lemma 5.1 . . . 113

IV Computation with PIEs 115

6 Building Digital Twin with PIEs 117
6.1 Introduction . . . 118
6.2 PI Operators, PIEs and PIETOOLS . . . 120
6.2.1 Introduction to PI Operators and PIETOOLS . . . 120
6.2.2 Partial Integral Equations . . . 121
6.2.3 Properties of Partial Integral Operators . . . 121
6.2.4 Positivity of PI Operators . . . 122
6.2.5 Linear PI Inequalities (LPIs) . . . 124
6.3 Thermo-Fluidic Processes and PDE-ODE Systems . . . 125
6.4 Representation of Thermo-Fluidic Processes Using PIEs . . . 127
6.4.1 Finding the Unitary State Transformation . . . 128
6.4.2 PIEs are Equivalent to PDE-ODE Coupled Systems . . . 130
6.5 Analysis and Control of Thermo-Fluidic Processes Using PIEs . . . 133
6.5.1 Verifying Exponential Stability of Thermo-Fluidic Processes 134
6.5.2 Determining the Worst Case Disturbance Amplification . . . 135
6.5.3 Verifying Input-to-State Stability of Thermo-Fluidic Processes 138
6.6 Synthesis of $\mathcal{H}_\infty$ Optimal State Estimator . . . 141
6.6.1 Designing Luenberger-Type State Estimator . . . 142
6.6.2 Implementation of PIE State Estimator . . . 146
6.7 Closing Remarks . . . 149

Appendices **151**
6.A Proof of Theorem 6.1 151
6.B Proof of Theorem 6.3 152
6.C Proof of Theorem 6.3 155
6.D Proof of Theorem 6.4 160
6.E Proof of Theorem 6.5 161
6.F Proof Of Theorem 6.6 162
6.G Proof of Theorem 6.7 165

V Applications to Industrial Benchmarks **167**

7 Industrial Application of Lumped Digital Twin: DoD Inkjet Printhead **169**
7.1 Introduction 170
7.2 Building a Digital Twin of Inkjet Printhead 170
7.3 Calibration of Soft Sensor 173
7.4 Experimental Validation 177
7.4.1 Experimental Setup 177
7.4.2 Experiment Design 178
7.4.3 Setting Up the Digital Twin for Simulation 178
7.4.4 Results 179
7.5 In Situ Controller: A Proof of Principle 180
7.5.1 Concept of In Situ Sensing-Actuation 180
7.5.2 Performance Specifications 181
7.5.3 Generating Voltage Pulse for Heating Actuators 181
7.5.4 Implementation and Illustration of MPC Scheme 182
7.6 Closing Remarks 187

8 Estimation of Moisture in Composite Materials in Fixation Process **189**
8.1 Introduction 190
8.2 Overview of the Fixation Process 190
8.3 Problem 1: Modeling and Simulation of Diffusive Thermo-Fluidic Processes in Composite Material 191
8.3.1 Graph Theoretic Modeling of Fixation Process 191
8.4 Scaling of Spatial Domain for Individual Spatially Distributed Node 194
8.5 Simulation of Heat and Moisture Diffusion in a Composite Material 195
8.5.1 Simulation of Coupled Heat and Moisture Transport in Three-Layered Composite 196
8.6 Application of PIEs and PIETOOLS for Estimating Moisture Content 197
8.7 Closing Remarks 201

VI Epilogue **203**

9 Conclusions and Future Outlook **205**

9.1 Viewpoint of the Thesis 206

9.2 Part I: A Framework to Build Thermo-Fluidic Models for the Digital Twin 206

9.3 Part II: Computation with Lumping for Controlling Thermo-Fluidic Process 208

9.4 Part III: Computation with Approximation for Controlling Diffusive Thermo-Fluidic Process 209

9.5 Part IV: Computation with PIEs for Controlling Thermo-Fluidic Process 211

9.6 Part V: Applications to Industrial Benchmarks 213

9.7 Answer to the Research Question 214

Bibliography **217**

List of Symbols and Notation **225**

List of Abbreviations **227**

Acknowledgments **229**

About the Author **233**

Part I

Prologue

1 CHAPTER

Introduction

Digitization is changing the world in many aspects. The industrial sector is one of the recipients of the digital revolution that allows the consumer market to transcend towards a cheaper, faster, more efficient and environmentally more sustainable future. This thesis explores how a digital twin can harness the power of digital computers and contribute to the future product development process in an industrial enterprise. The digital twin provides a virtual representation of a physical asset in an industrial process. In many physical assets, such as inkjet printers, thermo-fluidic processes often limit the asset's best achievable performance. This chapter motivates how building a digital twin can leverage future product development and achieve improved performance in the presence of thermo-fluidic processes. After a detailed discussion on digital twins and thermo-fluidic processes, this chapter lays down the fundamental research objectives and scientific contributions of this thesis.

Outline

1.1 Digitization of Industrial Processes **4**
1.2 Functionalities and Features of a Digital Twin **5**
1.3 Application: Inkjet Printing **7**
1.4 Building a Digital Twin for DoD Inkjet Printer **10**
1.5 Research Objective **13**
1.6 Subproblems and Approaches to Solve Them **13**
1.7 Organization of the thesis **19**

1.1 Digitization of Industrial Processes

In the advent of the 18th century, the first industrial revolution started when machine tools and mechanized factories were introduced to aid in producing goods. Three centuries later, through the era of electricity, electronic devices, computers and automation, industries have now entered the phase of the fourth revolution (commonly known as industry 4.0). Thanks to the world wide web, the abundance of data and artificial intelligence, industries are experiencing a paradigm shift in the way commercial assets are conceptualized, manufactured, monitored, troubleshot and delivered to end-users. A crucial aspect of industry 4.0 is the usage of digital computers and software as an enabling technology during the entire life cycle of the asset (see [61]). From the very initial phase of conceptualization to its sale in the market, the development process is leveraged by computer-aided tools. This juxtaposition of computer tools and connectivity is creating a digital transformation on industries to produce goods in a smarter, safer, cleaner and more efficient manner.

With increasing competition in the consumer market, industries are striving for better design, improved performance, higher efficiency of a product or a service; however, at a faster rate and lower retail price. As a result, the time window between the concept phase to delivery and service for customers is becoming smaller. In addition to that, for environmental reasons, factories and fabrication laboratories have to be more sustainable and recyclable with almost zero waste. The digitization and virtualization of industries have stemmed from this urgency of reducing costs, time spans and carbon footprint by transferring the entire development cycle to a computer-aided digital environment.

Digital Twin: Future of Digitized Industries

A *digital twin* is a virtual platform dedicated to an industrial process or a product for simulating its design, operation, functionality and life cycle by using a digital computer and software. The framework of digital twins helps to bring an industrial process out of the four walls of a laboratory to perform experiments, modifications, analysis, prediction and diagnostics without requiring its physical form. As a digital twin has the potential to transform a large part of the product development completely virtual, a great deal of physical equipment and time for manual experiments can be avoided for research and development. Moreover, in case of unexpected problems at any stage of the product cycle, the response time to resolve the design, optimization or technical problems can significantly be improved with minimum financial repercussion.

Potential Impact of Digital Twins for the Future Industries

This thesis identifies three key areas where digital twin can be instrumental in bringing the next revolution in the industrial sector:

1. *Improving performance*: The digital twin allows one to explore all potential scenarios that the industrial process may undergo in its life-cycle and provide the guarantee of its performance. This is a crucial factor in obtaining legal certification under mission-critical operations. Moreover, one can also predict the possible occurrence of failures by utilizing the data collected over the previous life cycles of the same product and prescribe measures to avoid them beforehand.
2. *Faster productivity*: By utilizing the connectivity of the digital twin via the world wide web, engineers can collaborate and perform different tasks at the same time on a single digital twin utilizing its different functionalities. As a result, the probationary phase of conceptualization and development of the first prototype can be significantly shortened.
3. *Mass customization and accessibility*: Future production is moving towards a new era where customers dictate many aspects of the functionality and design of the product they intend to purchase. The digital twin has the potential to leverage such personalized customization by means of allowing end-users to interact with the design process directly.

1.2 Functionalities and Features of a Digital Twin

Computer scientist David Gelernter has claimed that a digital twin can 'put the universe in a shoebox' (the interested readers may find it inspiring to read the visionary monograph by [38]). By merely being a program or software on a desk, the digital twin has infinite potential and endless scope in every avenue of technological development. In [45], a detailed and up-to-date survey is provided about the core functionalities and characteristics of the digital twin in various sectors of industry. It has been shown that the core architecture of a digital twin typically includes a multitude of functionalities that are interlinked with each other and these functionalities can be customizable based on industry-specific requirements. In [54], the survey on the digital twin's architecture and functionalities are restricted to the manufacturing process.

Functionalities a Digital Twin Must Have

Despite the depth and breadth of a digital twin, from a commercial standpoint, the core functionalities of a digital twin must include the following aspects:

1. *Performance analysis and process optimization*: Main functionality of the digital twin is to provide a quantitative measure on how a physical asset or a specific part of the industrial process perform in a real-world environment. At the same time, to be ahead of the competition, the virtual framework of a digital twin can be used to further optimize various aspects of the asset to obtain better performance. In this way, the asset's productivity and performance can reach the target goal.

2. *Prediction and visualization of product's life cycle*: Another functionality of the digital twin is to predict how a physical asset or a specific part of the industrial process evolves during the entire life cycle. Understanding and visualizing the assets evolution enables better product design and effective management of the supply-chain.

3. *Monitoring and diagnosis*: During usage of the physical asset by an end-user, one functionality of the digital twin is to monitor its performance by means of collecting data from the physical asset. By using this data, diagnostic tools can be developed that help to identify and resolve faults (even predicting future occurrences of fault to prevent them from taking place). Such predictive analysis helps in faster and cheaper troubleshooting the product.

4. *Conceptualizing future design*: The digital twin provides a virtual framework to explore novel designs and operations of the physical asset that may appear in future. Such virtual conceptualization can be rigorously tested and validated in the digital twin before it is built in its physical form; hence, it can significantly reduce the time and money. Moreover, owing to its virtual presence, the digital twin can allow the customers to be directly involved in the product development process.

Qualities a Digital Twin Must Have

With wide variations in functionalities and applications, the underlying framework of a digital twin is multifaceted. Hence, there are four features a digital twin should facilitate:

1. *Flexibility*: At any stage, in a digital twin, the virtual representation of the physical asset or process should be flexible so that a user can make adjustments on it at any stage of the development cycle.

2. *Versatility*: As the functionalities of a digital twin are quite diverse, it has to be versatile accommodating all the functionalities under software tool that the digital twin represents.

3. *Modularity*: The digital twin has to be modular so that, in future, when a new component or service is added to the physical asset or process, the digital twin is modular to adapt its architecture accordingly.

4. *Integrebility*: The entire enterprise of a commercial industry consists of numerous physical assets and processes. A digital twin that is dedicated to a specific asset or process has to be meaningfully integrable to the industry's entire operational pipeline.

The discussion in this section leads to Figure 1.1 that embodies the crucial aspects concerning the development of a digital twin.

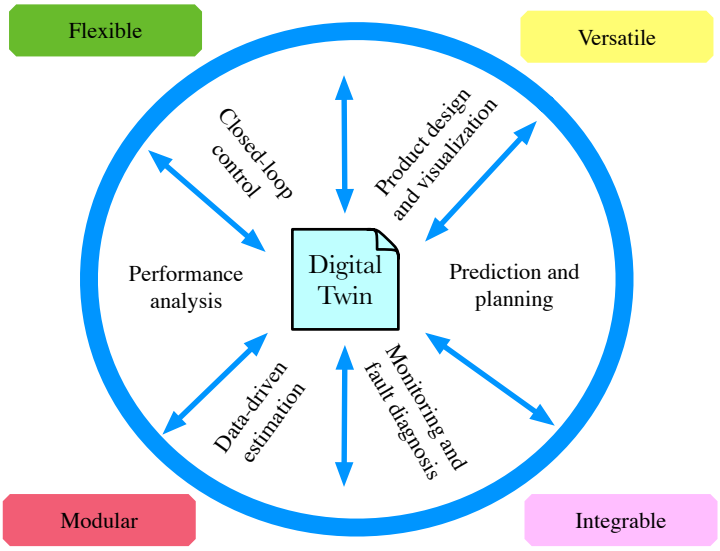


Figure 1.1: Core functionalities of a digital twin.

1.3 Application: Inkjet Printing

From the Gutenberg Press back in the fifteenth century till today, the technology associated with printing has preserved the intellectual, social, historical, economical and cultural contributions of humankind. Perhaps, the printing industry is one of the few technological sectors that has witnessed the entire evolution of the industry. From the manual press machines to the recent Drop on Demand (DoD) digital inkjet technology, the printing industry has always evolved and adapted to the newer technological paradigms and become faster, more cost-effective and energy-efficient.

Lately, digital technologies have a profound impact on making the outreach of printing beyond books and journals on sheets of paper. By harnessing the power of computers and automation, digital printing has also attracted versatile applications. By using its non-contact and additive nature, inkjet printing is now applicable to a wide range of materials including polymers and metals and also various media like textile, wood, and circuit boards [87]. Moreover, the process of printing is now completely automated; the user defines a print-job (a set of to-be-printed images), and the digital printer is responsible for automatically printing the images on a solid medium in the right sequence, at the right position, and with the best possible print quality. Thanks to digital printing, we are now gradually transcending 'from the world of paper printing to printing the world'.

Two Components of DoD Inkjet Printing

One can consider the process of printing as the physical integration of a solid medium (e.g. sheets of paper, cardboard, etc.) and liquid material (droplet of ink).

Such integration between solid material and liquid occurs in two separate stages in a printer. These two stages are known as a) jetting and b) fixation, as depicted in Figure 1.2.

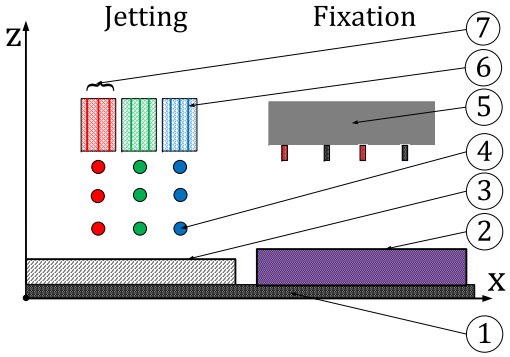


Figure 1.2: Two stages of a printing process. (1) Support. (2) Printed product. (3) Print medium. (4) Droplet of liquid material. (5) Actuators for fixation unit. (6) Nozzles for jetting. (7) Printhead.

- *Jetting unit*: During the process of jetting, liquid droplets at a pre-specified temperature, volume and jetting speed are deposited from a set of nozzles at the printhead on the solid material (known as a print medium). Deposition of liquid droplets solely depends on the specifics of the image that a user decides to print and this is known as the DoD mechanism. Figure 1.3 shows the simplified description of one printhead. The flow of

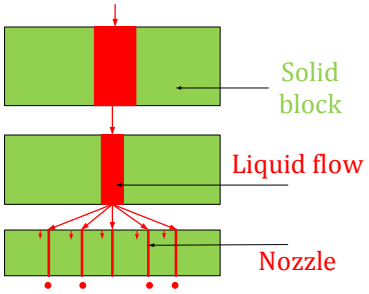


Figure 1.3: Schematics of one printhead.

liquid is from top to bottom. The lowest level consists of a set of vertical nozzles. Every nozzle is equipped with a piezoelectric actuator that produces a pressure wave in the printhead chamber so as to jet a drop of ink towards the medium. Based on the to-be-printed image, once a voltage is applied, the piezoelectric actuator ejects a droplet of liquid from the nozzle and deposits it on the print medium.

- *Fixation unit*: Once an image is printed on the print medium, the printed product gets dried (or 'fixed') in a fixation unit. A set of heater and air

impingement units are used to extract excess moisture from the individual freshly printed medium. The fixation unit is depicted in Figure 1.4.

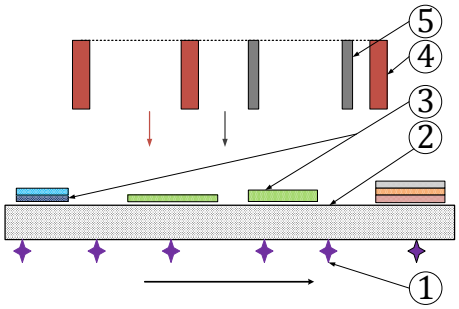


Figure 1.4: Overview of the fixation unit. (1) Temperature sensors. (2) Support (solid platform). (3) Printed products. (4) Heating unit. (5) Air impingement unit.

Four unresolved challenges in DoD inkjet printing

Printer manufactures are facing a critical period ahead. On environmental grounds, enterprises are moving away from the usage of paper-based printing (which has always been the primary market for printing). As a result, printer manufacturers must explore new application domains where printing can establish a new market. This means that the printer's architecture, design, and performance requirements may change and this change must reach customer imminently. Moreover, customers constantly demand better print quality (defined by how accurately the printed product resembles the image). Hence, printing technology must cope with ever-increasing demands for improved print quality, as well as be flexible to accommodate new designs and functionalities. In this context, the printing process, as a combination of jetting and fixation, imposes the following challenges as open problems that will further enhance print quality as well as make printing technology prepare for future changes in the market.

1. Both jetting and fixation processes can be affected by unknown inputs in terms of ambient temperature, movement of the printers, relative humidity etc. The problem is that:

   there does not exist any tool to verify how these disturbances affect the performance and quality of printing.

2. For optimal drying (fixation), the moisture content of the printed product is a crucial factor. The intensity of dryness (fixation) depends on the amount of moisture in the printed medium. Also, depending on the type of medium and the print job, the moisture level in a printed medium varies over location and time during the fixation process. However, there is no dedicated sensor for measuring moisture content for every individual printed product. This

makes the entire fixation process inherently challenging to control. And, the question remains open that:

how to estimate the moisture content of every individual printed product without installing any additional sensors in the fixation unit.

3. The temperature of the liquid ink is not consistent at all times during jetting. The temperature fluctuation of droplets causes its physiochemical properties (these properties are viscosity, surface tension, temperature, and the speed of sound inside the nozzle) deviate from desired values. This causes deterioration of print quality. Hence, the challenge remains unresolved:

   how to maintain a consistent temperature at every droplet of liquid during jetting without installing any additional sensors and actuators at the printhead.

4. To cope with the ever-increasing demands on throughput, in the future design, the number of printed products a printer can print within a given time span must increase. For instance, in future, design of the printhead may accommodate an arbitrarily large number of nozzles to ensure more substantial numbers of jetted droplets to be deposited on the medium at the same instant. Hence, the challenge is that:

   the state of the art for design, simulation, process optimizations and diagnosis are not readily adaptable if the number of nozzles changes.

1.4 Building a Digital Twin for DoD Inkjet Printer

A digital twin, as the virtual representation of a printer's functionality and its life cycle, can have a significant impact on today's printing industry. From designing a new printer or new equipment and emulating the dynamic changes of a printer (or a specific part) over its life cycle to optimizing the print quality and diagnosing faults, a digital twin can perform all these tasks in a virtual environment without utilizing a physical laboratory. For example, the challenges mentioned in the previous section cover a wide variety of domains encompassing design, estimation, control, performance analysis, diagnosis and monitoring. If a digital twin is considered, it can offer a flexible, versatile, modular and integrable tool to tackle every one of these problems and potentially prescribe solutions under a unique framework. In this way, by utilizing the digital twin, printer manufacturer can deliver to customers a faster, cheaper and greener printer with improved performance and print quality.

Thermo-Fluidic Processes in Inkjet Printing

As the applicability of the to-be-built digital twin is quite diverse, the process of jetting and fixation provides a generic description of the DoD inkjet printing process without concentrating on any application-specific scenario. Therefore, the digital twin of an inkjet printer must represent the processes of jetting and

fixation. In both the processes, a common aspect is the role of interaction among fluid substances (liquid or gas) and solid substances. As solids and fluids are distinct states of matter, their physical properties are different, so as the influence of external thermal energy on them. For example, during jetting, the solid structure of the printhead and liquid ink interact over their common boundaries. Moreover, if not all the nozzles are simultaneously used for jetting, a gradient in liquid temperature takes place among jetting and non-jetting nozzles. During fixation, by applying heat and air flux, the evaporation of excess moisture takes out additional heat from the printed medium. As a result, along with the moisture, the temperature of printed product changes. Such interaction between solid and fluid substances under the influence of external thermal energy (input heat flux) is referred as a *thermo-fluidic process* (see Figure 1.5). And, the thermo-fluidic process is key to describe DoD inkjet printing and affect the formation of printed products. Therefore, the digital twin of a DoD inkjet printer must incorporate thermo-fluidic processes.

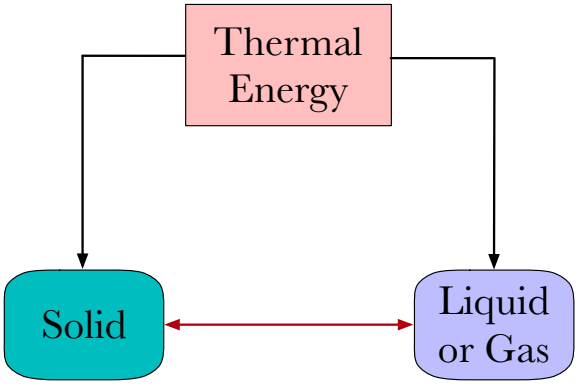


Figure 1.5: General description of thermo-fluidic processes.

Building a digital twin based on thermo-fluidic processes

If one aims to build a digital twin for the assets that are governed by thermo-fluidic processes, there are a few fundamental physical properties and characteristics that need to be taken into account. These are

1. *Spatially interconnected structure*: The physical layout of the printing process as an example, the printheads and the fixation unit demonstrate a spatially interconnected structure consisting of thermo-fluidic processes print materials, support, air, heat and liquid that are distributed over a spatial configuration and physically interconnected.

2. *Spatio-temporal dynamics*: As an immediate consequence, systems governed by thermo-fluidic processes exhibit dynamics that evolve both over space and time.

3. *Lack of sensors and actuators*: Because of lack of space, physical limitations on the printhead's dimension, and inhibited costs, there is a persistent lack of

sensor information and actuating power to control thermo-fluidic processes in the printhead.

Role of models in building a digital twin for thermo-fluidic processes

To virtually represent the thermo-fluidic processes, the digital twin needs to rely on a model which is a replica of the jetting and fixation as well as which describes its underlying phenomena and working mechanisms. As a result, building a model stands out to be the core of a digital twin. These models typically involve mathematical equations that are either derived from physical laws (e.g. conservation of mass, momentum and energy) or identified using the data collected from previous life cycles of the process. The model must also exhibit the underlying dynamical behavior of the process while being relevant over time, i.e. be able to capture the thermo-fluidic processe's evolution over the entire life cycle.

Role of systems and control in building a digital twin for thermo-fluidic processes

With the increasing complexity of the design, thermo-fluidic processes often result from a complex architecture of interconnected components or systems of systems. The field of systems and control plays a significant role in designing, optimizing and developing a physical entity that constitutes of interrelated and interdependent parts. Moreover, the digital twin must facilitate improving all aspects related to the design, optimization, estimation, control, virtual testing, prediction, maintenance and fault diagnosis of the inkjet printing process. In fact, the key difference of a digital twin, in comparison to a simulator, is such diverse functionalities that are already enabled in the core operation of a digital twin.

Role of computation in building a digital twin for thermo-fluidic processes

The role of computation is ubiquitous in any industrial process. In particular, to build a digital twin, computation is an enabling factor for its functionalities. The computation is involved not only in simulating the model that governs the thermo-fluidic processes or predicting its evolution over space and time in future, but also in providing quantitative certification to the processe's performance and in optimizing the performance to obtain the best achievable level. As a result, computation is crucial in offering a market valuation of the printers which a commercial enterprise, such as a printing manufacturing company, strongly demands.

1.5 Research Objective

The task of the digital twin is to address all aspects related to design, optimization, estimation, control, virtual testing, prediction, maintenance of an industrial asset that is governed by the thermo-fluidic processes. Moreover, unlike a conventional simulator, the digital twin has well-defined objectives in terms of analyzing the processe's performance under unforeseen disturbances, estimating unmeasured physical parameters (spatially varying) as well as physical quantities (spatio-temporally varying), enabling the synthesis of controllers for the thermo-fluidic processes. These functionalities must be achieved without installing any additional sensors or actuators while ensuring that the required performance criterion is met. Moreover, the digital twin must be flexible, versatile, modular and integrable. To this end, this thesis seeks an answer to the following question:

How to develop a digital twin for an industrial asset that is governed by thermo-fluidic processes and guarantee that the asset achieves a predefined performance without adding new sensors or actuators?

- The usage and functionalities of a digital twin are wider than the scope of inkjet printing. As a result, attention is given towards making the design procedure of building a digital twin generic such that other applications of thermo-fluidic processes can also be adopted.
- *Addition of new sensors or actuators are restricted*. The digital twin can be used to investigate novel techniques for controlling the thermo-fluidic processes that do not require new sensors or actuators. This aspect again highlights the importance of versatility in a digital twin.
- The *performance metric* is typically *predefined*. For instance, in DoD inkjet printing, one may choose print quality as the performance metric and it can be quantified based on the underlying thermo-fluidic processes. For example, in jetting, a suitable performance metric would be to keep the temperature difference among jetting and non-jetting nozzles below a certain value. In fixation, the performance metric can be the amplification factor at which an external disturbance (in terms of humidity, or heat flux) affects the estimation of moisture for every individual printed medium.

1.6 Subproblems and Approaches to Solve Them

In order to achieve the formulated research goal, a set of relevant subproblems is formulated in this section. Specific theoretical approach and practical considerations are taken to solve every individual subproblem. The solutions of all subproblems can then be combined, and collectively, they can satisfy the

ultimate goal of this research. In the subsequent paragraphs, the relevant subproblems are formulated, corresponding challenges are found and approaches are summarized to circumvent the challenges and solve each subproblem.

A framework for modeling and representing thermo-fluidic processes

The first subproblem is to answer the following question:

How to develop a modeling framework that allows users to describe and represent a spatially interconnected thermo-fluidic process such that the model is integrable to the core functionalities of the digital twin?

The answer to this question is given in Chapter 2. Here two aspects of the thermo-fluidic processes are considered: a) spatial interconnection among different components with distinct physical properties, b) spatio-temporal evolution of the thermo-fluidic dynamics. As a result, the governing model has the following features:

1. The governing dynamics may involve partial differential equations (PDEs) involving physical quantities that are functions of space as well as time.
2. The governing dynamics may also involve ordinary differential equations (ODEs) involving physical quantities that are only functions of time.
3. Because of the spatially interconnected structures, there are spatial interconnections among PDEs or ODEs or among PDEs and ODEs.
4. The disturbances or inputs typically occur at the boundaries of a spatial configuration.

To ensure that these four properties of the thermo-fluidic processes are preserved in defining the model of the digital twin, a graph-theoretic modeling framework is used to describe the spatial interconnection of PDEs and ODEs under arbitrary topology. The interconnections among various components are modularized, and specific constraints are formulated that preserves the energy exchange across the spatial domain between two mutually interacting processes. Exploiting the graph structure, incorporating or removing new components as a part of the graph as well as accommodating new features for a specific component are performed efficiently.

Once the underlying thermo-fluidic model is defined, the core functionalities of the digital twin are built on the basis of this model. Moreover, every functionality involves the computation and evaluation of thermo-fluidic processes. For building any functionality regarding simulation, analysis, prediction, diagnosis, estimation or control of thermo-fluidic processes, the spatial interconnection among PDEs and ODEs allow one to use any one the three computational approaches, depicted in Figure 1.6.

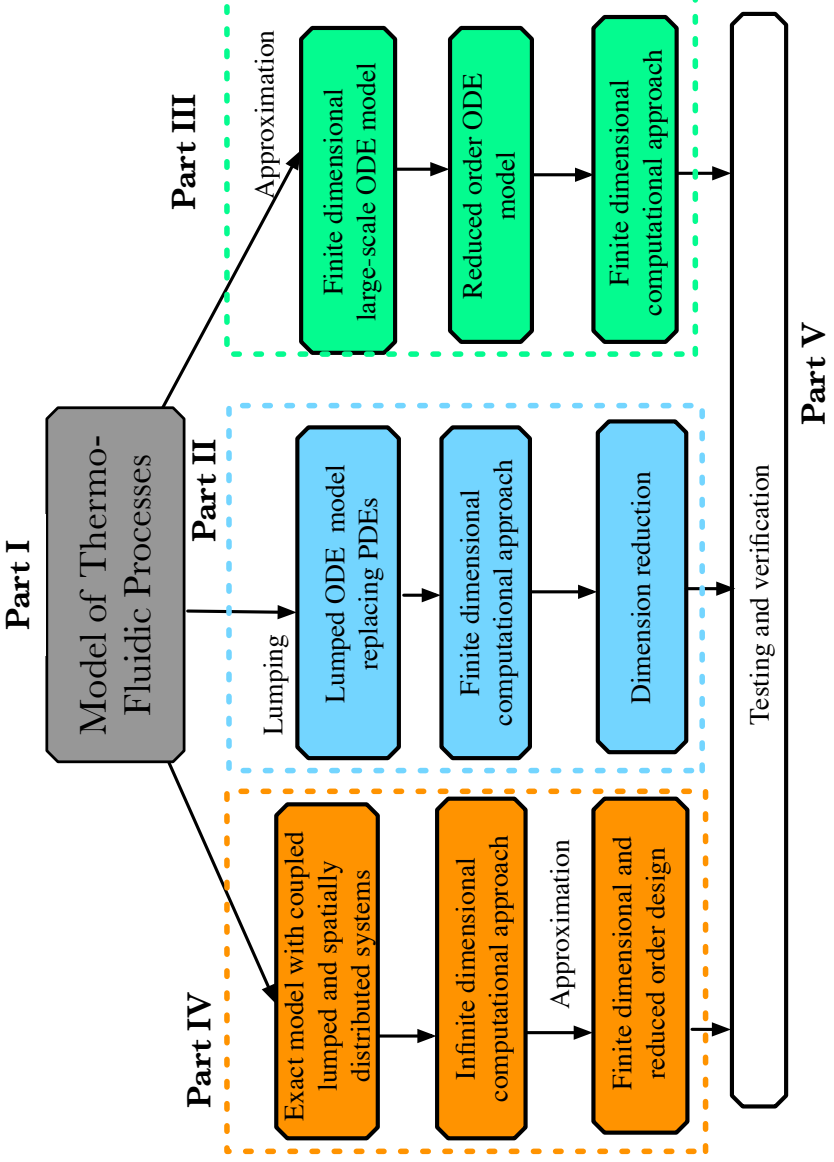


Figure 1.6: Three distinct computational approaches to build functionalities on the digital twin based on thermo-fluidic processes.

Part II: Computation with Lumping

This subproblem amounts to answering the following question:

How to utilize a lumping technique to represent, monitor and control spatially interconnected thermo-fluidic processes without adding new sensors and actuators while meeting user-specific demand?

The answer to this question is given in Chapter 3. In many cases, the dimensions and physical properties underlying components allow neglecting the spatial distribution of physical quantities. This means that one can describe them as ODEs. As a result, the entire thermo-fluidic process can be described by a lumped network of systems governed by ODEs. However, the lumping approach may not always provide sufficiently accurate behavior for all the components. To this end, in this thesis, the digital twin is made flexible in such a way that, at any stage of the development, a specific lumped component can be modified to improve accuracy.

As installing additional sensors or actuators is strictly prohibited, the diagnosis and monitoring functionalities of the digital twin are enabled by presenting a soft-sensor. Similar to inkjet printheads, often, liquid components are already equipped with piezoelectric elements. By using the self-sensing capability of the piezoelectric elements and the lumped model, a real-time algorithm is developed that estimates the temperature of the liquid component without adding new sensors. This estimation mechanism and the underlying lumped model enables the digital twin to synthesize an optimal controller that predicts the change in the dynamical behavior of the process and controls it to achieve a predefined performance criterion.

- This part of the research is based on the following work:

  —— A. Das, M. Princen, M. Shokrpour, A. Khalate, S. Weiland (2019). Soft Sensing Based In Situ Control of Thermo-Fluidic Processes in DoD Inkjet Printing. Accepted for publication in IEEE Transactions of Control Systems Technology, 2019. (DOI of the preprint: 10.20944/preprints202003.0355.v2).

Part III: Computation with Approximation

This subproblem amounts to answering the following question:

How to numerically approximate a spatially distributed diffusive thermo-fluidic process while satisfying the boundary conditions in the presence of boundary inputs and make the approximated model useful for performing various functionalities of the digital twin?

The answer to this question is found in Chapter 4 and 5. Here, an important subclass of thermo-fluidic processes is considered that are governed by diffusion-transport-reaction of coupled mass and temperature. In this thesis, they are defined as *diffusive thermo-fluidic process*. The model of a diffusive thermo-fluidic process is restricted to be a set of spatially interconnected PDEs

that are mutually interconnecting over boundaries. A computational approximation technique is presented that explicitly preserves the exchange of energy across the boundaries, as well as the presence of inputs at the boundaries. Once the approximation technique reduces the spatially interconnected PDEs to a set of coupled ODEs, the functionalities developed in Part II (lumping-then-design) are directly applicable.

In many applications of diffusive thermo-fluidic processes, the physical parameters of a particular component are often unknown in terms of their values as well as their spatial profile. A data-driven algorithm is presented to estimate such spatially varying physical parameters based on the available data from a set of location-specific sensors. In this way, measured data that are being accumulated over the previous life cycles of the thermo-fluidic processes allow the digital twin to perform validation of the underlying mode or improve the model if necessary.

- This part of the research is based on the following works:
  - —— A. Das, L. Iapichino, S. Weiland (2018). Model Approximation of Thermo-Fluidic Diffusion Processes in Spatially Interconnected Structures. Proc. 2018 European Control Conference (ECC), Limassol, pp. 2653-2658, DOI: 10.23919/ECC.2018.8550146.
  - —— A. Das, M. Van Berkel, S. Weiland (2019). Frequency Domain Estimation of Spatially Varying Parameters in Heat and Mass Transport. Proc. 2019 IEEE American Control Conference (ACC), Philadelphia, pp. 600-605, DOI: 10.23919/ACC.2019.8814465,.
  - —— A. Das, S. Weiland (2020). Modeling and Approximation of Networked Thermo-Fluidic Processes Under Arbitrary Spatial Topology. To be submitted for publication in International Journal of Systems Science.

Part IV: Computation with PIEs

This subproblem amounts to answering the following question:

How to develop a computational framework that allows to analyze and synthesize estimator based optimal controllers directly on the spatially interconnected thermo-fluidic processes without depending on lumping or approximation techniques while providing a quantifiable performance guarantee?

The answer to this question is found in Chapter 6. Here, the goal is to directly utilize the spatio-temporal dynamics with no prior approximation or lumping. An alternative representation of spatially interconnected thermo-fluidic processes is derived by using a class of operators called *Partial Integral (PI) operators*. These PI operators share many similarities with matrices. By using the PI operators, an equivalent representation of thermo-fluidic processes is determined that are called *Partial Integral Equations (PIEs)*. The PIE representation enables to develop a computational framework that directly utilizes the original

thermo-fluidic model for analysis, estimation and control purposes. By using the proposed representation, stability, input-output properties are tested for the thermo-fluidic processes. Moreover, an estimator is synthesized or thermo-fluidic processes that utilize finite-dimensional measured data to provide an estimate of the unmeasured physical quantities (spatio-temporally varying) while guaranteeing an optimal performance bound in terms of worst-case disturbance amplification in the estimation error.

- This part of the research is based on the following works:
  - —— A. Das, S. Shivakumar, M. Peet, S. Weiland (2020). Robust Analysis of Uncertain ODE-PDE Systems Using PI Multipliers, PIEs and LPIs. Accepted for presentation at the 2020 IEEE Conference on Decision and Control (CDC), Jeju Island.
  - —— S.Shivakumar, A. Das, M. Peet (2020). PIETOOLS: A Matlab Toolbox for Manipulation and Optimization of Partial Integral Operators. Proc. 2020 IEEE American Control Conference (ACC), Denver. DOI: 10.23919/ACC45564.2020.9147712
  - —— A. Das, S. Shivakumar, S. Weiland, M. Peet (2019). $\mathcal{H}_\infty$ Optimal Estimation for Linear Coupled PDE Systems. Proc. 2019 IEEE Conference on Decision and Control (CDC), Nice, 2019, pp. 262-267. DOI: 10.1109/CDC40024.2019.9029595
  - —— S. Shivakumar, A. Das, S. Weiland, M. Peet (2019). A Generalized LMI Formulation for Input-Output Analysis of Linear Systems of ODEs Coupled with PDEs. Proc. 2019 IEEE Conference on Decision and Control (CDC), Nice, 2019, pp. 280-285, DOI: 10.1109/CDC40024.2019.9030224
  - —— M. Peet, S. Shivakumar, A. Das, S. Weiland (2019). Discussion Paper: A New Mathematical Framework for Representation and Analysis of Coupled PDEs. Proc. 2019 IFAC Workshop on Control of Systems Governed by Partial Differential Equations (CPDE), Oaxaca, pp. 132-137. https://doi.org/10.1016/j.ifacol.2019.08.023
  - —— S. Shivakumar, A. Das, S. Weiland, M. Peet (2020). PIE Representation of ODEs Coupled with Higher Order PDE Systems. Submitted for publication in IEEE Transactions on Automatic Control.
  - —— A. Das, S. Shivakumar, S. Weiland, M. Peet (2020). $\mathcal{H}_\infty$ Optimal State Estimation for Coupled ODE-PDE Systems Using LPIs. To be submitted for publication in Automatica.

Application to industrial benchmarks

This subproblem amounts to answering the following question:

How to apply the developed methodologies in building a digital twin for controlling the jetting and fixation process in a DoD inkjet printer ?

The answer to this question is in Chapter 7 and Chapter 8.

1. The functionalities of the digital twin are utilized to develop a self-sensing based in situ controller that maintains a consistent liquid temperature at every individual nozzle without additional sensors and actuators.
2. The functionalities of the digital twin are utilized to develop an $\mathcal{H}_\infty$ optimal estimator that estimates the average moisture level of a printed product in fixation process only based on the measurement from a thermocouple below the solid support.

- This part of the research is based on the following work:

—— A. Das, M. Princen, M. Shokrpour, A. Khalate, S. Weiland (2019). Soft Sensing Based In Situ Control of Thermo-Fluidic Processes in DoD Inkjet Printing. Accepted for publication in IEEE Transactions of Control Systems Technology, 2019. (DOI of the preprint: 10.20944/preprints202003.0355.v2).

1.7 Organization of the thesis

The organization of the thesis is depicted in Figure 1.7.

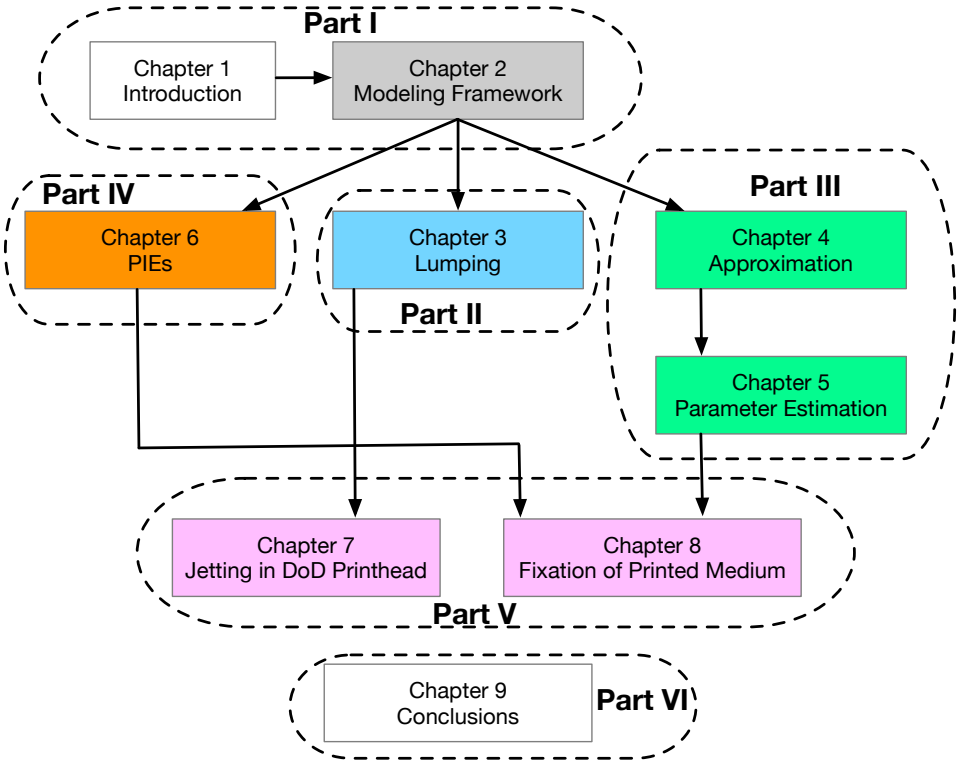


Figure 1.7: Organization of the thesis. Arrows indicate *read-before* relations. List of symbols is provided at the back of this thesis.

2 CHAPTER

Digital Twin's Framework to Model Thermo-Fluidic Processes

This chapter introduces a modeling framework that is in the core of a digital twin to describe and represent thermo-fluidic processes. To this end, a graph-theoretic framework is formulated as a collection of nodes and edges under a user-defined topology. Nodes are dynamical systems governed by either linear coupled partial differential equations or linear coupled ordinary differential equations. The edges define the interconnection among adjacent nodes by enforcing algebraic constraints on the signals that these mutually interconnected nodes share. Apart from the graph-theoretic definition, the digital twin's modeling framework also provides a few alternative representations of the thermo-fluidic models that are behaviorally equivalent. A user has the freedom to choose any one of those representations to describe the application-specific thermo-fluidic process and build a relevant functionality of the digital twin.

Outline

2.1 Introduction **22**
2.2 Abstract Definition of Spatially Interconnected Thermo-Fluidic Processes **22**
2.3 Building Thermo-Fluidic Model for the Digital Twin **24**
2.4 Equivalent Representations of Lumped Nodes **36**
2.5 Equivalent Representation of Spatially Distributed Nodes . . . **40**
2.6 Combined Representation of Spatially Distributed Nodes and Lumped Nodes **42**
2.7 Closing Remarks **46**

2.1 Introduction

The first part of building a digital twin is to represent the physical asset or the process virtually. Every engineering system can be viewed as an enhanced and goal-oriented application of physical laws of nature. Its virtual representation is typically abstracted by a mathematical model to provide an 'effective and unambiguous' description of these fundamental laws (c.f. [80]). In this thesis, the mathematical model offers a quantitative and qualitative description of the thermo-fluidic process. By definition, the model that describes thermo-fluidic processes must include the following aspects:

- Thermo-fluidic processes consist of multiple components that have distinct physical properties and are spatially interconnected.
- Some components are spatially distributed, and their governing dynamics evolve over space and time.
- Some components are lumped, and their governing dynamics only vary over time.
- Flexibility, versatility, modularity and integrability of a digital twin must reflect in the modeling framework.

Contribution of the Chapter

Incorporating multi-component, multi-physics, interconnected architecture of a physical asset while preserving the qualities of a digital twin is the primary objective of modeling thermo-fluidic processes and the main contribution of this chapter. By exploiting the spatial interconnection among different components, a graph-theoretic framework is presented to model and upscale the thermo-fluidic processes. Here, the graph consists of nodes and edges. A node describes the dynamical behavior of a locally defined thermo-fluidic process in a specific component. An edge defines the mutual interconnection of two adjacent components. Based on a specific use case, the user specifies the nodes, the edges, and topology on how components are interconnected to each other.

2.2 Abstract Definition of Spatially Interconnected Thermo-Fluidic Processes

Graph theory is widely used to model large-scale distributed and networked systems. Large-scale systems, known as *Spatially Interconnected Systems*, are decomposed and represented in a graph-theoretic framework in [60]. In [32, 58], a particular attention is given to exploit state space approach in representing spatially interconnected systems. A similar framework is also applied to the parameter varying systems interconnected through time-varying, arbitrary and

directed interaction topology in [41]. In this chapter, the work of [32] is generalized for a spatially interconnected system that consists of both lumped models with temporally varying dynamics as well as spatially distributed models with spatio-temporally varying dynamics.

Definition 2.1 *(Spatially interconnected thermo-fluidic process)*

The definition of a spatially interconnected thermo-fluidic process consists of following three objects.

D.1 Graph: A finite graph consists of a set of nodes and edges. It is defined by the following quadruple

$$\mathfrak{G} = (\mathfrak{N},\ \mathfrak{E},\ A, \mathbb{T}). \tag{2.1}$$

- *The time axis is denoted by the set* $\mathbb{T} \subseteq \mathbb{R}$.
- $\mathfrak{N}$ *is a set of all the nodes, and the set's cardinality is denoted by a scalar* $m > 0$. *Each object* $\mathfrak{N}_i$ *in the set* $\mathfrak{N}$, $i \in \mathbb{N}_{[1,m]}$, *is a node or component of the thermo-fluidic process with a specific physical property.*
- $A \in \mathbb{N}^{m\times m}$ *is an adjacency matrix that has either zero or one as its elements depending on whether the nodes* $\mathfrak{N}_i$ *and* $\mathfrak{N}_j$ *are physically attached or not. Its entries are*

$$A_{i,j} = \begin{cases} 1, & \text{if } \mathfrak{N}_i \text{ is connected to } \mathfrak{N}_j \\ 0, & \text{otherwise.} \end{cases}$$

- *The set of edges* $\mathfrak{E} = \{\mathfrak{E}_{i,j} \mid \text{for all } (i,j) \text{ with } A_{i,j} = 1\}$ *describes the interconnection between a particular node and its neighboring nodes.*

D.2 Nodes: Every individual node $\mathfrak{N}_i \in \mathfrak{N}$ *is defined by the following sextuple:*

$$\mathfrak{N}_i = (\mathbb{X}_i, \mathbb{X}_i^{bc}, \mathscr{S}_i, \mathscr{P}_i, \mathscr{P}_i^{bc}, \mathscr{B}_i). \tag{2.2}$$

- *The geometric configuration of every node* $\mathfrak{N}_i$ *is characterized by a bounded set* $\mathbb{X}_i \subset \mathbb{R}^d$. *Its boundary is defined by the limit-points as*

$$\mathbb{X}_i^{bc} = \mathbb{X}_i \setminus \mathbb{X}_i^{\circ},$$

  where $\mathbb{X}^{\circ}$ *denotes the interior of* $\mathbb{X}_i$. *The dynamics of* $\mathfrak{N}_i$ *is confined in the spatial domain* $\mathbb{X}_i^{\circ}$ *and is constrained by the boundary conditions in* $\mathbb{X}_i^{bc}$.
- *There are four categories of signals that are associated to a node* $\mathfrak{N}_i$. *These signals belong to a subspace* $\mathscr{S}_i$ *and consist of*
  - *(a) The internal state variables which are either multi-variable functions;* $\mathbf{x}_i : \mathbb{X}_i \times \mathbb{T} \to \mathscr{X}_i$ *or single variable functions* $x_i : \mathbb{T} \to \mathscr{X}_i$, *where* $\mathscr{X}_i$ *is the state space.*
  - *(b) The control signals which are single-variable functions;* $c_i : \mathbb{T} \to \mathscr{C}_i$, *where* $\mathscr{C}_i$ *is the control signal space.*

(c) *The parameter valued signals which are single variable functions* $d_i : \mathbb{T} \to \mathscr{D}_i$*, where* $\mathscr{D}_i$ *is the parameter signal space.*

(d) *The interconnection signals which are single variable functions* $l_i : \mathbb{T} \to \mathscr{L}_i$*, where* $\mathscr{L}_i$ *is the interconnection signal space.*

As a result, $\mathscr{S}_i = \mathscr{X}_i^{\mathbb{X}_i \times \mathbb{T}} \times \mathscr{C}_i^{\mathbb{T}} \times \mathscr{D}_i^{\mathbb{T}} \times \mathcal{L}_i^{\mathbb{T}}$.

- *The behavior of* $\mathfrak{N}_i$ *is captured by a relation among all the signals* $(x_i, \mathbf{x}_i, c_i, d_i, l_i)$ *and defines a subspace* $\mathscr{P}_i \subset \mathscr{S}_i$.
- *The behavior* $\mathscr{P}_i^{\text{bc}}$ *describes the constraints that need to be imposed on the behavior* $\mathscr{P}_i$ *when defined on* $\mathbb{X}_i^{\text{bc}}$*. In particular,* $\mathscr{P}_i^{\text{bc}} \subset \mathscr{P}_i$.
- *The behavior the parameter valued signals* d_i *defines a subspace* $\mathscr{B}_i \subset (\mathscr{D}_i)^{\mathbb{T}}$.

D.3 Edges: Every individual edge $\mathfrak{E}_{i,j} \in \mathfrak{E}$ *is defined by the following triple:*

$$\mathfrak{E}_{i,j} = (\mathbb{X}_{i,j}^I, \mathscr{L}_{i,j}^I, \mathscr{M}_{i,j}). \tag{2.3}$$

- *Each* $\mathfrak{E}_{i,j}$ *describes the interconnection between two adjacent nodes (*$\mathfrak{N}_i$ *and* $\mathfrak{N}_j$*) on the their common boundary which is defined by* $\mathbb{X}_{i,j}^I \subseteq \mathbb{X}_i \cap \mathbb{X}_j$.
- *Wherever* $\mathfrak{N}_i$ *is interconnected to* $\mathfrak{N}_j$*, i.e.* $A_{i,j} = 1$*, there are interconnection signals associated with the edge. Let these signals be single variable functions* $(l_{i,j}, l_{j,i}) : \mathbb{T} \to \mathscr{L}_{i,j}^I$*, respectively. Additionally,* $l_i = \text{col}(l_{i,k})_{k \in \mathbb{I}_{i,j}}$ *with* $\mathbb{I}_{i,j} := \{j \mid A_{i,j} = 1\}$.
- *The relation among interconnection signals* $l_{i,j}$ *and* $l_{j,i}$ *defines a subspace* $\mathscr{M}_{i,j} \subset (\mathscr{L}_{i,j}^I)^{\mathbb{T}}$.

2.3 Building Thermo-Fluidic Model for the Digital Twin

The digital twin's model of thermo-fluidic processes is built on the basis of Definitions (D.1)-(D.3) whose specifications are given in the sequel.

2.3.1 Specifying Physical Asset's Topology

A finite graph defines a physical asset that consists of spatially interconnected components with known topology according to (D.1).

1. Each node represents a solid or fluid component. There is m number of components as a part of the asset's underlying thermo-fluidic processes.
2. Every individual node $\mathfrak{N}_i$ represents either a lumped model (involving thermo-fluidic processes that solely vary over time) or spatially distributed model (involving thermo-fluidic processes that vary over space and time).

To distinguish between lumped nodes from spatially distributed ones, a user may define the spatial domain $\mathbb{X}_i, \mathbb{X}_i^{\text{bc}}$ to be empty and, in that case, the specific node is automatically considered to be lumped.

Let $\mathbb{P}$ and $\mathbb{O}$ be the two sets of indices corresponding the spatially distributed nodes and lumped nodes respectively. Precisely, $\mathbb{P} := \{i \mid i \in \mathbb{N}_{[1,m]} \wedge \mathbb{X}_i \neq \emptyset\}$, $\mathbb{O} := \{j \mid j \in \mathbb{N}_{[1,m]} \wedge \mathbb{X}_j = \emptyset\}$.

3. The adjacency matrix $A \in \mathbb{R}^{m \times m}$ is defined based on the physical architecture of how various components interact with each other in the asset's thermo-fluidic process.

   There may be multiple identical nodes (in terms of their model and interaction with rest of the graph), e.g. the liquid nozzles in a printhead, or identically printed paper sheet on a fixation tray. In such cases, it is sufficient to specify their total numbers and the graph-theoretic model automatically replicates them.

4. All the signals are considered to be in either continuous time with $\mathbb{T} \subseteq [0, \infty)$ or discrete time with $\mathbb{T} := \{kt_d \mid k \in \mathbb{N} \cup \{0\}\}$, where the sampling period t_d is a fixed scalar.

2.3.2 Specifying Spatial Configuration

The spatial configuration in which a spatially distributed component is confined and the boundaries through which it interacts with its neighbors are defined as follows.

Spatial Domain

In this thesis, all the thermo-fluidic processes are considered to be in one spatial dimension. Therefore, the set $\mathbb{X}_i$ is either empty or an interval, $[a_i, b_i]$ for some $b_i > a_i$.

Boundary

In one dimensional domain, the boundaries are the extremum points. Hence, boundary of a spatially distributed node consists of two points, i.e., $\mathbb{X}_i^{\text{bc}} = \{a_i, b_i\}$. When $\mathbb{X}_i$ is empty, $\mathbb{X}_i^{\text{bc}}$ is empty as well.

Interconnection Domain

When $A_{i,j} = 1$, interconnections are between either two lumped nodes, or two spatially distributed nodes, or else one spatially distributed node and one lumped node. Hence, whenever $A_{i,j} = 1$, the interconnection domain $\mathbb{X}_{i,j}^I \subseteq \mathbb{X}_i \cap \mathbb{X}_j$ is of three kinds:

1. *Interconnection between two lumped nodes*: If $\mathbb{X}^I_{i,j}$ is empty, the interconnection is between two lumped nodes.
2. *Interconnection between two spatially distributed nodes*: If both $\mathbb{X}_i$ and $\mathbb{X}_j$ are non-empty, then the interconnection is between two spatially distributed nodes. Such interconnection takes place only via common boundary points between $\mathfrak{N}_i$ and $\mathfrak{N}_j$, i.e. $\mathbb{X}^I_{i,j} := \mathbb{X}^{\text{bc}}_i \cap \mathbb{X}^{\text{bc}}_j$. In other words, $\mathbb{X}^I_{i,j}$ consists of a single point, either $a_i = b_j$ or $b_i = a_j$.
3. *Interconnection between a lumped node and a spatially distributed node*: When either $\mathbb{X}_i$ or $\mathbb{X}_j$ is empty, the interconnection is between a lumped node and a spatially distributed node. In that case, interconnection domain is either on the spatial domain or on the boundary of the spatially distributed node. In that case $\mathbb{X}^I_{i,j}$ is left to be defined by the user.

2.3.3 Specifying Signals

The signals are either functions of space and time or functions of time alone depending on whether they are part of the lumped node or spatially distributed node. For two mutually interconnected nodes $\mathfrak{N}_i$ and $\mathfrak{N}_j$, the signals are defined as follows.

States

Depending on whether $\mathbb{X}_i$ is empty or not, the states are either spatio-temporal functions denoted by $\mathbf{x}_i$ or temporal functions denoted by x_i. They are defined as $\mathbf{x}_i : \mathbb{X}_i \times \mathbb{T} \to \mathbb{R}^{n^i_p}$ and $x_i : \mathbb{T} \to \mathbb{R}^{n^i_x}$. Hence, $\mathscr{X}_i = \mathbb{R}^{n^i_p}$ or $\mathscr{X}_i = \mathbb{R}^{n^i_x}$.

Moreover, the state variables $\mathbf{x}_i$ may have restrictions in terms of differentiability with respect to the independent variable s_i (up to second order). Hence, these states are partitioned as $\mathbf{x}_i := \text{col}(\mathbf{x}_{0i}, \mathbf{x}_{1i}, \mathbf{x}_{2i})$ with $\mathbf{x}_{ji} : [a_i, b_i] \times \mathbb{R}^+ \to \mathbb{R}^{n^i_j}$ $(j = 0, 1, 2)$, where $n^i_p = n^i_0 + n^i_1 + n^i_2$. Here, $\mathbf{x}_{0i}$ are with no restrictions on differentiability, states $\mathbf{x}_{1i}$ with restrictions of first order differentiability with respect to its independent variable $s_i \in \mathbb{X}_i$ and states $\mathbf{x}_{2i}$ with restrictions on second order differentiability with respect to its independent variable $s_i \in \mathbb{X}_i$.

The state variables typically describe the spatio-temporal or temporal evolution of specific physical quantities in a thermo-fluidic process (e.g. temperature, mass concentration, the flow of liquid) in the sense that the future behavior of the node's thermo-fluidic processes entirely depends on states together with its future inputs such as actuations or noise (which can be externally manipulated).

Control Signals

For every node $\mathfrak{N}_i$, the control signals c_i are typically classified into inputs and outputs. Irrespective of whether $\mathbb{X}_i$ is empty or not, there are four types of control signals and all of them are considered to be functions of time.

1. The control (to-be-manipulated) inputs are $u_i : \mathbb{T} \to \mathbb{R}^{n_u^i}$.
2. The regulated (to-be-controlled) outputs are $z_i : \mathbb{T} \to \mathbb{R}^{n_z^i}$.
3. The external disturbance inputs (they affect the node's dynamics and, sometimes, they are exogenous) are $w_i : \mathbb{T} \to \mathbb{R}^{n_w^i}$.
4. The measured outputs (using available sensors) are $y_i : \mathbb{T} \to \mathbb{R}^{n_y^i}$.

As a result, $c_i = \mathrm{col}\,(u_i, z_i, w_i, y_i)$ and $\mathscr{C}_i = \mathbb{R}^{n_u^i} \times \mathbb{R}^{n_z^i} \times \mathbb{R}^{n_w^i} \times \mathbb{R}^{n_y^i}$. As the control signals are finite dimensional, in a spatially distributed node, they can be invoked either inside the spatial domain or at the boundaries.

Parametric Signals

In a lumped node ($\mathbb{X}_i$ is empty), there is a provision for defining parameters or part of a user's specifications that may vary over time (e.g. the print demand, set by the user in jetting). The signals associated to such parametric variations in a lumped node are classified into inputs $p_i : \mathbb{T} \to \mathbb{R}^{n_p^i}$ and outputs $q_i : \times\mathbb{T} \to \mathbb{R}^{n_q^i}$. Hence, $d_i = \mathrm{col}\,(p_i, q_i)$, and $\mathscr{D}_i = \mathbb{R}^{n_p^i} \times \mathbb{R}^{n_q^i}$. It is assumed that, the parametric signals are not present in a spatially distributed node (when $\mathbb{X}_i$ is nonempty).

Interconnection Signals

Whenever $A_{i,j} = 1$, the interconnection signals between $\mathfrak{N}_i$ and $\mathfrak{N}_j$ are categorized into inputs and outputs. The interconnections signals are assumed to be functions of time. Based on the three types of interconnections, these signals are attributed as follows.

1. *Interconnection between two lumped nodes*: When $\mathbb{X}_{i,j}^I$ is empty, the interconnection between two lumped nodes is characterized by inputs $m_{i,j}^{xx} : \mathbb{T} \to \mathbb{R}^{n_{mxx}^{ij}}$ and outputs $n_{i,j}^{xx} : \mathbb{T} \to \mathbb{R}^{n_{nxx}^{ij}}$.
2. *Interconnection between two spatially distributed nodes*: When both $\mathbb{X}_i$ and $\mathbb{X}_j$ are non-empty, the interconnection between two spatially distributed nodes are characterized at a common boundary point by inputs $m_{i,j}^{pp} : \mathbb{T} \to \mathbb{R}^{n_{mpp}^{ij}}$ and outputs $n_{i,j}^{pp} : \mathbb{T} \to \mathbb{R}^{n_{npp}^{ij}}$.
3. *Interconnection between a lumped node and a spatially distributed node*: When either $\mathbb{X}_i$ or $\mathbb{X}_j$ is empty, the interconnection between a spatially distributed node and a lumped node is characterized by inputs $m_{i,j}^{px} : \mathbb{T} \to \mathbb{R}^{n_{mpx}^{ij}}$ and outputs $n_{i,j}^{px} : \mathbb{T} \to \mathbb{R}^{n_{npx}^{ij}}$.

Hence the interconnection signal description differs based on the type of interconnection. Precisely, $l_{i,j} := \mathrm{col}(m_{i,j}^k, n_{i,j}^k)$ and $\mathscr{L}_{i,j}^I = \mathbb{R}^{n_{mk}^{ij}} \times \mathbb{R}^{n_{nk}^{ij}}$; $k \in \{pp, px, xx\}$.

Collection of All Interconnection Signals for An Individual Node

For a node $\mathfrak{N}_i$, all the interconnection signals are collectively defined as $l_i := \text{col}\left((m_i^{pp}, m_i^{px}, m_i^{xx}), (n_i^{pp}, n_i^{px}, n_i^{xx})\right)$.

1. The inputs are $m_i^{pp}(t) \in \mathbb{R}^{n^i_{mpp}}$ and outputs are $n_i^{pp}(t) \in \mathbb{R}^{n^i_{npp}}$, where $m_i^{pp} := \text{col}(m_{i,k}^{pp})_{k\in\mathbb{I}_{i,j}^{pp}}$ and $n_i^{pp} := \text{col}(n_{i,k}^{pp})_{k\in\mathbb{I}_{i,j}^{pp}}$ with $\mathbb{I}_{i,j}^{pp} := \{j \mid A_{i,j} = 1; i, j \in \mathbb{P}\}$.

2. The inputs are $m_i^{px}(t) \in \mathbb{R}^{n^i_{mpx}}$ and outputs are $n_i^{px}(t) \in \mathbb{R}^{n^i_{npx}}$, where $m_i^{px} := \text{col}(m_{i,k}^{px})_{k\in\mathbb{I}_{i,j}^{px}}$ and $n_i^{px} := \text{col}(n_{i,k}^{px})_{k\in\mathbb{I}_{i,j}^{px}}$ with $\mathbb{I}_{i,j}^{px} := \{j \mid A_{i,j} = 1;$ either $i \in \mathbb{O}$ or $j \in \mathbb{O}\}$.

3. The inputs are $m_i^{xx}(t) \in \mathbb{R}^{n^i_{mxx}}$ and outputs are $n_i^{xx}(t) \in \mathbb{R}^{n^i_{nxx}}$, where $m_i^{xx} := \text{col}(m_{i,k}^{xx})_{k\in\mathbb{I}_{i,j}^{xx}}$ and $m_i^{xx} := \text{col}(n_{i,k}^{xx})_{k\in\mathbb{I}_{i,j}^{xx}}$ with $\mathbb{I}_{i,j}^{xx} := \{j \mid A_{i,j} = 1; i, j \in \mathbb{O}\}$.

   Hence, $\mathscr{L}_i := \mathbb{R}^{n^i_{mpp}} \times \mathbb{R}^{n^i_{mpx}} \times \mathbb{R}^{n^i_{mxx}} \times \mathbb{R}^{n^i_{npp}} \times \mathbb{R}^{n^i_{npx}} \times \mathbb{R}^{n^i_{nxx}}$.

Figure 2.1 gives an illustration of how the spatial domains and signals are assigned for two mutually interconnected nodes.

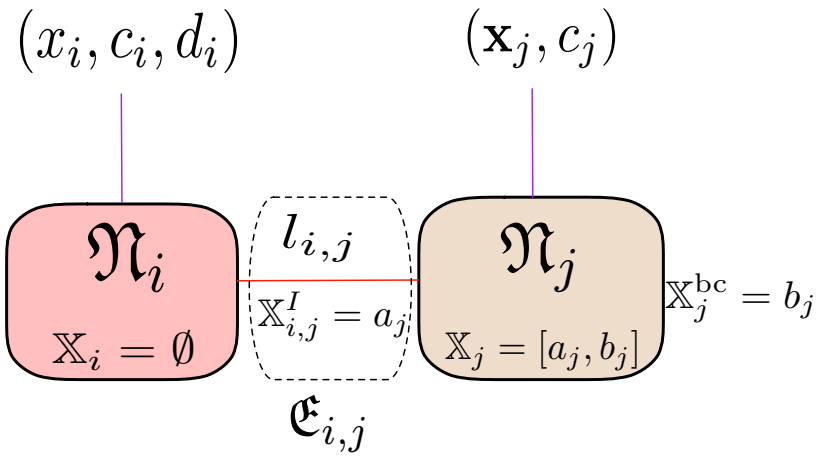


Figure 2.1: Illustration of two mutually interconnected nodes. Among them $\mathfrak{N}_i$ is lumped and $\mathfrak{N}_j$ is spatially distributed. The edge $\mathfrak{E}_{i,j}$ is defined at the boundary point a_j.

2.3.4 Specifying Edges

The interconnection describes the exchange of energy among adjacent component's underlying thermo-fluidic processes. The energy exchange is either conductive (between two solid components) or convective (between a solid and liquid component or between two liquid components). Typically, a bidirectional

exchange of energy occurs between two solid or a solid and a liquid component. A unidirectional exchange of energy occurs between two adjacent liquid channels towards the direction of the flow. Hence, a distinction is made between an edge $\mathfrak{E}_{i,j}$ and another edge $\mathfrak{E}_{j,i}$ to distinguish between bidirectional and unidirectional interconnections. Whenever $A_{i,j} = 1$, there are three types of interconnection between adjacent components $\mathfrak{N}_i$ and $\mathfrak{N}_j$. They are specified as follows.

Interconnection Between Two Lumped Nodes

The interconnection between two lumped nodes occurs when $A_{i,j} = 1$ along with the condition that $i, j \in \mathbb{O}$, i.e. both $\mathbb{X}_i$ and $\mathbb{X}_j$ are empty. Based on the conductive or convective thermal exchange, point-wise in time, the subspace $\mathscr{M}_{i,j}$ defines either uni-direction or bi-direction interconnection relations according to

$$\mathscr{M}_{i,j} := \left\{(l_{i,j}, l_{j,i}) \mid m^{xx}_{j,i}(t) = M^{xx}_{i,j} n^{xx}_{i,j}(t), A_{i,j} = 1, i, j \in \mathbb{O}\ \forall t \in \mathbb{T}\right\}. \tag{2.4}$$

Wherever $A_{i,j} = 1$, the user specifies the relation (2.4) by defining a constant matrix $M^{xx}_{i,j}$ of suitable dimension. In this way, interconnection between two lumped nodes is restricted by an algebraic constraint.

Interconnection Between a Spatially Distributed Node and a Lumped Node

When either $\mathbb{X}_i$ or $\mathbb{X}_j$ is empty and $A_{i,j} = 1$, interconnection occurs between a spatially distributed node and a lumped node. The subspace $\mathscr{M}_{i,j}$ defines either uni-direction or bi-direction interconnection relations according to

$$\mathscr{M}_{i,j} := \left\{(l_{i,j}, l_{j,i}) \mid m^{px}_{j,i}(t) = M^{px}_{i,j} n^{px}_{i,j}(t), A_{i,j} = 1, i \in \mathbb{O} \text{ or } j \in \mathbb{O}\ \forall t \in \mathbb{T}\right\}. \tag{2.5}$$

Wherever $A_{i,j} = 1$, the user specifies the relation (2.5) by defining a constant matrix $M^{px}_{i,j}$ of suitable dimension. Hence, interconnection between one lumped node and one spatially distributed node is restricted by an algebraic constraint.

Interconnection Between Two Spatially Distributed Nodes

The interconnection between two spatially distributed nodes occurs when $A_{i,j} = 1$ along with the condition that $i, j \in \mathbb{P}$, i.e. both $\mathbb{X}_i$ and $\mathbb{X}_j$ are non-empty. Now the corresponding interconnection relation is defined according to

$$\mathscr{M}_{i,j} := \left\{(l_{i,j}, l_{j,i}) \mid m^{pp}_{j,i}(t) = M^{pp}_{i,j} n^{pp}_{i,j}(t), A_{i,j} = 1, i, j \in \mathbb{P}\ \forall t \in \mathbb{T}\right\}. \tag{2.6}$$

The signals $(l_{i,j}, l_{j,i})$ are related to the state variables, $\mathbf{x}_i$ and $\mathbf{x}_j$, when evaluated at the interconnection boundary. In that case, one may assume the existence of

constant matrices, $B^n_{i,j}$ and $B^m_{j,i}$ of appropriate dimensions such that

$$n^{pp}_{i,j}(t) = B^n_{i,j}\begin{bmatrix}\mathbf{x}_{1i}(s^I_{i,j},t)\\ \mathbf{x}_{2i}(s^I_{i,j},t)\\ \partial_{s_i}\mathbf{x}_{2i}(s^I_{i,j},t)\end{bmatrix}, \qquad m^{pp}_{j,i}(t) = B^m_{j,i}\begin{bmatrix}\mathbf{x}_{1j}(s^I_{i,j},t)\\ \mathbf{x}_{2j}(s^I_{i,j},t)\\ \partial_{s_j}\mathbf{x}_{2j}(s^I_{i,j},t)\end{bmatrix}, \tag{2.7}$$

where $s^I_{i,j}$ the geometric location at the common boundary of the $\mathfrak{N}_i$ and $\mathfrak{N}_j$, i.e. $s^I_{i,j} \in \mathbb{X}^I_{i,j} \subseteq \{a_i = b_j, b_i = a_j\}$. Using (2.7) one can rewrite the interconnection constraint in (2.6) as the following boundary conditions at the point $s^I_{i,j} \in \mathbb{X}^I_{i,j}$.

$$\begin{bmatrix}B^m_{j,i} & -M^{pp}_{i,j}B^n_{i,j}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{1i}(s^I_{i,j},t)\\ \mathbf{x}_{2i}(s^I_{i,j},t)\\ \partial_{s_i}\mathbf{x}_{2i}(s^I_{i,j},t)\\ \mathbf{x}_{1j}(s^I_{i,j},t)\\ \mathbf{x}_{2j}(s^I_{i,j},t)\\ \partial_{s_j}\mathbf{x}_{2j}(s^I_{i,j},t)\end{bmatrix} = 0. \tag{2.8}$$

2.3.5 Specifying Time-Varying Parametric Relation

To make every lumped node adaptable with time-varying user-defined demand, they may include time-varying parameters (e.g. volumetric flow of liquid) that are not directly captured in (2.10). Such parameters are stored in a time-varying matrix $\Theta(t)$ for all lumped nodes. To allocate these parameters for every individual lumped node, an operator $\mathcal{T}_j$ is defined such that $\Theta_j(t) = (\mathcal{T}_j(\Theta))(t)$, where $\Theta_j(t)$ is the allocated time-varying parameter for the node $\mathcal{N}_j$.

The function $\Theta_j(t)$ relates the a specific functionality of a lumped node via parametric inputs (e.g. in terms of convective heat flux) $p_i(t)$ and outputs $q_i(t) \in \mathbb{R}^{n^i_q}$ (e.g. in terms of thermal energy over unit volume of liquid) according to the following algebraic relation point-wise in time:

$$\mathscr{B}_i := \{(p_i, q_i) \mid p_i(t) = \Theta_i(t)q_i(t)\ \forall t \in \mathbb{T}\}\,. \tag{2.9}$$

2.3.6 Specifying Node Dynamics

Dynamical behavior of a node depicts the temporal or spatio-temporal relationships among state variables, control signals, parametric signals and interconnection signals. The governing equations to describe a node's thermo-fluidic behavior are derived using balance laws of mass, momentum and energy. In case of lumped node, the dynamics is governed by coupled ordinary differential equations (ODEs) and in case of spatially distributed nodes the dynamics is governed by coupled partial differential equations (PDEs).

Temporal Node Dynamics

When $\mathbb{X}_j$ is empty for a node $\mathfrak{N}_j$, its lumped dynamics is governed by coupled ODEs. In this thesis, the ODEs are considered to be linear. In particular,

$$\mathscr{P}_j := \left\{ \begin{array}{l} \operatorname{col}\left((m_j^{px}, m_j^{xx}, p_j, w_j, u_j), (x_j), (n_j^{px}, n_j^{xx}, q_j, z_j, y_j)\right) \mid \forall j \in \mathbb{O}, t \in [0,\infty), \\ \\ \begin{bmatrix} n_j^{px}(t) \\ n_j^{xx}(t) \\ q_j(t) \\ z_j(t) \\ y_j(t) \\ \dot{x}_j(t) \end{bmatrix} = \begin{bmatrix} E_{nm1j} & E_{nm2j} & E_{npj} & E_{nwj} & E_{nuj} & G_{nxj} \\ F_{nm1j} & F_{nm2j} & F_{npj} & F_{nwj} & D_{nuj} & H_{nxj} \\ D_{qm1j} & D_{qm2j} & D_{qpj} & D_{qwj} & D_{quj} & C_{qxj} \\ D_{zm1j} & D_{zm2j} & D_{zpj} & D_{zwj} & D_{zuj} & C_{zxj} \\ D_{ym1j} & D_{ym2j} & D_{ypj} & D_{ywj} & D_{yuj} & C_{yxj} \\ B_{xm1j} & B_{xm2j} & B_{xpj} & B_{xwj} & B_{nuj} & A_{xxj} \end{bmatrix} \begin{bmatrix} m_j^{px}(t) \\ m_j^{xx}(t) \\ p_j(t) \\ w_j(t) \\ u_j(t) \\ x_j(t) \end{bmatrix} \end{array} \right\}, \tag{2.10}$$

where E_{nm1j}, E_{nm2j}, E_{npj}, E_{nwj}, E_{nuj}, G_{nxj}, F_{nm1j}, F_{nm2j}, F_{npj}, F_{nwj}, D_{nuj}, H_{nxj}, D_{qm1j}, D_{qm2j}, D_{qpj}, D_{qwj}, D_{quj}, C_{qxj}, D_{zm1j}, D_{zm2j}, D_{zpj}, D_{zwj}, D_{zuj}, C_{zxj}, D_{ym1j}, D_{ym2j}, D_{ypj}, D_{ywj}, D_{yuj}, C_{yxj}, B_{xm1j}, B_{xm2j}, B_{xpj}, B_{xwj}, B_{nuj}, and A_{xxj} are constant matrices of appropriate dimensions. The matrix A_{xxj} is typically derived using conservation laws of mass, momentum and energy over time. The other matrices are left to be defined by the user.

Spatio-Temporal Node Dynamics

When $\mathbb{X}_i = [a_i, b_i]$, the node dynamics of $\mathfrak{N}_i$ is governed by Partial Differential Equations (PDEs). In this thesis, the PDEs are considered to be linear and are defined in one spatial dimension. Hence,

$$\mathscr{P}_i := \left\{ \begin{array}{c} \operatorname{col}\left((m_i^{px}, w_i, u_i), (\mathbf{x}_i), (n_i^{px}, z_i, y_i)\right) \mid \forall i \in \mathbb{P}, t \in [0,\infty), \\ \mathbf{x}_i \text{ is Fréchet differentiable,} \\ \begin{bmatrix} n_i^{px}(t) \\ z_i(t) \\ y_i(t) \\ \dot{\mathbf{x}}_i(t) \end{bmatrix} = \begin{bmatrix} E_{nmi} & E_{nwi} & E_{nui} & \mathcal{E}_{n\mathbf{x}i} \\ D_{zmi} & D_{zwi} & D_{zui} & \mathcal{C}_{z\mathbf{x}i} \\ D_{ymi} & D_{ywi} & D_{yui} & \mathcal{C}_{y\mathbf{x}i} \\ \mathcal{B}_{\mathbf{x}mi} & \mathcal{B}_{\mathbf{x}wi} & \mathcal{B}_{\mathbf{x}ui} & \mathcal{A}_{pi} \end{bmatrix} \begin{bmatrix} m_i^{px}(t) \\ w_i(t) \\ u_i(t) \\ \mathbf{x}_i(t) \end{bmatrix} \end{array} \right\}. \tag{2.11}$$

In (2.11), all the operators are defined as in the following items.

- *Adding Inputs to the PDE Dynamics*:

For all $s_i \in [a_i, b_i]$ and $\mathbf{x}_i := \operatorname{col}(\mathbf{x}_{0i}, \mathbf{x}_{1i}, \mathbf{x}_{2i})$, $(\mathcal{B}_{\mathbf{x}mi}\mathbf{x}_i)(s_i) := B_{\mathbf{x}mi}(s_i)\mathbf{x}_i(s_i)$, $(\mathcal{B}_{\mathbf{x}wi}\mathbf{x}_i)(s_i) := B_{\mathbf{x}wi}(s_i)\mathbf{x}_i(s_i)$, $(\mathcal{B}_{\mathbf{x}ui}\mathbf{x}_i)(s_i) := B_{\mathbf{x}ui}(s_i)\mathbf{x}_i(s_i)$, where $B_{\mathbf{x}mi}(s_i)$, $B_{\mathbf{x}wi}(s_i)$, $B_{\mathbf{x}ui}(s_i)$ are matrix-valued functions of appropriate dimensions. Furthermore, E_{nmi}, E_{nwi}, E_{nui}, D_{zmi}, D_{zwi}, D_{zui}, D_{ymi}, D_{ywi}, D_{yui} are constant matrices of appropriate dimensions. These matrices are left to be defined by the user.

- *Including Parabolic and Hyperbolic PDEs, and Boundary Values in the Dynamics*:

For all $s_i \in [a_i, b_i]$, $\mathcal{A}_{pi}$ is defined as

$$\begin{aligned}(\mathcal{A}_{pi}\mathbf{x}_i)(s_i) :=& A_{0i}(s_i)\begin{bmatrix}\mathbf{x}_{0i}\\ \mathbf{x}_{1i}\\ \mathbf{x}_{2i}\end{bmatrix}(s_i) + A_{1i}(s_i)\partial_{s_i}\begin{bmatrix}\mathbf{x}_{1i}\\ \mathbf{x}_{2i}\end{bmatrix}(s_i) + A_{2i}(s_i)\partial_{s_i}^2\left[\mathbf{x}_{2i}\right](s_i)\\ &+\int_{a_i}^{s_i} A_{2li}(s_i,\theta_i)\begin{bmatrix}\mathbf{x}_{0i}\\ \mathbf{x}_{1i}\\ \mathbf{x}_{2i}\\ \partial_{s_i}\mathbf{x}_{1i}\\ \partial_{s_i}\mathbf{x}_{2i}\\ \partial_{s_i}^2\mathbf{x}_{2i}\end{bmatrix}(\theta_i)\mathrm{d}\theta_i + \int_{s_i}^{b_i} A_{2ui}(s_i,\theta_i)\begin{bmatrix}\mathbf{x}_{0i}\\ \mathbf{x}_{1i}\\ \mathbf{x}_{2i}\\ \partial_{s_i}\mathbf{x}_{1i}\\ \partial_{s_i}\mathbf{x}_{2i}\\ \partial_{s_i}^2\mathbf{x}_{2i}\end{bmatrix}(\theta_i)\mathrm{d}\theta_i\\ &+E_{2i}(s_i)\begin{bmatrix}\mathbf{x}_{1i}(a_i)\\ \mathbf{x}_{1i}(b_i)\\ \mathbf{x}_{2_i}(a_i)\\ \mathbf{x}_{2_i}(b_i)\\ \partial_{s_i}\mathbf{x}_{2i}(a_i)\\ \partial_{s_i}\mathbf{x}_{2i}(b_i)\end{bmatrix}. \end{aligned} \tag{2.12}$$

Here, $A_{0i}(s_i)$, $A_{1i}(s_i)$, $A_{2i}(s_i)$, $E_{2i}(s_i)$, $A_{2li}(s_i,\theta_i)$, $A_{2ui}(s_i,\theta_i)$ are matrix-valued functions of appropriate dimensions. These functions represent spatially varying (as functions of s_i) physical coefficients related to the thermo-fluidic process in a spatially distributed component. These coefficients often depend on the material properties, dimensions and the underlying physical process occurred in a component. The first three terms on the right hand side are related to the parabolic or hyperbolic PDEs. The next two terms are related to integrals of all the state variables and the last term is related to including boundary values into the dynamics.

- *Including boundary values in the measured, regulated and interconnection outputs*:

For all $s_i \in [a_i, b_i]$, operators $\mathcal{C}_{y\mathbf{x}i}$, $\mathcal{C}_{z\mathbf{x}i}$, $\mathcal{E}_{n\mathbf{x}i}$ are defined as follows.

$$\begin{aligned}(\mathcal{C}_{y\mathbf{x}i}\mathbf{x}_i)(s_i) :=& \int_{a_i}^{b_i}\Bigg(C_{ay\mathbf{x}i}(s_i)\begin{bmatrix}\mathbf{x}_{0i}\\ \mathbf{x}_{1i}\\ \mathbf{x}_{2i}\end{bmatrix}(s_i) + C_{by\mathbf{x}i}(s_i)\partial_{s_i}\begin{bmatrix}\mathbf{x}_{1i}\\ \mathbf{x}_{2i}\end{bmatrix}(s_i)\\ &+ C_{cy\mathbf{x}i}(s_i)\partial_{s_i}^2\left[\mathbf{x}_{2i}\right](s_i)\Bigg)\mathrm{d}s_i + C_{y0i}\begin{bmatrix}\mathbf{x}_{1i}(a_i)\\ \mathbf{x}_{1i}(b_i)\\ \mathbf{x}_{2i}(a_i)\\ \mathbf{x}_{2i}(b_i)\\ \partial_{s_i}\mathbf{x}_{2i}(a_i)\\ \partial_{s_i}\mathbf{x}_{2i}(b_i)\end{bmatrix}. \end{aligned} \tag{2.13}$$

$$(\mathcal{C}_{z\mathbf{x}i}\mathbf{x}_i)(s_i) := \int_{a_i}^{b_i} \Bigg(C_{az\mathbf{x}i}(s_i) \begin{bmatrix} \mathbf{x}_{0i} \\ \mathbf{x}_{1i} \\ \mathbf{x}_{2i} \end{bmatrix} (s_i) + C_{bz\mathbf{x}i}(s_i)\partial_{s_i} \begin{bmatrix} \mathbf{x}_{1i} \\ \mathbf{x}_{2i} \end{bmatrix} (s_i) + C_{cz\mathbf{x}i}(s_i)\partial_{s_i}^2 \left[\mathbf{x}_{2i}\right](s_i) \Bigg) \mathrm{d}s_i + C_{z0i} \begin{bmatrix} \mathbf{x}_{1i}(a_i) \\ \mathbf{x}_{1i}(b_i) \\ \mathbf{x}_{2i}(a_i) \\ \mathbf{x}_{2i}(b_i) \\ \partial_{s_i}\mathbf{x}_{2i}(a_i) \\ \partial_{s_i}\mathbf{x}_{2i}(b_i) \end{bmatrix}. \tag{2.14}$$

$$(\mathcal{E}_{n\mathbf{x}i}\mathbf{x}_i)(s_i) := \int_{a_i}^{b_i} \Bigg(E_{an\mathbf{x}i}(s_i) \left[\mathbf{x}_{0i}\right](s_i) + E_{bn\mathbf{x}i}(s_i)\partial_{s_i} \left[\mathbf{x}_{1i}\right](s_i) + E_{cn\mathbf{x}i}(s_i)\partial_{s_i}^2 \left[\mathbf{x}_{2i}\right](s_i) \Bigg) \mathrm{d}s_i + E_{n0i} \begin{bmatrix} \mathbf{x}_{1i}(a_i) \\ \mathbf{x}_{1i}(b_i) \\ \mathbf{x}_{2i}(a_i) \\ \mathbf{x}_{2i}(b_i) \\ \partial_{s_i}\mathbf{x}_{2i}(a_i) \\ \partial_{s_i}\mathbf{x}_{2i}(b_i) \end{bmatrix}. \tag{2.15}$$

Here, C_{y0i}, C_{z0i} and E_{n0i} are constant matrix of appropriate dimensions. Moreover, $C_{ay\mathbf{x}i}$, $C_{by\mathbf{x}i}$, $C_{cy\mathbf{x}i}$, $C_{az\mathbf{x}i}$, $C_{bz\mathbf{x}i}$, $C_{cz\mathbf{x}i}$, $E_{an\mathbf{x}i}$, $E_{bn\mathbf{x}i}$, $E_{cn\mathbf{x}i}$ are matrix-valued functions of appropriate dimensions. Based on a specific application and physical set-up of the component, these matrices are left to be defined by the user.

- *Interconnection Boundary Conditions*:

For a spatially distributed node $\mathfrak{N}_i$ (when $i \in \mathbb{P}$), grouping all its interconnection relations (2.8) with other spatially distributed nodes, one obtains the following boundary constraints that restricts the evolution of state variables $\mathbf{x}_i$ on the points $s_{i,k}^I \in (\cup_{k\in\mathbb{I}_{i,j}^{pp}} \mathbb{X}_{i,k}^I)$ with $\mathbb{I}_{i,j}^{pp} := \{j \mid A_{i,j} = 1; i, j \in \mathbb{P}\}$.

$$\underbrace{\mathrm{diag}\left(\begin{bmatrix} B_{j,i}^m & -M_{i,j}^{pp} B_{i,j}^n \end{bmatrix} \right)_{k\in\mathbb{I}_{i,j}^{pp}}}_{:=B_i^I} \underbrace{\mathrm{col}\left(\begin{bmatrix} \mathbf{x}_{1i}(s_{i,k}^I, t) \\ \mathbf{x}_{2i}(s_{i,k}^I, t) \\ \partial_{s_i}\mathbf{x}_{2i}(s_{i,k}^I, t) \\ \mathbf{x}_{1k}(s_{i,k}^I, t) \\ \mathbf{x}_{2k}(s_{i,k}^I, t) \\ \partial_{s_k}\mathbf{x}_{2k}(s_{i,k}^I, t) \end{bmatrix} \right)_{k\in\mathbb{I}_{i,j}^{pp}}}_{:=\Lambda_{s_i}^I(\mathbf{x}_i(t),\mathbf{x}_k(t))} = 0, \tag{2.16}$$

and which can be compactly written as

$$B_i^I \Lambda_{s_i}^I(\mathbf{x}_i(t), \mathbf{x}_k(t)) = 0. \tag{2.17}$$

- *External Boundary Conditions with Periodic Conditions and Inputs*:

The external boundary conditions are defined on the boundary points that are excluded from the corresponding set of interconnection points. They are allowed to be perturbed by inputs in the form of disturbances w_i, control inputs u_i and interconnection signals from the lumped nodes m_i^{px}. Moreover, it is possible to include periodic boundary conditions. These boundary conditions are defined on every point $s_i^{\mathrm{bc}} \in \mathbb{X}_i^{\mathrm{bc}} \setminus \cup_{k \in \mathbb{I}_{i,j}^{pp}} \mathbb{X}_{i,k}^{I}$ with $\mathbb{I}_{i,j}^{pp} := \{j \mid A_{i,j} = 1; i, j \in \mathbb{P}\}$ according to

$$\begin{aligned} B_i^{\mathrm{bc}} \overbrace{\operatorname{col}\left(\begin{bmatrix} \mathbf{x}_{1i}(s_i^{\mathrm{bc}}, t) \\ \mathbf{x}_{2i}(s_i^{\mathrm{bc}}, t) \\ \partial_{s_i}\mathbf{x}_{2i}(s_i^{\mathrm{bc}}, t) \end{bmatrix}\right)_{s_i^{\mathrm{bc}} \in \mathbb{X}_i^{\mathrm{bc}} \setminus \cup_{k \in \mathbb{I}_{i,j}^{pp}} \mathbb{X}_{i,k}^{I}}}^{:=\Lambda_{s_i}^{\mathrm{bc}}(\mathbf{x}_i(t))} &= B_{pxi} m_i^{px}(t) + B_{wi} w_i(t) + B_{u_i} u_i(t) \\ &\quad + \underbrace{\int_{a_i}^{b_i} B_{xxi}(s_i) \begin{bmatrix} \mathbf{x}_{0i} \\ \partial_{s_i}\mathbf{x}_{1i} \\ \partial_{s_i}^2 \mathbf{x}_{2i} \end{bmatrix} (s_i, t) \mathrm{d}s_i}_{:=\mathcal{B}_i^{\mathrm{bc}}(\mathbf{x}_i(t))}, \end{aligned} \tag{2.18}$$

and which can be compactly written as

$$\begin{bmatrix} B_{pxi} & B_{wi} & B_{ui} & -B_i^{\mathrm{bc}} \end{bmatrix} \begin{bmatrix} m_i^{px}(t) \\ w_i(t) \\ u_i(t) \\ \Lambda_{s_i}^{\mathrm{bc}}(\mathbf{x}_i(t)) \end{bmatrix} + \mathcal{B}_i^{\mathrm{bc}}(\mathbf{x}_i(t)) = 0. \tag{2.19}$$

For a spatially distributed node $\mathfrak{N}_i$, the number of points denoted by s_i^{bc} can be either one or two depending on whether it is interconnected with any other spatially distributed node or not. If there are r_i number of points in the set $(\mathbb{X}_i^{\mathrm{bc}} \setminus \cup_{k \in \mathbb{I}_{i,j}^{pp}} \mathbb{X}_{i,k}^{I})$, then $B_i^{\mathrm{bc}} \in R^{v_i \times r_i(n_1^i + 2n_2^i)}$ where v_i is the number of external boundary conditions. The other constant matrices B_{pxi}, B_{wi}, B_{ui} are defined accordingly. Moreover, $B_{xxi}(s_i)$ is a matrix valued function of appropriate dimension. Hence, the entire set of boundary conditions defines a subspace $\mathscr{P}_i^{\mathrm{bc}} \subset \mathscr{P}_i$ follows.

$$\mathscr{P}_i^{\mathrm{bc}} := \left\{ \begin{array}{c} \operatorname{col}\Big((m_i^{px}, w_i, u_i), (\mathbf{x}_i), (\mathbf{x}_k)\Big) \mid \forall i, k \in \mathbb{P}, t \in [0, \infty), \\ B_i^I \Lambda_{s_i}^I(\mathbf{x}_i(t), \mathbf{x}_k(t)) = 0, \\ \begin{bmatrix} B_{pxi} & B_{wi} & B_{ui} & -B_i^{\mathrm{bc}} \end{bmatrix} \begin{bmatrix} m_i^{px}(t) \\ w_i(t) \\ u_i(t) \\ \Lambda_{s_i}^{\mathrm{bc}}(\mathbf{x}_i(t)) \end{bmatrix} + \mathcal{B}_i^{\mathrm{bc}}(\mathbf{x}_i(t)) = 0 \end{array} \right\}. \tag{2.20}$$

Remark 2.1 *(Restriction on the number of boundary conditions) To avoid ill-posedness, the number of boundary conditions must be restricted to a fixed value. In one dimensional spatial interval, there are two boundary points for an*

individual node. Based on the definition of PDEs' state variables $\mathbf{x}_i$*, on these two boundary points, there must be exactly* n_i *boundary conditions related to* $\mathbf{x}_{1i}$ *and* $2(n_2^i)$ *boundary conditions related to* $\mathbf{x}_{2i}$*. In this graph-theoretic framework, the boundary points are divided into two mutually exclusive categories; the boundary points are either interconnected with an another spatially distributed node or exposed to external conditions (including interconnection with a lumped node). Therefore, for a spatially distributed node* $\mathfrak{N}_i$*, grouping these two types of boundary conditions according to* (2.16)-(2.18)*, one must ensure that the matrix* $\begin{bmatrix} B_i^{\mathrm{bc}} & 0 \\ 0 & B_i^I \end{bmatrix}$ *has* $(n_1^i + 2n_2^i)$ *independent rows.*

In summary, building a digital twin for thermo-fluidic processes amounts to specifying the items of Section 2.3 in terms of

1. Adjacency matrix to specify the topology of a spatially interconnected thermo-fluidic process. Time axis can be either continuous or discrete.

2. Spatial intervals of all spatially distributed nodes. Interconnection spaces and the boundary spaces are automatically determined using the adjacency matrix.

3. Dimensions of all the signal spaces (states, control signals, interconnection signals, parametric signals).

4. Matrices in the interconnection relations (2.4)-(2.8) as well as the time-varying matrices in the parametric relations (2.7) .

5. Matrices in (2.10) for every individual lumped node, matrices as well as matrix-valued functions to define the operators in (2.11) and matrices as well as matrix-valued functions to define the external boundary conditions in (2.20).

The graph-theoretic definitions (D.1)-(D.3) makes the modeling framework modular and generic with decisive advantages in terms of flexibility, scalability, versatility and integrability to build the digital twin. In particular, Object-Oriented Programming (OOP) can be utilized to develop software tools that can upscale the thermo-fluidic model on the basis of user-defined specifications. For instance, if there exists more than one node that is identical in terms of dynamics, boundary conditions, and interconnection with rest of the topologies, simply specifying the number of such identical nodes suffice to replicate their thermo-fluidic model and integrate with the digital twin's architecture. In figure 2.2, an illustration is provided to depict the graph-theoretic setting for mutual interconnection among two lumped nodes and two spatially distributed nodes.

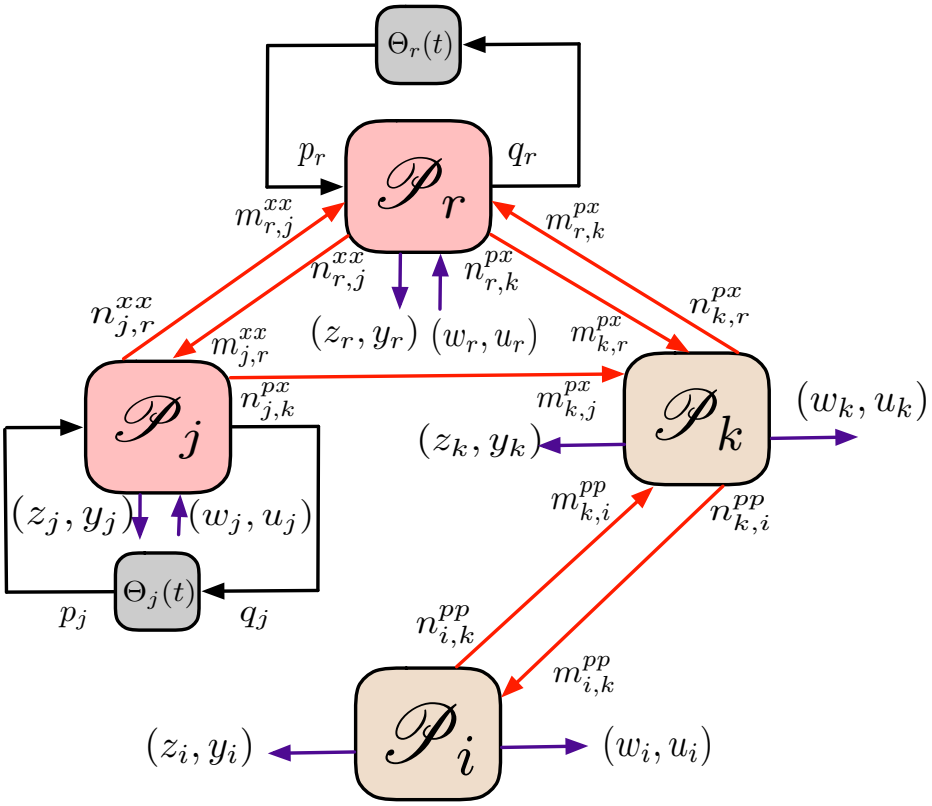


Figure 2.2: Illustration of four mutually interconnected nodes. ■ denotes the lumped nodes and ■ denotes the spatially distributed nodes. Interconnections can be unidirectional or bi-directional. A lumped node has additional provision to incorporate time-varying parameters (given by $\Theta_i(t), \Theta_j(t)$). ▬ denotes the parametric signals, ▬ denotes the interconnection signals, ▬ denotes the control signals.

2.4 Equivalent Representations of Lumped Nodes

Despite the merits and user-friendly implementation of graph-theoretic model of thermo-fluidic processes, a compact multi-input and multi-output (MIMO) representation may appear to be useful for design optimization, prediction, maintenance or synthesizing an estimator-based controller. To make the digital twin flexible and ensure its versatile functionalities, the digital twin's modeling framework offers various alternative representations of the thermo-fluidic processes that are equivalent to the graph-theoretic model. In the first part, only the lumped nodes are considered. By rearranging signals and performing algebraic operations, there are three ways to equivalently represent the part of a graph that consists of only lumped nodes.

By Stacking Node Signals

$$\breve{\mathscr{P}}_{\mathrm{o}} := \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{m}^{xx}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{n}^{xx}, \breve{q}, \breve{z}, \breve{y})\right) \mid \forall t \in [0,\infty), \\ \breve{m}^{xx}(t) = \breve{M}^{xx}\breve{n}^{xx}(t), \\ \breve{p}(t) = \breve{\Theta}(t)\breve{q}(t), \\ \begin{bmatrix} \breve{n}^{px}(t) \\ \breve{n}^{xx}(t) \\ \breve{q}(t) \\ \breve{z}(t) \\ \breve{y}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \breve{E}_{nm1} & \breve{E}_{nm2} & \breve{E}_{np} & \breve{E}_{nw} & \breve{E}_{nu} & \breve{G}_{nx} \\ \breve{F}_{nm1} & \breve{F}_{nm2} & \breve{F}_{np} & \breve{F}_{nw} & \breve{D}_{nu} & \breve{H}_{nx} \\ \breve{D}_{qm1} & \breve{D}_{qm2} & \breve{D}_{qp} & \breve{D}_{qw} & \breve{D}_{qu} & \breve{C}_{qx} \\ \breve{D}_{zm1} & \breve{D}_{zm2} & \breve{D}_{zp} & \breve{D}_{zw} & \breve{D}_{zu} & \breve{C}_{zx} \\ \breve{D}_{ym1} & \breve{D}_{ym2} & \breve{D}_{yp} & \breve{D}_{yw} & \breve{D}_{yu} & \breve{C}_{yx} \\ \breve{B}_{xm1} & \breve{B}_{xm2} & \breve{B}_{xp} & \breve{B}_{xw} & \breve{B}_{xu} & \breve{A}_{xx} \end{bmatrix} \begin{bmatrix} \breve{m}^{px}(t) \\ \breve{m}^{xx}(t) \\ \breve{p}(t) \\ \breve{w}(t) \\ \breve{u}(t) \\ x(t) \end{bmatrix} \end{array} \right\}. \tag{2.21}$$

Here, $x(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_x^j}$, $\breve{y}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_y^j}$, $\breve{z}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_z^j}$, $\breve{w}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_w^j}$, $\breve{u}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_u^j}$, $\breve{p}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_p^j}$, $\breve{q}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_q^j}$, $\breve{m}^{xx}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_{mxx}^j}$, $\breve{n}^{xx}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_{nxx}^j}$, $\breve{m}^{px}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_{mpx}^j}$, and $\breve{n}^{px}(t) \in \mathbb{R}^{\sum_{j\in\mathbb{O}} n_{npx}^j}$. Matrix $\breve{M}^{xx}$ is constructed by stacking $m_{k,j}^{xx} = M_{j,k}^{xx} n_{j,k}^{xx}$, whenever $A_{j,k} = 1; j,k \in \mathbb{O}$. Moreover, $\breve{\Theta}(t) = \mathrm{diag}(\Theta_j(t))_{j\in\mathbb{O}}$. All the other constant matrices in (2.21) is defined by diagonally stacking the respective matrices that are defined for every individual lumped node.

By Eliminating Interconnection Relations Among Lumped Nodes

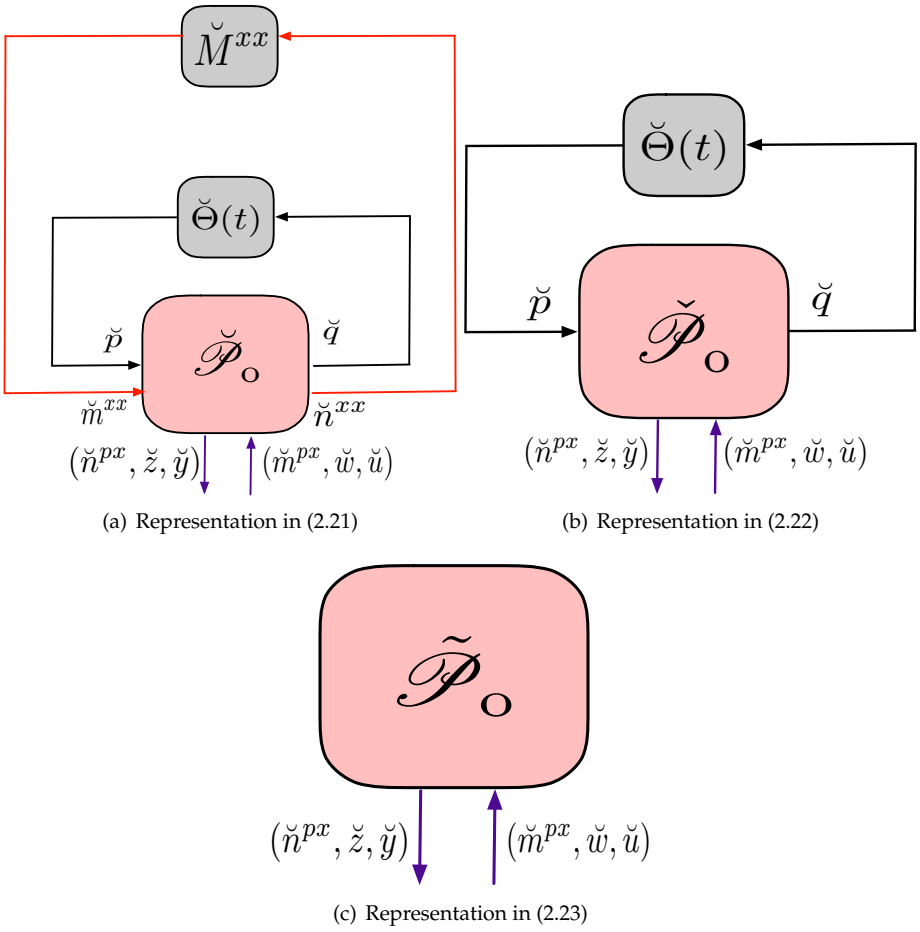
$$\check{\mathscr{P}}_{\mathrm{o}} := \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{q}, \breve{z}, \breve{y})\right) \mid \forall t \in [0,\infty), \\ \breve{p}(t) = \breve{\Theta}(t)\breve{q}(t), \\ \begin{bmatrix} \breve{n}^{px}(t) \\ \breve{q}(t) \\ \breve{z}(t) \\ \breve{y}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \check{E}_{nm} & \check{E}_{np} & \check{E}_{nw} & \check{E}_{nu} & \check{G}_{nx} \\ \check{D}_{qm} & \check{D}_{qp} & \check{D}_{qw} & \check{D}_{qu} & \check{C}_{qx} \\ \check{D}_{zm} & \check{D}_{zp} & \check{D}_{zw} & \check{D}_{zu} & \check{C}_{zx} \\ \check{D}_{ym} & \check{D}_{yp} & \check{D}_{yw} & \check{D}_{yu} & \check{C}_{yx} \\ \check{B}_{xm} & \check{B}_{xp} & \check{B}_{xw} & \check{B}_{xu} & \check{A}_{xx} \end{bmatrix} \begin{bmatrix} \breve{m}^{px}(t) \\ \breve{p}(t) \\ \breve{w}(t) \\ \breve{u}(t) \\ x(t) \end{bmatrix} \end{array} \right\}. \tag{2.22}$$

By Eliminating Parametric Relations

$$\tilde{\mathscr{P}}_{\mathrm{o}} := \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right) \mid \forall t \in [0,\infty), \\ \begin{bmatrix} \breve{n}^{px}(t) \\ \breve{z}(t) \\ \breve{y}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \tilde{E}_{nm}(t) & \tilde{E}_{nw}(t) & \tilde{E}_{nu}(t) & \tilde{G}_{nx}(t) \\ \tilde{D}_{zm}(t) & \tilde{D}_{zw}(t) & \tilde{D}_{zu}(t) & \tilde{C}_{zx}(t) \\ \tilde{D}_{ym}(t) & \tilde{D}_{yw}(t) & \tilde{D}_{yu}(t) & \tilde{C}_{yx}(t) \\ \tilde{B}_{xm}(t) & \tilde{B}_{xw}(t) & \tilde{B}_{xu}(t) & \tilde{A}_{xx}(t) \end{bmatrix} \begin{bmatrix} \breve{m}^{px}(t) \\ \breve{w}(t) \\ \breve{u}(t) \\ x(t) \end{bmatrix} \end{array} \right\}. \tag{2.23}$$

In (2.22)-(2.23), dimensions of all matrices are understood by their respective domains.

The above three representations equivalently represent thermo-fluidic processes that consist of lumped components only. These three representations can be visually illustrated in figure 2.3.



(a) Representation in (2.21)

(b) Representation in (2.22)

(c) Representation in (2.23)

Figure 2.3: Three equivalent representations of graph-theoretic thermo-fluidic model that consists of lumped components only.

One can also view $\breve{\mathscr{P}}_{\mathrm{o}}$ and $\tilde{\breve{\mathscr{P}}}_{\mathrm{o}}$ as projections of $\breve{\mathscr{P}}_{\mathrm{o}}$ onto the subspaces defined by the signals $\mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right)$ and $\mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right)$, respectively. To represent such projections using (2.22) and (2.23) requires elimination of interconnection signals $\breve{m}^{xx}(t), \breve{n}^{xx}(t)$ and parametric signals $\breve{p}(t), \breve{q}(t)$ from (2.21). Note that deriving (2.21) from (D.1)-(D.3) simply requires rearrangement of various signals. However, deriving (2.22) and (2.23) are conditional and Theorem 2.1 provides conditions under which such elimination of signals is possible.

Theorem 2.1 ***(Equivalence of lumped models under signal elimination)*** *1. If* $(I - \breve{F}_{nm2}\breve{M}^{xx})$ *is invertible, then*

$$\breve{\mathscr{P}}_{\mathrm{o}} = \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{q}, \breve{z}, \breve{y})\right) \mid \exists\, \mathrm{col}\,(\breve{m}^{xx}, \breve{n}^{xx}) \text{ such that,} \\ \mathrm{col}\left((\breve{m}^{px}, \breve{m}^{xx}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{n}^{xx}, \breve{q}, \breve{z}, \breve{y})\right) \in \breve{\mathscr{P}}_{\mathrm{o}} \end{array} \right\}.$$

Moreover,

$$\begin{bmatrix} \breve{E}_{nm} & \breve{E}_{np} & \breve{E}_{nw} & \breve{E}_{nu} & \breve{G}_{nx} \\ \breve{D}_{qm} & \breve{D}_{qp} & \breve{D}_{qw} & \breve{D}_{qu} & \breve{C}_{qx} \\ \breve{D}_{zm} & \breve{D}_{zp} & \breve{D}_{zw} & \breve{D}_{zu} & \breve{C}_{zx} \\ \breve{D}_{ym} & \breve{D}_{yp} & \breve{D}_{yw} & \breve{D}_{yu} & \breve{C}_{yx} \\ \breve{B}_{xm} & \breve{B}_{xp} & \breve{B}_{xw} & \breve{B}_{nu} & \breve{A}_{xx} \end{bmatrix} = \begin{bmatrix} \breve{E}_{nm1} & \breve{E}_{np} & \breve{E}_{nw} & \breve{E}_{nu} & \breve{G}_{nx} \\ \breve{D}_{qm1} & \breve{D}_{qp} & \breve{D}_{qw} & \breve{D}_{qu} & \breve{C}_{qx} \\ \breve{D}_{zm1} & \breve{D}_{zp} & \breve{D}_{zw} & \breve{D}_{zu} & \breve{C}_{zx} \\ \breve{D}_{ym1} & \breve{D}_{yp} & \breve{D}_{yw} & \breve{D}_{yu} & \breve{C}_{yx} \\ \breve{B}_{xm1} & \breve{B}_{xp} & \breve{B}_{xw} & \breve{B}_{xu} & \breve{A}_{xx} \end{bmatrix}$$
$$+ \begin{bmatrix} \breve{E}_{nm2} \\ \breve{D}_{qm2} \\ \breve{D}_{zm2} \\ \breve{D}_{ym2} \\ \breve{B}_{xm2} \end{bmatrix} \breve{M}^{xx}(I - \breve{F}_{nm2}\breve{M}^{xx})^{-1} \begin{bmatrix} \breve{F}_{nm1}^{\top} \\ \breve{F}_{np}^{\top} \\ \breve{F}_{nw}^{\top} \\ \breve{D}_{nu}^{\top} \\ \breve{H}_{nx}^{\top} \end{bmatrix}^{\top}. \tag{2.24}$$

2. If $(I - \breve{D}_{qp}\breve{\Theta}(t))$ *is invertible for all* $t \in \mathbb{T}$*, then*

$$\begin{aligned} &\tilde{\breve{\mathscr{P}}}_{\mathrm{o}} \\ &= \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right) \mid \exists\, \mathrm{col}\,(\breve{p}, \breve{q}) \text{ such that,} \\ \mathrm{col}\left(\breve{m}^{px}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{q}, \breve{z}, \breve{y})\right) \in \breve{\mathscr{P}}_{\mathrm{o}} \end{array} \right\}. \\ &= \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right) \mid \exists\, \mathrm{col}\,((\breve{m}^{xx}, \breve{p}), (\breve{n}^{xx}, \breve{q})) \text{ such that,} \\ \mathrm{col}\left((\breve{m}^{px}, \breve{m}^{xx}, \breve{p}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{n}^{xx}, \breve{q}, \breve{z}, \breve{y})\right) \in \breve{\mathscr{P}}_{\mathrm{o}} \end{array} \right\} \end{aligned}$$

Moreover,

$$\begin{bmatrix} \tilde{E}_{nm}(t) & \tilde{E}_{nw}(t) & \tilde{E}_{nu}(t) & \tilde{G}_{nx}(t) \\ \tilde{D}_{zm}(t) & \tilde{D}_{zw}(t) & \tilde{D}_{zu}(t) & \tilde{C}_{zx}(t) \\ \tilde{D}_{ym}(t) & \tilde{D}_{yw}(t) & \tilde{D}_{yu}(t) & \tilde{C}_{yx}(t) \\ \tilde{B}_{xm}(t) & \tilde{B}_{xw}(t) & \tilde{B}_{xu}(t) & \tilde{A}_{xx}(t) \end{bmatrix} = \begin{bmatrix} \breve{E}_{nm} & \breve{E}_{nw} & \breve{E}_{nu} & \breve{G}_{nx} \\ \breve{D}_{zm} & \breve{D}_{zw} & \breve{D}_{zu} & \breve{C}_{zx} \\ \breve{D}_{ym} & \breve{D}_{yw} & \breve{D}_{yu} & \breve{C}_{yx} \\ \breve{B}_{xm} & \breve{B}_{xw} & \breve{B}_{xu} & \breve{A}_{xx} \end{bmatrix}$$
$$+ \begin{bmatrix} \breve{E}_{np} \\ \breve{D}_{zp} \\ \breve{D}_{yp} \\ \breve{B}_{xp} \end{bmatrix} \breve{\Theta}(t)(I - \breve{D}_{qp}\breve{\Theta}(t))^{-1} \begin{bmatrix} \breve{D}_{qm}^{\top} \\ \breve{D}_{qw}^{\top} \\ \breve{D}_{qu}^{\top} \\ \breve{C}_{qx}^{\top} \end{bmatrix}^{\top}. \tag{2.25}$$

Proof: Deriving (2.24) requires algebraic manipulations to uniquely express $\breve{m}^{xx}$ in terms of $(x, \breve{w}, \breve{u}, \breve{p})$ by using $m^{xx}(t) = \breve{M}^{xx}n^{xx}(t)$. The uniqueness of this expression requires the invertibility of $(I - \breve{F}_{nm2}\breve{M}^{xx})$. The same argument holds for deriving (2.25), where $(\breve{p}, \breve{q})$ is uniquely expressed in terms of $(x, \breve{w}, \breve{u})$. In this case, one requires the invertibility of $(I - \breve{D}_{qp}\breve{\Theta}(t))$ for all $t \in \mathbb{T}$. ■

2.5 Equivalent Representation of Spatially Distributed Nodes

In a similar fashion, the spatially distributed nodes can also have alternative representations. Here, one of them is shown.

2.5.1 Normalization of Spatial Domain

In the definition of every spatially distributed node $\mathfrak{N}_i$, the spatial domain is $\mathbb{X}_i = [a_i, b_i]$ and every point s_i takes value in the interval $[a_i, b_i]$. For convenience, each interval $[a_i, b_i]$ can also be scaled to a fixed domain $[a, b]$, making every individual spatially distributed node to be defined in a constant spatial interval $[a, b]$. In order to achieve that, the following formulas are used.

1. For $s_i \in [a_i, b_i]$, define $s_i = m_i s + c_i$, with $m_i = \frac{b_i - a_i}{b-a}$ and $c_i = \frac{a_i b - b_i a}{b-a}$. Note that $s_i \to a_i$, then $s \to a$, and $s_i \to b_i$, then $s \to b$.
2. $\tilde{\mathbf{x}}_{0i}(s) = \mathbf{x}_{0i}(m_i s + c_i)$, $\tilde{\mathbf{x}}_{1i}(s) = \mathbf{x}_{1i}(m_i s + c_i)$ and $\tilde{\mathbf{x}}_{2i}(s) = \mathbf{x}_{2i}(m_i s + c_i)$.
3. $\partial_{s_i}\left[\mathbf{x}_{1i}(s_i)\right] = \frac{1}{m_i}\partial_s\left[\tilde{\mathbf{x}}_{1i}(s)\right]$, $\quad \partial_{s_i}\left[\mathbf{x}_{2i}(s_i)\right] = \frac{1}{m_i}\partial_s\left[\tilde{\mathbf{x}}_{2i}(s)\right]$,
4. $\partial^2_{s_i}\left[\mathbf{x}_{2i}(s_i)\right] = \frac{1}{m_i^2}\partial^2_s\left[\tilde{\mathbf{x}}_{2i}(s)\right]$.

Scaling Operators in the Domain $[a, b]$

In the dynamic model of a spatially distributed node, scaling of spatial domain also scales the spatially varying coefficients in the definitions of $\mathcal{A}_{pi}$, $\mathcal{C}_{y\mathbf{x}i}$, $\mathcal{C}_{z\mathbf{x}i}$, $\mathcal{E}_{n\mathbf{x}i}$. Precisely, in (2.12)-(2.15), for all $s \in [a, b]$

$$\begin{aligned}
(\tilde{\mathcal{A}}_{pi}\tilde{\mathbf{x}}_i)(s) :=& (\mathcal{A}_{pi}\mathbf{x}_i)(m_i s + c_i), \qquad (\tilde{\mathcal{C}}_{y\mathbf{x}i}\tilde{\mathbf{x}}_i)(s) := (\mathcal{C}_{y\mathbf{x}i}\mathbf{x}_i)(m_i s + c_i),\\
(\tilde{\mathcal{C}}_{z\mathbf{x}i}\tilde{\mathbf{x}}_i)(s) :=& (\mathcal{C}_{z\mathbf{x}i}\mathbf{x}_i)(m_i s + c_i), \qquad (\tilde{\mathcal{E}}_{n\mathbf{x}i}\tilde{\mathbf{x}}_i)(s) := (\mathcal{E}_{n\mathbf{x}i}\mathbf{x}_i)(m_i s + c_i),\\
(\tilde{\mathcal{B}}_{\mathbf{x}mi}\tilde{\mathbf{x}}_i)(s) :=& (\mathcal{B}_{\mathbf{x}mi}\mathbf{x}_i)(m_i s + c_i), \qquad (\tilde{\mathcal{B}}_{\mathbf{x}wi}\tilde{\mathbf{x}}_i)(s) := (\mathcal{B}_{\mathbf{x}wi}\mathbf{x}_i)(m_i s + c_i),\\
& (\tilde{\mathcal{B}}_{\mathbf{x}ui}\tilde{\mathbf{x}}_i)(s) := (\mathcal{B}_{\mathbf{x}ui}\mathbf{x}_i)(m_i s + c_i). \qquad (2.26)
\end{aligned}$$

Scaling Boundary conditions on $\{a, b\}$

Due to spatial scaling, the boundary points of any spatially distributed node would always be the points $\{a, b\}$. Therefore, the interconnection boundary conditions in (2.16) can be transformed to the following relations on the points $s^I_{i,k} \in (\cup_{k \in \mathbb{I}^{pp}_{i,j}} \mathbb{X}^I_{i,k})$ with $\mathbb{I}^{pp}_{i,j} := \{j | A_{i,j} = 1; i, j \in \mathbb{P}\}$ and $\mathbb{X}^I_{i,k} \subseteq \{a, b\}$.

$$\tilde{B}^I_i\left[\Lambda^{Ii}_s(\tilde{\mathbf{x}}_i(t), \tilde{\mathbf{x}}_k(t))\right] = 0. \qquad (2.27)$$

Here, $\left[\Lambda_s^{I_i}(\tilde{\mathbf{x}}_i(t), \tilde{\mathbf{x}}_k(t))\right]$ is obtained from $\Lambda_{s_i}^I(\mathbf{x}_i(t), \mathbf{x}_k(t))$ after applying the appropriate scaling on the spatial domain. Moreover, $\tilde{B}_i^I$ is related to B_i^I in (2.20) by incorporating the constant factors due to scaling the operators $\partial_{s_i}(\cdot)$ into $\partial_s(\cdot)$.

On every point $s_i^{\mathrm{bc}} \in [a,b] \setminus \cup_{k\in\mathbb{I}_{i,j}^{pp}} \mathbb{X}_{i,k}^I$ with $\mathbb{I}_{i,j}^{pp} := \{j | A_{i,j} = 1, \mathbb{X}_i, \mathbb{X}_j \text{ are nonempty}\}$ and $\mathbb{X}_{i,k}^I \subseteq \{a,b\}$, the external boundary conditions in (2.18) are redefined as

$$\tilde{B}_i^{\mathrm{bc}}\left[\Lambda_s^{\mathrm{bc}}(\tilde{\mathbf{x}}_i(t))\right] - \tilde{\mathcal{B}}_i^{\mathrm{bc}}(\tilde{\mathbf{x}}_i(t)) = B_{pxi} m_i^{px}(t) + B_{wi} w_i(t) + B_{u_i} u_i(t). \tag{2.28}$$

Here, $\left[\Lambda_s^{\mathrm{bc}i}(\tilde{\mathbf{x}}_i(t))\right]$ is obtained from $\Lambda_{s_i}^{\mathrm{bc}}(\mathbf{x}_i(t))$ after applying the appropriate scaling on the spatial domain. Moreover, $\tilde{B}_i^{\mathrm{bc}}$ is related to B_i^{bc} in (2.20) by incorporating the constant factors due to scaling the operators $\partial_{s_i}(\cdot)$ into $\partial_s(\cdot)$. Also, the operator $\tilde{\mathcal{B}}_i^{\mathrm{bc}}$ is identically constructed from $\mathcal{B}_i^{\mathrm{bc}}$.

Equivalent Representation of Spatially Distributed Nodes on a Scaled Domain

Now, on the constant domain $[a,b]$, an alternative representation can be derived by grouping all the signals of the entire set of spatially distributed nodes.

$$\mathscr{P}_{\mathrm{p}} := \left\{ \begin{array}{c} \mathrm{col}\left((\bar{m}^{px}, \bar{w}, \bar{u}), (\mathbf{x}), (\bar{n}^{px}, \bar{z}, \bar{y})\right) \mid \forall t \in [0,\infty), \\ \mathbf{x} \text{ is Fréchet differentiable}, \\ \begin{bmatrix} \bar{m}^{px}(t) \\ \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}(t) \end{bmatrix} \in X_{\mathrm{dom}}^{\mathrm{p}}, \\ \begin{bmatrix} \bar{n}^{px}(t) \\ \bar{z}(t) \\ \bar{y}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} = \begin{bmatrix} \bar{E}_{nm} & \bar{E}_{nw} & \bar{E}_{nu} & \bar{\mathcal{E}}_{n\mathbf{x}} \\ \bar{D}_{zm} & \bar{D}_{zw} & \bar{D}_{zu} & \bar{\mathcal{C}}_{z\mathbf{x}} \\ \bar{D}_{ym} & \bar{D}_{yw} & \bar{D}_{zu} & \bar{\mathcal{C}}_{y\mathbf{x}} \\ \bar{\mathcal{B}}_{\mathbf{x}m} & \bar{\mathcal{B}}_{\mathbf{x}w} & \bar{\mathcal{B}}_{\mathbf{x}u} & \bar{\mathcal{A}}_p \end{bmatrix} \begin{bmatrix} \bar{m}^{px}(t) \\ \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}(t) \end{bmatrix} \end{array} \right\}. \tag{2.29}$$

with

$$X_{\mathrm{dom}}^{\mathrm{p}} := \left\{ \begin{array}{l} \begin{bmatrix} \bar{m}^{px}(t) \\ \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}(t) \end{bmatrix} \in \prod_{i\in\mathbb{P}} \left(\mathbb{R}^{n_{m_{px}}^i + n_w^i + n_u^i} \times L_2^{n_0^i}[a,b] \times H_1^{n_1^i}[a,b] \times H_2^{n_2^i}[a,b]\right) \mid \\ \\ \begin{bmatrix} \bar{B}_{px} & \bar{B}_w & \bar{B}_u & -\bar{B} \end{bmatrix} \begin{bmatrix} \bar{m}^{px}(t) \\ \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}_1(a,t) \\ \mathbf{x}_1(b,t) \\ \mathbf{x}_2(a,t) \\ \mathbf{x}_2(b,t) \\ \partial_s \mathbf{x}_2(a,t) \\ \partial_s \mathbf{x}_2(b,t) \end{bmatrix} + \int_a^b \bar{B}_{xx}(s) \begin{bmatrix} \mathbf{x}_0 \\ \partial_s \mathbf{x}_1 \\ \partial_s^2 \mathbf{x}_2 \end{bmatrix} (s,t) \mathrm{d}s = 0 \end{array} \right\}. \tag{2.30}$$

Here, $\bar{m}^{px}(t) \in \mathbb{R}^{\sum_{i\in\mathbb{P}} n^i_{px}}$, $\bar{w}(t) \in \mathbb{R}^{\sum_{i\in\mathbb{P}} n^i_{w}}$, $\bar{u}(t) \in \mathbb{R}^{\sum_{i\in\mathbb{P}} n^i_{u}}$. Identical construction is followed for obtaining the vectors $\bar{n}^{px}(t)$, $\bar{z}(t)$, $\bar{y}(t)$. In a similar fashion, $\mathbf{x}_0 = \operatorname{col}(\tilde{\mathbf{x}}_{0i})_{i\in\mathbb{P}}$. Similar definitions hold for $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\mathbf{x} := \operatorname{col}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$. To define the operators in (2.29)-(2.30), corresponding constant matrices and matrix-valued functions are obtained by appropriate diagonal stacking of (2.26)-(2.27) while using the definitions in (2.12)-(2.15) as well as (2.20) and are understood by their respective domains.

2.6 Combined Representation of Spatially Distributed Nodes and Lumped Nodes

In the digital twin's model, so far, various representations are derived for lumped nodes and spatially-distributed nodes separately. In this section, the digital twin's model is completed by combining the lumped and spatially distributed nodes together and derive equivalent representations of the entire network of thermo-fluidic process.

By Stacking MIMO Lumped Dynamics and Spatially Distributed Dynamics

By augmenting the subspaces $\mathscr{P}_{\mathrm{p}}$ in (2.29) and $\tilde{\mathscr{P}}_{\mathrm{o}}$ (2.23), an equivalent representation of the networked thermo-fluidic process is as follows.

$$
\tilde{\tilde{\mathscr{P}}} = \left\{ \begin{array}{c} \operatorname{col}\left((\breve{m}^{px}, \breve{m}^{px}, \breve{w}, \breve{w}, \breve{u}, \breve{u}), (x, \mathbf{x}), (\breve{n}^{px}, \breve{n}^{px}, \breve{z}, \breve{z}, \breve{y}, \breve{y})\right) \mid \forall t \in [0,\infty), \\ \mathbf{x} \text{ is Fréchet differentiable}, \\ \begin{bmatrix} \bar{m}^{px}(t) \\ \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}(t) \end{bmatrix} \in X^{\mathrm{p}}_{\mathrm{dom}}, \\ \begin{bmatrix} \breve{m}^{px}(t) \\ \bar{m}^{px}(t) \end{bmatrix} = M^{px} \begin{bmatrix} \breve{n}^{px}(t) \\ \bar{n}^{px}(t) \end{bmatrix}, \\ \begin{bmatrix} \breve{n}^{px}(t) \\ \bar{n}^{px}(t) \\ \breve{z}(t) \\ \bar{z}(t) \\ \breve{y}(t) \\ \bar{y}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} = \begin{bmatrix} \tilde{E}_{nm}(t) & 0 & \tilde{E}_{nw}(t) & 0 & \tilde{E}_{nu}(t) & 0 & \tilde{G}_{nx}(t) & 0 \\ 0 & \bar{E}_{nm} & 0 & \bar{E}_{nw} & 0 & \bar{E}_{nu} & 0 & \bar{\mathcal{E}}_{n\mathbf{x}} \\ \tilde{D}_{zm}(t) & 0 & \tilde{D}_{zw}(t) & 0 & \tilde{D}_{zu}(t) & 0 & \tilde{C}_{zx}(t) & 0 \\ 0 & \bar{D}_{zm} & 0 & \bar{D}_{zw} & 0 & \bar{D}_{zu} & 0 & \bar{\mathcal{C}}_{z\mathbf{x}} \\ \tilde{D}_{ym}(t) & 0 & \tilde{D}_{yw}(t) & 0 & \tilde{D}_{yu}(t) & 0 & \tilde{C}_{yx}(t) & 0 \\ 0 & \bar{D}_{ym} & 0 & \bar{D}_{yw} & 0 & \bar{D}_{yu} & 0 & \bar{\mathcal{C}}_{y\mathbf{x}} \\ \tilde{B}_{xm}(t) & 0 & \tilde{B}_{xw}(t) & 0 & \tilde{B}_{xu}(t) & 0 & \tilde{A}_{xx}(t) & 0 \\ 0 & \bar{\mathcal{B}}_{\mathbf{x}m} & 0 & \bar{\mathcal{B}}_{\mathbf{x}w} & 0 & \bar{\mathcal{B}}_{\mathbf{x}u} & 0 & \bar{\mathcal{A}}_{\mathbf{xx}} \end{bmatrix} \begin{bmatrix} \breve{m}^{px}(t) \\ \breve{m}^{px}(t) \\ \breve{w}(t) \\ \breve{w}(t) \\ \breve{u}(t) \\ \breve{u}(t) \\ x(t) \\ \mathbf{x}(t) \end{bmatrix} \end{array} \right\}. \tag{2.31}
$$

Equivalent PDE-ODE Coupled Representation of Thermo-Fluidic Processes

Combining the ODE dynamics of lumped nodes and PDE dynamics of spatially distributed nodes one can derive an MIMO representation as a time-varying PDE-ODE coupled system. Such a time-varying PDE-ODE coupled system belongs to

the class of models as defined in the Definition 2.2. Lemma 2.1 states the condition under which such equivalent conversion is possible.

Definition 2.2 ***(Behavior of Time-Varying PDE-ODE Coupled Systems)*** *Given matrix valued functions A_0,, A_1, A_2, A_{2l}, A_{2u}, E_2, E_a,, E_b, E_c, E_0, E, B_{21}, B_{22}, C_{ai}, C_{bi}, C_{ci}, for $i \in \{1,2\}$, as well as C_{10}, C_{20}, E_0, A, C_1, C_2, B_{11}, B_{12}, D_{11}, D_{12}, D_{21}, D_{22}, of appropriate dimensions, define*

$$
\begin{aligned}
(\mathcal{A}_p\mathbf{x})(s,t) :=& A_0(s,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t) + A_1(s,t)\partial_s\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t) + A_2(s,t)\partial_s^2\left[\mathbf{x}_2\right](s,t)\\
&+\int_a^s A_{2l}(s,\theta,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\\ \partial_s\mathbf{x}_1\\ \partial_s\mathbf{x}_2\\ \partial_s^2\mathbf{x}_2\end{bmatrix}(\theta,t)\mathrm{d}\theta + \int_s^b A_{2u}(s,\theta,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\\ \partial_s\mathbf{x}_1\\ \partial_s\mathbf{x}_2\\ \partial_s^2\mathbf{x}_2\end{bmatrix}(\theta,t)\mathrm{d}\theta\\
&+E_2(s,t)\begin{bmatrix}\mathbf{x}_1(a,t)\\ \mathbf{x}_1(b,t)\\ \mathbf{x}_2(a,t)\\ \mathbf{x}_2(b,t)\\ \partial_s\mathbf{x}_2(a,t)\\ \partial_s\mathbf{x}_2(b,t)\end{bmatrix}
\end{aligned}
\tag{2.32a}
$$

$$
\begin{aligned}
(\mathcal{C}_{1p}\mathbf{x})(s,t) :=& \int_a^b \left(C_{a1}(s,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t) + C_{b1}(s,t)\partial_s\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t)\right)\mathrm{d}s\\
&+\int_a^b C_{c1}(s,t)\partial_s^2\left[\mathbf{x}_2\right](s,t)\mathrm{d}s + C_{10}\begin{bmatrix}\mathbf{x}_1(a,t)\\ \mathbf{x}_1(b,t)\\ \mathbf{x}_2(a,t)\\ \mathbf{x}_2(b,t)\\ \partial_s\mathbf{x}_2(a,t)\\ \partial_s\mathbf{x}_2(b,t)\end{bmatrix},
\end{aligned}
\tag{2.32b}
$$

$$
\begin{aligned}
(\mathcal{C}_{2p}\mathbf{x})(s,t) :=& \int_a^b \left(C_{a2}(s,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t) + C_{b2}(s,t)\partial_s\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t)\right)\mathrm{d}s\\
&+\int_a^b C_{c2}(s,t)\partial_s^2\left[\mathbf{x}_2\right](s,t)\mathrm{d}s + C_{20}\begin{bmatrix}\mathbf{x}_1(a,t)\\ \mathbf{x}_1(b,t)\\ \mathbf{x}_2(a,t)\\ \mathbf{x}_2(b,t)\\ \partial_s\mathbf{x}_2(a,t)\\ \partial_s\mathbf{x}_2(b,t)\end{bmatrix},
\end{aligned}
\tag{2.32c}
$$

$$
(\mathcal{E}_p\mathbf{x})(s,t) := \int_a^b \left(E_a(s,t)\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t) + E_b(s,t)\partial_s\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(s,t)\right)\mathrm{d}s
$$

$$+\int_a^b E_c(s,t)\partial_s^2\left[\mathbf{x}_2\right](s,t)\mathrm{d}s + E_0\begin{bmatrix}\mathbf{x}_1(a,t)\\ \mathbf{x}_1(b,t)\\ \mathbf{x}_2(a,t)\\ \mathbf{x}_2(b,t)\\ \partial_s\mathbf{x}_2(a,t)\\ \partial_s\mathbf{x}_2(b,t)\end{bmatrix}, \tag{2.32d}$$

$$(\mathcal{E}x)(s,t) := E(s,t)x(t), (\mathcal{B}_{21}w)(s,t) := B_{21}(s,t)w(t), (\mathcal{B}_{22}u)(s,t) := B_{22}(s,t)u(t). \tag{2.32e}$$

Then for a full rank $B \in \mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$, the behavior of time-varying PDE-ODE coupled system is defined by the subspace $\bar{\mathscr{P}}_{\mathrm{p}}^t$ such that

$$\bar{\mathscr{P}}_{\mathrm{p}}^t := \left\{ \begin{array}{c} \mathrm{col}\left((z,y),(x,\mathbf{x}),(w,u)\right) \mid \forall t\in[0,\infty) \\ \mathrm{col}\left(z(t),y(t)\right)\in\mathbb{R}^{n_z+n_y},\ \mathbf{x}\ \textit{is Fréchet differentiable}, \\ \begin{bmatrix}w(t)\\ u(t)\\ x(t)\\ \mathbf{x}(t)\end{bmatrix}\in X_{\mathrm{dom}}, \\ \begin{bmatrix}z(t)\\ y(t)\\ \dot{x}(t)\\ \dot{\mathbf{x}}(t)\end{bmatrix} = \begin{bmatrix}D_{11} & D_{12} & C_1 & \mathcal{C}_{1p}\\ D_{21} & D_{22} & C_2 & \mathcal{C}_{2p}\\ B_{11} & B_{12} & A & \mathcal{E}_p\\ \mathcal{B}_{21} & \mathcal{B}_{22} & \mathcal{E} & \mathcal{A}_p\end{bmatrix}(t)\begin{bmatrix}w(t)\\ u(t)\\ x(t)\\ \mathbf{x}(t)\end{bmatrix} \end{array} \right\}, \tag{2.33}$$

with

$$X_{\mathrm{dom}} := \left\{ \begin{array}{l} \begin{bmatrix}w(t)\\ u(t)\\ x(t)\\ \mathbf{x}(t)\end{bmatrix} \in \mathbb{R}^{n_w+n_u+n_x}\times L_2^{n_0}[a,b]\times H_1^{n_1}[a,b]\times H_2^{n_2}[a,b] \mid \\ \qquad \begin{bmatrix}B_w(t) & B_u(t) & B_{x0}(t) & \mathcal{B}_c(t)\end{bmatrix}\begin{bmatrix}w(t)\\ u(t)\\ x(t)\\ \mathbf{x}(t)\end{bmatrix} = 0 \\ (\mathcal{B}_c\mathbf{x})(s,t) := \int_a^b B_{xx}(s,t)\begin{bmatrix}\mathbf{x}_0\\ \partial_s\mathbf{x}_1\\ \partial_s^2\mathbf{x}_2\end{bmatrix}(s,t)\mathrm{d}s - B(t)\begin{bmatrix}\mathbf{x}_1(a,t)\\ \mathbf{x}_1(b,t)\\ \mathbf{x}_2(a,t)\\ \mathbf{x}_2(b,t)\\ \partial_s\mathbf{x}_2(a,t)\\ \partial_s\mathbf{x}_2(b,t)\end{bmatrix} \end{array} \right\}. \tag{2.34}$$

Lemma 2.1 ***(Equivalence of Time-Varying PDE-ODE coupled system and thermo-fluidic processes)***

If $\left(I - \begin{bmatrix}\tilde{E}_{nm}(t) & 0\\ 0 & \bar{E}_{nm}\end{bmatrix}M^{px}\right)$ is invertible for all $t\in\mathbb{T}$, then

$$\left\{\begin{array}{c}\operatorname{col}\left((\breve{w},\breve{w},\breve{u},\breve{u}),(x,\mathbf{x}),(\breve{z},\breve{z},\breve{y},\breve{y})\right) \mid \exists\, \operatorname{col}\left((\breve{m}^{px},\breve{m}^{px}),(\breve{n}^{px},\breve{n}^{px})\right) \text{ such that,}\\ \operatorname{col}\left((\breve{m}^{px},\breve{m}^{px},\breve{w},\breve{w},\breve{u},\breve{u}),(x,\mathbf{x}),(\breve{n}^{px},\breve{n}^{px},\breve{z},\breve{z},\breve{y},\breve{y})\right) \in \breve{\bar{\mathscr{P}}}^{*}\end{array}\right\}$$
$$= \left\{\begin{array}{c}\bar{\mathscr{P}}^{t}_{\mathrm{p}} \mid \\ w=\operatorname{col}(\breve{w},\breve{w}), u=\operatorname{col}(\breve{u},\breve{u}), z=\operatorname{col}(\breve{z},\breve{z}), y=\operatorname{col}(\breve{y},\breve{y})\end{array}\right\}.$$

Moreover, there exist matrix-valued functions A_0, A_1, A_2, A_{2l}, A_{2u}, E_2, E_a, E_b, E_c, E_0, E, B_{21}, B_{22}, C_{ai}, C_{bi}, C_{ci}, *for* $i \in \{1,2\}$, *as well as* C_{10}, C_{20}, E_0, A, C_1, C_2, B_{11}, B_{12}, D_{11}, D_{12}, D_{21}, D_{22}, *for which* (2.32) *admits*

$$\begin{bmatrix} D_{11} & D_{12} & C_1 & \mathcal{C}_{1p} \\ D_{21} & D_{22} & C_2 & \mathcal{C}_{2p} \\ B_{11} & B_{12} & A & \mathcal{E}_p \\ \mathcal{B}_{21} & \mathcal{B}_{22} & \mathcal{E} & \mathcal{A}_p \end{bmatrix}(t) = \begin{bmatrix} \tilde{D}_{zw}(t) & 0 & \tilde{D}_{zu}(t) & 0 & \tilde{C}_{zx}(t) & 0 \\ 0 & \bar{D}_{zw} & 0 & \bar{D}_{zu} & 0 & \bar{\mathcal{C}}_{z\mathbf{x}} \\ \tilde{D}_{yw}(t) & 0 & \tilde{D}_{yu}(t) & 0 & \tilde{C}_{yx}(t) & 0 \\ 0 & \bar{D}_{yw} & 0 & \bar{D}_{yu} & 0 & \bar{\mathcal{C}}_{y\mathbf{x}} \\ \tilde{B}_{xw}(t) & 0 & \tilde{B}_{xu}(t) & 0 & \tilde{A}_{xx}(t) & 0 \\ 0 & \bar{B}_{\mathbf{x}w} & 0 & \bar{B}_{\mathbf{x}u} & 0 & \bar{\mathcal{A}}_{\mathbf{xx}} \end{bmatrix}$$
$$+ \begin{bmatrix} \tilde{E}_{nm}(t) & 0 \\ 0 & \bar{E}_{nm} \\ \tilde{D}_{zm}(t) & 0 \\ 0 & \bar{D}_{zm} \\ \tilde{D}_{ym}(t) & 0 \\ 0 & \bar{D}_{ym} \\ \tilde{B}_{xm}(t) & 0 \\ 0 & \bar{B}_{\mathbf{x}m} \end{bmatrix} \begin{bmatrix} W_1 & W_2 \\ W_3 & W_4 \end{bmatrix}(t) G(t). \quad (2.35)$$

Here,

$$\begin{bmatrix} W_1 & W_2 \\ W_3 & W_4 \end{bmatrix}(t) = M^{px}\left(I - \begin{bmatrix} \tilde{E}_{nm}(t) & 0 \\ 0 & \bar{E}_{nm} \end{bmatrix} M^{px}\right)^{-1}, \quad (2.36)$$
$$G(t) = \begin{bmatrix} \tilde{E}_{nm}(t) & 0 & \tilde{E}_{nw}(t) & 0 & \tilde{E}_{nu}(t) & 0 & \tilde{G}_{nx}(t) & 0 \\ 0 & \bar{E}_{nm} & 0 & \bar{E}_{nw} & 0 & \bar{E}_{nu} & 0 & \bar{\mathcal{E}}_{n\mathbf{x}} \end{bmatrix},$$

and

$$B_w(t) = \bar{B}_w + \bar{B}_{px} \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} W_1(t) & W_2(t) \\ W_3(t) & W_4(t) \end{bmatrix} \begin{bmatrix} \tilde{E}_{nw}(t) & 0 \\ 0 & \bar{E}_{nw} \end{bmatrix},$$
$$B_u(t) = \bar{B}_u + \bar{B}_{px} \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} W_1(t) & W_2(t) \\ W_3(t) & W_4(t) \end{bmatrix} \begin{bmatrix} \tilde{E}_{nu}(t) & 0 \\ 0 & \bar{E}_{nu} \end{bmatrix},$$
$$\begin{bmatrix} B_{x0}(t) & \mathcal{B}_c(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \mathbf{x}(t) \end{bmatrix} = \bar{B}_{px} \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} W_1(t) & W_2(t) \\ W_3(t) & W_4(t) \end{bmatrix} \begin{bmatrix} \tilde{G}_{nx}(t) & 0 \\ 0 & \bar{\mathcal{E}}_{n\mathbf{x}} \end{bmatrix} \begin{bmatrix} x(t) \\ \mathbf{x}(t) \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & \bar{\mathcal{B}}_c(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \mathbf{x}(t) \end{bmatrix},$$

$$(\bar{\mathcal{B}}_c\mathbf{x})(s) = \int_a^b \bar{B}_{xx}(s) \begin{bmatrix} \mathbf{x}_0 \\ \partial_s \mathbf{x}_1 \\ \partial_s^2 \mathbf{x}_2 \end{bmatrix} (s)ds - \bar{B} \begin{bmatrix} \mathbf{x}_1(a) \\ \mathbf{x}_1(b) \\ \mathbf{x}_2(a) \\ \mathbf{x}_2(b) \\ \partial_s \mathbf{x}_2(a) \\ \partial_s \mathbf{x}_2(b) \end{bmatrix}. \tag{2.37}$$

Proof: The proof is similar to the proof of Theorem 2.1. It requires eliminating $(\breve{m}^{px}(t), \bar{m}^{px}(t))$ and $(\breve{n}^{px}(t), \bar{n}^{px}(t))$ from the definition of $\bar{\tilde{\mathscr{P}}}$ in (2.31). ■

2.7 Closing Remarks

In this chapter, the graph-theoretic framework is built to model spatially interconnected thermo-fluidic processes. A graph consists of nodes that describe the local dynamics and edges that describes the interconnection among adjacent nodes. Either ODEs or PDEs describes the node dynamics. The interconnection among adjacent nodes is described by either boundary conditions or algebraic constraints. Using augmentation of signals, scaling of the spatial domain for a (spatially distributed) node, and elimination of signals, seven equivalent representations of thermo-fluidic processes are provided as a part of the digital twin and they are depicted in Figure 2.4. The user has complete freedom to choose any one of these equivalent representations to describe the model of thermo-fluidic digital twin and build model-based functionalities. In this way, the modeling framework is modular and generic to offer decisive advantages in terms of the digital twin's flexibility, scalability, computability and versatility.

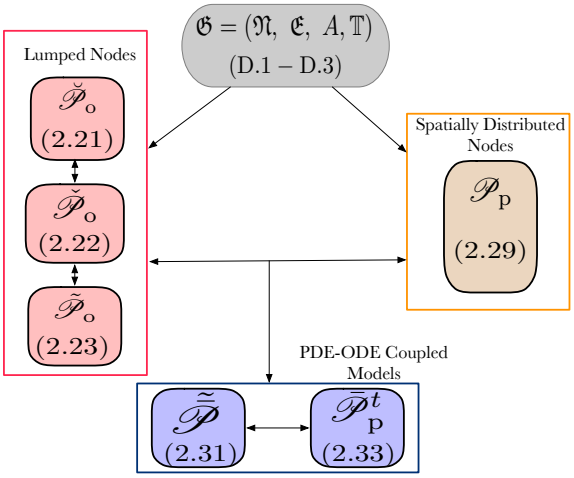


Figure 2.4: Seven equivalent representations of the thermo-fluidic model along with the references to their respective expressions.

Part II

Computation with Lumping

3 CHAPTER

Lumping Based Estimation and Control of Thermo-Fluidic Processes

This chapter discusses an abstraction of thermo-fluidic processes by neglecting the spatial variation of physical quantities. To this end, the method of lumping is introduced. Using the lumping technique, a version of digital twin is developed to describe the thermo-fluidic processes. The digital twin is based on a graph-theoretic framework where every individual edge describe the finite dimensional lumped dynamics of the thermo-fluidic process in a solid or liquid component and every edge describe the exchange of thermal energy among mutually interconnected components. A technique is developed for implementing soft sensor in the digital twin that estimates the liquid temperature solely using the presence of piezoelectric elements without installing additional devices. Using the digital twin, an output tracking anticipative feedback controller is synthesized. The control implementation is made flexible so that the user can vary the locations of the control inputs and the measured outputs depending on application specific requirements.

Outline

3.1 Introduction **50**
3.2 Specifications of Lumped Model **51**
3.3 Designing a Soft Sensor for Monitoring Thermo-Fluidic Processes **53**
3.4 Synthesizing a Model Predictive Controller for Thermo-Fluidic Processes **57**
3.5 Academic Case Study **63**
3.6 Closing Remarks **72**

3.1 Introduction

In 1873, James Clerk Maxwell introduced the idea of finding mechanical analogy of electrical circuits. Since then, using an isomorphic notion of energy or power, physical systems across electrical, mechanical, magnetic, and chemical domains are often abstracted using an equivalent and interchangeable set of physical quantities. For example, the voltage and current in electrical domain are equivalent to force and velocity respectively in mechanical domain, as well as, to chemical potential and molar rate respectively in chemical domain (c.f. [20]). The motivation behind *lumping* stems from simplifying the derivation of physical model by drawing such analogies among dynamical systems with equivalent notion of energy or power. Especially in thermo-fluidic processes, the lumping technique neglects the spatial distribution of the physical quantities in a system and, therefore, replaces the infinite dimensional partial integro-differential equations with finite dimensional ordinary differential equations. To this end, temperature difference and entropy rate in thermal domain, and pressure and volumetric flow rate in fluid domain respectively analogize the voltage and current in electrical domain. The lumping technique has been successfully applied to model and control climate inside a large-scale building in [83] and [46], for geo-thermal reservoir engineering in [40], and for gas processing plants in [71].

Concerning thermo-fluidic processes, one must capture the following three phenomena by the means of lumping

1. *Conduction*: Heat transfer within solid components or via contact between solid components.
2. *Convection*: Heat transfer between the solid component and liquid component.
3. *Advection*: Heat transfer via bulk movement of the liquid from one component to the other.

Similar to the RLC circuits, lumping based derivation of thermo-fluidic processes involves determining the heat storage and heat dissipation in these three phenomena and formulate a power balance among them. The governing model is finite dimensional and, therefore, can be easily implemented in software and hardware level. In Appendix 3.A, the equivalence between various thermo-fluidic aspects and electrical domain is summarized.

Remark 3.1 *(On the accuracy of lumping) Despite the simplicity of implementation, lumping based model is still an approximation of the spatio-temporal physical quantities. To verify the validity of neglecting spatial distribution, modeling practitioners often use a dimensionless parameter, called Biot number* (Bi)*. Typically the approximation error due to lumping is considered to be sufficiently small if the component's Biot number admits* $\mathrm{Bi} \leq 0.1$ *(c.f [11]). However, it has been demonstrated that the upper bound of* (Bi) *for which the lumping is accurate largely depends on the material properties, dimensions and the physical construct of the component (c.f. [52]).*

Contribution of this Chapter

This chapter contributes in the following aspects:

1. Based on the lumped model a data-drive soft sensor is developed that utilizes the piezoelectric actuators at liquid channels for estimating liquid temperature.

2. A reference-tracking model predictive controller is developed that makes the controlled process to obey a specific performance criterion. While synthesizing the controller, the user has the flexibility to change the locations of the control inputs and the measured outputs, leading to a flexible and user-friendly design tool.

3.2 Specifications of Lumped Model

In Chapter 2, the digital twin of thermo-fluidic processes has been modeled as a finite graph, according to (D.1)-(D.3) (on page 23-24), that consists of lumped nodes and spatially distributed nodes that are mutually interconnected in arbitrary spatial structure. Due to lumping, in this Chapter, a spatially distributed node also admits a lumped representation (may require experimental validation for that). Hence, it suffices to consider the entire graph consisting only lumped nodes. Following the notations and definitions In Chapter 2, this requires a few relaxations on (D.1)-(D.3):

- All the Nodes $\mathfrak{N}_i, i \in \mathbb{N}_{[1,m]}$ are lumped.
- $\mathbb{X}_i$ is empty for all $i \in \mathbb{N}_{[1,m]}$.
- Whenever $A_{i,j} = 1$, the interconnection is only among lumped nodes.
- The interconnections between spatially distributed nodes and lumped nodes are also omitted.

Remark 3.2 *(On partitioning lumped nodes for better accuracy) The accuracy of lumping largely depends on experimental verification on the values of Biot number. By merely lumping every individual spatially distributed node by a single lumped node, the accuracy of the model to describe the thermo-fluidic processes can be seriously compromised. In such cases, the modeling framework is built to be flexible towards further partitioning individual lumped components into multiple lumped sub-components. Taking interconnection signals into account, it can be partitioned into smaller identical nodes until the required model accuracy is achieved in the graph. Once the required model accuracy is achieved by partitioning, they are reconnected to the graph using interconnection relations. The method of partitioning can be extremely useful for lumping longer (with fast movement of liquid) liquid channels as shown in Figure 3.1.*

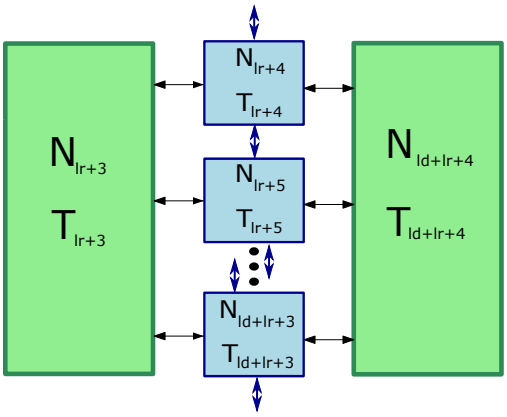


Figure 3.1: Example of partitioning of the liquid channels (given in blue rectangles). For node N_3, N_4 and N_5, there are additional indices (such as ld, lr) that can be user-defined and the digital twin automatically creates respective partitioned nodes. As a result of partitioning liquid channels, the adjacent solid blocks (given in green rectangle) are also partitioned.

Functionality of Feedback Based Lumped Analysis and Feedback Synthesis

As the underlying model is finite dimensional, there is a great body of well-established literature available for addressing these problems. For example, various system properties like well-posedness, and stability of spatially interconnected linear systems are analyzed in [32, 27, 58], and in[41, 105] similar analysis are performed for parameter varying systems. On the other hand, graph-theoretic framework is used for synthesis of distributed robust $\mathcal{H}_\infty$ controllers in [98], and controller design for identically modeled nodes is addressed in [65]. Robust controllers for interconnected systems can also be developed for discrete-time linear systems by following the framework in [32].

In this research, one of the major challenge is that controlling thermo-fluidic processes must be achieved without introducing additional sensors or additional actuators. Regardless of whether the aim is to design an estimator or a feedback controller, it is therefore relevant to determine what can be considered as measured outputs solely based on available resources. For example, consider the two applications in inkjet printing. The fixation process already facilitates temperature sensors that provides point-specific measurement of the temperature. As a result, in the absence of full state information, these point-specific temperature measurements can be utilized to design a state estimator for the thermo-fluidic process and, subsequently, a feedback controller. However, as seen in the jetting process, there is typically no sensor available to measure the liquid temperature. Due to the absence of measured output signals, implementing a feedback scheme to control the liquid temperature is not yet possible. In the next exposition, this problem is circumvented by introducing a

soft sensor based feedback scheme that can be applied to control liquid temperature without the need of installing additional sensors.

3.3 Designing a Soft Sensor for Monitoring Thermo-Fluidic Processes

For control, monitoring, and fault-diagnosis of the thermo-fluidic processes, it is necessary to acquire real-time information about the liquid temperature. However, this must be achieved without incorporating any additional sensors. It has already been seen that typically piezoelectric elements are placed at liquid channels for controlling the formation of droplets and their deposition (e.g. see the jetting process in inkjet printing). It is well-known that piezoelectric elements have self-sensing capabilities (c.f. [50, 5]). Therefore, a new role is assigned to the already installed piezoelectric elements to make a *soft sensing* device from which liquid temperature can be estimated. In this way, to control temperature of the jetting liquid in an inkjet printer, every individual nozzle can be equipped with collocated soft sensor.

Mechanism of Acoustic Sensing in Piezoelectric Element

Within the context of the jetting process, a piezoelectric element works in two operating modes. For formation and deposition liquid droplets, the piezoelectric element is in *actuation mode* where a sequence of trapezoidal voltage pulses allows droplet formation and ejection [57]. Using the same piezoelectric element, in *sensing mode*, one can capture the change in pressure oscillation and the nozzle acoustics during and after the actuation [97]. The mechanism is also known as *acoustic sensing* and the signal measured is called *acoustic signal* (Volt). The acoustic signal provides useful information about the dynamics inside a nozzle (c.f. [100, 49]).

By exploiting this self-sensing-actuation capability, in [100] and [50], two different approaches are presented to simultaneously operate the piezoelectric element in both actuation and sensing mode. However, these approaches require power-consuming and expensive electronic hardware that make it difficult for online implementation at every individual liquid channel [49](Chapter 3). Moreover, these measurements are typically used for fault-diagnosis and monitoring of the channel acoustics. For simpler hardware and its online application as a temperature sensor, in this research, a time delay (t_{delay}) is introduced between the actuation mode and sensing mode of the piezoelectric element. Once the actuation pulse is applied, after the time delay, an external trigger switches its operation to sensing mode, and the residual part of the acoustic signal is captured. Figure 3.2 depicts this sequential actuation-sensing scheme in an individual liquid channel.

To use the piezoelectric element at a liquid channel as a soft sensor for estimating liquid temperature, this paper uses the residual acoustic signal. Experiment

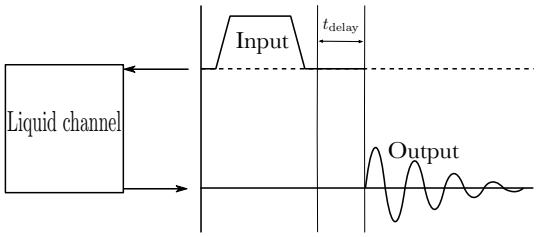


Figure 3.2: Sequential scheme for acoustic sensing.

shows that, at different temperature, there is a difference in the measured acoustic signals which indicates that that the acoustic sensing signals can be used for estimating liquid temperature (see figure 3.3).

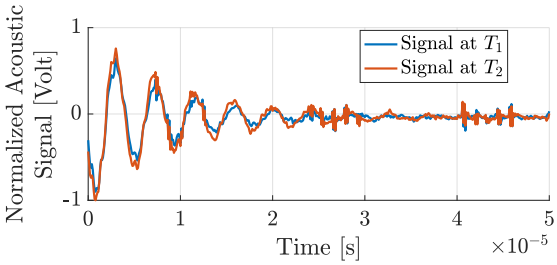


Figure 3.3: Measured acoustic signal in a particular liquid channel at two different liquid temperatures T_1°C, T_2°C, (see [28]).

Reconstructing Acoustic Sensing Signal

In view of figure 3.3, one can assume that the acoustic sensing signal consists of discrete samples of a damped sinusoidal function:

$$y(n) = \alpha e^{-\zeta n T_s} \sin(\omega n T_s + \phi) + \gamma; \qquad n \in \mathbb{N}. \tag{3.1}$$

With a fixed sampling time T_s, here, $\alpha > 0$ denotes the amplitude (Volt), $0 < \omega < \pi$ the frequency (rad/sample), $-\pi < \phi \leq \pi$ the phase shift (rad), γ the signal offset (Volt) and $\zeta \geq 0$ the damping factor.

Let the finite samples of acoustic sensing signal be given as the set $\{s_n \mid n \in \mathbb{N}_{[1,N]}\}$, with the length of the data-set as $0 < N < \infty$ and sampling period T_s is fixed. The aim is to estimate the parameters α, ζ, ω, γ and ϕ such that (3.1) optimally approximates the measured signal $\{s_n \mid n \in \mathbb{N}_{[1,N]}\}$. To this end, the following algorithmic scheme is developed

Algorithm 1 Estimating α, ζ, ω, γ and ϕ

Data: $\{s_n, n \in \mathbb{N}_{[1,N]}\}$
Unknowns: α, ζ, ω, γ, ϕ
Signal Generating model: $z^+ = A_s z + B_s w, \quad y = C_s z$

Step 1 – Finding poles of the signal generating model

Specify: $L \leftarrow \lfloor \frac{N}{3} \rfloor$
Hankel matrix: $H \leftarrow \begin{bmatrix} s_1 & \dots & s_{L-1} & s_L \\ s_2 & \dots. & s_L & s_{L+2} \\ \vdots & \ddots & \vdots & \vdots \\ s_{N-L} & \dots. & s_{N-2} & s_{N-1} \\ s_{N-L+1} & \dots & s_{N-1} & s_N \end{bmatrix}$
Optimal 2-rank approximation of H (singular value decomposition):

$$H \approx U \Sigma V^{\mathrm{H}}, \;\; \Sigma := \mathrm{diag}(\sigma_1, \sigma_2), \;\; \sigma_1, \sigma_2 > 0$$

Define: $U^1 \leftarrow U[1 : N-L, :]$
Define: $U^2 \leftarrow U[2 : N-L+1, :]$
Determine: $\tilde{A}_s \leftarrow \Sigma^{-\frac{1}{2}} U^{1\dagger} U^2 \Sigma^{\frac{1}{2}}$
for $k = 1$ and 2 **do**
    Solve eigen value ρ_k and right eigen vector v_k: $(\tilde{A}_s - \rho_k) v_k = 0$
    return ρ_k
end for

Step 2 – Determine ω and ζ

for $k = 1$ or 2 **do**
    $\omega \leftarrow \mathrm{Im}(\ln \rho_k)$
    $\zeta \leftarrow \mathrm{Re}(\ln \rho_k)$
end for
return ω, ζ

Step 3 – Determine α, ϕ, and γ

Define: $\mathbf{s} \leftarrow \mathrm{col}(s_1, \cdots, s_N)$
for $n \in \mathbb{N}_{[1,N]}$ **do**
    $\mathbf{\Gamma}[n, 1] \leftarrow (e^{j(\omega + j\zeta)nT_s} + e^{-j(\omega - j\zeta)nT_s})$
    $\mathbf{\Gamma}[n, 2] \leftarrow j(e^{j(\omega + j\zeta)nT_s} - e^{-j(\omega - j\zeta)nT_s})$
    $\mathbf{\Gamma}[n, 3] \leftarrow 1$
end for
Solve $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} ||\mathbf{\Gamma}\mathbf{x} - \mathbf{s}||_2$:

$$\hat{\mathbf{x}} \leftarrow (\mathbf{\Gamma}^{\mathrm{H}} \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^{\mathrm{H}} \mathbf{s}$$

$\alpha \leftarrow 2\sqrt{\mathrm{Re}(\hat{\mathbf{x}}[1] + j\hat{\mathbf{x}}[2])^2 + \mathrm{Im}(\hat{\mathbf{x}}[1] + j\hat{\mathbf{x}}[2])^2}$
$\phi \leftarrow \mathrm{Im}(\ln(\hat{\mathbf{x}}[1] + j\hat{\mathbf{x}}[2]))$.
$\gamma \leftarrow \hat{\mathbf{x}}[3]$
return α, ϕ, γ

Proof: Proof of the algorithm is given in Appendix 3.B. ■

Remark 3.3 *(on the accuracy of Algorithm 3.1) One can compare the developed algorithm against the conventional fast Fourier Transform (FFT) to judge the quality of reconstructing $y(n)$. Figure 3.4 shows this comparison.*

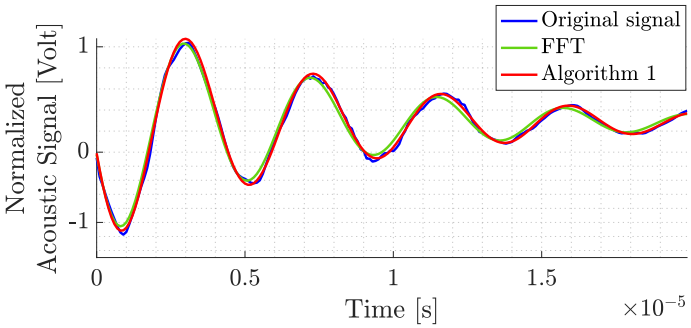


Figure 3.4: Comparison of FFT and proposed algorithm to reconstruct $y(n)$.

In Table 3.1, the estimated key parameters are compared. In spite of a little larger computational time, due to increased accuracy, the proposed algorithm is preferred over FFT.

Table 3.1: Key Parameters from the FFT, and Algorithm 3.1

Method	ω (Mrad/s)	α (Volt)	ζ (-)	ϕ (rad)	γ (Volt)	time (s)
FFT	0.1475	166.49	0.0135	-2.8561	-0.8530	0.006
Algorithm 3.1	0.1460	173.26	0.0127	-2.8798	-0.7129	0.008

Reconstruction of (3.1) acts as a digital filter to improve the noise level on the measured signal. Furthermore, the estimation of α, ζ, ω, γ and ϕ also offers a online algorithm to monitor the dynamic behaviour inside liquid channels.

Estimation of Liquid Temperature Using Acoustic Energy-Temperature Curve

Experiments show that the liquid temperature has the closest relation with the acoustic energy which is defined as the squared 2-norm of the estimated $y(n)$, $||y(n)||_2^2$. One expects lower acoustic energy in a lower temperature, and higher acoustic energy in a higher liquid temperature. To demonstrate that, for a specific liquid channel, a characteristic relation between acoustic energy and liquid temperature is established by using simulated data-sets at four operating points of liquid temperature and executing Algorithm 3.1 to obtain acoustic energy $||y(n)||_2^2$. Figure 3.5 shows the corresponding energy-temperature curve.

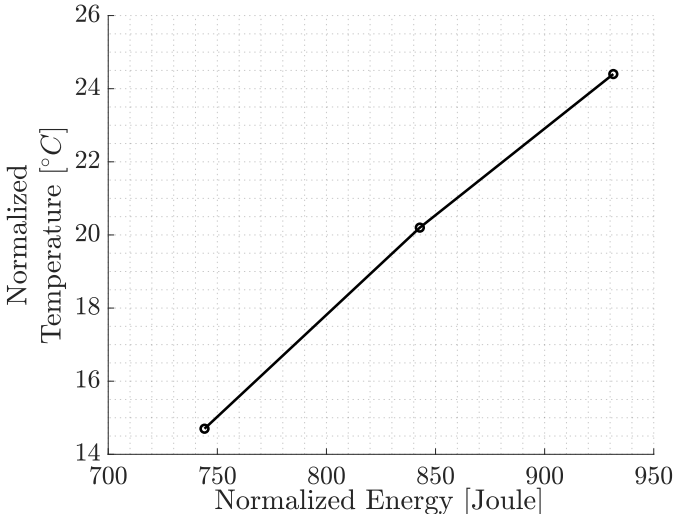


Figure 3.5: Example energy-temperature curve estimated with three operating points of temperature.

Based on the energy-temperature curves, one can determine a parametric linear model relating the acoustic energy and the liquid temperature. For an individual liquid channel that is equipped with piezoelectric element, let the energy-temperature characteristic curve be modeled as:

$$x_i = m_i \phi_i + c_i. \tag{3.2}$$

Here, for i^{th} channel, x_i is the liquid temperature and ϕ_i is the acoustic energy. The unknown parameters $m_i, c_i \in \mathbb{R}$ are obtained by fitting the respective energy-temperature curve. Every time a liquid droplet is jetted from a nozzle, thereafter, it can be followed by measuring the acoustic signal. The acoustic signal is modeled as (3.1) and its energy ϕ_i is determined using Algorithm 3.1. Subsequently, the corresponding liquid temperature x_i is obtained from (3.2). As a result, with the help of piezoelectric elements, this procedure allows to measure liquid temperature without installing additional sensors.

3.4 Synthesizing a Model Predictive Controller for Thermo-Fluidic Processes

One of the performance limiting aspects of thermo-fluidic processes is the fluctuation in liquid temperature among adjacent liquid channels (e.g. temperature fluctuation among nozzles in a printhead and among adjacent printheads). The only resources the controller has to use are a) the prior knowledge about the flow pattern, b) the lumping based digital twin to

anticipate and predict the evolution of thermo-fluidic process, and c) the soft sensor as feedback information on the temperature at specific liquid channels.

Model for Control

Here, one can directly consider the following behavior, defined in (2.23) on page 37.

$$\tilde{\mathscr{P}}_{\mathrm{o}} := \left\{ \begin{array}{c} \mathrm{col}\left((\breve{m}^{px}, \breve{w}, \breve{u}), (x), (\breve{n}^{px}, \breve{z}, \breve{y})\right) \mid \forall t \in [0, \infty), \\ \begin{bmatrix} \breve{n}^{px}(t) \\ \breve{z}(t) \\ \breve{y}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \tilde{E}_{nm}(t) & \tilde{E}_{nw}(t) & \tilde{E}_{nu}(t) & \tilde{G}_{nx}(t) \\ \tilde{D}_{zm}(t) & \tilde{D}_{zw}(t) & \tilde{D}_{zu}(t) & \tilde{C}_{zx}(t) \\ \tilde{D}_{ym}(t) & \tilde{D}_{yw}(t) & \tilde{D}_{yu}(t) & \tilde{C}_{yx}(t) \\ \tilde{B}_{xm}(t) & \tilde{B}_{xw}(t) & \tilde{B}_{xu}(t) & \tilde{A}_{xx}(t) \end{bmatrix} \begin{bmatrix} \breve{m}^{px}(t) \\ \breve{w}(t) \\ \breve{u}(t) \\ x(t) \end{bmatrix} \end{array} \right\}. \quad (3.3)$$

Here,

- $\breve{m}^{px}$ and $\breve{n}^{px}$ are set to be empty.
- $\tilde{D}_{zw}(t), \tilde{D}_{zu}(t)$ are set to zero and $\tilde{C}_{zx}(t)$ is set to a constant matrix H.

By performing time discretization of (3.3) on the sampled set $\mathbb{T} = \{kt_d | k \in \mathbb{N} \cup \{0\}\}$ with a fixed sampling period $t_d > 0$, $\{\tilde{\mathscr{P}}_{\mathrm{o}} \mid \breve{m}^{px} = \varnothing, \breve{n}^{px} = \varnothing, \tilde{D}_{zw}(t), \tilde{D}_{zu}(t) = 0, \tilde{C}_{zx}(t) = H, \forall t \in \mathbb{T}\}$ can re-written as the following discrete-times state space model

$$\begin{bmatrix} Q_t(x)(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \tilde{A}(t) & \tilde{B}(t) \\ \tilde{C}(t) & \tilde{D}(t) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} G(t) + W(t)d(t) \\ J(t) \end{bmatrix}. \quad (3.4)$$

Now, the vector $x(t)$ has to be understood as discrete samples and $Q_t(x)(t) := x(t + t_d), t \in \{kt_d | k \in \mathbb{N} \cup \{0\}\}$. In (3.4), $\tilde{A}(t)$, $\tilde{B}(t)$, $\tilde{C}(t)$, $\tilde{D}(t)$ are discretized version of $\tilde{D}_{yu}(t)$, $\tilde{C}_{yx}(t)$, $\tilde{B}_{xu}(t)$, and $\tilde{A}_{xx}(t)$.

Moreover, the discretized version of $\tilde{D}_{yw}(t)\breve{w}(t)$ is $J(t)$ and the discretized version of $\tilde{B}_{xw}(t)\breve{w}(t)$ is $G(t) + W(t)d(t)$. Purpose of such definitions is required to make the method applicable for applications related to inkjet printing. For example, during jetting, $d(t) \in \mathbb{R}^{n_d}$ are the known disturbances in terms of thermal power (in Watt) exerted by every individual piezoelectric actuator. Their values are known or can be experimentally determined.

The digital twin is made flexible such that the user may choose among all the node on which ones control inputs are applied, which ones' states are sensed, and on which ones external disturbances are applied. Therefore, the dimension of control input signals $u(t)$, measured output signals $y(t)$ and disturbance signals $d(t)$ are considered to be time-dependent based on the user's choice that may vary over application, performance specifications or operational constraints. In particular, at all times $t \in \mathbb{T}$, the digital twin has the flexibility for the user to allocate the following four time-varying set-valued maps.

1. $\mathbb{S}_p(t)$: The set of nodes on which disturbances are acted on at time t. The cardinality of the set is $n_p(t)$.

2. $\mathbb{S}_h(t)$: The set of all nodes on which control inputs are applied at time t. The cardinality of the set is $n_h(t)$.

3. $\mathbb{S}_s(t)$: The set of all nodes whose outputs are sensed at time t. The cardinality of the set is $n_s(t)$.

4. $\mathbb{S}_u(t)$: The set of all nodes that are neither in any of the above modes at time t. The cardinality of the set is $n_u(t)$.

To map the total number of states to its allocated modes of operation, binary matrices $S^p(t) \in \mathbb{R}^{n_u \times n_p(t)}$, $S^h(t) \in \mathbb{R}^{n_d \times n_h(t)}$, and $S^s(t) \in \mathbb{R}^{n_s(t) \times n_y}$ are introduced. Their (i,j)-th entries are:

$$S^p_{i,j}(t) = \begin{cases} 1, & i \in \mathbb{S}_p(t), \quad j \in \mathbb{N}_{[1,n_p(t)]} \\ 0, & \text{otherwise.} \end{cases},$$

$$S^h_{i,j}(t) = \begin{cases} 1, & i \in \mathbb{S}_h(t), \quad j \in \mathbb{N}_{[1,n_h(t)]} \\ 0, & \text{otherwise.} \end{cases},$$

$$S^s_{i,j}(t) = \begin{cases} 1, & i \in \mathbb{N}_{[1,n_s(t)]}, \quad j \in \mathbb{S}_s(t) \\ 0, & \text{otherwise.} \end{cases},$$

As a result, (3.4) is modified as

$$\begin{bmatrix} (Q_t(x))(t) \\ y^s(t) \end{bmatrix} = \begin{bmatrix} \tilde{A}(t) & \tilde{B}(t)S^h(t) \\ S^s(t)\tilde{C}(t) & S^s(t)\tilde{D}(t)S^h(t) \end{bmatrix} \begin{bmatrix} x(t) \\ u^h(t) \end{bmatrix} + \begin{bmatrix} G(t) + W(t)S^p(t)d^p(t) \\ J(t) \end{bmatrix}. \tag{3.5}$$

Here, at time $t \in \mathbb{T}$, the control inputs $u^h(t) \in \mathbb{R}^{n_h(t)}$ are applied by the set of heating nodes (in terms of thermal power, Watt). The disturbance $d^p(t) \in \mathbb{R}^{n_p(t)}$ are applied in terms of thermal power (Watt). The sensed outputs $y^s(t) \in \mathbb{R}^{n_s(t)}$ are sensed liquid temperature by the set of sensing nodes (in terms of temperature, °C).

Formulating Operational Constraints

Let $u_{\max} \in \mathbb{R}^{n_u}$ be the maximum admissible thermal power of a control actuator. Since colling is not allowed in many applications, at every time instant $t \in \mathbb{T}$, control input must satisfy the constraint

$$0 \leq S^h(t)u^h(t) \leq u_{\max}. \tag{3.6}$$

No additional constraints on the state variable $x(t)$ has been enforced.

Specifying Control Criterion

Based on the user-defined performance criterion, let a vector of to-be-controlled variables be defined as

$$z(t) = Hx(t), \tag{3.7}$$

where, $H \in \mathbb{R}^{n_z \times n_x}$ is a matrix depending on the choice of to-be-controlled variables. The objective of the controller is to achieve a steady state reference value $z^r \in \mathbb{R}^{n_z}$ for the to-be-controlled variable for all time $t \in \mathbb{T}$.

At every time instant $t = kt_d$ with $k \in \mathbb{N} \cup \{0\}$, for a finite horizon of future time instants $t \in \mathbb{T}_N^k$ where $\mathbb{T}_N^k := \{kt_d \mid k \in \mathbb{N}_{[k,N+k]}\}$ and $N > 0$, a reference tracking problem is formulated to minimize the following cost functional:

$$J_k := \sum_{t \in \mathbb{T}_N^k} ||x(t) - x^r||^2_{Q_k(t)} + \sum_{t \in \mathbb{T}_{N-1}^k} ||u^h(t) - u^r(t)||^2_{R_k(t)}. \tag{3.8}$$

Here, for $t \in \mathbb{T}_N^k$, the reference trajectories $(x^r, u^r(t))$ are pre-determined as the solution of the following linear equations:

$$\begin{aligned} x^r &= \tilde{A}(t)x^r + \tilde{B}(t)S^h(t)u^r(t) + G(t) + W(t)S^p(t)d^p(t), \\ z^r &= Hx^r. \end{aligned} \tag{3.9}$$

Moreover, for all $t \in \mathbb{T}_N^k$, one must pre-define the time-varying weights $Q_k(t) \in \mathbb{R}^{n_x \times n_x}$, $R_k(t) \in \mathbb{R}^{n_h(t) \times n_h(t)}$ in (3.8) to penalize the deviation of states and inputs from their respective references.

Formulating Model Predictive Control Scheme

A digital controller is required that minimizes (3.8) by using future prediction on the states in (3.5) while satisfying the constraints in (3.6). To this end, a reference-tracking model predictive control (MPC) scheme is presented.

At every time instant $t = kt_d$ with $k \in \mathbb{N} \cup \{0\}$, let any function $f(t)$ be defined as $f_{0|k} := f(kt_d)$. From its initial value $f_{0|k}$, over a finite horizon of future time instants $t \in \mathbb{T}_N^k$ with $N > 0$, the value of $f(t)$ at instant $k + i$ is denoted by $f_{i|k} := f((k + i)t_d)$. Starting from the time step kt_d, the N-horizon MPC scheme amounts to minimizing the following cost functional (c.f. [63])

$$\begin{aligned} J(k, \bar{x}_k, \bar{u}_k^h) := &||x_{N|k} - x^r||^2_{Q_k} + \\ &\sum_{i=0}^{N-1} \Big(||x_{i|k} - x^r||^2_Q + ||u^h_{i|k} - u^r_{i|k}||^2_{R_{i|k}} \Big), \end{aligned} \tag{3.10}$$

subject to the following constraints:

$$x_{i+1|k} = \tilde{A}_{i|k}x_{i|k} + \tilde{B}_{i|k}S^h_{i|k}u^h_{i|k} + \tilde{G}_{i|k}, i \in \mathbb{N}_{[0,N-1]}, \tag{3.11a}$$

$$E_{i|k}u^h_{i|k} \leq b, \quad i \in \mathbb{N}_{[0,N-1]}, \tag{3.11b}$$
$$F_{N|k}x_{N|k} \leq b_{N|k}, \tag{3.11c}$$
$$x_{0|k} = x(kt_d). \tag{3.11d}$$

Here, for $i \in \mathbb{N}_{[0,N-1]}$,

$$\begin{aligned}\tilde{G}_{i|k} =& G_{i|k} + W_{i|k}S^p_{i|k}d^p_{i|k},\\ E_{i|k} =& \text{col}(-S^h_{i|k}, S^h_{i|k}),\\ b =& \text{col}(0, u_{\max}).\end{aligned}$$

For all $t \in \mathbb{T}^k_N$, the reference trajectories (x^r, u^r) are determined by solving (3.9). The weights $Q \in \mathbb{R}^{n_x \times n_x}, R_{i|k} \in \mathbb{R}^{n_h(t)\times n_h(t)}$ are user-defined matrices. The terminal penalty $Q_k \in \mathbb{R}^{n_x \times n_x}$ in the cost (3.10) and $F_{N|k}, b_{N|k}$ in the terminal constraint (3.11c) are specifically chosen to ensure stability of the closed-loop system [68].

The above constrained optimization problem is solved for every k over the horizon $kt_d, \dots (k+N)t_d$ of N future time samples. The sequence of predicted states and future inputs are denoted as $\bar{x}_k := \{x_{1|k}, \cdots, x_{N|k}\}$ and $\bar{u}^h_k := \{u^h_{0|k}, \cdots, u^h_{N-1|k}\}$ respectively. If the minimizer of the optimal control problem (3.10)-(3.11) is denoted by $\bar{u}^{h*}_k := \{u^{h*}_{0|k}, \cdots, u^{h*}_{N-1|k}\}$, then its first entry $u^{h*}_{0|k}$ is implemented as control input at time step $t = kt_d$. Subsequently, on a receding horizon, the optimal control problem is solved again at step $(k+1)t_d$ using $x_{0|k+1} = x((k+1)t_d)$ as its initial state. In a receding horizon fashion, this procedure is continued iteratively over all time $t \in \{kt_d \mid k \in \mathbb{N} \cup \{0\}\}$ [62].

One may substitute the equality constraint (3.11a) in (3.10) to eliminate the variables $\{x_{0|k}, \cdots, x_{N|k}\}$ from the optimization problem. This results in the following dense linearly constrained quadratic programming (LCQP) problem:

Problem Description 3.1 *(Dense LCQP)*

$$\arg\min_{U_k} \frac{1}{2}U_k^\top G_k U_k + U_k^\top W_k \tag{3.12a}$$
$$\text{subject to } L_k U_k \leq V_k, \tag{3.12b}$$

where,

$$\begin{aligned}G_k &= 2(\Gamma_k^\top \Omega_k \Gamma_k + \Psi_k),\\ W_k &= (2\Gamma_k^\top \Omega_k \Phi_k)x_{0|k} + 2\Gamma_k^\top \Omega_k \Xi_k - \Lambda_k,\\ \Lambda_k &= -2\Gamma_k^\top \Omega_k X^r_k - 2\Psi_k U^r_k,\\ L_k &= \bar{M}\Gamma_k + \bar{E}_k,\\ V_k &= \bar{b} - \bar{M}\Phi_k x_{0|k} - \bar{M}\Xi_k,\\ x_{0|k} &= x(k),\end{aligned}$$

with

$$\Phi_k = \begin{bmatrix} \tilde{A}_{0|k} \\ \tilde{A}_{1|k}\tilde{A}_{0|k} \\ \vdots \\ \tilde{A}_{N-1|k}\cdots\tilde{A}_{0|k} \end{bmatrix}, \bar{M} = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & F_{N|k} \end{bmatrix},$$

$$\bar{E}_k = \begin{bmatrix} E_{0|k} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & E_{N-1|k} \\ 0 & \cdots & 0 \end{bmatrix}, \bar{b} = \begin{bmatrix} b \\ b \\ \vdots \\ b_{N|k} \end{bmatrix},$$

$$\Gamma_k = \begin{bmatrix} \tilde{B}_{0|k} & 0 & \cdots & 0 \\ \tilde{A}_{1|k}B_{0|k} & \tilde{B}_{1|k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_{N-1|k}\cdots\tilde{A}_{1|k}\tilde{B}_{0|k} & \tilde{A}_{N-2}\cdots\tilde{A}_{2|k}\tilde{B}_{1|k} & \cdots & \tilde{B}_{N-1|k} \end{bmatrix}$$

$$\Xi_k = \begin{bmatrix} \tilde{G}_{0|k} & 0 & \cdots & 0 \\ \tilde{A}_{1|k}\tilde{G}_{0|k} & \tilde{G}_{1|k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_{N-1|k}\cdots\tilde{A}_{1|k}\tilde{G}_{0|k} & \tilde{A}_{N-2|k}\cdots\tilde{A}_{2|k}\tilde{G}_{1|k} & \cdots & \tilde{G}_{N-1|k} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Here, the decision variables are the to-be-applied control input $U_k = \mathrm{col}(u^h_{0|k}, \cdots, u^h_{N-1|k})$. Moreover, $X^r_k = \mathrm{col}(x^r, \cdots, x^r)$ and $U^r_k = \mathrm{col}(u^r_{0|k}, \cdots, u^r_{N-1|k})$ are the reference values for states and inputs respectively. Moreover, $\Psi_k := \mathrm{diag}(R_{0|k}, \cdots, R_{N-1|k})$ and $\Omega_k := \mathrm{diag}(Q, \cdots, Q, Q_k)$.

The following result summarizes convergence properties of the resulting controlled system.

Theorem 3.1 ***(Stability of MPC)*** *Let the matrices $R_{i|k} \in \mathbb{S}^{n_h(t)}_{\succ 0}$, $Q \in \mathbb{S}^{n_x}_{\succ 0}$ are given for all time $t \in \mathbb{T}^k_N$. Moreover, let $X_k \in \mathbb{S}^{n_x}_{\succ 0}$ and $Y_k \in \mathbb{R}^{n_h(t)\times n_x}$ satisfy the following linear matrix inequality (LMI)*

$$\begin{bmatrix} -X_k & 0 & \tilde{A}_{N|k}X_k + \tilde{B}_{N|k}Y_k & 0 \\ 0 & -R^{-1}_{N|k} & Y_k & 0 \\ (\tilde{A}_{N|k}X_k + \tilde{B}_{N|k}Y_k)^\top & Y_k^\top & -X_k & X_k \\ 0 & 0 & X_k & -Q^{-1} \end{bmatrix} \preccurlyeq 0. \quad (3.13)$$

Furthermore, let (3.12) *have a unique minimizer $U^*_k := \{u^{h*}_{0|k}, \cdots, u^{h*}_{N-1|k}\}$ under the following conditions:*

1. *The terminal weight matrix Q_k admits $Q_k = X_k^{-1}$*

2. *The terminal constraint* $F_{N|k}x_{N|k} \leq b_{N|k}$ *admits*

$$F_{N|k} = E_{N|k}K_k; \quad b_{N|k} = b - u^r_{N|k} + E_{N|k}K_k x^r,$$

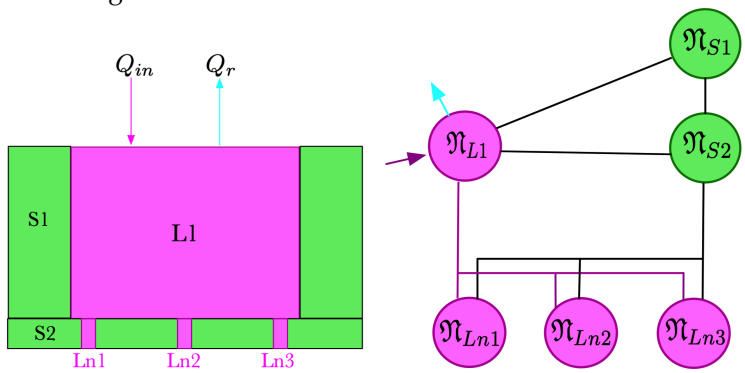
where, $K_k = Y_k X_k^{-1}$, $E_{N|k} = col(-S^h_{N|k}, S^h_{N|k})$, *and* $b = col(0, u_{max})$.

Then, setting $u^h(kt_d) = u^{h*}_{0|k}$, *there exists a* $\delta > 0$ *for which every initial condition* $||x(0) - x^r||_2 < \delta$ *and every solution to* (3.5), $k \to x(kt_d)$, *admits* $\lim_{k\to\infty} ||x(kt_d) - x^r||_2 = 0$.

Proof: The proof is included in the Appendix 3.C. ■

3.5 Academic Case Study

The methodologies developed in this chapter are demonstrated by developing a digital twin and applying the soft-sensor based controller on the academic case study shown in Figure 3.6.



(a) Cross-sectional Schematics (b) Graph Structure

Figure 3.6: Example of a thermo-fluidic process. ● represents a solid component, ● represents a liquid component for inlet. ▬ is an edge describing conductive exchange of thermal energy between two solid components or a solid and liquid component. ▬ is an edge describing convective thermal energy due to the inflow of liquid along the inlet channels. ▬ is an edge describing convective thermal energy due to the re-circulation of liquid along the return channels.

Here, the liquid tank $L1$ has a time-varying inflow of $Q_{in}(t)$ m^3/s and outflow of $Q_r(t)$. Every individual nozzle $Ln1$, $Ln2$, $Ln3$ has liquid return flow with a time-varying inflow rate $Q_{Ni}(t)$, and return flow rate $Q_{Oi}(t)$ for $i \in \{1, 2, 3\}$. For assigning the job-specific flow parameter, liquid channels are considered as pipes.

The fluid exhibits in-compressible nature and the continuous flow of the liquid causes no change of the liquid level inside each channel. Thus the mass balance equation results in the following algebraic relations.

$$\begin{aligned} Q_{in}(t) =& Q_r(t) + \sum_{i=1}^{3} Q_{Ni}(t), \\ Q_{Ni}(t) =& Q_{Oi}(t); \quad i = 1, \cdots, 3. \end{aligned} \tag{3.14}$$

3.5.1 Implementing the Digital Twin

Graph

There are six nodes; $\mathfrak{N} := \{\mathfrak{N}_{S1}, \mathfrak{N}_{S2}, \mathfrak{N}_{L1}, \mathfrak{N}_{Ln1}, \mathfrak{N}_{Ln2}, \mathfrak{N}_{Ln3}\}$.

Topology

The adjacent matrix $A \in \mathbb{R}^{6\times 6}$ is defined as follows

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.15}$$

It is evident that the topology of Lni is identical for $i = \{1, 2, 3\}$. In other words, topologically, there is no difference in the interconnection between any of the three nozzles and the rest of the graph. This is a major advantage for building graph-theoretic model.

Node dynamics

Using the equivalence between electrical and thermo-fluidic domain in Appendix 3.A, in the following items, the thermo-fluidic processes are derived per node:

$\mathfrak{N}_{S1}$: Its temperature T_{S1} ($^\circ C$) is influenced by convection with $\mathfrak{N}_{L1}$ and conduction with $\mathfrak{N}_{S1}$ according to the following equation of energy conservation

$$C_{S1}\frac{dT_{S1}}{dt} = H_{d,1}(T_{S2} - T_{S1}) + H_{v,1}(T_{L1} - T_{S1}).$$

Substituting the expressions of $H_{d,1}, H_{v,1}$, one obtains

$$\underbrace{m_{S1}c_{S1}\frac{dT_{S1}}{dt}}_{Thermal\ storage} = \underbrace{\frac{1}{R_{tot}}}_{Conduction}(T_{S2}-T_{S1}) + \underbrace{\frac{Nu_{L1}\cdot k_{L1}}{D_{hL1}}A_{S1-L1}}_{Convection}(T_{L1}-T_{S1}). \tag{3.16}$$

Here, the equivalent thermal resistance is $R_{tot} = \frac{l_{S1}}{A_{S1-S2}k_{S1}} + \frac{l_g}{A_{S1-S2}k_g} + \frac{l_{S2}}{A_{S1-S2}k_{S2}}$.

$\mathfrak{N}_{L1}$: Its temperature T_{L1} ($^\circ C$) is influenced by advection and convection according to the following equation of energy conservation

$$C_{L1}\frac{dT_{L1}}{dt} = H_{v,1}(T_{S1}-T_{L1}) + H_{v,2}(T_{S2}-T_{L1}) + H_{a,1}(t)(T_{in}-T_{L1}).$$

In other words,

$$\begin{aligned}\underbrace{m_{L1}c_{L1}\frac{dT_{L1}}{dt}}_{Thermal\ storage} =& \underbrace{\frac{Nu_{L1}\cdot k_{L1}}{D_{hL1}}A_{S1-L1}}_{Convection}(T_{S1}-T_{L1}) + \underbrace{\frac{Nu_{L1}\cdot k_{L1}}{D_{hL1}}A_{S2-L1}}_{Convection}(T_{S2}-T_{L1}) \\ &+ \underbrace{\rho_{in}c_{in}(Q_r(t)+\sum_{i=1}^{3}Q_{Ni}(t)}_{Advection}(T_{in}-T_{L1})).\end{aligned} \tag{3.17}$$

$\mathfrak{N}_{Lni}$: For individual liquid nozzle $\mathfrak{N}_{Lni}$ $i \in \{1,2,3\}$, the governing equation is identical. Hence, as an example, the governing equation of temperature T_{Lni} ($^\circ C$) for node $\mathfrak{N}_{Lni}$ is

$$C_{Lni}\frac{dT_{Lni}}{dt} = H_{v,ni}(T_{S2}-T_{Lni}) + H_{a,ni}(t)(T_{L1}-T_{Lni}).$$

$$\underbrace{m_{Lni}c_{Lni}\frac{dT_{Lni}}{dt}}_{Thermal\ storage} = \underbrace{\frac{Nu_{Lni}\cdot k_{Lni}}{D_{hLni}}A_{S2-Lni}}_{Convection}(T_{S2}-T_{Lni}) + \underbrace{\rho_{L1}c_{L1}Q_{Ni}(t)}_{Advection}(T_{L1}-T_{Lni}). \tag{3.18}$$

To obtain the thermo-fluidic model of every individual nozzle, one simply has to repeat (3.18) by substituting the physical parameters with index $i \in \{1,2,3\}$.

$\mathfrak{N}_{S2}$: This node is connected with all of the other nodes. Similar to $\mathfrak{N}_{S1}$, it temperature, T_{S2} ($^\circ C$), is governed by the following equation of energy conservation:

$$C_{S2}\frac{dT_{S2}}{dt} = H_{d,1}(T_{S1}-T_{S2}) + H_{v,2}(T_{L1}-T_{S2}) + \sum_{i=1}^{3}H_{v,ni}(T_{Lni}-T_{S2})$$

$$\underbrace{m_{S2}c_{S2}\frac{dT_{S2}}{dt}}_{Thermal\ storage} = \underbrace{\frac{1}{R_{tot}}}_{Conduction}(T_{S1}-T_{S2}) + \underbrace{\frac{Nu_{L1}\cdot k_{L1}}{D_{hL1}}A_{S2-L1}}_{Convection}(T_{L1}-T_{S2}) + \sum_{i=1}^{3}\underbrace{\frac{Nu_{Lni}\cdot k_{Lni}}{D_{hLni}}A_{S2-Lni}}_{Convection}(T_{Lni}-T_{S2}) \quad (3.19)$$

Remark 3.4 *By construction, modularity is a key attribute in the thermo-fluidic process. As the dynamics and topological interconnection of every individual nozzle is identical with respect to the entire graph, the digital twin simply requires repetition of identical models based on the number of nozzles.*

State-space representation of a node

For example, for $\mathfrak{N}_{S1}$, the temperature evolution can be re-written as

$$\begin{bmatrix}\dot{T}_{S1}\\ n_{S1,S2}\\ n_{S1,L1}\\ q_{S1}\end{bmatrix} = \begin{bmatrix}\frac{(H_{d,1}+H_{v,1})}{C_{S1}} & \frac{H_{d,1}}{C_{S1}} & \frac{H_{v,1}}{C_{S1}} & 1 & 0\\ 1 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}T_{S1}\\ m_{S1,S2}\\ m_{S1,L1}\\ p_{S1}\end{bmatrix},$$

$$\begin{bmatrix}m_{S1,S2}\\ m_{S1,L1}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}n_{S1,S2}\\ n_{S1,L1}\\ n_{S2,S1}\\ n_{L1,S1}\end{bmatrix},$$

$$p_{S1} = q_{S1}. \quad (3.20)$$

Interconnection Structure

The interconnection relation $\breve{m}^{xx} = \breve{M}^{xx}\breve{n}^{xx}$ is given below.

$$\underbrace{\begin{bmatrix}m_{S1,S2}\\ m_{S1,L1}\\ m_{S2,S1}\\ m_{S2,L1}\\ m_{S2,Ln1}\\ m_{L1,S_1}\\ m_{L1,S2}\\ m_{Lni,S2}\\ m_{Lni,L1}\end{bmatrix}}_{\breve{m}^{xx}} = \underbrace{\begin{bmatrix}0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0\\ 1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0\end{bmatrix}}_{\breve{M}^{xx}}\underbrace{\begin{bmatrix}n_{S1,S2}\\ n_{S1,L1}\\ n_{S2,S1}\\ n_{S2,L1}\\ n_{S2,Ln1}\\ n_{L1,S1}\\ n_{L1,S2}\\ n_{L1,Lni}\\ n_{Lni,S2}\end{bmatrix}}_{\breve{n}^{xx}} \quad (3.21)$$

For brevity, only one nozzle, $\mathfrak{N}_{Lni}$, is considered and it can be repeated for $i \in \{1,2,3\}$. This is again a major advantage for building the digital twin. Irrespective of the number of nozzles, as they have identical dynamics and identical topology, up-scaling the model is a straightforward task.

Now, one can easily build the representation by eliminating signals $(\breve{m}^{xx}, \breve{n}^{xx})$ or $(\breve{p}, \breve{q})$.

Determining an equivalent representation

$$\begin{bmatrix} \dot{T}_{S1}(t) \\ \dot{T}_{S2}(t) \\ \dot{T}_{L1}(t) \\ \dot{T}_{Lni}(t) \end{bmatrix} = \begin{bmatrix} -\frac{(H_{d,1}+H_{v,1})}{C_{S1}} & 0 & 0 & 0 \\ 0 & -\frac{H_{d,1}+H_{v,2}+H_{v,n1}}{C_{S2}} & 0 & 0 \\ 0 & 0 & -\frac{H_{v,1}+H_{v,2}}{C_{L1}} & 0 \\ 0 & 0 & 0 & -\frac{H_{v,ni}}{C_{Lni}} \end{bmatrix} \begin{bmatrix} T_{S1}(t) \\ T_{S2}(t) \\ T_{L1}(t) \\ T_{Lni}(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \breve{p}(t)$$

$$+ \begin{bmatrix} \frac{H_{d,1}}{C_{S1}} & \frac{H_{v,1}}{C_{S1}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{H_{d,1}}{C_{S2}} & \frac{H_{v,2}}{C_{S2}} & \frac{H_{v,n1}}{C_{S2}} & \frac{1}{C_{S2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{H_{v,1}}{C_{L1}} & \frac{H_{v,2}}{C_{L1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{H_{v,ni}}{C_{Lni}} & 0 & \frac{1}{C_{Lni}} & \end{bmatrix} \begin{matrix} \breve{m}^{xx}(t), \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \tag{3.22}$$

$$\breve{q}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\rho_{L1}c_{L1}}{C_{L1}} & 0 \\ 0 & 0 & 0 & -\frac{\rho_{L1}c_{L1}}{C_{Lni}} \end{bmatrix} \begin{bmatrix} T_{S1}(t) \\ T_{S2}(t) \\ T_{L1}(t) \\ T_{Lni}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\rho_{L1}c_{L1}}{C_{Lni}} & 0 & \end{bmatrix} \begin{matrix} \\ \\ \breve{m}^{xx}(t), \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \tag{3.23}$$

$$\breve{n}^{xx}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{S1}(t) \\ T_{S2}(t) \\ T_{L1}(t) \\ T_{Lni}(t) \end{bmatrix}, \breve{y}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_{S1}(t) \\ T_{S2}(t) \\ T_{L1}(t) \\ T_{Lni}(t) \end{bmatrix}, \breve{p}(t) = \underbrace{\begin{bmatrix} 1 & 0 & & & 0 \\ 0 & 1 & & & 0 \\ & & & & \\ 0 & 0 & Q_r(t) + \sum_{i=1}^{3} Q_{Ni}(t) & & 0 \\ & & & & \\ 0 & 0 & & & Q_{Ni}(t) \end{bmatrix}}_{\breve{\Theta}(t)} \breve{q}(t). \tag{3.24}$$

3.5.2 Soft Sensing Based Model Predictive Control

The MPC scheme is applied to the digital twin of the system described in figure 3.6. The objective of the controller is to achieve a fixed temperature of 70°C at at each of the three nodes $\{\mathfrak{N}_{Ln1}, \mathfrak{N}_{Ln2}, \mathfrak{N}_{Ln3}\}$. To this end, the following specifications are considered:

Control Scenarios

Two scenarios are conceptualized in terms of choosing on which nodes the control input (in terms of thermal power, Watt) is applied:

- Scenario 1: Individual heating inputs are applied on the solid node $\mathfrak{N}_{S1}$.
- Scenario 2: Individual heating inputs are applied on each of the three liquid nodes $\{\mathfrak{N}_{Ln1}, \mathfrak{N}_{Ln2}, \mathfrak{N}_{Ln3}\}$.

Setting Up the Digital Twin of the Model

1. The job-specific flow parameter are given in figure 3.7. The recirculation flow rate $Q_r(t)$ is considered to be zero.

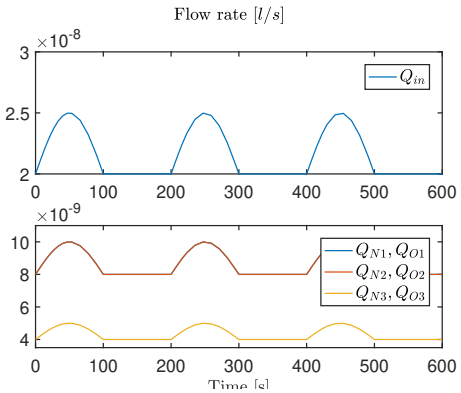


Figure 3.7: Assigned flow parameters. For $\mathfrak{N}_{Ln1}$ and $\mathfrak{N}_{Ln2}$, the flow parameters are chosen to be identical

2. With known parameter trajectories, the model of every individual node is considered in discrete time by choosing the $\mathbb{T} = \{kt_d \mid k \in \mathbb{N} \cup \{0\}\}$ with $t_d = 0.01$ seconds. Euler's approach is used for time discretization due to its sparse and structure preserving implementation ([92], page 4).

3. For implementing the soft-sensor, for every individual $\mathfrak{N}_{Lni}$, same energy-temperature curve as shown in figure 3.5 is parametrized as (3.2). As there is no real-time data yet available in the simulation, the digital twin simulates

the thermo-fluidic model over time, and the temperature of every individual nozzle is subsequently cross-evaluated using the soft-sensor.

4. The sets $\mathbb{S}_p(t)$ and $\mathbb{S}_s(t)$ is chosen to include every individual $\mathfrak{N}_{Lni}$. Based on whether $\mathfrak{N}_{S1}$ or every individual $\mathfrak{N}_{Lni}$ is heated for control, $\mathbb{S}_h(t)$ is determined. This determines the matrices $S^h(t), S^p(t)$, $S^s(t), S^u(t)$. The model for control (3.5) is built subsequently. The unmeasured states are automatically replaced by the corresponding state-updates from the model.

5. The maximum allowable control input is chosen to be $u_{\max} = 20$ Watt in control scenario 1. In control scenario 2, $u_{\max} = 1$ Watt.

Solving MPC Scheme

Using (3.5), The prediction model is built over the time horizon $t \in \mathbb{T}_N^k$ where $\mathbb{T}_N^k := \{kt_d \mid k \in \mathbb{N}_{[k,N+k]}\}$. Here, $N = 10$. In (3.10), the weights Q and $R_{i|k}$ are chosen as diagonal matrices. Here, the diagonal entries in $R_{i|k}$ are chosen significantly higher than that of Q to strictly penalize the deviation of inputs from its reference values.

The optimization problem in (3.12) is solved by the freely available `mpcqpsolver` using the interior point method. Once the optimization yields the optimal control inputs, they are applied to the heating piezoelectric actuators by means of non-jetting voltage pulses. The MPC then repeats the same procedure over the entire bit-map iteratively.

Results

Once the MPC is applied, the closed loop responses are shown in figure 3.8 and figure 3.9 for control scenario 1 and 2 respectively. It demonstrates that the performance specifications are met while satisfying the constraints. Comparing the results of control scenario 1 and 2, it is observed that scenario 2 outperforms scenario 1 due to the presence of collocated configuration of sensor and actuator at every individual $\mathfrak{N}_{Lni}$. This is also evident by comparing the tracking error in Table 3.2. It has not been yet explored what can be a suitable actuation mechanism that needs to be employed for heating every individual $\mathfrak{N}_{Lni}$. Nevertheless, the conceptualization of control scenario 2 and its better performance will be crucial for successful application of the developed methods in inkjet printing.

Table 3.2: Comparison of performance between two control scenarios

Controller Scenario	Root Mean Square Error [$^\circ C$]			Max Absolute Error [$^\circ C$]		
	$\mathfrak{N}_{Ln1}$	$\mathfrak{N}_{Ln2}$	$\mathfrak{N}_{Ln3}$	$\mathfrak{N}_{Ln1}$	$\mathfrak{N}_{Ln2}$	$\mathfrak{N}_{Ln3}$
1	0.767	0.953	0.849	0.89	1.1210	0.9212
2	0.0026	0.0027	0.0072	0.0069	0.0102	0.0158

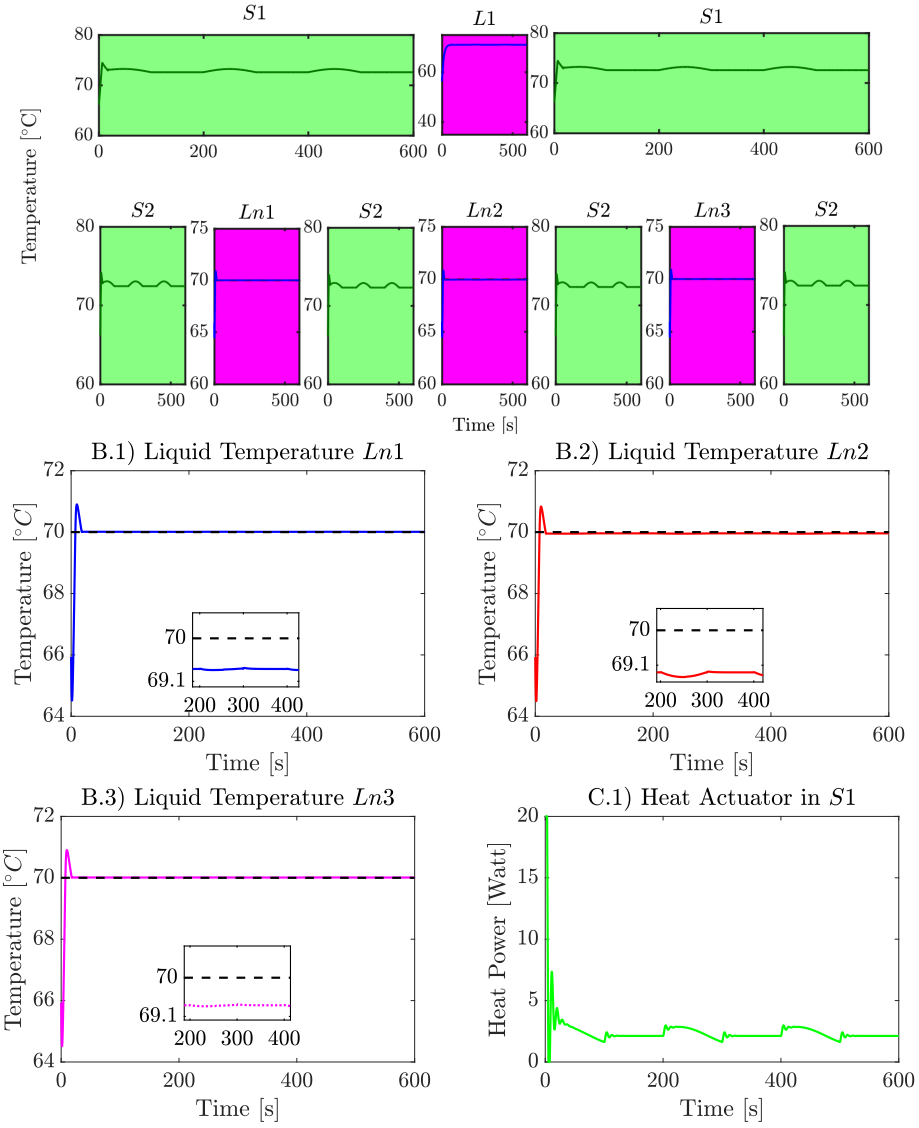


Figure 3.8: Closed loop simulation results of control scenario 1 with heating inputs applied to $\mathfrak{N}_{S1}$

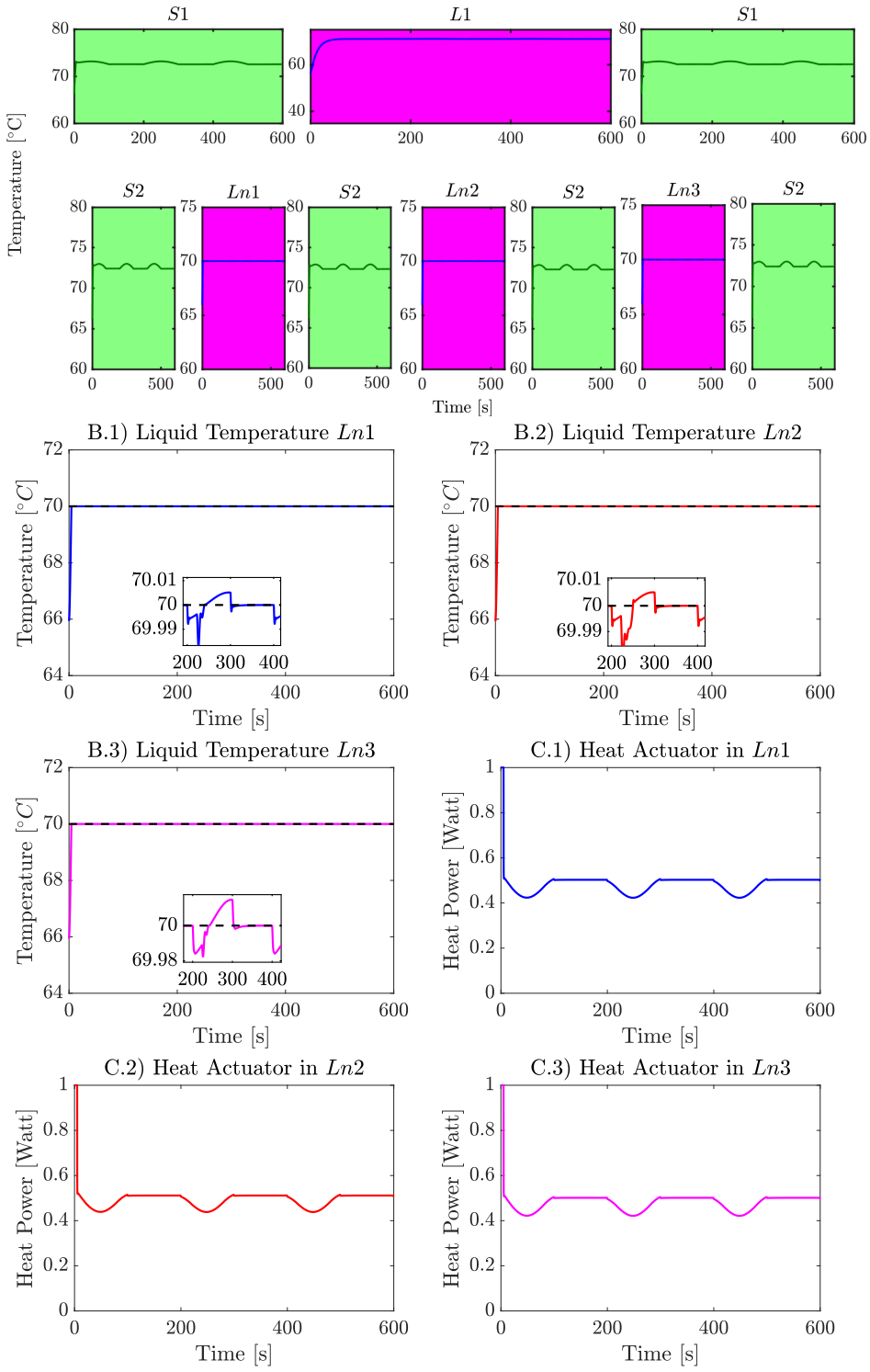


Figure 3.9: Closed loop simulation results of control scenario 2 with heating inputs applied to every individual $\mathfrak{N}_{Lni}$.

3.6 Closing Remarks

Lumping based finite dimensional model is used in graph-theoretic framework to develop a modular, flexible and versatile digital twin of the thermo-fluidic processes. Owing to modular structure of the graph, upscaling the digital twin in case of identically modeled nodes is a major advantage of the presented methodology. Moreover, the digital twin is equipped with four equivalent representations of the thermo-fluidic model and the user has the complete freedom to choose any one of these for simulations, fault diagnosis, monitoring and control.

One major challenge of the thermo-fluidic processes is the restrictions of not using additional sensors and additional actuators. To realize the possibility of applying feedback controller in case of no measured outputs, a soft sensing mechanism is presented that uses the presence of collocated piezoelectric elements to estimate liquid temperature in a corresponding channel. Using the developed soft-sensors, a reference tracking model predictive control scheme is developed for the digital twin. While implementing the controller, user has the flexibility to change the locations of the control inputs and the measured outputs. The digital twin is implemented for the academic case study and the controller is applied. It is observed that collocated sensing and actuation in every individual liquid channel provides better performance than applying control inputs to the solid components.

Appendices

3.A Equivalence Between Thermo-Fluidic and Electrical Domain

Table 3.A1 summarizes the types of thermal power in a thermo-fluidic process and corresponding equivalence in electrical domain.

Table 3.A1: Equivalence between thermo-fluidic and electrical domain

Type	Phenomena	Electrical current	Electrical component
Dissipation	Conduction	$\frac{\Delta T_s}{(\frac{l_s}{k_s A_s})}$	Resistance = $\frac{l_s}{k_s A_s}$
	Convection	$\frac{\Delta T_{sl}}{(\frac{D_l}{Nuk_l A_l})}$	Resistance = $\frac{D_l}{Nuk_l A_l}$
	Advection	$\frac{\Delta T_l}{(\frac{1}{\rho_l c_l Q_l})}$	Resistance = $\frac{1}{\rho_l c_l Q_l}$
Storage	Capacity	$m_s c_s \frac{dT_s}{dt}$ or $m_l c_l \frac{dT_l}{dt}$	Capacitance = $m_s c_s$ or $m_l c_l$
Dissipation	Valve	$\frac{\Delta p}{R}$	Resistance = $\frac{8\mu l}{\pi r^4}$
Storage	Tank	$\frac{A}{\rho g}\frac{dp}{dt}$	Capacitance = $\frac{A}{\rho g}$
Storage	Mass	$\frac{\rho L}{A}\frac{dQ}{dt}$	Inductance = $\frac{\rho L}{A}$

Here, ΔT_s, ΔT_l, and ΔT_{sl} represents the difference of temperature between two solid components, between two liquid components and between a solid and a liquid component respectively. The temperature of solid component and the liquid component are T_s and T_l. For a fluid, Δp denotes the gradient of pressure p and Q denotes the volumetric flow. The physical parameters for solid and liquid (s, l) are either known for a specific material or derived experimentally (e.g. see [72]).

3.B Proof of Algorithm 3.1

Proof: Consider a class of stable, second-order, single-input-single-output systems that has the following representation:

$$\begin{aligned} z(n+1) =& A_s z(n) + B_s w(n), \\ y(n) =& C_s z(n). \end{aligned} \tag{3.25}$$

Here, for $n \in \mathbb{N} \cup \{0\}$, $z(n) \in \mathbb{R}^2$ is a vector with two internal states and $w(n) \in \mathbb{R}$ is the applied impulsive input. The output signal $y(n) \in \mathbb{R}$ is modeled as (3.1).

$$y(n) = \alpha e^{-\zeta n T_s} \sin(\omega n T_s + \phi) + \gamma; \qquad n \in \mathbb{N}.$$

Step 1: Determining the poles

The first task is to find an estimate of the state matrix $A_s \in \mathbb{R}^{2\times 2}$. To this end, the Hankel matrix is constructed using the data $\{s_n, n \in \mathbb{N}_{[1,N]}\}$ as follows:

$$H = \begin{bmatrix} s_1 & \cdots & s_{L-1} & s_L \\ s_2 & \cdots. & s_L & s_{L+2} \\ \vdots & \ddots & \vdots & \vdots \\ s_{N-L} & \cdots. & s_{N-2} & s_{N-1} \\ s_{N-L+1} & \cdots & s_{N-1} & s_N \end{bmatrix} = \mathcal{O}_{N-L+1}\mathcal{R}_L,$$

where, the observability matrix $\mathcal{O}_{N-L+1}$ and the reachability matrix $\mathcal{R}_L$ are defined as

$$\mathcal{R}_L := \begin{bmatrix} B_s & A_s B_s & \cdots & A_s^{L-1} B_s \end{bmatrix}, \mathcal{O}_{N-L+1} := \begin{bmatrix} C_s \\ C_s A_s \\ \vdots \\ C_s A_s^{N-L} \end{bmatrix}.$$

Remark 3.5 *(shift property of observability matrix)*

$$\mathcal{O}^1_{N-L+1} A_s = \mathcal{O}^2_{N-L+1},$$

where, $\mathcal{O}^1_{N-L+1} = \begin{bmatrix} C_s \\ C_s A_s \\ \vdots \\ C_s A_s^{N-L-1} \end{bmatrix}$ *and* $\mathcal{O}^2_{N-L+1} = \begin{bmatrix} C_s A_s \\ \vdots \\ C_s A_s^{N-L} \end{bmatrix}$.

As, we are interested in a second order system, let the optimal 2-rank approximation of the Hankel H be given by

$$H \approx U \Sigma V^{\mathrm{H}}, \;\; \Sigma := \mathrm{diag}(\sigma_1, \sigma_2), \;\; \sigma_1, \sigma_2 > 0.$$

In other words,

$$H \approx \underbrace{U\Sigma^{\frac{1}{2}}}_{\mathcal{O}_{N-L+1}} \underbrace{\Sigma^{\frac{1}{2}}V^{\mathrm{H}}}_{\mathcal{R}_L}.$$

Now, using the shift property we obtain:

$$\tilde{A}_s = \Sigma^{-\frac{1}{2}}U^{1\dagger}U^2\Sigma^{\frac{1}{2}},$$

where $\tilde{A}_s = T^{-1}A_sT$ for an unknown matrix T. As the poles of the system does not change of under similarity transformation, the two poles of A_s are found by solving the eigen value ρ_k and right eigen vector v_k, $k = 1, 2$:

$$(\tilde{A}_s - \rho_k)v_k = 0$$

Determine ω and ζ

Using the system poles, we find the frequency ω and the damping ζ as follows:

- Natural Frequency: $\omega = \mathrm{Im}(\ln\rho_k), k = 1$ or 2.
- Damping : $\zeta = \mathrm{Re}(\ln\rho_k), k = 1$ or 2.

Determine α, ϕ, and γ

The amplitude α, the phase ϕ and the shift γ are found by reconstructing the signal s_n using computed values of ω, ζ. In particular:

$$\begin{aligned} y(n) &= \alpha e^{-\zeta nT_s}\sin(\omega nT_s + \phi) + \gamma = (a+jb)e^{j(\omega+j\zeta)nT_s} + (a+jb)^* e^{-j(\omega-j\zeta)nT_s} + \gamma. \\ &= a\Big(e^{j(\omega+j\zeta)nT_s} + e^{-j(\omega-j\zeta)nT_s}\Big) + jb\Big(e^{j(\omega+j\zeta)nT_s} - e^{-j(\omega-j\zeta)nT_s}\Big) + \gamma. \end{aligned} \tag{3.26}$$

Here, $a = \frac{\alpha}{2}\sin\phi$ and $a = \frac{\alpha}{2}\cos\phi$ are the unknowns coefficients and γ is the unknown offset. Using the data at samples $\{0, \cdots, N-1\}$, the unknowns can be found by solving the following linear equations:

$$\mathbf{s} = \mathbf{\Gamma}\mathbf{x}, \tag{3.27}$$

Here, the unknowns are $\mathbf{x} := \mathrm{col}(a, b, \gamma)$, $\mathbf{s} := \mathrm{col}(s_1, \cdots, s_N)$ is the data. Moreover the matrix is defined as

$$\mathbf{\Gamma} := \mathrm{col}\Big(\begin{bmatrix} (e^{j(\omega+j\zeta)nT_s} + e^{-j(\omega-j\zeta)nT_s}) & j(e^{j(\omega+j\zeta)nT_s} - e^{-j(\omega-j\zeta)nT_s}) & 1 \end{bmatrix} \Big)_{n\in\mathbb{N}_{[1,N]}}$$

that depends on ω and ζ, and . An unbiased minimum variance solution of (3.27), $\hat{\mathbf{x}} := \mathrm{col}(\hat{a}, \hat{b}, \hat{\gamma})$, accepts the following analytic expression:

$$\hat{\mathbf{x}} = (\mathbf{\Gamma}^{\mathrm{H}}\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^{\mathrm{H}}\mathbf{s}. \tag{3.28}$$

Using computed $\hat{\mathbf{x}}$, the amplitude (α) and phase (ϕ) of the signal (3.1) are computed as follows:

- Amplitude: $\alpha = 2\sqrt{\text{Re}(\hat{a} + j\hat{b})^2 + \text{Im}(\hat{a} + j\hat{b})^2}$.
- Phase : $\phi = \text{Im}(\ln(\hat{a} + j\hat{b}))$.
- Offset : $\gamma = \hat{\gamma}$.

The computed values of α, ζ, ω, γ and ϕ parametrize and reconstruct the signal $y(n)$ in (3.1). ■

3.C Proof of Theorem 3.2

Proof: Regarding the presented tracking control problem, asymptotic stability is understood with respect to the following non-autonomous error system:

$$\Big(x((k+1)td) - x^r\Big) = \tilde{A}(kt_d)\Big(x(kt_d) - x^r\Big) + \tilde{B}(kt_d)\Big(u^h(kt_d) - u^r(kt_d)\Big). \tag{3.29}$$

The asymptotic stability of the (3.29) in closed-loop amounts to verifying whether there exists $\delta > 0$ such that $\lim_{k\to\infty} ||x(kt_d) - x^r|| = 0$ for all initial condition $||x(0) - x^r|| < \delta$ while $u^h(kt_d)$ is applied by solving the MPC problem. To this end, using Lyapunov theory (c.f. [18]), we show that the MPC cost functional $J(k, \cdot, \cdot)$ at instant k, once substituted with unique minimizer $\bar{x}^*(k) := \text{col}(x^*_{1|k}, \cdots x^*_{N|k})$ and $\bar{u}^{h*}_k := \{u^{h*}_{0|k}, \cdots, u^{h*}_{N-1|k}\}$, is a candidate Lyapunov function.

It is now possible to construct an input and state trajectory $\bar{x}^{\text{f}}(k+1), \bar{u}^{\text{f}}(k+1)$ such that the MPC problem at time $k+1$ is feasible (not necessarily optimal) with cost $J^{\text{f}}(k+1, \bar{x}^{\text{f}}(k+1), \bar{u}^{\text{f}}(k+1))$. Here,

$$\bar{x}^{\text{f}}(k+1) = \text{col}\Big(x^*_{2|k}, \cdots x^*_{N|k}, x^{\text{f}}_{N|k+1}\Big),$$

$$\bar{u}^{\text{f}}(k+1) = \text{col}\Big(u^{h*}_{1|k}, \cdots u^{h*}_{N-1|k}, u^{\text{f}}_{N-1|k+1}\Big),$$

where, $x^{\text{f}}_{N|k+1}, u^{\text{f}}_{N-1|k+1}$ are future state and input to be determined by MPC at iteration $k+1$. Similar to the dual mode formulation as proposed in [90], we construct the predicted terminal input at time step $k+1$ as a stabilizing state feedback law for (3.29). In other words,

$$(x^{\text{f}}_{N|k+1} - x^r) = \Big(\tilde{A}_{N|k} + \tilde{B}_{N|k}K_k\Big)(x^*_{N|k} - x^r).$$

$$u^{\mathrm{f}}_{N-1|k+1} = K_k(x^*_{N|k} - x^r) + u^r_{N|k}.$$

Owing to the convexity and positivity of the objective function, for proving stability, it is sufficient to show that the feasible cost functional is contractive over time samples. In other words,

$$J^{\mathrm{f}}(k+1, \bar{x}^{\mathrm{f}}(k+1), \bar{u}^{\mathrm{f}}(k+1)) < J(k, \bar{x}^*(k), \bar{u}^{h*}(k))$$

Substituting the respective expressions, we obtain

$$\begin{aligned}
&J^{\mathrm{f}}(k+1, \bar{x}^{\mathrm{f}}(k+1), \bar{u}^{\mathrm{f}}(k+1)) - J(k, \bar{x}^*(k), \bar{u}^{h*}(k)) \\
&= -||x_{0|k} - x^r||^2_Q - ||u^h_{0|k} - u^r_{0|k}||^2_{R_{0|k}} - ||x^*_{N|k} - x^r||^2_{P_k} \\
&\quad + ||x^*_{N|k} - x^r||^2_Q + ||K_k(x^*_{N|k} - x^r)||^2_{R_{N|k}} + ||(A_{N|k} + B_{N|k}K_k)(x^*_{N|k} - x^r)||^2_{P_k}.
\end{aligned} \tag{3.30}$$

Since the first two terms in the RHS of (3.30) is negative, we require to satisfy the following inequality for the contraction of the cost functional:

$$(\tilde{A}_{N|k} + \tilde{B}_{N|k}K_k)^\top P_k(\tilde{A}_{N|k} + \tilde{B}_{N|k}K_k) - P_k \preccurlyeq -Q - K_k^\top R_{N|k}K_k, \qquad P_k \succ 0$$

Using changes of variables $X_k = P_k^{-1}$, $Y_k = K_k P_k^{-1}$ and using the rule of Schur complement we obtain the following LMI:

$$\begin{bmatrix}
-X_k & 0 & \tilde{A}_{N|k}X_k + \tilde{B}_{N|k}Y_k & 0 \\
0 & -R^{-1}_{N|k} & Y_k & 0 \\
(\tilde{A}_{N|k}X_k + \tilde{B}_{N|k}Y_k)^\top & Y_k^\top & -X_k & X_k \\
0 & 0 & X_k & -Q^{-1}
\end{bmatrix} \preccurlyeq 0, \qquad X_k \succ 0.$$

The corresponding control input $u^{\mathrm{f}}_{N-1|k+1} = K_k(x^*_{N|k} - x^r) + u^r_{N|k}$ should also be an admissible control action. In other words, $u^{\mathrm{f}}_{N-1|k+1}$ should satisfy the following constraint:

$$E_{N|k}(K_k(x^*_{N|k} - x^r) + u^r_{N|k}) < b$$

This amounts to the following terminal constraint on $x_{N|k}$:

$$E_{N|k}K_k x_{N|k} < b - u^r_{N|k} + E_{N|k}K_k x^r$$

■

Part III

Computation with Approximation

4 CHAPTER

Approximating Diffusive Thermo-Fluidic Processes

This work presents a novel and generic framework to describe, model and solve a network of thermo-fluidic processes over an arbitrary interconnection in space. Here, a graph-theoretic framework to model thermo-fluidic diffusion processes are viewed as a network of distributed parameter systems. A three-step procedure has been proposed to reduce such a network to a finite dimensional system while preserving the boundary conditions. The first step in this method is to separate the effect of external boundary inputs and in-domain inputs in the solution. In the second stage, multivariable solutions are decomposed and approximated as spectral expansions of spatially and temporally varying functions. The third step involves deriving reduced order solutions using approximation techniques. We specifically exploit the orthogonality of the basis functions to obtain an optimal reduced order model. The viability of the developed procedure is illustrated through simulation results under physically realistic scenario.

Outline

4.1 Introduction **82**
4.2 Specifications for Diffusive Thermo-Fluidic Processes **83**
4.3 Finite-Dimensional Approximation of Diffusive Thermo-Fluidic Processes **85**
4.4 Academic Case Study: Heat Diffusion in Composite **93**
4.5 Closing Remarks **97**

4.1 Introduction

The model that describes thermo-fluidic processes is often derived using conservation laws of mass and energy. However, by definition, these processes involve coupled behavior of solid and fluid substances (see [66]). Due to the difference in their physical structure, the energy exchange between a solid body and fluid is significant when their physical interaction is allowed. In this chapter, the focus is on a specific class of such thermo-fluidic processes where the mutual interaction among various solids and fluids is dominantly diffusive. The diffusive process is typically defined by a phenomenon in which matter is transported from one part of the matter (higher concentration) to another part (lower concentration) as a result of random motions of the molecules (see [56]). The diffusive thermo-fluidic processes are observed in numerous engineering applications. For example, in the commercial printing and packaging industries, the drying of paper or cardboard typically involves heat and moisture diffusion through composite materials [9]. In food processing technology, the drying of food items consists of the diffusion of liquid ingredients under the influence of external heat sources. In semiconductor industries, the doping of impurities in an intrinsic semiconductor involves diffusion of n or p-type species in the different layer of semiconducting materials [75]. In these applications in mind, there are three salient features of the underlying processes.

- The processes are dominantly diffusive in nature.
- There exist interactive spatio-temporal phenomena among multiple physical quantities that are possibly coupled.
- There exists simultaneous interaction of various materials (solids and fluids) in a spatially distributed structure.

The models that describe the diffusive thermo-fluidic processes are governed by a set of coupled linear partial differential equations (PDEs), well-known as diffusion-transport-reaction equations (see [66]). A detailed derivation of such PDEs using conservation laws of mass and energy can be found in [16]. In practice, these PDEs are commonly approximated using numerical schemes that result in a finite-dimensional model that approximates the infinite-dimensional behavior of the underlying process. Moreover, due to the parabolicity of these models, approximation techniques are typically consistent and convergent, resulting in a well-posed finite-dimensional model [82]. As a result, once these models are approximated into finite-dimensional systems, engineers have numerous well-developed methods at their disposal to choose from and build the digital twin's functionalities on the basis of an approximated model of diffusive thermo-fluidic processes.

Despite the well-established theories on the analysis and approximation of diffusion-transport-reaction type PDEs, little research has been carried out that explicitly takes the spatial interconnection and corresponding energy exchange into account during approximation. For example, in [43], [10], the model for

chemical processes is modeled as a diffusion equation. But the coupled effect is neglected. In [21], [79], the coupled heat and moisture diffusion is considered. Still, this model has not been extended to the diffusion processes in spatially interconnected networks. In [7] and the references therein, a detailed account is provided of various numerical approximation and reduction techniques. However, most of these techniques are focused on methods that are only applicable to the decoupled, single variable models and the models without spatial interconnection.

Contribution of this Chapter

This chapter proposes a computational procedure to approximate diffusive thermo-fluidic processes in an arbitrary geometrical configuration. When the physical parameters of the thermo-fluidic processes are known, a three-step procedure is developed to derive an approximated model that preserves the interaction among connected thermo-fluidic processes. Specifically, developed numerical scheme is consistent with respect to the boundary conditions that describes the mutual interactions among different thermo-fluidic processes. Once the approximated model is developed, it can be used for building functionalities of the digital twin for the assets that are governed by diffusive thermo-fluidic processes.

4.2 Specifications for Diffusive Thermo-Fluidic Processes

In Chapter 2, the digital twin's model describing thermo-fluidic processes is formulated as a finite graph, according to (D.1)-(D.3) (on page 23-24). The graph consists of lumped nodes and spatially distributed nodes that are mutually interconnected in arbitrary spatial structure. In this Chapter, the graph is considered to have only spatially distributed components and every individual of them is governed by coupled diffusion-transport-reaction equations. Following the notations and definitions In Chapter 2, the following relaxations are imposed on (D.1)-(D.3):

- All the Nodes $\mathfrak{N}_i, i \in \mathbb{N}_{[1,m]}$ are spatially distributed nodes.
- $\mathbb{X}_i$ is non empty and $\mathbb{X}_i^{\text{bc}} = \{a_i, b_i\}$ for all $i \in \mathbb{N}_{[1,m]}$.
- Whenever $A_{i,j} = 1$, the interconnection domain $\mathbb{X}_{i,j}^I \subseteq \mathbb{X}_i \cap \mathbb{X}_j$ is defined on the common boundary points. Hence, $\mathbb{X}_{i,j}^I \subseteq \{a_i = b_j, b_i = a_j\}$.
- For every individual node $\mathfrak{N}_i$ according to (D.2), state-variables only consist of functions that maximally twice differentiable with respect to the spatial variable. Hence, all the governing PDEs are parabolic.
- The effect of boundary conditions in the dynamics are excluded.

- Output operators (both measured and regulated) only include bounded integral operators, excluding the boundary values.
- From the boundary conditions, periodic boundary conditions are excluded.
- There is no interconnection between lumped and spatially distributed nodes.
- Between $\mathfrak{N}_i, \mathfrak{N}_j$, only bidirectional interconnection takes place in terms of interconnection boundary conditions (see (2.8) on page 30).

Following the scaling of spatial domain and grouping all the node and edge signals as per Chapter 2, the coupled representation of thermo-fluidic processes are given on a uniform domain $[a, b]$ by $\{\mathscr{P}_{\mathrm{p}} \mid \bar{m}^{px}, \bar{n}^{px}, \mathbf{x}_1, \mathbf{x}_0 = \varnothing\}$, where $\mathscr{P}_{\mathrm{p}}$ is defined in (2.29) on page 41.

Definition 4.1 *(Behavior of diffusive thermo-fluidic processes)*

Given matrix valued functions $\bar{A}_0$, $\bar{A}_1$, $\bar{A}_2$, $\bar{B}_{\mathbf{x}w}$, $\bar{B}_{\mathbf{x}w}$, $\bar{C}_{ai}$, $\bar{C}_{bi}$, $\bar{C}_{ci}$ for $i \in \{z\mathbf{x}, y\mathbf{x}\}$, and constant matrices $\bar{D}_{zw}$, $\bar{D}_{zu}$, $\bar{D}_{yw}$, $\bar{D}_{zu}$, define

$$
\begin{aligned}
(\bar{\mathcal{A}}_p \mathbf{x}_2)(s) &:= \bar{A}_0(s)\left[\mathbf{x}_2\right](s) + \bar{A}_1(s)\partial_s\left[\mathbf{x}_2\right](s) + \bar{A}_2(s)\partial_s^2\left[\mathbf{x}_2\right](s) \\
(\bar{\mathcal{C}}_{z\mathbf{x}} \mathbf{x}_2)(s) &:= \int_a^b \left(\bar{C}_{az\mathbf{x}}(s)\left[\mathbf{x}_2\right](s) + \bar{C}_{bz\mathbf{x}}(s)\partial_s\left[\mathbf{x}_2\right](s) + \bar{C}_{cz\mathbf{x}}(s)\partial_s^2\left[\mathbf{x}_2\right](s) \right) ds \\
(\bar{\mathcal{C}}_{y\mathbf{x}} \mathbf{x}_2)(s) &:= \int_a^b \left(\bar{C}_{ay\mathbf{x}}(s)\left[\mathbf{x}_2\right](s) + \bar{C}_{by\mathbf{x}}(s)\partial_s\left[\mathbf{x}_2\right](s) + \bar{C}_{cy\mathbf{x}}(s)\partial_s^2\left[\mathbf{x}_2\right](s) \right) ds \\
(\bar{\mathcal{B}}_{\mathbf{x}w}\bar{w})(s) &:= \bar{B}_{\mathbf{x}w}(s)\bar{w}, (\bar{\mathcal{B}}_{\mathbf{x}u}\bar{u})(s) := \bar{B}_{\mathbf{x}u}(s)\bar{u}.
\end{aligned} \tag{4.1}
$$

Then, the behavior of diffusive thermo-fluidic processes are defined by the subspace $\mathscr{P}_{\mathrm{d}}$ such that

$$
\mathscr{P}_{\mathrm{d}} = \left\{ \begin{array}{c}
\mathrm{col}\left((\bar{w}, \bar{u}), (\mathbf{x}_2), (\bar{z}, \bar{y})\right) \mid \forall t \in [0, \infty), \\
\mathrm{col}\,(\bar{z}, \bar{y}) \in \mathbb{R}^{\sum_{i=1}^m n_z^i} \times \mathbb{R}^{\sum_{i=1}^m n_y^i},\ \mathbf{x} \textit{ is Fréchet differentiable}, \\
\begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}_2(t) \end{bmatrix} \in X_{\mathrm{dom}}^{\mathrm{d}}, \\
\begin{bmatrix} \bar{z}(t) \\ \bar{y}(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \bar{D}_{zw} & \bar{D}_{zu} & \bar{\mathcal{C}}_{z\mathbf{x}} \\ \bar{D}_{yw} & \bar{D}_{zu} & \bar{\mathcal{C}}_{y\mathbf{x}} \\ \bar{\mathcal{B}}_{\mathbf{x}w} & \bar{\mathcal{B}}_{\mathbf{x}u} & \bar{\mathcal{A}}_p \end{bmatrix} \begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}_2(t) \end{bmatrix}
\end{array} \right\}, \tag{4.2}
$$

and

$$X_{\mathrm{dom}}^{\mathrm{d}} = \left\{ \begin{array}{c} \begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}_2(t) \end{bmatrix} \in \prod_{i=1}^{m} \left(\mathbb{R}^{n_w^i + n_u^i} \times H_2^{n_2^i}[a,b] \right), \\ \\ \begin{bmatrix} \bar{B}_w & \bar{B}_u & -\bar{B} \end{bmatrix} \begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \\ \mathbf{x}_2(a,t) \\ \mathbf{x}_2(b,t) \\ \partial_s \mathbf{x}_2(a,t) \\ \partial_s \mathbf{x}_2(b,t) \end{bmatrix} = 0 \end{array} \right\}. \tag{4.3}$$

where, the rank of B *is* $\sum_{i}^{m} 2n_2^i$.

4.3 Finite-Dimensional Approximation of Diffusive Thermo-Fluidic Processes

The purpose of describing the physical model and input-state-output representation as in (4.2) is to utilize the model and systematically build functionalities for diffusive thermo-fluidic processes. To this end, this chapter focuses explicitly on determining a finite-dimensional approximation of behavior (4.2) such that standard results from systems and control theory of finite-dimensional system can be directly used.

Here, the main challenge is that the boundary conditions on the external boundary points as well as the interface points have to be satisfied in the finite-dimensional behavior. Moreover, the external boundary conditions are perturbed by external inputs $w(t), u(t)$ that adds additional difficulties. If one refers to the state of the art finite element methods, few approaches tackle these issues. For example, to handle mixed boundary conditions and inputs at the boundaries a common practice is to use lifting functions to approximate (non-homogeneous) boundary conditions (see [82], Chapter 3). To handle spatially interconnected PDEs, recently domain decomposition approach has been widely used (see [42], Chapter 3 for an overview). However, the boundary conditions at the interface are not explicitly invoked in the finite element method; instead, they are approximated by using a locally dense mesh around the interface boundary.

In approximation methods, the fundamental idea is to project the infinite-dimensional PDE model onto a finite-dimensional subspace. This requires a) projection of infinite-dimensional signals (in this case, functions $\mathbf{x}_2$) on to a subspace denoted by $\mathscr{V}$, with $\mathscr{V} \subset X_{\mathrm{dom}}^{\mathrm{d}}$, b) projection of PDE dynamics (4.1) on to another subspace denoted by $\mathscr{V}$, with $\mathscr{W} \subset X_{\mathrm{dom}}^{\mathrm{d}}$. Definition of $\mathscr{V}$ and $\mathscr{W}$ requires a specific choice on basis functions that span the respective subspaces. As $X_{\mathrm{dom}}^{\mathrm{d}}$ restricts the evolution of functions by enforcing boundary conditions (4.3) as explicit constraints, a meaningful approximation (projection)

can only be guaranteed when the basis spanning the subspace $\mathscr{V}$ and $\mathscr{W}$ also satisfies (4.3). Moreover, the boundary conditions are perturbed by inputs that are user-defined, making the boundary conditions even more crucial while developing approximation methods for PDEs. In the remainder of this section, a three step procedure is developed that can be followed to select specific basis functions that span $X^{\mathrm{d}}_{\mathrm{dom}}$ satisfying the boundary conditions, to find a finite dimensional subspace on which the signals as well the PDEs can be projected.

4.3.1 Step 1: Separation of Solution and Homogenization

The main difficulty arises from the fact that external inputs are present at the boundary conditions (4.3). In the first step, the task would be to rewrite the spatio-temporal functions $\mathbf{x}_2$ in (4.1) by means of direct sum of two separate functions: one of them is explicitly governed by the boundary inputs and the other one is governed by the PDE model (4.1) when no input is present at the boundary.

Theorem 4.1 ***(Separation of Solution)***

Let $\bar{w} \in C^1(\mathbb{T}, \mathbb{R}^{\sum_{i=1}^m n_w^i})$, $\bar{u} \in C^1(\mathbb{T}, \mathbb{R}^{\sum_{i=1}^m n_u^i})$.

Then, $\operatorname{col}\Big((\bar{w}, \bar{u}), (\mathbf{x}_2), (\bar{z}, \bar{y})\Big) \in \mathscr{P}_{\mathrm{d}}$ *if* $\mathbf{x}_2(s,t)$ *admits the following decomposition*

$$\mathbf{x}_2(s,t) = \mathbf{z}(s,t) + \mathbf{f}(s,t) \tag{4.4}$$

if the following two statements hold true:

(i). There exists a linear map $\operatorname{col}(\bar{w}, \bar{u}) \mapsto \mathbf{f}$ *such that*

$$\mathbf{f}(s,t) = F(s) \begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \end{bmatrix}, \tag{4.5}$$

where, $F : [a,b] \to \mathbb{R}^{(\sum_{i=1}^m n_2^i) \times (\sum_{i=1}^m n_w^i + \sum_{i=1}^m n_w^i)}$ *is defined as linear functions in* s*, i.e.* $F(s) = F_1 s + F_2$*. Here,* F_1, F_2 *are full-rank constant matrices that satisfies the following matrix equalities*

$$\bar{B} \begin{bmatrix} F_1 a + F_2 \\ F_1 b + F_2 \\ F_1 \\ F_1 \end{bmatrix} = \begin{bmatrix} \bar{B}_w & \bar{B}_u \end{bmatrix}. \tag{4.6}$$

(ii). There exists function $\mathbf{z} : [0, \infty) \to \prod_{i=1}^{m} H_2^{n_2^i}[a,b]$ *that satisfies the*

following differential equations

$$\dot{\mathbf{z}}(s,t) = (\mathcal{A}_p\mathbf{z})(s,t) + B_1(s)\begin{bmatrix}\bar{w}(t)\\ \bar{u}(t)\end{bmatrix} + B_2(s)\begin{bmatrix}\dot{\bar{w}}(t)\\ \dot{\bar{u}}(t)\end{bmatrix} \tag{4.7}$$

where $\bar{\mathcal{A}}_p$ *is defined according to* (4.1), $B_1(s) = \bar{A}_1(s)F_1 + \bar{A}_0(s)F_1 s + \bar{A}_0(s)F_2 + \begin{bmatrix}\bar{B}_{\mathbf{x}w}(s) & \bar{B}_{\mathbf{x}u}(s)\end{bmatrix}$ *and* $B_2(s) = F_1 s + F_2$.

Moreover, (4.7) *is subject to the following constraints*

$$\bar{B}\begin{bmatrix}\mathbf{z}(a,t)\\ \mathbf{z}(b,t)\\ \partial_s\mathbf{z}(a,t)\\ \partial_s\mathbf{z}(b,t)\end{bmatrix} = 0. \tag{4.8}$$

Proof: The proof is obtained by substituting $\mathbf{x}_2(t) = \mathbf{z}(t) + \mathbf{f}(t)$ into (4.2)-(4.3). ■

Theorem 4.1 suggests an explicit splitting of the solution trajectories $\mathbf{f}(t)$ and $\mathbf{z}(t)$ while each of their evaluation is completely independent from each other with neither of them having any boundary constraint affected by inputs. Precisely, $\mathbf{z}(s,t)$ must satisfy (4.7) which is also a variant of coupled diffusion-transport-reaction equations, however, perturbed by external inputs $\bar{w}(t), \bar{u}(t)$ as well as their time derivatives $\dot{\bar{w}}(t), \dot{\bar{u}}(t)$ while constrained by the boundary conditions(4.8) (independent of boundary inputs). On the other hand, $\mathbf{f}(t)$ is considered to be linear functions that can be obtained by finding F_1 and F_2 using set of linear matrix equalities (4.6) (which are independent of inputs).

4.3.2 Step 2: Spectral Decomposition of $\mathbf{z}(s,t)$

The analytical expression of $\mathbf{f}(s,t)$ can be explicitly found by solving the linear matrix equalities (4.6). Now, in this subsection, the focus is on determining the functions $\mathbf{z}(s,t)$.

To this end, first, the spectral properties of (4.7) is investigated

Lemma 4.1 *For* $s \in [a,b]$, $t \in [0,\infty)$, *let* $\mathbf{z} : [0,\infty) \to \mathrm{dom}(\bar{\mathcal{A}}_p)$ *satisfy*

$$(\bar{\mathcal{A}}_p\mathbf{z})(s,t) = \bar{A}_0(s)\left[\mathbf{z}\right](s,t) + \bar{A}_1(s,t)\partial_s\left[\mathbf{z}\right](s,t) + \bar{A}_2(s)\partial_s^2\left[\mathbf{z}\right](s,t), \tag{4.9}$$

where $\mathcal{A}_p$ is a self-adjoint operator with

$$\mathrm{dom}(\bar{\mathcal{A}}_p) := \left\{ \begin{array}{c} \mathbf{z}(t) \in \prod_{i=1}^{m} H_2^{n_2^i}[a,b], \\ \\ \bar{B} \begin{bmatrix} \mathbf{z}(a,t) \\ \mathbf{z}(b,t) \\ \partial_s \mathbf{z}(a,t) \\ \partial_s \mathbf{z}(b,t) \end{bmatrix} = 0, \\ \\ \bar{B} \in \mathbb{R}^{2\sum_{i=1}^{m} n_2^i \times 4\sum_{i=1}^{m} n_2^i} \text{ has full rank} \end{array} \right\}. \tag{4.10}$$

Then there exists $\omega_n \in \mathbb{C}$ and functions $0 \neq \Phi_n \in \mathrm{dom}(\bar{\mathcal{A}}_p)$ such that the following equality holds true for every $n \in \mathbb{N}$

$$\bar{\mathcal{A}}_p \Phi_n = \omega_n \Phi_n. \tag{4.11}$$

Moreover,

$$\mathrm{Re}(\omega_1) \geq \mathrm{Re}(\omega_2) \geq \cdots \geq \cdots \geq \mathrm{Re}(\omega_\infty).$$

Proof: The proof can be found in [36]. Also see [23], Chapter 4. ■

Remark 4.1 *Note that, for general class of PDEs as described in Chapter 2 to define spatially distributed nodes do not facilitate the outcome of Proposition 4.1. Even, when a diffusion-transport reaction equation is defined on semi-infinite domain, the spectrum of operator $\bar{\mathcal{A}}_p$ is not separable anymore ([23], Chapter 4).*

Proposition 4.1 has two crucial implications on finding a solution of $\mathbf{z}(s,t)$.

1. Every single eigenvalue ω_n and eigenfunctions Φ_n satisfy (4.8) and can be treated independently form other $n \in \mathbb{N}$.
2. Eigenfunctions $\{\Phi_n \mid n = 1, \cdots, \infty\}$ defines basis for the space $\mathrm{dom}(\bar{\mathcal{A}}_p)$.

Now, the functions $\mathbf{z}(s,t)$ are rewritten as an infinite sum where each component of the sum is a product of spatial basis functions (which is chosen to be the eigen function) and a time-varying coefficient as follows

$$\mathbf{z}(s,t) = \sum_{n=1}^{\infty} \Theta_n(t) \Phi_n(\omega_n, s). \tag{4.12}$$

As $\mathbf{span}\{\Phi_n \mid n = 1, \cdots, \infty\} = \mathrm{dom}(\bar{\mathcal{A}}_p)$, every individual term in the solution decomposition (4.12) explicitly satisfies the boundary conditions (4.8). In principle, for every $n \in \mathbb{N}$, $\Phi_n(\omega_n, s)$ is determined by solving the eigenvalue problem (4.11) subject to the constraints (4.10). However, for general definition of (4.9), finding actual values for ω_n and an analytical expression of $\Phi_n(s)$ is not always possible.

Parametrization of Basis Functions Φ_n

Instead of directly solving the eigenvalue problem (4.11) subject to the constraints (4.10), a specific structure is enforced on $\Phi_n(s)$ that allows one to express them using any user-defined analytic functions while still satisfying (4.10).

For every $n \in \mathbb{N}$, Φ_n and Θ_n are defined as

$$\Phi_n(\omega_n, s) := \operatorname{col}\left(\phi_{ni}(\omega_n, s)\right)_{i \in \mathbb{N}_{[1, \sum_{i=1}^m n_2^i]}}, \quad \Theta_n(t) := \operatorname{diag}\left(\theta_{ni}(t)\right)_{i \in \mathbb{N}_{[1, \sum_{i=1}^m n_2^i]}}. \tag{4.13}$$

Here, every scalar function ϕ_{ni} is defined according to

$$\phi_{ni}(\omega_n, s) = \alpha_{ni} f_{ni}(\omega_n, s) + \beta_{ni} g_{ni}(\omega_n, s). \tag{4.14}$$

Remark 4.2 *In the definition of ϕ_{ni}, f_{ni} and g_{ni} are twice differentiable and anlaytic functions and are left to be chosen by the user. On the other hand,* (4.10) *imposes $2(\sum_{i=1}^m n_2^i)$ boundary conditions on $(\sum_{i=1}^m n_2^i)$ number of $\phi_{ni}(\omega_n, s)$ leaving two boundary conditions for every individual $\phi_{ni}(\omega_n, s)$. As a result, in* (4.14)*, $\{a_{ni}, b_{ni}\}$ are kept as free scalar parameters (can also be viewed as two independent degrees of freedom) that can be solved to enforce the boundary conditions in* (4.10)*.*

Based on the paraemtrization of $\Phi_n(s)$ and $\Theta_n(t)$, $z(s,t) = \sum_{n=1}^{\infty} \Theta_n(t)\Phi_n(\omega_n, s)$ consists of the following items for every $n \in \mathbb{N}$

- The user-defined or preselected basis functions $f_{ni}(\omega_n, s)$ and $g_{ni}(\omega_n, s)$.
- The unknwon scalar ω_n.
- The unknown coefficients α_{ni}, β_{ni}. For every $n \in \mathbb{N}$, there are $2(\sum_{i=1}^m n_2^i)$ coefficients that are unknown.
- The unknwon temporal functions $\theta_{ni}(t)$. For every $n \in \mathbb{N}$, there are $\sum_{i=1}^m n_2^i$ temporal functions that are unknown.

Solving for ω_n

The boundary conditions (4.8) are restrictions on the spatial distribution on the solution $z(s,t) = \sum_{n=1}^{\infty} \Theta_n(t)\Phi_n(\omega_n, s)$ that must hold for all $t \in [0, \infty)$. Hence,

using superposition principle, for every $n \in \mathbb{N}$, one obtains:

$$B \begin{bmatrix} \Theta_n(t)\Phi_n(\omega_n, a) \\ \Theta_n(t)\Phi_n(\omega_n, b) \\ \Theta_n(t)\partial_s\Phi_n(\omega_n, a) \\ \Theta_n(t)\partial_s\Phi_n(\omega_n, b) \end{bmatrix} = 0 \tag{4.15}$$

The collection of all the algebraic relations (4.15) ensures that there are in total $\sum_{i=1}^{m} 2n_2^i$ number of equations. Rearranging all of these $\sum_{i=1}^{m} 2n_2^i$ equations together, one gets

$$\Gamma_n(\omega_n)\Xi_n(t) = 0. \tag{4.16}$$

Here $\Gamma_n(\omega_n)$ is a square matrix nonlinearly parametrized by ω_n. The time dependent vector and $\Xi_n(t) := \mathrm{col}\Big(\theta_{ni}(t)\alpha_{ni}, \theta_{ni}(t)\beta_{ni}\Big)_{i\in\mathbb{N}_{[1,\sum_{i=1}^{m} n_2^i]}}$. Therefore, all the unknwon ω_n are found in the matrix Γ_n and all the unknwon $\theta_{ni}(t)\alpha_{ni}, \theta_{ni}(t)\beta_{ni}$ are found inside the vector Ξ_n. Let $\Xi_n(t)$ be uniquely defined for every $n \in \mathbb{N}$. Then, $\Xi_n(t)$ must be the non-trivial one dimensional nullspace of $\Gamma(\omega_n)$. This essentially suggests that the determinant of $\Gamma_n(\omega_n)$ is zero. Therefore, for every $n \in \mathbb{N}$, finding ω_n amounts to the following minimization problem:

$$\min_{\omega_n \in \mathbb{C}} \quad |\det\Big(\Gamma_n(\omega_n)\Big)| \tag{4.17}$$

In spectral analysis, the equation $\det(\Gamma_n(\omega_n)) = 0$ is often referred to as the transcendental equation. And ω_n, $n \in \mathbb{N}$ is one of its roots (there are infinitely many of them).

In numerical analysis, finding roots of $\det(\Gamma_n(\omega_n)) = 0$ is typically solved by standard root finding routines (see [104] for a detailed overview about solving transcendental equations). An iterative bisection search is used in this chapter on a coarse grid over the complex plane. This allows to search for points that approximately equates $\det(\Gamma_n(\omega_n))$ to zero.

Solving for α_{ni}, β_{ni}

Once individual ω_n is determined, the unknwon vector $\Xi_n(t)$ is uniquely defined according to (4.16). However, finding every α_{ni}, β_{ni} from $\Xi_n(t)$ depends on the choice of structuring $\Xi_n(t) := \mathrm{col}\Big(\theta_{ni}(t)\alpha_{ni}, \theta_{ni}(t)\beta_{ni}\Big)_{i\in\mathbb{N}_{[1,\sum_{i=1}^{m} n_2^i]}}$. In fact, the choice of parametrization for Φ_n and Θ_n is important for a feasible and unique set of α_{ni}, β_{ni} for all $t \in \mathbb{T}$. What is shown next is that arbitrary choice of parametrization for Φ_n and Θ_n does not necessarily provide unique α_{ni}, β_{ni} by solving (4.16). To verify that the feasibility of uniquely determining α_{ni}, β_{ni} is tested for three distinct parametrizations of Φ_n and Θ_n.

- Case 1

A possible structure of (4.12) would be to consider identical basis functions for every element in $\Phi_n(s)$. In other words, $\phi_{ni}(\omega_n, s) = \phi_{nk}(\omega_n, s) = \phi_n(\omega_n, s)$ for

any i, k and for all $s \in [a, b]$. In other words,

$$\mathbf{z}(s,t) = \sum_{n=1}^{\infty} \Theta_n(t)\phi_n(\omega_n, s). \tag{4.18}$$

- Case 2

Another choice of parametrization is to keep the same structure as (4.13), i.e. different basis functions and different temporal coefficients for all the elements in $\mathbf{z}$. Precisely,

$$\mathbf{z}(s,t) = \sum_{n=1}^{\infty} \Theta_n(t)\Phi_n(\omega_n, s). \tag{4.19}$$

- Case 3

The third choice would be to consider identical temporal functions $\theta_n(t)$ for each element in $\Theta_n(t)$ and keep different basis functions in $\Phi_n(\omega_n, s)$. In other words, $\theta_{ni}(t) = \theta_{nk}(t) = \theta_n(t)$ for any i, k. As a result $\mathbf{z}$ takes the following form

$$\mathbf{z}(s,t) = \sum_{n=1}^{\infty} \theta_n(t)\Phi_n(\omega_n, s). \tag{4.20}$$

To assess feasbility of solving unique α_{ni}, β_{ni} from (4.16), let the following definitions be introduced.

Definition 4.2 *(Feasible parametrization)*

If there exists a unique and constant Ξ_n(t independent) that satisfies (4.16) *for all $t \in [0, \infty)$, then the unique α_{ni}, β_{ni} defines a feasible parametrization of basis $\Phi_n(\omega_n, s)$ according to* (4.13)-(4.14).

> **Lemma 4.2** *Among various cases, the solution decomposition in* (4.20) *leads to a feasible parametrization $\forall t \in [0, \infty)$*

Proof: In Case 1, $\phi_n(s) = \alpha_n f(\omega_n, s) + \beta_n g(\omega_n, s)$. Hence,

$\Xi_n(t) := \text{col}\Big(\theta_{ni}(t)\alpha_n, \theta_{ni}(t)\beta_n\Big)_{i \in \mathbb{N}_{[1, \sum_{i=1}^{m} n_2^i]}}$. Evidently, it is not feasible to uniquely solve for constant two parameters α_n, β_n.

In Case 2, $\Xi_n(t) := \text{col}\Big(\theta_{ni}(t)\alpha_{ni}, \theta_{ni}(t)\beta_{ni}\Big)_{i \in \mathbb{N}_{[1, \sum_{i=1}^{m} n_2^i]}}$. It is only possible to solve for unique and constant α_{ni}, β_{ni} if and only if all $\theta_{ni}(t)$ are known. Hence, the decomposition (4.19) leads an over parametrization and, hence, not feasible.

In Case 3, as $\theta_n(t)$ is a scalar, $\Xi_n(t) := \mathrm{col}\Big(\alpha_{ni}, \beta_{ni}\Big)_{i\in\mathbb{N}_{[1,\sum_{i=1}^m n_2^i]}}$ for all $t \in [0,\infty)$. Now, one has $\sum_{i=1}^m 2n_2^i$ unknwons and they can be found uniquely by solving (4.16) for every $n \in \mathbb{N}$. ■

As a result, the expansion in (4.20), where all temporal functions are identical, is a suitable choice for finding unique basis functions $\{\Phi_n | n \in \mathbb{N}\}$. This completes the determination of all basis functions $\{\Phi_n | n \in \mathbb{N}\}$ and only remaining unknowns are $\theta_n(t)$.

4.3.3 Step 3: Approximation of Solution and Finding $\theta_n(t)$

So far, a suitable method has been found to determine and parametrize the solution $\mathbf{z}(t)$ according to

$$\mathbf{z}(s,t) = \sum_{n=1}^{\infty} \theta_n(t)\Phi_n(s). \tag{4.21}$$

The basis functions $\Phi_n(s)$ are defined according to (4.13) and, at this stage, they are known. However, the scalar function $\theta_n(t)$ are still unknown and there are infinitely many of them. Here, such infinite-dimensional solution expansion is projected onto a finite-dimensional basis using Galerkin projection (see [8]).

To this end, define a residual operator $\mathbf{R}$ according to

$$\mathbf{R}(\mathbf{z}, \bar{w}, \bar{u}, \dot{w}, \dot{u}) := \frac{\partial \mathbf{z}}{\partial t} - (\bar{\mathcal{A}}_p \mathbf{z})(t) - B_1 \begin{bmatrix} \bar{w}(t) \\ \bar{u}(t) \end{bmatrix} - B_2 \begin{bmatrix} \dot{\bar{w}}(t) \\ \dot{\bar{u}}(t) \end{bmatrix} = 0. \tag{4.22}$$

A finite-dimensional model is obtained by projecting both $\mathbf{z}$ in (4.21) and $\mathbf{R}$ in (4.22) on a finite dimensional subspace Z_H of $\prod_{i=1}^m H_2^{n_2^i}[a,b]$. To this end, let the inner product on $\prod_{i=1}^m H_2^{n_2^i}[a,b]$ is defined as

$$\langle F, G \rangle_{L_2} := \int_a^b J^\top(s) G(s) \mathrm{d}s$$

for all functions $J(s), G(s) \in \prod_{i=1}^m H_2^{n_2^i}[a,b]$.

Let, Z_H be a H dimensional subspace spanned by the first H basis functions, i.e. $\{\Phi_n(s) \mid n = 1, \cdots, H\}$. Hence, the finite-dimensional approximation of (4.22) is given by solution $\hat{z}(s,t)$

$$\hat{z}(s,t) = \sum_{n=1}^{H} \theta_n(t)\Phi_n(s). \tag{4.23}$$

that satisfies

$$\left\langle \Phi_m, \mathbf{R}(\hat{z}, \bar{w}, \bar{u}, \dot{w}, \dot{u}) \right\rangle_{L_2} = 0; \quad \forall m \in \{1, \cdots, H\}. \tag{4.24}$$

Here, the equation (4.24) represents a H-dimensional state space model of (4.22). Equation (4.24) can be rewritten as a finite dimensional system that has the following representation

$$\dot{x}(t) = \bar{A}x(t) + \bar{B}_1 \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} + \bar{B}_2 \begin{bmatrix} \dot{w}(t) \\ \dot{u}(t) \end{bmatrix}, \tag{4.25}$$

where, $x(t) := \operatorname{col}(\theta_n(t))_{n\in\{1,\cdots,H\}}$, and $\bar{A} := \left\langle \Phi_m, \bar{\mathcal{A}}_p \hat{z} \right\rangle_{L_2} \ \forall m \in \{1, \cdots, H\}$. Similar definition goes for $\bar{B}_1$ and $\bar{B}_2$. One can now solve for $x(t) := \operatorname{col}(\theta_n(t))_{n\in\{1,\cdots,H\}}$ from (4.25) using any stable time-marching method. Once, $\hat{z}$ is found, the approximation of original solution $\mathbf{x}_2$ for which $\operatorname{col}\Big((\bar{w}, \bar{u}), (\mathbf{x}_2), (\bar{z}, \bar{y})\Big) \in \mathscr{P}_{\mathrm{d}}$ is $\hat{x}(s,t)$ such that

$$\hat{x}(s,t) = \sum_{n=1}^{H} \theta_n(t)\Phi_n(s) + F_1(s) + F_2. \tag{4.26}$$

Note that the output equations in the definition of $\mathscr{P}_{\mathrm{d}}$ can also be performed in a similar fashion.

Once a finite dimensional approximated mode is found, the entire set of tools presented in Chapter 3 is directly applicable.

4.4 Academic Case Study: Heat Diffusion in Composite

To demonstrate the discussed methodologies, approximation of one dimensional heat diffusion model in a composite material is considered. The composite material is a non-homogeneous substance which is constructed by interconnecting layers of different materials [9]. The simulation of heat diffusion in such a structure allows to analyze the thermal effects of different materials when they are perturbed by external thermal disturbances.

Figure 1 illustrates a three-layered composite material in a spatially interconnected structure.

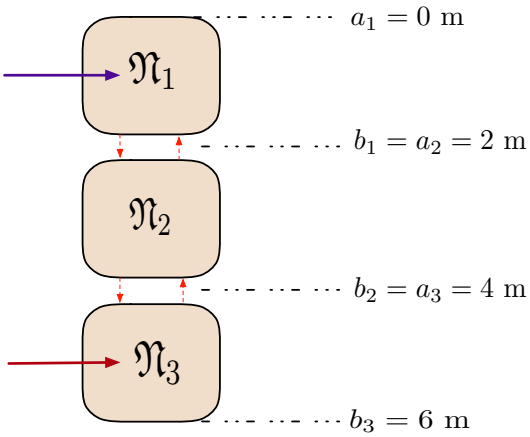


Figure 1: Graph structure to study diffusion problem in a three-layered composite material.

Topology

A finite and connected graph is defined as

$$\mathcal{G} = (\mathfrak{N},\ \mathfrak{E},\ A, \mathbb{T}).$$

Here, $\mathbb{T} = [0, \infty)$, with $\mathcal{N} = \{\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3\}$ and $\mathcal{E} = \{\mathfrak{E}_{1,2}, \mathfrak{E}_{2,1}, \mathfrak{E}_{2,3}, \mathfrak{E}_{3,2}\}$. The adjacency matrix $A \in \mathbb{R}^{4\times4}$ is symmetric and defined as following:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Spatially Distributed Nodes

Every individual node $\mathfrak{N}_i \in \mathfrak{N}$, $i \in \mathbb{N}_{[1,3]}$ is specified in following items

- $\mathbb{X}_1 = [0, 2]$ (in m), $\mathbb{X}_2 = [2, 4]$ (in m), $\mathbb{X}_1 = [4, 6]$ (in m).
- $\mathbb{X}_1^{\text{bc}} = \{0\}$ (in m), $\mathbb{X}_2^{\text{bc}} = \emptyset$, $\mathbb{X}_3^{\text{bc}} = 6$(in m).

Node Dynamics

The mathematical model of this heat conduction problem is given by the following partial differential equation for each of the 3 layers for $s_i \leq s \leq s_{i+1}$ with $i = 1, 2, 3$ and $t \geq 0$ as below:

$$\frac{\partial^2 T_i(s,t)}{\partial s^2} + \frac{1}{K_i} Q_i(s,t) = \frac{1}{D_i^2} \frac{\partial T_i(s,t)}{\partial t}. \tag{4.27}$$

Here,

$T_i(s,t)$ is the temperature in ith section.

$D_i^2 = \frac{K_i}{\rho_i C_{p,i}}$ is the thermal diffusivity in the ith section.

K_i is the thermal conductivity of the ith section.

ρ_i is the density in the ith section.

$C_{p,i}$ is the specific heat in the ith section.

Boundary Conditions

The corresponding boundary conditions on $\mathbb{X}_1^{\text{bc}}$ and $\mathbb{X}_3^{\text{bc}}$ are

$$K_1 \frac{\partial T_1(0,t)}{\partial s} = h_1[T_1(0,t) - T_0], \tag{4.28}$$

$$K_3 \frac{\partial T_3(6,t)}{\partial s} = -h_1[T_3(6,t) - T_{out}]. \tag{4.29}$$

At the internal boundaries $i = 1, 2$, the media are subjected to following boundary conditions:

$$-K_i \frac{\partial T_i(s_{i+1},t)}{\partial s} = h_{i+1}[T_i(s_{i+1},t) - T_{i+1}(s_{i+1},t)], \tag{4.30}$$

$$K_i \frac{\partial T_i(s_{i+1},t)}{\partial s} = K_{i+1} \frac{\partial T_{i+1}(s_{i+1},t)}{\partial s}. \tag{4.31}$$

Physical Parameters and Input Specifications

Table 1: Parameter values

Parameters	$i=1$	$i=2$	$i=3$
$\rho_i(g/m^3)$	11.08	2.71	7.4
$C_{p,i}(cal/g^\circ C)$	0.031	0.181	0.054
$K_i(cal/m^\circ C)$	297.64	1741.18	565.51
$h_i(cal/m^{2\circ}C)$	300	24000	15000

The external force is a distributed function $Q_i(s,t) = l(s)q_i(t)$ where $l(s)$ is an indicator function representing how the heat is distributed over each layer. For simplicity, we take uniform distribution of heat flow over each layer. That is

$$\begin{aligned} l(s) &= 1;\ s \in \{[s_1, s_2] \cup [s_3, s_4]\}, \\ &= 0;\ \text{elsewhere.} \end{aligned} \tag{4.32}$$

Equation (4.32) indicates that we allow uniform heat flux over first and last layer, where in middle layer no heat flux is provided.

Finding the Eigen Functions

The basis functions $\phi_{ni}(\omega_n, s)$ in (4.14) are solved by the following differential equations

$$D_i^2 \frac{d^2\phi_{ni}(s)}{ds^2} + \omega_n^2 \phi_{ni}(s) = 0. \tag{4.33}$$

One can prove that prove that $\phi_{ni}(\omega_n, s)$ is an orthogonal function that admits the following expression.

$$\phi_{ni}(\omega_n, s) = \alpha_{ni} \cos\left(\frac{\omega_n}{D_i} s\right) + \beta_{ni} \sin\left(\frac{\omega_n}{D_i} s\right). \tag{4.34}$$

Simulation

To simulate the spatio-temporal evolution of the temperature, presented three-step method is compared with respect to the solution produced by PDE toolbox in MATLAB that utilizes finite difference method (cf. [101]) on a discretized domain. The comparison between these two simulation results is given in Figure 2. Additionally, Figure 3 shows the temperature difference between the three-step method and the finite difference method.

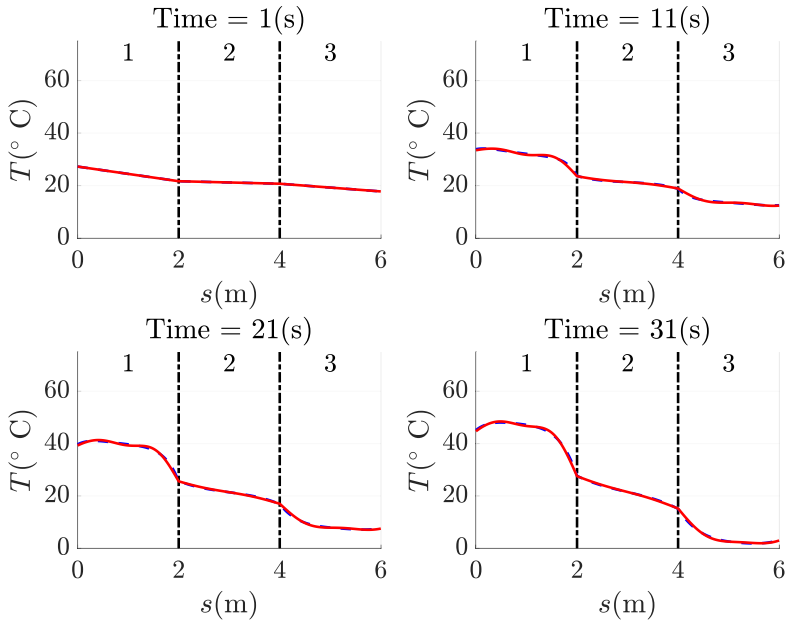


Figure 2: Temperature distribution of three-step method with dimension 20 (red line) compared to the finite difference approach of dimension 50 (blue dotted line) at four time instances. $\mathfrak{N}_1$ is heated up with +50 watt/m^2 and $\mathfrak{N}_3$ is cooled down with -50 watt/m^2.

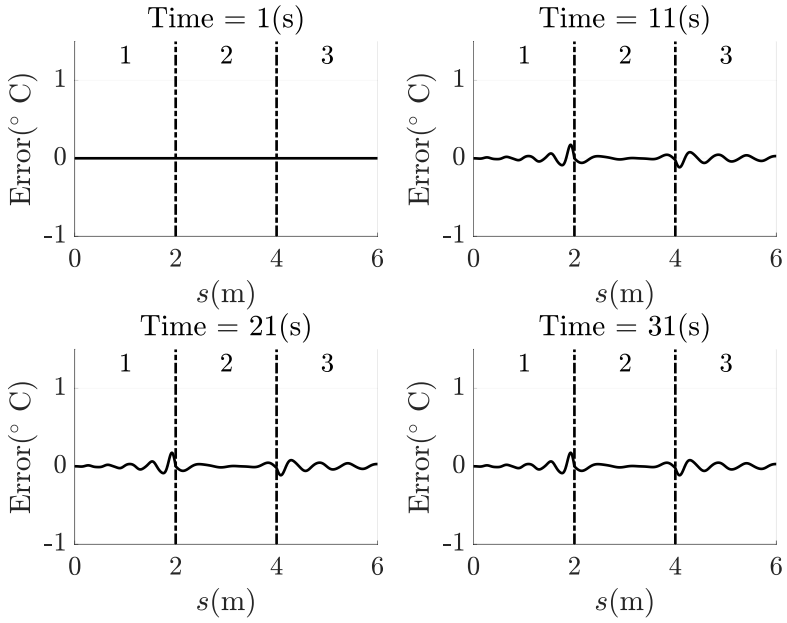


Figure 3: The temperature difference between the proposed method with finite difference method.

Based on the Figure 2 and 3, the difference in temperature between two methods is below ± 0.2°C. At the same time, with the proposed method, we are able to achieve significant reduction in the dimension (20 compared to 50). The proposed method also captures the mutual effect of simultaneously heating and cooling the different layers.

4.5 Closing Remarks

A three step procedure is presented to approximate the infinite dimensional behavior of diffusive thermo-fluidic processes in the presence of external inputs that act on boundaries. The approximated finite dimensional model satisfies all the boundary conditions by having two degrees of freedom on every basis functions. As a result of this computational method of approximation, one can apply analysis and control tools for finite dimensional systems on the approximated version of diffusive thermo-fluidic processes.

5 CHAPTER

Estimation of Unknown Physical Parameters in Diffusive Thermo-Fluidic Processes

This chapter presents a frequency domain approach to estimate spatially varying physical parameters of a one dimensional diffusion-transport-reaction equation. A non-linear least squares optimization of a frequency relevant criterion is proposed on the basis of measurements from a limited number of sensors. Analytic expressions of the Jacobian of the criterion function are exploited in an efficient numerical scheme. The proposed method exploits the sparsity of the underlying discretized model for a fast computation of the system parameters. The performance of the proposed procedure is illustrated by a number of simulation results. We demonstrate that the proposed method is able to estimate a spatially varying profile of unknown physical parameters in a realistic scenario.

Outline

5.1 Introduction **100**
5.2 Setting Up the Parameter Estimation Problem **101**
5.3 Grey-Box Identification in Frequency Domain **102**
5.4 Academic Case Study **108**
5.5 Closing Remarks **112**

5.1 Introduction

Modeling and approximation of the diffusive thermo-fluidic processes requires the knowledge of the spatially varying transport coefficients that are typically unknown. As emphasized before, for instance, the printing process is localized in space and time, which implies that diffusion, transport and reaction properties are non-homogeneous over the material and vary with space. An experiment based data driven approach is one way to determine the spatial distribution of the unknown parameters using input-output data. To this end, in this chapter, we propose a frequency domain approach to estimate the spatially varying profile of diffusion-transport-reaction coefficients as well as the spatial distribution of the input using measurements from a limited number of sensors. The proposed method does not necessarily require the boundary conditions and initial conditions to be known in advance. Moreover, the proposed method provides a fast computation of the estimated physical parameters using algebraic computation of the matrices. We utilize a) the signals in frequency domain, b) the sparsity of a specific discretization of the model and c) the analytic expressions of the Jacobian of the cost functional.

The application of diffusion-transport-reaction phenomena can be found in a wide range of applications. For example, thermo-fluidic transport in composite materials. In inkjet printing, the transport of ink through the printing media involves diffusion phenomena with convective action [66]. In hydrology, the study of the groundwater-surfacewater system involves the study of diffusion-transport of water fluxes across the stream-bed (c.f. [86], [6]). In all these applications, the diffusion-transport-reaction phenomena possess few salient attributes. Namely, a) the physical parameters, e.g. diffusion coefficient, transport coefficient and reaction coefficient, vary over the spatial configuration and are typically unknown, b) the boundary conditions and the initial conditions are often difficult to measure, and c) the spatial distribution of the external inputs is unknown. The later problem is also known as source localization problems [4].

Many methods have led to the estimation of parameter profiles of PDEs. An overview of these estimation methods is discussed in [13], [14] and the references therein, and mainly focus on methods to identify constant and uniform system parameters in time domain. Using the methodology introduced in [24], it is possible to transfer these estimation problems into the frequency domain. This method converts a PDE to a parameterized set of ordinary differential equations (ODEs) which can be subsequently approximated or, in some cases, be solved analytically. However, the unknown parameters are often non-linearly related to these equations leading to a mathematically challenging problem. A new methodology was developed in [96] to identify parameter profiles by implementing sample maximum likelihood estimations. This method takes the noise of both the input and the output into account but only considers the estimation of constant parameter values.

Contribution of this Chapter

This chapter contributes on the following aspects:

1. We consider the estimation of the spatially varying parameters in a single one dimensional PDE describing diffusion, transport and reaction phenomena.. The proposed method estimates spatially varying profiles of the parameters using measurements from a limited number of sensors. The method is versatile to simultaneously estimate the diffusion, transport and reaction coefficients as well as the spatial distribution of the in-domain input.
2. A frequency domain criterion is used which is optimized for the estimation of all spatially distributed coefficients of the discretized PDE. The computation is numerically efficient and completely algebraic involving operations on sparse matrices. An analytic expression is derived for the Jacobian of the cost criterion. This avoids the numerical calculation of descent directions and significantly reduces computation time. In addition, the use of the Jacobian increases the numerical stability of the estimation procedure as well.
3. In frequency domain, the transient response due to the initial conditions can be either compensated or we can wait till the transient response diminishes. As a result, unknown prior knowledge about the boundary conditions and the initial conditions does not affect the estimation algorithm. This makes the proposed method realistic and easily applicable for experimental set-ups.

5.2 Setting Up the Parameter Estimation Problem

Model

We consider diffusion-transport-reaction phenomena in a one dimensional (1D) configuration space. The spatial domain is a bounded $\mathbb{X} := [a,\ b] \subset \mathbb{R}$ and the time axis is a subset of non-negative real numbers $\mathbb{T} := [0,\ \infty) \subseteq \mathbb{R}_{\geq 0}$. For any given pair $(s,t) \in (\mathbb{X} \times \mathbb{T})$, the variable $\mathbf{x}_2(s,t)$ is the governing physical quantity (e.g., temperature, mass concentration etc.) that evolves over space and time according to a generalized version of the 1D diffusion-transport-reaction equation defined by the following linear parabolic PDE

$$\frac{\partial \mathbf{x}_2(s,t)}{\partial t} = \bar{A}_2(s)\frac{\partial^2 \mathbf{x}_2(s,t)}{\partial s^2} + \bar{A}_1(s)\frac{\partial \mathbf{x}_2(s,t)}{\partial s} + \bar{A}_0(s)\mathbf{x}_2(s,t) + \bar{B}_{\mathbf{x}w}(s)\bar{w}(t). \quad (5.1)$$

The physical parameters $\bar{A}_2(s)$, $\bar{A}_1(s)$ and $\bar{A}_0(s)$ are sufficiently regular. Precisely, $\bar{A}_2(s)$ is the *diffusion coefficient*, $\bar{A}_1(s)$ is the *transport coefficient* and $\bar{A}_0(s)$ is the *reaction coefficient*. Similarly, the input profile $\bar{B}_{\mathbf{x}w}(s)$ indicates the spatial distribution of the in-domain actuators and $\bar{w}(t)$ is the corresponding variation of inputs in time.

Boundary conditions

The corresponding boundary conditions at $s = a$, $s = b$ are considered to be unknown.

Initial Conditions

The initial condition $\mathbf{x}_2(s, 0)$ is also considered to be unknown. It is assumed that the initial condition of the problem is compatible with the model and its boundaries (c.f.[26]).

Measurements

The system is equipped with M sensors that measure $\mathbf{x}_2$ at specific locations $\mathbb{X}^M \subset \mathbb{X}$. In particular, a specific measured output is $y^m(t) \in \{\mathbf{x}_2(s^1, t), \cdots, \mathbf{x}_2(s^M, t)\}$ where $s^m \in \mathbb{X}^M$ is the location of a sensor with $m \in \{1, \cdots, M\}$. In case of unknown initial conditions, the transient response can be removed from the forced response using advanced frequency domain techniques such as the local polynomial method (e.g. see [39], or local rational method [33].

Problem Definition

In this chapter, the problem amounts to estimating

$$\gamma(s) := \text{col}(\bar{A}_2(s), \bar{A}_1(s), \bar{A}_0(s), \bar{B}_{\mathbf{x}w}(s))$$

as a function of $s \in \mathbb{X}$ from the input-output data $\{(\bar{w}(t), y^m(t)) \mid t \in \mathbb{T}, m \in \mathbb{N}_{[1,M]}\}$ using the model (5.1). One may assume that $\gamma : \mathbb{X} \to \mathbb{R}^4$ belongs to a function space Γ. For estimation purposes, we assume that Γ is parametrized by a surjective mapping $\Pi : \Theta \to \Gamma$ defined on a real-valued parameter space Θ so that

$$\gamma(s) := \Big[\Pi(\theta)\Big](s) \tag{5.2}$$

belongs to Γ for any $\theta \in \Theta$. Therefore, the problem to determine the functions in $\gamma(s)$ amounts to estimating $\theta \in \Theta$ in (5.2) using the measured output $y^m(t)$, and the model that is governed by (5.1). The problem set-up is visualized in Figure 1.

5.3 Grey-Box Identification in Frequency Domain

In this section, the methodology to estimate $\theta \in \Theta$ is described. In [13], the above problem is solved in the time domain while the unknown parameters are considered to be constant (uniformly distributed) over $\mathbb{X}$. In this chapter, we solve the equivalent problem using a frequency domain optimization criterion to

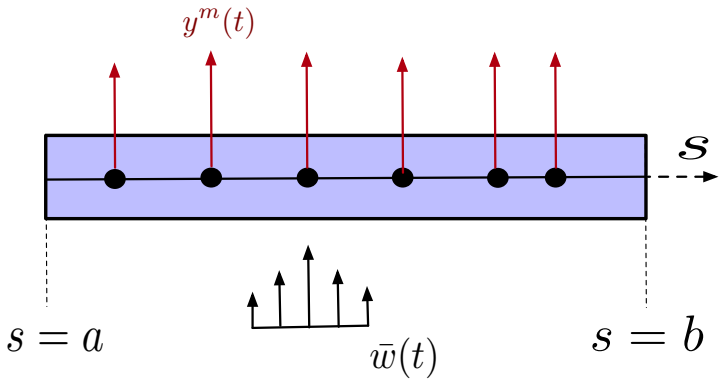


Figure 1: Schematic of the problem set-up. • indicates the location of the sensors.

estimate non-uniform (spatially varying) distributions of the physical parameters. Here, a frequency domain approach offers important advantages over time domain approaches, especially for periodic inputs. Indeed,

- The periodicity of the measured output is implied by the periodicity of the input due to the linear and time invariant nature of (5.1). Moreover, as a direct consequence of the linearity in (5.1), the frequencies appeared in the outputs retain the exact frequencies that are present in the excitation input.
- In the frequency domain, the signals are averaged over their periods leading to a significant improvement on the signal to noise ratio. Moreover, incorporating colored noise is easier in the frequency domain (but is not considered in this chapter and kept as a future work).

Based on these observations, we formulate an optimal estimation problem in the frequency domain. For the actuators, we select a finite number of frequency points ω_ℓ, $\ell \in \{1 \cdots, H\}$ to design the excitation signal $\bar{w}(t)$. As the governed PDE in (5.1) is linear, the same frequency points appear in the measurements y^m as well. However, directly computing of the non-rational transfer functions and estimating $\gamma \in \Gamma$ will be computationally challenging as the unknown parameters typically shows non-linear dependencies on the transfer function (for an example see pp. 148 [95]). In this chapter, we propose to cast the estimation problem into a finite dimensional discretized model of the PDE. The estimation problem comprises four relevant aspects which are discussed in the following subsections.

5.3.1 Parametrization of the unknown functions

In this section, we provide an explicit parametrization of $\gamma(s)$. To this end, the mapping Π in (5.2) is defined as an infinite sum of basis functions spanning Γ.

Precisely,

$$\gamma(s) := \Big[\Pi(\theta)\Big](s) = \sum_{r=1}^{\infty} B_r(s)\theta_r, \tag{5.3}$$

where $\theta_r \in \Theta$ with $\Theta \subset \mathbb{R}^4$. Then Π defines Γ by uniquely parametrizing individual functions $\bar{A}_2(s)$, $\bar{A}_1(s)$, $\bar{A}_0(s)$, and $\bar{B}_{\mathbf{x}w}(s)$. In particular, $B_r(x) := \text{diag}(B_r^D(s), B_r^U(s), B_r^K(s), B_r^P(s))$ are assumed to be given for all $r \in \mathbb{Z}^+$ and $\theta_r := \text{col}(\theta_r^D, \theta_r^U, \theta_r^K, \theta_r^P) \in \Theta$ for all $r \in \mathbb{Z}^+$. Since this amounts to estimating infinite sequences of θ_r, (5.3) is approximated by considering the first R terms

$$\gamma(s) \approx \sum_{r=1}^{R} B_r(s)\theta_r. \tag{5.4}$$

This means that the parameter set $\{\theta_r \in \Theta \mid 1, \cdots, R\}$ has become finite dimensional. As a result, the parameters we need to estimate are $\bar{\theta} := \text{col}(\theta_1, \cdots, \theta_R) \in \bar{\Theta} \subset \mathbb{R}^{4R}$.

5.3.2 Discretization procedure

A spatial discretization of (5.1) is presented to obtain an approximate finite dimensional model. To this end, our goal is to achieve a sparse structure in the finite dimensional model with a simpler parametric dependency so as to numerically enhance the computation of the estimation problem. It is important to note that the discretization grid can be increased to arbitrary precision. Instead of to other methods, we show that the finite difference method (see [82]) leads to a sparse state space model with linear-affine dependency of the unknown parameters.

In a finite difference method, the spatial domain is discretized in N equidistant points with a (homogeneous) grid sample $\Delta_s > 0$. In principle, this discretization grid can be increased to arbitrary precision. Now, consider the sequence of discrete points s_i with $i \in \{1, \cdots, N\}$ belonging to the sample set $\mathbb{X}^D \subset \mathbb{X}$. Additionally, we assume that $\mathbb{X}^D$ contains $\mathbb{X}^M$, i.e., $\mathbb{X}^D \cap \mathbb{X}^M = \mathbb{X}^M$. Furthermore, since the boundary conditions are considered to be unknown, we select we select them as additional inputs applied on the boundary of $\mathbb{X}$. This results in $N-2$ first order ordinary differential equations when using a central finite difference [82]. This leads to a finite dimensional state space model to discretize (5.1) in the following form

$$\begin{aligned} \dot{z} &= A(\bar{\theta})z + B(\bar{\theta})q, \\ y^m &= C^m z, \end{aligned} \tag{5.5}$$

Where the state vector $z := \text{col}(z_2, \cdots, z_{N-1}) = \text{col}(\mathbf{x}_2(s_2, t), \cdots, \mathbf{x}_2(s_{N-1}, t))$ represents $\mathbf{x}_2(\cdot, t)$ evaluated at each grid-point s_i for $i = 2, \cdots, N-1$. The extended input vector is $q(t) = \text{col}(\bar{w}(t), \mathbf{x}_2(s^1, t), \mathbf{x}_2(s^M, t))$. The matrix C^m has

dimension $1 \times (N-2)$ and measures $\mathbf{x}(\cdot, t)$ at the m^{th} sensor location. Precisely,

$$A(\bar{\theta}) = \sum_{r=1}^{R} \Big[\theta_r^D L_r^D + \theta_r^U L_r^U + \theta_r^K L_r^K\Big], \tag{5.6}$$

$$\begin{aligned} B(\bar{\theta})q(t) = \sum_{r=1}^{R} \Big[\theta_r^D g_r^D + \theta_r^U g_r^U\Big]\mathbf{x}_2(s^1, t) \\ + \sum_{r=1}^{R} \Big[\theta_r^D h_r^D + \theta_r^U h_r^U\Big]\mathbf{x}(s^M, t) + \sum_{r=1}^{R} \Big[\theta_r^P f_r^P\Big]\bar{w}(t). \end{aligned} \tag{5.7}$$

Furthermore,

$$L_r^D = \frac{1}{(\Delta_s)^2}\tilde{B}_r^D \begin{bmatrix} -2 & 1 & 0 & \cdots & \cdots \\ 1 & -2 & 1 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & 1 & -2 \end{bmatrix}, \tag{5.8}$$

$$L_r^U = \frac{1}{2\Delta_s}\tilde{B}_r^U \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots \\ -1 & 0 & 1 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & -1 & 0 \end{bmatrix}, \tag{5.9}$$

$$L_r^K = \tilde{B}_r^K. \tag{5.10}$$

Here, $\tilde{B}_r^D$, $\tilde{B}_r^U$ and $\tilde{B}_r^K$ are diagonal matrices of dimension $(N-2) \times (N-2)$ with the diagonal entries $B_r^D(s_i)$, $B_r^U(s_i)$, $B_r^K(s_i)$ evaluated at each grid point s_i for $i = 2, \cdots, N-1$.

$$\begin{aligned} f_r^P &= \text{col}\Big(B_r^P(s_2), \cdots, B_r^P(s_{N-1})\Big), \\ g_r^D &= \text{col}\Big(\frac{B_r^D(s_2)}{(\Delta_s)^2}, 0, \cdots, 0\Big), \\ g_r^U &= \text{col}\Big(-\frac{B_r^U(s_2)}{2\Delta_s}, 0, \cdots, 0\Big), \\ h_r^D &= \text{col}\Big(0, \cdots, 0, \frac{B_r^D(s_{N-1})}{(\Delta_s)^2}\Big), \\ h_r^U &= \text{col}\Big(0, \cdots, 0, \frac{B_r^U(s_{N-1})}{2\Delta_s}\Big). \end{aligned} \tag{5.11}$$

The approximated Multi-input-single-output (MISO) rational transfer function

mapping q to y^m can be derived according to

$$G^m\left(\omega_\ell, \bar{\theta}\right) = C^m\left(j\omega_\ell I - A(\bar{\theta})\right)^{-1} B(\bar{\theta}). \tag{5.12}$$

As a result, the combination of a frequency domain approach and a suitable approximation scheme facilitate such affine and sparse structure that can be fully exploited while performing the optimization.

5.3.3 Specification of the cost functional

For each frequency point ω_ℓ, $\ell \in \{1, \cdots, H\}$, we have $\{(q(\omega_l), y^m(\omega_l)) \mid m \in \mathbb{N}_{[1,M]}\}$ as available information from the experiments. On the other hand, we can model the outputs of (5.1) using (5.12). The cost functional can be formulated in terms of $\bar{\theta}$ to minimize a weighted sum of squared absolute errors between the measured output and the modeled output. In other words,

$$\tilde{V}(\bar{\theta}) := \sum_{\ell=1}^{H} \sum_{m=2}^{M-1} \frac{\mid y^m(\omega_\ell) - G^m(\omega_\ell, \bar{\theta})\, q(\omega_\ell) \mid^2}{w^m w_\ell}. \tag{5.13}$$

With some abuse of notation, $y^m(\omega)$ denotes the Fourier transformation of scalar signal $y^m(t)$. We have also incorporated user-defined weights $w^m > 0$ and $w_\ell > 0$ to penalize the relative error due to a specific sensor and a specific frequency point. In the case of faulty or missing measurements, these weights can be adjusted to incorporate prior information.

5.3.4 Optimization Procedure

In order to find an optimal $\bar{\theta}$ that minimizes (5.13), numerical solvers can be employed that iteratively evaluate (5.13) starting from an initial guess of $\bar{\theta}$ and finding the minimizer of (5.13) in a specific descent direction. Any gradient based algorithm requires the Jacobian of (5.13) to be calculated iteratively for finding the descent directions (see [74]). However, the Jacobian of (5.13) is typically computed using numerical approximations. Here, we determine an analytic expression of the Jacobian. Together with the sparsity of the matrices in (5.5), this considerably enhances the numerical efficiency of gradient-based methods to minimize (5.13).

Analytic expression of Jacobian

The cost functional in (5.13) can be rewritten as

$$\tilde{V}(\bar{\theta}) = \sum_{\ell=1}^{H} \sum_{m=2}^{M-1} \frac{1}{w^m w_\ell} (e_\ell^m)^* (e_\ell^m), \tag{5.14}$$

where per frequency ω_ℓ, the point-wise error is

$$e_\ell^m := y^m(\omega_\ell) - C^m(j\omega_\ell I - A(\bar{\theta}))^{-1}B(\bar{\theta})\, q(\omega_\ell), \tag{5.15}$$

and $(e_\ell^m)^*$ is its adjoint. The Jacobian of $\tilde{V}(\bar{\theta})$ is defined as

$$J(\bar{\theta}) := \frac{\partial \tilde{V}(\bar{\theta})}{\partial \bar{\theta}} = \begin{bmatrix} J_1(\bar{\theta}) \cdots & J_r(\bar{\theta}) & \cdots & J_R(\bar{\theta}) \end{bmatrix},$$
$$J_r(\bar{\theta}) := \begin{bmatrix} \frac{\partial \tilde{V}}{\partial \theta_r^D} & \frac{\partial \tilde{V}}{\partial \theta_r^U} & \frac{\partial \tilde{V}}{\partial \theta_r^K} & \frac{\partial \tilde{V}}{\partial \theta_r^P} \end{bmatrix}. \tag{5.16}$$

Lemma 5.1 *For given indices $r \in \{1, \cdots, R\}, 1 < R \in \mathbb{N}$ and $k \in \{D, U, K, P\}$, the Jacobian of $\tilde{V}$ with respect to θ_r^k can be expressed as*

$$\frac{\partial \tilde{V}}{\partial \theta_r^k} = \sum_{\ell=1}^{H} \sum_{m=2}^{M-1} \frac{1}{w^m w_\ell} \left[\mathcal{G}_r^{k^*} + \mathcal{G}_r^k \right], \tag{5.17}$$

with

$$\begin{aligned} \mathcal{G}_r^D &= -(e_\ell^m)^* C^m F(\bar{\theta}) L_r^D F(\bar{\theta}) B(\bar{\theta}) q \\ &\quad - (e_\ell^m)^* C^m F(\bar{\theta}) \begin{bmatrix} 0_{(N-2)\times 1} & g_r^D & h_r^D \end{bmatrix} q, \\ \mathcal{G}_r^U &= -(e_\ell^m)^* C^m F(\bar{\theta}) L_r^U F(\bar{\theta}) B(\bar{\theta}) q \\ &\quad - (e_\ell^m)^* C^m F(\bar{\theta}) \begin{bmatrix} 0_{(N-2)\times 1} & g_r^U & h_r^U \end{bmatrix} q, \\ \mathcal{G}_r^K &= -(e_\ell^m)^* C^m F(\bar{\theta}) L_r^K F(\bar{\theta}) B(\bar{\theta}) q, \end{aligned} \tag{5.18}$$

$$\mathcal{G}_r^P = -(e_\ell^m)^* C^m F(\bar{\theta}) \begin{bmatrix} f_r^P & 0_{(N-2)\times 1} & 0_{(N-2)\times 1} \end{bmatrix} q. \tag{5.19}$$

Here, $F(\bar{\theta}) := [j\omega_\ell I - A(\bar{\theta})]^{-1}$ and $\mathcal{G}_r^{k^*}$ represents the adjoint of $\mathcal{G}_r^k$. □

Proof: The proof is included in the Appendix 5.A. ■

Another important aspect in calculating the Jacobian is the inverse in the definition of $F(\bar{\theta})$. Here, the tri-diagonal structure of $A(\bar{\theta})$ can be exploited to calculate a computationally efficient inverse using LU-factorization techniques [31].

Algorithmic Aspects

To minimize the cost functional (5.13), we employ the Levenberg-Marquardt algorithm (see [17], [64]). This algorithm is a combination of the gradient descent method and the Gauss-Newton method to improve its robustness [17]. At a particular iteration i, the descent direction $h_{\text{lm},i}$ is formulated based on the evaluated Jacobian and satisfies

$$(J^* J + \lambda_i I) h_{\text{lm},i} = J^* \tilde{V}(\bar{\theta}_i). \tag{5.20}$$

The non-negative scalars λ_i can be adjusted over each iteration to adaptively switch between a gradient descent method and the Gauss-Newton method [17]. A detailed analysis about this algorithm, its computational aspects and local convergence is provided in [17]. Here, the analytic computation of the Jacobian plays an important role. By exploiting the sparsity of the state space matrices, the analytic expression of the Jacobian can produce a fast computation of descent direction and accelerate the optimization.

5.4 Academic Case Study

This section describes the implementation of the proposed estimation algorithm, presents simulation results, and briefly discusses the design choices. The estimation algorithm is implemented in Matlab®, which is computationally optimized for solving sparse linear matrices [67]. The algorithm has been implemented with two variations: a) with unknown but static boundary conditions (without boundary inputs) and b) using the measurements from the outer two sensors as approximated boundary inputs. Owing to the affine parametrization of the basis functions in the state space model, the implemented algorithm facilitates a modular structure such that any user-defined basis functions can be easily implemented and tested.

5.4.1 Data Generation for the Case Study

In order to test the developed algorithm, we consider a simulation study that mimics a real experiment for generating the input-output data. Here, we are interested in estimating the thermal properties of a heterogeneous solid object. In particular, for all $x \in [0, 1]$, we are interested in determining the spatial profiles of the diffusion coefficient $D(x)$, the transport coefficient $V(x)$, the reaction coefficient $K(x)$ and the locations of two adjacent heat sources $P(x)$ on the basis of the following PDE.

$$\frac{\partial z}{\partial t} = D(x)\frac{\partial^2 z}{\partial x^2} + U(x)\frac{\partial z}{\partial x} + K(x)z + P(x)p(t). \tag{5.21}$$

with boundary conditions $\frac{\partial z}{\partial x}(0, t) = 0$ and $z(1, t) = 0$. The in-domain input $p(t)$ is a block-wave excitation signal (symmetric pulse train). We use the first four Fourier modes of the block-wave.

In practice, the thermal source is often represented by a Gaussian function and the thermal transport coefficients are represented by polynomial functions while considering slab geometry (c.f. [34]). For the data generation using (5.21), we consider the following profiles for $\{D, U, K\}$,

$$D^{\text{sim}}(x) := 5x^3 - 0.005x + 5, \tag{5.22}$$

$$U^{\text{sim}}(x) := 15x^2 - 0.005, \tag{5.23}$$

$$K^{\text{sim}}(x) := -3x. \tag{5.24}$$

Additionally, we consider two heat sources located around $x = 0.35$ and $x = 0.6$ and defined by setting

$$P^{\text{sim}}(x) := 0.2 + \frac{7}{\sqrt{\pi}} \mathrm{e}^{\frac{-(x-0.35)^2}{(0.1)^2}} + \frac{5.6}{\sqrt{\pi}} \mathrm{e}^{\frac{-(x-0.6)^2}{(0.1)^2}}. \tag{5.25}$$

The problem set-up is equipped with $M = 8$ sensors inside the spatial domain with the extremum two senors placed at $x = 0.1967$ and $x = 0.8967$, respectively. These two extremum sensors are also considered as additional two inputs. The input-output data can be generated by simulating the model (5.21) along with (5.22)-(5.25) on a discretized spatial domain $\mathbb{X}_d$ and evaluating it on the measurement set $\mathbb{X}^M$. Specifically, the discretization grid $\mathbb{X}_d$ is chosen such that the measurement set $\mathbb{X}^M$ is a subset of $\mathbb{X}_d$. As the estimation procedure considers the initial conditions to be unknown, the measurement data is collected after the transient is settled.

5.4.2 Estimation Results

For estimating the real-valued functions D, U, K, P defined on $\mathbb{X} = [0, 1]$, we use the simulated input-output data that is generated in the previous subsection. The estimation procedure has no knowledge about the initial and boundary conditions of the data generating system. In principle, the implementation of the estimation algorithm is generic such that any user-defined basis can be incorporated as a shape function for the unknown parameters. Here, we directly include the prior knowledge about the shape of the unknown functions in the estimation of $\{D, U, K, P\}$. This prior knowledge about the shape of the physical coefficients is a reasonable consideration as far as practical applications are concerned. In practice such an estimation procedure is preceded by various physical experiments that provide useful information about the profile of the unknown parameters. Therefore, using only the prior knowledge about the shape of the unknown parameters, we choose polynomial basis functions to parameterize $\{D, U, K\}$. To parametrize P, we use b-spline functions where the basis functions $B_r^s(x)$ are designed using de Boor's recursion formula [30] with 13 control points. Precisely,

$$D(x) = \sum_{r=1}^{8} x^{r-1}\theta_r^D,\ U(x) = \sum_{r=1}^{6} x^{r-1}\theta_r^U, K(x) = \sum_{r=1}^{3} x^{r-1}\theta_r^K,\ P(x) = \sum_{r=1}^{13} B_r^s(x)\,\theta_r^P.$$

As the shape of the basis functions are known a-priori, the unknowns are given in a vector $\bar{\theta} \in \mathbb{R}^{30}$ that has to be estimated by minimizing (5.13). The state space model in (5.5) is derived by spatial discretization on 211 discrete points. The boundary conditions and the initial conditions are unknown to the estimation problems. Figure 2-5 compares the simultaneously estimated $\{D, U, K, P\}$ with respect to space starting from an initial guess. Based on the simulation results in

Figure 2-5, the estimated profile of $\{D, U, K, P\}$ closely matches with the original functions except small errors near the extremum measurements.

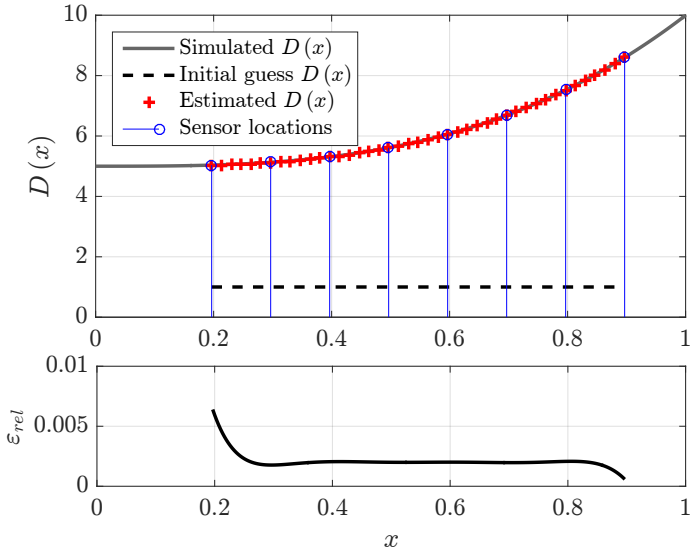


Figure 2: Estimated $D(x)$ over $\mathbb{X}^M$ and the relative error ε_{rel} between $D^{\text{sim}}(x)$ and estimated $D(x)$.

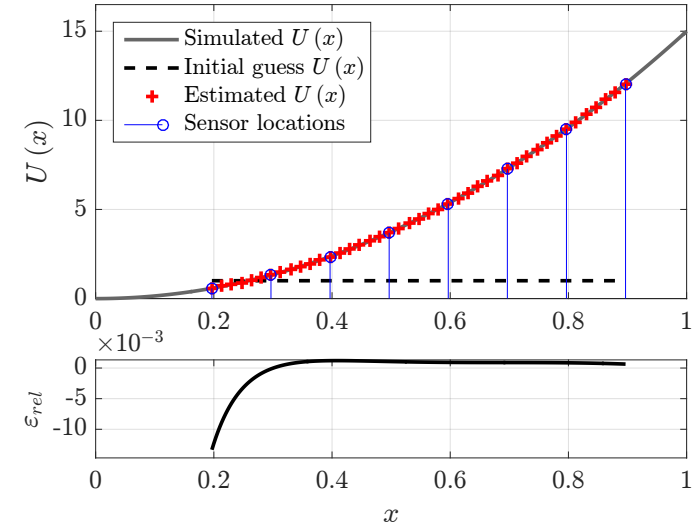


Figure 3: Estimated $U(x)$ over $\mathbb{X}^M$ and the relative error ε_{rel} between $U^{\text{sim}}(x)$ and estimated $U(x)$.

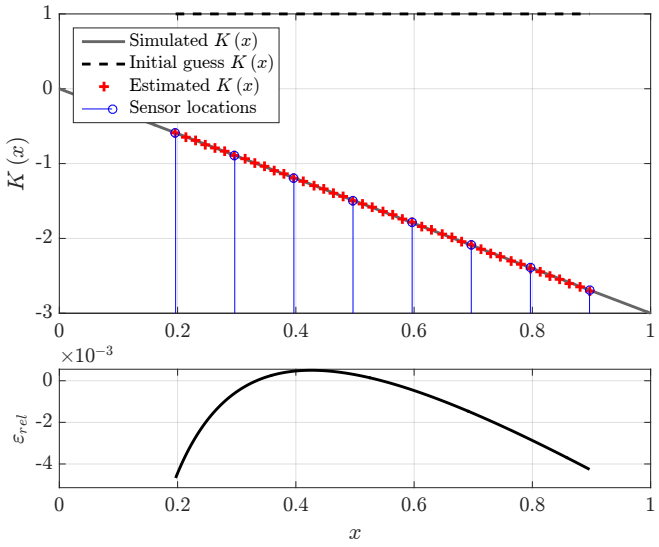


Figure 4: Estimated $K(x)$ over $\mathbb{X}^M$ and the relative error ε_{rel} between $K^{\text{sim}}(x)$ and estimated $K(x)$.

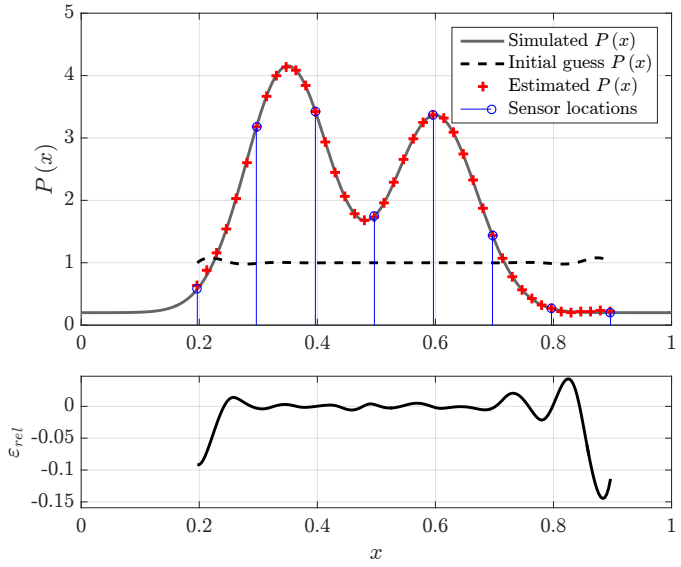


Figure 5: Estimated $P(x)$ over $\mathbb{X}^M$ and the relative error ε_{rel} between $P^{\text{sim}}(x)$ and estimated $P(x)$.

5.5 Closing Remarks

A frequency domain approach has been proposed to estimate the spatial profile of the physical parameters of a diffusion-transport-reaction model described by a PDE with distributed coefficients and using measurements from a limited number of sensors. The method uses a finite difference technique to discretize the model and non-linear least square technique in combination with a-priori choices of basis functions to estimate the unknown parameters. The spatial discretization of the model preserves the linear affine relation of the state space matrices with respect to the unknown parameters. The sparsity structure of the state space matrices is another distinctive feature of the discretized model that has been exploited throughout to offer computational benefit. For the optimization, the required Jacobian is analytically derived for fast computation exploiting its sparse structure. The simulation results show that the method optimally estimates diffusion, transport, reaction and input distributions simultaneously.

The discrepancy between the basis functions used in simulation and the parameter estimator can cause serious estimation error, especially at the boundaries. This issue has to be dealt with in future. Moreover, robustness with respect to the measurement uncertainty, i.e., the noise has currently not been investigated. However, weighting factors for taking colored output measurement noise into account have already been implemented (not considered in this chapter).

Appendices

5.A Proof of Lemma 5.1

Proof: Here, we derive the analytic expression of (5.17). The term $\mathcal{G}_r^k$ can be expanded by substituting the expression of e_ℓ^m and $G^m(\omega_\ell, \bar{\theta})$. Precisely,

$$\mathcal{G}_r^k = -(e_\ell^m)^* \Big[\frac{\partial G^m(\omega_\ell, \bar{\theta})}{\partial \theta_r^k}\Big] q. \tag{5.26}$$

The general expression of $\frac{\partial G^m}{\partial \theta_r^k}$ with $k \in E$ can be computed using chain rule. Precisely, substituting the expression of G^m we obtain

$$\frac{\partial G^m}{\partial \theta_r^k} = C^m \frac{\partial}{\partial \theta_r^k}\Big[j\omega_\ell I - A(\bar{\theta})\Big]^{-1} B(\bar{\theta}) \tag{5.27}$$

$$+ C^m \Big[j\omega_\ell I - A(\bar{\theta})\Big]^{-1} \frac{\partial B(\bar{\theta})}{\partial \theta_r^k}. \tag{5.28}$$

At this stage we use the identity $\frac{\partial \mathcal{M}^{-1}}{\partial \theta_r^k} \equiv -\mathcal{M}^{-1} \frac{\partial \mathcal{M}}{\partial \theta_r^k} \mathcal{M}^{-1}$. Subsequently, using the expression of $A(\bar{\theta})$ and $B(\bar{\theta})$ for $k \in \{D, U, K\}$, we write

$$\frac{\partial G^m}{\partial \theta_r^k} = C^m \Big[j\omega_\ell I - A(\bar{\theta})\Big]^{-1} L_r^k \Big[j\omega_\ell I - A(\bar{\theta})\Big]^{-1} B(\bar{\theta})$$

$$+ C^m \Big[j\omega_\ell I - A(\bar{\theta})\Big]^{-1} \frac{\partial B(\bar{\theta})}{\partial \theta_r^k}. \tag{5.29}$$

Moreover, $\frac{\partial B(\bar{\theta})}{\partial \theta_r^k}$ can be computed for $k \in \{D, U, K\}$.

$$\frac{\partial B(\bar{\theta})}{\partial \theta_r^D} = \begin{bmatrix} 0_{(N-2)\times 1} & g_r^D & h_r^D \end{bmatrix}, \tag{5.30}$$

$$\frac{\partial B(\bar{\theta})}{\partial \theta_r^V} = \begin{bmatrix} 0_{(N-2)\times 1} & g_r^U & h_r^U \end{bmatrix}, \tag{5.31}$$

$$\frac{\partial B(\bar{\theta})}{\partial \theta_r^K} = 0_{(N-2)\times 3}. \tag{5.32}$$

For θ_r^P, the Jacobian can be computed as

$$\frac{\partial G^m}{\partial \theta_r^P} = C^m \Big[j\omega_\ell I - A(\bar{\theta}) \Big]^{-1} \frac{\partial B(\bar{\theta})}{\partial \theta_r^P}, \tag{5.33}$$

with

$$\frac{\partial B(\bar{\theta})}{\partial \theta_r^P} = \begin{bmatrix} f_r^P & 0_{(N-2)\times 1} & 0_{(N-2)\times 1} \end{bmatrix}. \tag{5.34}$$

The above derivations can be used to find the analytic expression of $\mathcal{G}_r^k$ for every $r \in \{1, \cdots, R\}$ and $k \in \{D, U, K, P\}$. In a similar fashion,

$$\mathcal{G}_r^{k^*} = -q^* \frac{\partial}{\partial \theta_r^k} \Big[G^m(\omega_\ell, \bar{\theta}) \Big]^* e_\ell^m$$

$$\begin{aligned} \frac{\partial (G^m)^*}{\partial \theta_r^k} = & \frac{\partial B^\top(\bar{\theta})}{\partial \theta_r^k} \Big[-j\omega_\ell I - A^\top(\bar{\theta}) \Big]^{-1} C^{m\top} \\ & + B^\top(\bar{\theta}) \frac{\partial}{\partial \theta_r^k} \Big[-j\omega_\ell I - A^\top(\bar{\theta}) \Big]^{-1} C^{m\top}. \end{aligned}$$

We can follow the derivation of $\mathcal{H}_l^m$ to derive a similar analytic expression of $\mathcal{G}_l^m$. ■

Part IV

Computation with PIEs

6 CHAPTER

Building Digital Twin with PIEs

The main computational bottleneck in building model-based functionalities on the digital twin is the presence of spatially distributed nodes in a thermo-fluidic process. A spatially distributed node's dynamics is governed by PDEs. So far, it has been shown that one must use a lumping or numerical approximation technique to first approximate the PDE models and only then is the model-based simulation, analysis, prediction, diagnosis, estimation or control possible. Despite tremendous strides in the field of computation, directly utilizing infinite-dimensional systems (such as PDEs) for simulation, analysis, prediction, diagnosis, estimation or control is an open problem. In this chapter, this problem is circumvented by utilizing a class of linear bounded operators, known as Partial Integral (PI) operators and linear equations known as Partial Integral Equations (PIEs). It is shown that model of thermo-fluidic processes can be equivalently represented using PIEs. Moreover, analysis of stability, input-output properties, and synthesis of optimal state estimators can be formulated as Linear PI equalities (LPIs) that can be solved using Linear Matrix equalities (LMIs), a mature computational method.

Outline

6.1 Introduction 118
6.2 PI Operators, PIEs and PIETOOLS 120
6.3 Thermo-Fluidic Processes and PDE-ODE Systems 125
6.4 Representation of Thermo-Fluidic Processes Using PIEs 127
6.5 Analysis and Control of Thermo-Fluidic Processes Using PIEs . 133
6.6 Synthesis of $\mathcal{H}_\infty$ Optimal State Estimator 141
6.7 Closing Remarks 149

6.1 Introduction

In Part II and III, the digital twins are a finite-dimensional approximation of spatio-temporal thermo-fluidic processes. Once the model is approximated, standard control theory has a plethora of methods and computational tools that can be used for analysis and control. However, finite-dimensional approximations of systems, that are originally infinite-dimensional in nature do not necessarily represent the exact behavior to a quantified level of accuracy [69]. As a result, serious errors may appear in analyzing stability, input-output properties or synthesizing estimator based controller by solely using approximated models. For example, by spectral decomposition of spatio-temporal systems in Part II, it is inevitable to ignore high-frequent modes. In case of an unforeseen perturbation of the neglected modes, the controlled system often suffers from instability, commonly known as the spillover effect [12]. Another key disadvantage of approximating an infinite-dimensional system with a finite-dimensional model is the fact that such approximations are prone to numerical instabilities and curse of dimensionality. In fact, increased demands on accuracy may require the dimension of the approximated system to be so large (in the range of few million state variables) that solving analysis and synthesis problems become computationally intractable (c.f. [109]).

To circumvent the problems with finite-dimensional approximations, infinite dimensional PDEs can be directly utilized for analysis and control. Semigroup theory (see [25]) offers a theoretical framework to represent,and analyze systems on Banach space. Although semigroup theory lays the foundation for understanding the properties of dynamical systems beyond finite dimension, there is no computational framework that can aid in translating semigroup theory to computable tests that one can implement in a digital environment.

Another widely used computational method for analysis and control is the Backstepping approach (see [55], [1] for PDEs and [89] for PDE-ODE coupled systems). The backstepping technique does not require any approximation of the infinite-dimensional model; however, it may not provide a quantitative guarantee on the system's optimal performance. As a result, for finding induced L_2 gain or designing optimal estimator based controller (e.g. in $\mathcal{H}_\infty$ sense), backstepping method is not typically used (the only exception, according to the best of our knowledge, is the work on stability analysis by [84]). At the same time, various methods are well-established that use analytical solutions of an application-specific distributed parameter systems (for more details, see the works of [48], [59], [15]. For port-Hamiltonian systems see [99]). Spectral analysis was used for designing optimal estimators for specific PDEs in [81, 108]. However, the state of the art involves either prior approximation of infinite dimensional systems (which suffers from the curse of dimensionality and numerical instability), norm-bounding methods (e.g., Poincare inequality, Cauchy-Schwartz inequality, Wirtinger's inequality, etc.) or integration-by-parts (application-specific). The major disadvantage of all these methods is that they are problem-specific and their workings very much depend on the type of PDEs describing the model. For example, treatment of a wave equation and a coupled

heat-wave equations are completely different. Hence, for every new PDE model, one has to take a different approach to solve the same analysis or control related problem. As a result, translating currently available methods into a scalable algorithm with either minimal or no user intervention is not yet possible.

Recently, using the extension of Lyapunov theory to the infinite-dimensional space by [29], many attempts were made to use Sum-of-Squares (SOS) optimization methods for constructing Lyapunov functions for PDE-ODE coupled systems. Some of the notable works include but not limited to [76], [35], [94] and [3]. Furthermore, numerous problems related to the analysis of stability and robustness and controller-estimator synthesis are addressed in [70], [37] etc. Standard SOS techniques provide a computational tool for analyzing systems' performance in terms of LMI tests. However, these works are still limited to specific types of PDEs, not always applicable to the general class of thermo-fluidic processes considered in this thesis. In particular, different boundary conditions (e.g. Dirichlet and Neumann) require distinct treatment to construct an associated Lyapunov functions. Moreover, cases where boundary inputs perturb the PDEs, are considered to be an open problem.

Contribution of this Chapter

In this chapter, the goal is to develop a scalable framework that allows the use of LMIs for analyzing stability and input-output properties or synthesizing estimator based controller for infinite-dimensional thermo-fluidic processes without any approximation. This computational framework relies on an equivalent representation of thermo-fluidic process using Partial Integral Equations (PIEs). PIEs are linear differential equations involving a class of bounded linear operators known as Partial Integral (PI) operators. Motivated by matrices, PI operators are closed under the algebraic operations of addition, concatenation, composition and adjoint, and their construction inherits the structure of matrices. As a result, analogous to the usage of LMIs in a finite-dimensional control system, there are Linear PI inequalities (LPIs) for analysis and control of PIEs. Furthermore, LPIs can be solved using LMIs by exploiting the inherent matrix structure of PI operators. The main contribution of this chapter can be divided into three aspects:

- *Conversion from PDE-ODE coupled systems to PIEs*:

    1. The equivalent PDE-ODE coupled representation of a thermo-fluidic process in this thesis admits an equivalent representation in terms of PIEs.

- *Analysis and Control of PIEs Using LPIs*:

    1. Analysis of exponential stability, input-to state stability, and the worst case disturbance amplification of thermo-fluidic processes can be formulated using LPIs

2. $\mathcal{H}_\infty$ optimal state estimator synthesis can also be formulated in terms of LPIs.

- *PIEs and LPIs are solved using* PIETOOLS:

1. PIETOOLS is developed as a scalable computational tool for manipulation and optimization of PI operators and PIEs [1].
2. Converting PDE-ODE coupled systems to PIEs, formulating LPIs and solving them for analysis and control of PIEs is performed using PIETOOLS.

6.2 PI Operators, PIEs and PIETOOLS

Any operator that maps from a finite dimensional space to another finite dimensional space is bounded and can be written as a matrix. On infinite dimensional inner product spaces, such as Hilbert spaces, such a generalization does not exist. However, there exists a class of bounded operators, called Partial integral (PI) operators, that do facilitate such properties similar to matrices.

6.2.1 Introduction to PI Operators and PIETOOLS

A PI operator is a map from $\mathbb{R}^m \times L_2^n[a,b]$ to $\mathbb{R}^p \times L_2^q[a,b]$ and is defined as follows.

Definition 6.1 *(PI Operator) A PI operator is a bounded linear operator that maps from $\mathbb{R}^m \times L_2^n[a,b]$ to $\mathbb{R}^p \times L_2^q[a,b]$ and is parametrized as*

$$\mathcal{P}\begin{bmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{bmatrix}\begin{bmatrix} x \\ \mathbf{y} \end{bmatrix}(s) = \begin{bmatrix} Px + \int_a^b Q_1(s)\mathbf{y}(s)ds \\ Q_2(s)x + \mathcal{P}_{\{\{R_0,R_1,R_2\}\}}\mathbf{y}(s) \end{bmatrix} \tag{6.1}$$

where $P \in \mathbb{R}^{p\times m}$ is a matrix, $Q_1 : [a,b] \to \mathbb{R}^{p\times n}$, $Q_2 : [a,b] \to \mathbb{R}^{q\times m}$, $R_0 : [a,b] \to \mathbb{R}^{q\times n}$, and $R_1, R_2 : [a,b]\times[a,b] \to \mathbb{R}^{q\times n}$ are bounded integrable functions and $\mathcal{P}_{\{R_0,R_1,R_2\}} : L_2^n[a,b] \to L_2^q[a,b]$ is another PI operator of the form

$$\left(\mathcal{P}_{\{R_0,R_1,R_2\}}\mathbf{x}\right)(s) := R_0(s)\mathbf{x}(s) + \int_a^s R_1(s,\theta)\mathbf{x}(\theta)d\theta + \int_s^b R_2(s,\theta)\mathbf{x}(\theta)d\theta.$$

Declaring PI Operators in PIETOOLS

PIETOOLS is built as a MATLAB toolbox along with semidefinite programming (SDP) solvers e.g. SeDumi, MOSEK, SDPT3 etc. In PIETOOLS, the PI operator class is declared used the command `opvar`

[1] PIETOOLS is co-developed by the author in collaboration with Sachin Shivakumar and Dr. Matthew Peet at the Arizona State University. For learning how to use PIETOOLS and its functionalities visit `http://control.asu.edu/pietools/`

Table 1: List of properties in `opvar` class and their descriptions in PIETOOLS

Property	Description
`pvar s, theta`	Declare the independent variables s, θ
`P`	A matrix P with dimensions $p \times m$
`Q1`	A matrix-valued polynomial $Q_1(s)$ object with dimensions $p \times n$
`Q2`	A matrix-valued polynomial $Q_2(s)$ object with dimensions $q \times m$
`R.R0`	A matrix-valued polynomial $R_0(s)$ object with dimensions $q \times n$
`R.R1,R.R2`	Matrix-valued polynomial $R_1(s,\theta), R_2(s,\theta)$ objects with dimensions $q \times n$
`I`	The spatial interval defined as `[a,b]`
`dim`	A matrix storing the dimensions of mapped spaces `[p,m;q,n]`

6.2.2 Partial Integral Equations

Partial Integral Equations (PIEs) are set of linear differential equations that are parametrized by PI operators. The general form of a PIE is

$$\begin{aligned} \mathcal{T}\dot{\mathbf{v}}(t) + \mathcal{T}_{Bw}\dot{w}(t) + \mathcal{T}_{Bu}\dot{u}(t) &= \mathcal{A}\mathbf{v}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t), \\ z(t) &= \mathcal{C}_1\mathbf{v}(t) + \mathcal{D}_{11}w(t) + \mathcal{D}_{12}u(t), \\ y(t) &= \mathcal{C}_2\mathbf{v}(t) + \mathcal{D}_{21}w(t) + \mathcal{D}_{22}u(t), \\ \mathbf{v}(0) &= \mathbf{v}_0 \in \mathbb{R}^m \times L_2^n[a,b], \end{aligned} \tag{6.2}$$

where $\mathcal{T}, \mathcal{A} : \mathbb{R}^m \times L_2^n[a,b] \to \mathbb{R}^m \times L_2^n[a,b]$, $\mathcal{T}_{Bw} : \mathbb{R}^p \to \mathbb{R}^m \times L_2^n[a,b]$, $\mathcal{T}_{Bu} : \mathbb{R}^q \to \mathbb{R}^m \times L_2^n[a,b]$, $\mathcal{B}_1 : \mathbb{R}^p \to \mathbb{R}^m \times L_2^n[a,b]$, $\mathcal{B}_2 : \mathbb{R}^q \to \mathbb{R}^m \times L_2^n[a,b]$, $\mathcal{C}_1 : \mathbb{R}^m \times L_2^n[a,b] \to \mathbb{R}^k$, $\mathcal{C}_2 : \mathbb{R}^m \times L_2^n[a,b] \to \mathbb{R}^l$, $\mathcal{D}_{11} \in \mathbb{R}^{k\times p}$, $\mathcal{D}_{12} \in \mathbb{R}^{k\times q}$, $\mathcal{D}_{21} \in \mathbb{R}^{l\times p}$ and $\mathcal{D}_{22} \in \mathbb{R}^{l\times q}$ are PI operators.

6.2.3 Properties of Partial Integral Operators

PI operators have the following properties:

- **Addition of two PI operators is also a PI operator**

Suppose $A, L \in \mathbb{R}^{m\times p}$ and $B_1, M_1 : [a,b] \to \mathbb{R}^{m\times q}$, $B_2, M_2 : [a,b] \to \mathbb{R}^{n\times p}$, $C_0, N_0 : [a,b] \to \mathbb{R}^{n\times q}$, $C_i, N_i : [a,b] \times [a,b] \to \mathbb{R}^{n\times q}$, for $i \in \{1,2\}$, are matrix valued polynomials. If

$$\mathcal{P}\begin{bmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{bmatrix} = \mathcal{P}\begin{bmatrix} A, & B_1 \\ B_2, & \{\{C_0, C_1, C_2\}\} \end{bmatrix} + \mathcal{P}\begin{bmatrix} L, & M_1 \\ M_2, & \{\{N_0, N_1, N_2\}\} \end{bmatrix},$$

then $\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right]$ is parametrized by matrix valued polynomials.

- **Composition of two PI operators is also a PI operator**

Suppose $\mathcal{P}\left[\begin{smallmatrix} A, & B_1 \\ B_2, & \{\{C_0, C_1, C_2\}\} \end{smallmatrix}\right] : \mathbb{R}^l \times L_2^k[a,b] \to \mathbb{R}^m \times L_2^n[a,b]$ and $\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right] : \mathbb{R}^p \times L_2^q[a,b] \to \mathbb{R}^l \times L_2^k[a,b]$ are PI operators parametrized by matrix valued polynomials. Then $\mathcal{P}\left[\begin{smallmatrix} \hat{P}, & \hat{Q}_1 \\ \hat{Q}_2, & \left\{\{\hat{R}_0, \hat{R}_1, \hat{R}_2\}\right\} \end{smallmatrix}\right]$ is a PI operator parametrized by matrix valued polynomials where

$$\mathcal{P}\left[\begin{smallmatrix} \hat{P}, & \hat{Q}_1 \\ \hat{Q}_2, & \left\{\{\hat{R}_0, \hat{R}_1, \hat{R}_2\}\right\} \end{smallmatrix}\right] = \mathcal{P}\left[\begin{smallmatrix} A, & B_1 \\ B_2, & \{\{C_0, C_1, C_2\}\} \end{smallmatrix}\right] \mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right].$$

- **Adjoint of a PI operator is also a PI operator**

Suppose $P \in \mathbb{R}^{m\times p}$ and $Q_1 : [a,b] \to \mathbb{R}^{m\times q}$, $Q_2 : [a,b] \to \mathbb{R}^{n\times p}$, $R_0 : [a,b] \to \mathbb{R}^{n\times q}$, $R_1, R_2 : [a,b]^2 \to \mathbb{R}^{n\times q}$ are bounded functions. Then for any $x \in \mathbb{R}^m \times L_2^n[a,b], y \in \mathbb{R}^p \times L_2^q[a,b]$,

$$\left\langle \mathcal{P}\left[\begin{smallmatrix} \hat{P}, & \hat{Q}_1 \\ \hat{Q}_2, & \left\{\{\hat{R}_0, \hat{R}_1, \hat{R}_2\}\right\} \end{smallmatrix}\right] x, y \right\rangle_{\mathbb{R}\times L_2} = \left\langle x, \mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right] y \right\rangle_{\mathbb{R}\times L_2}. \tag{6.3}$$

With these properties, PI operators form a *-subalgebra with binary operations of addition and composition.

Operations on PI Operators in PIETOOLS

Operations on PI operators in PIETOOLS are executed in a similar manner to matrices

Table 2: List of operations on PI operators in PIETOOLS

Property	Description
+	Summation of two PI operators
*	Composition of two PI operators
'	Adjoint of a PI operator
[A B]	Horizontal concatenation of two PI operators
[A;B]	Vertical concatenation of two PI operators

6.2.4 Positivity of PI Operators

The main advantage of using a PI operator is that the positivity of a PI operator can be verified using LMIs.

Theorem 6.1 ***(Positivity of PI Operators)*** *For any integrable functions $Z_1 : [a,b] \to \mathbb{R}^{d_1\times n}$, $Z_2 : [a,b]\times[a,b] \to \mathbb{R}^{d_2\times n}$, if $g(s) \geq 0$ for all $s \in [a,b]$ and*

$$
\begin{aligned}
P &= T_{11}\int_a^b g(s)ds,\\
Q(\eta) &= g(\eta)T_{12}Z_1(\eta) + \int_\eta^b g(s)T_{13}Z_2(s,\eta)ds + \int_a^\eta g(s)T_{14}Z_2(s,\eta)ds,\\
R_1(s,\eta) &= g(s)Z_1(s)^\top T_{23}Z_2(s,\eta) + g(\eta)Z_2(\eta,s)^\top T_{42}Z_1(\eta)\\
&\quad + \int_s^b g(\theta)Z_2(\theta,s)^\top T_{33}Z_2(\theta,\eta)d\theta\\
&\quad + \int_\eta^s g(\theta)Z_2(\theta,s)^\top T_{43}Z_2(\theta,\eta)d\theta + \int_a^\eta g(\theta)Z_2(\theta,s)^\top T_{44}Z_2(\theta,\eta)d\theta,\\
R_2(s,\eta) &= g(s)Z_1(s)^\top T_{32}Z_2(s,\eta) + g(\eta)Z_2(\eta,s)^\top T_{24}Z_1(\eta)\\
&\quad + \int_\eta^b g(\theta)Z_2(\theta,s)^\top T_{33}Z_2(\theta,\eta)d\theta\\
&\quad + \int_s^\eta g(\theta)Z_2(\theta,s)^\top T_{34}Z_2(\theta,\eta)d\theta + \int_a^s g(\theta)Z_2(\theta,s)^\top T_{44}Z_2(\theta,\eta)d\theta,\\
R_0(s) &= g(s)Z_1(s)^\top T_{22}Z_1(s).
\end{aligned}
\tag{6.4}
$$

where

$$
\begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14}\\ T_{21} & T_{22} & T_{23} & T_{24}\\ T_{31} & T_{32} & T_{33} & T_{34}\\ T_{41} & T_{42} & T_{43} & T_{44}\end{bmatrix} \succcurlyeq 0,
$$

then $\left\langle x, \mathcal{P}\left[\begin{smallmatrix} P, & Q_1\\ Q_2, & \{\{R_0, R_1, R_2\}\}\end{smallmatrix}\right]x\right\rangle_{\mathbb{R}\times L_2} \geq 0$ *for all* $x \in \mathbb{R}^m \times L_2^n[a,b]$.

Proof: The proof is included in Appendix 6.A. ■

Remark 6.1 *Using Theorem 6.1, the positivity of* $\mathcal{P}\left[\begin{smallmatrix} P, & Q_1\\ Q_2, & \{\{R_0, R_1, R_2\}\}\end{smallmatrix}\right]$ *becomes a feasibility test defined by LMIs. If Q, R_0, R_1, R_2 are matrix-valued polynomials, typical choices for Z_1, Z_2 are matrices whose rows are monomial basis of degree d_1, d_2 respectively. Moreover, to constrain* $\mathcal{P}\left[\begin{smallmatrix} P, & Q_1\\ Q_2, & \{\{R_0, R_1, R_2\}\}\end{smallmatrix}\right]$ *to be positive on the domain $[a,b]$, $g(s)$ is chosen to be the combination of $g(s) = 1$ and $g(s) = (s-a)(b-s)$.*

To search for a feasible $\mathcal{P}\left[\begin{smallmatrix} P, & Q_1\\ Q_2, & \{\{R_0, R_1, R_2\}\}\end{smallmatrix}\right] \succcurlyeq 0$ using Theorem 6.1, define the cone of positive PI operators with polynomial multipliers and kernels.

Definition 6.2 *Let $P \in \mathbb{R}^{m\times m}$ be a constant real-valued matrix and $Q : [a,b] \to \mathbb{R}^{m\times n}$, $S : [a,b] \to \mathbb{R}^{n\times n}$, and $R_1, R_2 : [a,b]\times[a,b] \to \mathbb{R}^{n\times n}$ be*

matrix-valued polynomials. Then, the cone of the positive operators $\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0,R_1,R_2\}\}\end{smallmatrix}\right]$ *is defined by*

$$\Xi_{d_1}^{d_2} := \left\{ \begin{array}{c} \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0,R_1,R_2\}\}\end{smallmatrix}\right] + \mathcal{P}\left[\begin{smallmatrix} M, & N \\ N^\top, & \{\{S_0,S_1,S_2\}\}\end{smallmatrix}\right] \\ (P,Q,Q^\top,R_0,R_1,R_2) \text{ and } (M,N,N^\top,S_0,S_1,S_2) \text{ satisfies Theorem 6.1 with,} \\ Z_1:[a,b]\to\mathbb{R}^{d_1\times n}, Z_2:[a,b]\times[a,b]\to\mathbb{R}^{d_2\times n} \\ g(s)=1 \text{ and } g(s)=(b-s)(s-a) \end{array} \right\}. \tag{6.5}$$

With this definition, the constraint $\mathcal{P}\left[\begin{smallmatrix} A, & B \\ B^\top, & \{\{C_0,C_1,C_2\}\}\end{smallmatrix}\right] \in \Xi_{d_1}^{d_2}$ is understood to be an LMI constraint on the coefficients of the polynomial functions $(A,B,B^\top,C_0,C_1,C_2)$ and guarantees $\mathcal{P}\left[\begin{smallmatrix} A, & B \\ B^\top, & \{\{C_0,C_1,C_2\}\}\end{smallmatrix}\right] \succcurlyeq 0$.

Declaring Positive PI Operator in PIETOOLS

For declaring a positive PI operator, the corresponding commands are given in Table 3. Consult the PIETOOLS website for more details.

Table 3: Commands to declare positive PI operators

Command	Description
`p=sosprogram([s,t])`	1. Initialize a LPI program named `p` 2. `s` and `t` are declared as `pvar` objects
`[p,T] = poslpivar(p,n,I,d)`	1. `p`: an sosprogram that stores all decision variables and constraints 2. `n`: 2×1 vector specifying the dimensions of `T` 3. `I`: 2×1 vector specifying the interval of `T` 4. `d` (optional): specifies the degrees of PI operator. `d` is cell structure of the form $\{a,[b0,b1,b2]\}$ where a is the degree of s in $Z_1(s)$, $b0$ is degree of s in $Z_2(s,\theta)$, $b1$ degree of θ in $Z_2(s,\eta)$ and $b2$ is degree of s and η combined in $Z_2(s,\eta)$. (see Theorem 6.1)

6.2.5 Linear PI Inequalities (LPIs)

Using the algebra of PI operators, one can set up linear operator inequalities (LOIs), now, involving PI operators. These inequalities when defined by PI

operators are called Linear PI Inequalities or LPIs.

Definition 6.3 *For given PI operators $\{\mathcal{E}_{i,j}, \mathcal{F}_{i,j}, \mathcal{G}_i\}$ and convex linear functional $\mathcal{L}(\cdot)$, a linear PI inequality is a convex optimization of the following form*

$$\min_{P_i, Q_{1i}, Q_{2i}, R_{0i}, R_{1i}, R_{2i}} \mathcal{L}(\{P_i, Q_{1i}, Q_{2i}, R_{0i}, R_{1i}, R_{2i}\})$$
$$\sum_{j=1}^{K} \mathcal{E}_{ij}^* \mathcal{P}\left[\begin{smallmatrix} P_i, & Q_{1i} \\ Q_{2i}, & \{\{R_{0i}, R_{1i}, R_{2i}\}\} \end{smallmatrix}\right] \mathcal{F}_{ij} + \mathcal{G}_i \succcurlyeq 0 \tag{6.6}$$

Declaring and Solving LPIs in PIETOOLS

For declaring and solving LPIs in PIETOOLS following commands are useful. For more details, consult the PIETOOLS website.

Table 4: Commands to declare and solve LPIs

Command	Description
`[p,T] = lpivar(p,n,I,d)`	declares an unknown indefinite PI operator
`p = lpi_eq(p,T)`	Declares $\mathcal{T} = 0$
`p = lpi_ineq(p,T)`	declares $\mathcal{T} \succcurlyeq 0$
`p=sosdecvar(p, gamma)`	declares `gamma` as a decision variable
`p=sossetobj(p, gamma)`	cost function to minimize `gamma`
`p=sossolve(p)`	solves the optimization problem

6.3 Thermo-Fluidic Processes and PDE-ODE Systems

In Part II, those thermo-fluidic processes are considered whose spatial distributions are negligible. In Part III, those thermo-fluidic models are considered whose nodes are solely governed by diffusion-transport-reaction type of PDEs. In contrast, this chapter takes into account the entire class of thermo-fluidic processes as defined in (D.1)-(D.3) (on page 23-24). In this chapter, the representation of time-varying PDE-ODE coupled systems in (2.2) on page 43 under the following specification.

- For every lumped node $\mathfrak{N}_j$, it is assumed that the parameters are, i.e. $\Theta_j = \Theta, \forall t \in \mathbb{T}$.

As a result, in (2.21) on page 37, $\breve{\Theta}(t) = \breve{\Theta}, \forall t \in \mathbb{T}$. As a result, all the operators in (2.33) on page 44 are time-invariant and can be defined by the subspace $\bar{\mathscr{P}}_{\mathrm{p}}$ such

that

$$\bar{\mathscr{P}}_{\mathrm{p}}:=\left\{\begin{array}{c}\operatorname{col}\Big((z,y),(x,\mathbf{x}),(w,u)\Big)\mid\forall t\in[0,\infty)\\ \operatorname{col}\left(z(t),y(t)\right)\in\mathbb{R}^{n_z+n_y},\ \mathbf{x}\text{ is Fréchet differentiable},\\ \begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}(t)\end{bmatrix}\in X_{\mathrm{dom}},\\ \begin{bmatrix}z(t)\\y(t)\\\dot{x}(t)\\\dot{\mathbf{x}}(t)\end{bmatrix}=\begin{bmatrix}D_{11}&D_{12}&C_1&\mathcal{C}_{1p}\\D_{21}&D_{22}&C_2&\mathcal{C}_{2p}\\B_{11}&B_{12}&A&\mathcal{E}_p\\\mathcal{B}_{21}&\mathcal{B}_{22}&\mathcal{E}&\mathcal{A}_p\end{bmatrix}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}(t)\end{bmatrix}\end{array}\right\},\tag{6.7}$$

with

$$X_{\mathrm{dom}}:=\left\{\begin{array}{c}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}(t)\end{bmatrix}\in\mathbb{R}^{n_w+n_u+n_x}\times L_2^{n_0}[a,b]\times H_1^{n_1}[a,b]\times H_2^{n_2}[a,b]\mid\\ \begin{bmatrix}B_w&B_u&B_{x0}&\mathcal{B}_c\end{bmatrix}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}(t)\end{bmatrix}=0\\ (\mathcal{B}_c\mathbf{x})(s):=\int_a^b B_{xx}(s)\begin{bmatrix}\mathbf{x}_0\\\partial_s\mathbf{x}_1\\\partial_s^2\mathbf{x}_2\end{bmatrix}(s)\mathrm{d}s-B\begin{bmatrix}\mathbf{x}_1(a)\\\mathbf{x}_1(b)\\\mathbf{x}_2(a)\\\mathbf{x}_2(b)\\\partial_s\mathbf{x}_2(a)\\\partial_s\mathbf{x}_2(b)\end{bmatrix}\end{array}\right\}.\tag{6.8}$$

Here, matrix valued functions A_0,, A_1, A_2, A_{2l}, A_{2u}, E_2, E_a,, E_b, E_c, E_0, E, B_{21}, B_{22}, C_{ai}, C_{bi}, C_{ci}, for $i\in\{1,2\}$, and constant matrices C_{10}, C_{20}, E_0, A, C_1, C_2, B_{11}, B_{12}, D_{11}, D_{12}, D_{21}, D_{22}, of appropriate dimensions are given. Moreover, $B\in\mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$ has full rank. Furthermore,

$$\begin{aligned}(\mathcal{A}_p\mathbf{x})(s):=&A_0(s)\begin{bmatrix}\mathbf{x}_0\\\mathbf{x}_1\\\mathbf{x}_2\end{bmatrix}(s)+A_1(s)\partial_s\begin{bmatrix}\mathbf{x}_1\\\mathbf{x}_2\end{bmatrix}(s)+A_2(s)\partial_s^2\begin{bmatrix}\mathbf{x}_2\end{bmatrix}(s)\\&+\int_a^s A_{2l}(s,\theta)\begin{bmatrix}\mathbf{x}_0\\\mathbf{x}_1\\\mathbf{x}_2\\\partial_s\mathbf{x}_1\\\partial_s\mathbf{x}_2\\\partial_s^2\mathbf{x}_2\end{bmatrix}(\theta)\mathrm{d}\theta+\int_s^b A_{2u}(s,\theta)\begin{bmatrix}\mathbf{x}_0\\\mathbf{x}_1\\\mathbf{x}_2\\\partial_s\mathbf{x}_1\\\partial_s\mathbf{x}_2\\\partial_s^2\mathbf{x}_2\end{bmatrix}(\theta)\mathrm{d}\theta\\&+E_2(s)\begin{bmatrix}\mathbf{x}_1(a)\\\mathbf{x}_1(b)\\\mathbf{x}_2(a)\\\mathbf{x}_2(b)\\\partial_s\mathbf{x}_2(a)\\\partial_s\mathbf{x}_2(b)\end{bmatrix},\end{aligned}\tag{6.9}$$

$$(\mathcal{C}_{1p}\mathbf{x})(s) := \int_a^b \left(C_{a1}(s) \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s) + C_{b1}(s)\partial_s \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s) + C_{c1}(s)\partial_s^2 \left[\mathbf{x}_2\right](s) \right) \mathrm{d}s + C_{10} \begin{bmatrix} \mathbf{x}_1(a) \\ \mathbf{x}_1(b) \\ \mathbf{x}_2(a) \\ \mathbf{x}_2(b) \\ \partial_s\mathbf{x}_2(a) \\ \partial_s\mathbf{x}_2(b) \end{bmatrix}, \tag{6.10}$$

$$(\mathcal{C}_{2p}\mathbf{x})(s) := \int_a^b \left(C_{a2}(s) \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s) + C_{b2}(s)\partial_s \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s) + C_{c2}(s)\partial_s^2 \left[\mathbf{x}_2\right](s) \right) \mathrm{d}s + C_{20} \begin{bmatrix} \mathbf{x}_1(a) \\ \mathbf{x}_1(b) \\ \mathbf{x}_2(a) \\ \mathbf{x}_2(b) \\ \partial_s\mathbf{x}_2(a) \\ \partial_s\mathbf{x}_2(b) \end{bmatrix}, \tag{6.11}$$

$$(\mathcal{E}_p\mathbf{x})(s) := \int_a^b \left(E_a(s) \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s) + E_b(s)\partial_s \left[\mathbf{x}_1\mathbf{x}_2\right](s) + E_c(s)\partial_s^2 \left[\mathbf{x}_2\right](s) \right) \mathrm{d}s + E_0 \begin{bmatrix} \mathbf{x}_1(a) \\ \mathbf{x}_1(b) \\ \mathbf{x}_2(a) \\ \mathbf{x}_2(b) \\ \partial_s\mathbf{x}_2(a) \\ \partial_s\mathbf{x}_2(b) \end{bmatrix}, \tag{6.12}$$

$$(\mathcal{E}x)(s) := E(s)x, (\mathcal{B}_{21}w)(s) := B_{21}(s)w, (\mathcal{B}_{22}u)(s) := B_{22}(s)u. \tag{6.13}$$

6.4 Representation of Thermo-Fluidic Processes Using PIEs

The primary motivation behind searching for a new representation of the thermo-fluidic processes, in contrast to (6.7), is to circumvent the technical difficulties regarding the presence inhomogeneous boundary conditions and unbounded operators. In Chapter 4, it has been already pointed out that, even for approximating diffusion-transport-reaction systems, these two difficulties are the major hurdle in analysis and control. The objective of this section is to utilize PI operators to formulate an equivalent PIE representation of (6.7) which may potentially resolve the aforementioned difficulties. To this end, remainder of this section follows two steps

1. Find a unitary PI operator that maps between the functions space X_{dom} in

(6.8) and the space of square integrable functions; $\mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$.

2. Determine an equivalent PIE representation of (6.7).

6.4.1 Finding the Unitary State Transformation

The space X_{dom} in (6.8) is constrained by the boundary conditions as well as order of differentiability a function must admit. On the other hand, the space $\mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$, is independent of such constraints. A transformation between these spaces can be found using the fundamental theorem of calculus which is stated in Lemma 6.1.

Lemma 6.1 *Let $\mathbf{x}_0 \in L_2^{n_0}[a,b]$, $\mathbf{x}_1 \in H_1^{n_1}[a,b]$, $\mathbf{x}_2 \in H_2^{n_2}[a,b]$. Then, for all $s \in [a,b]$, the following identities hold true:*

$$\begin{bmatrix} \mathbf{x}_0(s) \\ \mathbf{x}_1(s) \\ \mathbf{x}_2(s) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0(s) \\ \mathbf{x}_1(a) + \int_a^s \partial_s \mathbf{x}_1(\theta) d\theta \\ \mathbf{x}_2(a) + (s-a)\partial_s \mathbf{x}_2(a) + \int_a^s (s-a)\partial_s^2 \mathbf{x}_2(\theta) d\theta \end{bmatrix} \tag{6.14}$$

Proof: The proof is given in [78], page 9. ■

Lemma 6.1 shows that, in an interval, the value of a function at any arbitrary point $s \in [a,b]$ is linearly related to the boundary values of the functions and the functions' higher order derivatives. Using this idea, Theorem 6.2 determines a unitary map from $\mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$ to X_{dom}.

Theorem 6.2 ***(State Transformation)*** *Suppose, in (6.8), $B \in \mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$ and*

$$B \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix} \text{ is invertible.}$$

Then, for any $\operatorname{col}(w,u,x,\mathbf{x}) \in X_{\text{dom}}$, the following identity holds

$$\begin{bmatrix} x \\ \mathbf{x} \end{bmatrix} = \left[\begin{array}{cc|c} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{array}\right] \left[\begin{array}{c} w \\ u \\ \hline x \\ \mathbf{x}_f \end{array}\right]. \tag{6.15}$$

where, $\mathbf{x}_f := \begin{bmatrix} \mathbf{x}_0 \\ \partial_s \mathbf{x}_1 \\ \partial_s^2 \mathbf{x}_2 \end{bmatrix}$,

$$\begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} = \mathcal{P}\left[\begin{smallmatrix} [0 & 0 & I], \\ [\mathbf{Q}_{2f}], & & \end{smallmatrix} \begin{smallmatrix} 0 \\ \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\} \end{smallmatrix} \right], \tag{6.16}$$

and,

$$\mathbf{Q}_{2f}(s) = K(s)(BT)^{-1}\begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, \tag{6.17}$$
$$\mathbf{R}_{2f}(s,\theta) = K(s)(BT)^{-1}(B_{xx}(\theta) - BQ(\theta)),$$
$$\mathbf{R}_{1f}(s,\theta) = L_1(s,\theta) + K(s)(BT)^{-1}(B_{xx}(\theta) - BQ(\theta)), \mathbf{R}_{0f}(s) = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$L_1(s,\theta) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}, K(s) = \begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix},$$
$$T = \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix}, Q(s) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & I & (b-s)I \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{6.18}$$

Proof: For a detailed proof, see Appendix 6.B. ■

The state transformation map is unitary

Corollary 6.1 *Suppose, in* (6.8), $B \in \mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$ *has and*

$$B \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix} \text{ is invertible.}$$

Then, $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathcal{T}_{Bw} & \mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix} : \mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b] \to X_{\mathrm{dom}}$ *is unitary where* $\mathcal{T}_{Bw}, \mathcal{T}_{Bw}, \mathcal{T}$ *are defined in Theorem 6.2.*

Proof: Define $\bar{\mathcal{T}} := \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathcal{T}_{Bw} & \mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix}$, and $\bar{\boldsymbol{\partial}}_s := \begin{bmatrix} I & & & & 0 \\ & I & & & \\ & & I & & \\ 0 & & & \partial_s & \\ & & & & \partial_s^2 \end{bmatrix}$.

The inner product $\langle , \rangle_{X_{\rm dom}}$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle_{X_{\rm dom}} := \langle \bar{\boldsymbol{\partial}}_s \mathbf{x}, \bar{\boldsymbol{\partial}}_s \mathbf{y} \rangle_{\mathbb{R} \times L_2}$ for all $\mathbf{x}, \mathbf{y} \in X_{\rm dom}$. Then:

(i). Using Theorem 6.2, one gets

$$\left\langle \bar{\boldsymbol{\partial}}_s \bar{\boldsymbol{\mathcal{T}}} \mathbf{z}_f, \bar{\boldsymbol{\partial}}_s \bar{\boldsymbol{\mathcal{T}}} \mathbf{z}_f \right\rangle_{\mathbb{R} \times L_2} = \left\langle \mathbf{z}_f, \mathbf{z}_f \right\rangle_{\mathbb{R} \times L_2},$$

(ii). Theorem 6.2 also shows that, for any $\mathbf{x} \in X_{\rm dom}$, there exists a $\mathbf{x}_f \in \mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$ such that $\mathbf{x} = \bar{\boldsymbol{\mathcal{T}}} \mathbf{x}_f$, i.e. $\bar{\boldsymbol{\mathcal{T}}}$ is surjective. Furthermore, it has been now proven that for any $\mathbf{x}_f, \mathbf{y}_f \in \mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$, $\langle \bar{\boldsymbol{\mathcal{T}}} \mathbf{x}_f, \bar{\boldsymbol{\mathcal{T}}} \mathbf{y}_f \rangle_{X_{\rm dom}} = \langle \mathbf{x}_f, \mathbf{y}_f \rangle_{\mathbb{R} \times L_2}$. Hence, $\bar{\boldsymbol{\mathcal{T}}}$ is unitary.

■

Remark 6.2 *(X_{dom} is a Hilbert space) Since $\mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$ is a Hilbert space and $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathcal{T}_{Bw} & \mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix}$ is unitary, $X_{\rm dom}$ is a Hilbert space.*

6.4.2 PIEs are Equivalent to PDE-ODE Coupled Systems

In the previous section, a state transformation is found that allows one to equivalently represent any function that belongs to $X_{\rm dom}$using a newly defined set of functions that belong to $\mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$. Moreover, the corresponding map is a unitary PI operator. In this section, the objective is to leverage this map and the algebra of PI operators to determine an equivalent PIE representation of (6.7).

Recalling (6.2), one can define the behavior of a PIE as follows:

Definition 6.4 *(Behavior of PIEs) The behavior of a Partial Integral Equation*

(PIE), is defined as the following subspace $\mathscr{P}_{\mathrm{f}}$.

$$\mathscr{P}_{\mathrm{f}}:=\left\{\begin{array}{c}\mathrm{col}\left((w,u),(x,\mathbf{x}_f)(y,z)\right)\mid\forall t\in[0,\infty),\\ \mathrm{col}\left(z(t),y(t)\right)\in\mathbb{R}^{n_z+n_y},\\ \mathbf{x}_f \text{ is Fréchet differentiable},\\ \begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}_f(t)\end{bmatrix}\in\mathbb{R}^{n_w+n_u+n_x}\times L_2^{n_p}[a,b],\\ \left[\begin{array}{cccc|c}I&0&0&0&0\\0&I&0&0&0\\\hline 0&0&\mathcal{T}_{Bw}&\mathcal{T}_{Bu}&\mathcal{T}\end{array}\right]\begin{bmatrix}z(t)\\y(t)\\\dot{w}(t)\\\dot{u}(t)\\\hline\dot{x}(t)\\\dot{\mathbf{x}}_f(t)\end{bmatrix}=\left[\begin{array}{cc|c}\mathcal{D}_{11}&\mathcal{D}_{12}&\mathcal{C}_1\\\mathcal{D}_{21}&\mathcal{D}_{22}&\mathcal{C}_2\\\hline\mathcal{B}_1&\mathcal{B}_2&\mathcal{A}\end{array}\right]\begin{bmatrix}w(t)\\u(t)\\\hline x(t)\\\mathbf{x}_f(t)\end{bmatrix}\end{array}\right\},\tag{6.19}$$

where, $\mathcal{D}_{11},\mathcal{D}_{12},\mathcal{C}_1,\mathcal{D}_{21},\mathcal{D}_{22},\mathcal{C}_2,\mathcal{B}_1,\mathcal{B}_2,\mathcal{A},\mathcal{T},\mathcal{T}_{Bw},\mathcal{T}_{Bu}$ *are PI operators.*

Theorem 6.3 ***(Equivalence between PDE-ODE and PIEs)*** *Suppose, in* (6.8)*,* $B\in\mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$ *and*

$$B\begin{bmatrix}I&0&0\\I&0&0\\0&I&0\\0&I&(b-a)I\\0&0&I\\0&0&I\end{bmatrix}\text{ is invertible.}$$

Moreover, suppose $\mathrm{col}\left(x,\mathbf{x},w,u,y,z\right)\in\mathscr{P}_{\mathrm{p}}$ *as defined in* (6.7)*. Then* $\mathrm{col}\left(x,\mathbf{x}_f,w,u,y,z\right)\in\mathscr{P}_{\mathrm{f}}$ *according to* (6.4) *if the PI operators* $\mathcal{D}_{11},\mathcal{D}_{12},\mathcal{C}_1,\mathcal{D}_{21},\mathcal{D}_{22},\mathcal{C}_2,\mathcal{B}_1,\mathcal{B}_2,\mathcal{A},\mathcal{T},\mathcal{T}_{Bw},\mathcal{T}_{Bu}$ *in* (6.4) *are defined as follows.*

$$\left[\begin{array}{ccccc}I&0&0&0&0\\0&I&0&0&0\\0&0&\mathcal{T}_{Bw}&\mathcal{T}_{Bu}&\mathcal{T}\end{array}\right]=\mathcal{P}\left[\begin{array}{cc}\begin{bmatrix}I&0&0&0&0\\0&I&0&0&0\\0&0&0&0&I\end{bmatrix},&0\\ \begin{bmatrix}0&0&\mathbf{Q}_{2f}\end{bmatrix},&\{\{\mathbf{R}_{0f},\mathbf{R}_{1f},\mathbf{R}_{2f}\}\}\end{array}\right],\tag{6.20}$$

$$\begin{bmatrix}\mathcal{D}_{11}&\mathcal{D}_{12}&\mathcal{C}_1\\\mathcal{D}_{21}&\mathcal{D}_{22}&\mathcal{C}_2\\\mathcal{B}_1&\mathcal{B}_2&\mathcal{A}\end{bmatrix}=\mathcal{P}\left[\begin{array}{cc}\mathbf{P},&\mathbf{Q}_1\\\mathbf{Q}_2,&\{\{\mathbf{R}_0,\mathbf{R}_1,\mathbf{R}_2\}\}\end{array}\right]\mathcal{P}\left[\begin{array}{cc}\hat{\mathbf{P}}_f,&\hat{\mathbf{Q}}_{1f}(s)\\\hat{\mathbf{Q}}_{2f},&\left\{\{\hat{\mathbf{R}}_{0f},\hat{\mathbf{R}}_{1f},\hat{\mathbf{R}}_{2f}\}\right\}\end{array}\right],\tag{6.21}$$

where

$$\mathbf{P}=\begin{bmatrix} D_{11} & D_{12} & C_1 & C_{10} \\ D_{21} & D_{22} & C_2 & C_{20} \\ B_{11} & B_{12} & A & E_0 \end{bmatrix}, \qquad \mathbf{Q}_1(s)=\begin{bmatrix} C_{a1}(s) & C_{b1}(s) & C_{c1}(s) \\ C_{a2}(s) & C_{b2}(s) & C_{c2}(s) \\ E_a(s) & E_b(s) & E_c(s) \end{bmatrix},$$

$$\mathbf{Q}_2(s)=\begin{bmatrix} B_{21}(s) & B_{22}(s) & E(s) & E_2(s) \end{bmatrix},$$

$$\mathbf{R}_0(s)=\begin{bmatrix} A_0(s) & A_1(s) & A_2(s) \end{bmatrix}, \quad \mathbf{R}_1(s,\theta)=A_{2l}(s,\theta), \quad \mathbf{R}_2(s,\theta)=A_{2u}(s,\theta),$$

$$K(s)=\begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}, \quad L_0=\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad L_1(s,\theta)=\begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix},$$

$$\mathbf{R}_{0f}(s)=L_0, \quad \mathbf{R}_{1f}(s,\theta)=L_1(s,\theta)+\mathbf{R}_{2f}(s,\theta),$$

$$\mathbf{R}_{2f}(s,\theta)=K(s)(BT)^{-1}(B_{xx}(\theta)-BQ(\theta)),$$

$$\mathbf{Q}_{2f}=K(s)(BT)^{-1}\begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, \hat{\mathbf{Q}}_{2f}=\left(\mathbf{I}_1+\mathbf{I}_2\partial_s+\mathbf{I}_3\partial_s^2\right)\mathbf{Q}_{2f},$$

$$\hat{\mathbf{R}}_{1f}=\left(\mathbf{I}_1+\mathbf{I}_2\partial_s+\mathbf{I}_3\partial_s^2\right)\mathbf{R}_{1f}, \hat{\mathbf{R}}_{2f}=\left(\mathbf{I}_1+\mathbf{I}_2\partial_s+\mathbf{I}_3\partial_s^2\right)\mathbf{R}_{2f},$$

$$\hat{\mathbf{R}}_{0f}=\mathbf{I}_1\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}+\mathbf{I}_2\begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}+\mathbf{I}_3\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}, \tag{6.22}$$

and

$$\mathbf{I}_1=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{I}_2=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{I}_3=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix},$$

$$\hat{\mathbf{P}}_f=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ T(BT)^{-1}B_w & T(BT)^{-1}B_u & T(BT)^{-1}B_{xo} \end{bmatrix}, T=\begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix},$$

$$\hat{\mathbf{Q}}_{1f}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ Q(s)+T(BT)^{-1}(B_{xx}(s)-BQ(s)) \end{bmatrix}, Q(s)=\begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (b-s) \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{6.23}$$

Proof: For a detailed proof, see Appendix 6.C. ■

Converting PDE-ODE coupled system to PIEs in PIETOOLS

As the formulae given in Theorem 6.3 are defined by algebraic operations on PI operator, one can convert the entire PDE-ODE coupled systems in (6.7) to a PIE.

In PIETOOLS, the command is simply a one-line script:

```
convert_PIETOOLS_PDE;
```

6.5 Analysis and Control of Thermo-Fluidic Processes Using PIEs

It has been shown that PDE-ODE representation of thermo-fluidic processes in (6.7) are behaviorally equivalent to PIEs in (6.19). The PIEs have the following advantages.

1. PIEs are defined by PI operators. Similar to matrices, the algebra of PI operators is closed under concatenation, composition, adjoint and addition.

2. The evolution of a PIE is not constrained by boundary conditions.

3. Solving convex optimization problems defined by Linear PI Inequalities (LPIs) can be performed using Linear Matrix Inequalities (LMIs).

In the remainder of this chapter, the advantages of the PIE representation and the algebra of PI operators are utilized to solve three analysis and synthesis related problems for the thermo-fluidic processes.

1. *Application of LPIs for analysis of stability and input-output properties*

   - Find a computational test for verifying *exponential stability'* of thermo-fluidic processes.
   - Find a computational test for verifying *input-to-state stability* of thermo-fluidic processes.
   - Find a computational test for determining worst-case disturbance amplification for the behavior of thermo-fluidic processes.

2. *Application of LPIs for synthesizing state estimators*

   - Find a computational algorithm to synthesize and implement an $\mathcal{H}_\infty$ optimal state estimator for thermo-fluidic processes.

6.5.1 Verifying Exponential Stability of Thermo-Fluidic Processes

In [25], exponential stability of an autonomous infinite dimensional system is defined by the exponential decay of the corresponding semigroup, i.e. when all the unforced solutions of the system converge to the origin exponentially fast over time. For $\{\bar{\mathscr{P}}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ where $\bar{\mathscr{P}}_{\mathrm{p}}$ is defined in (6.7), exponential stability is defined as follows

Definition 6.5 *(Exponential stability (ES); convergence of autonomous behavior) The behavior* $\{\bar{\mathscr{P}}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ *is defined to be* ***exponentially stable*** *if there exist constants* M, $\alpha > 0$ *such that any* $\operatorname{col}((w,u),(x,\mathbf{x}),(y,z)) \in \{\bar{\mathscr{P}}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ *satisfies the following inequality for all* $t \in [0,\infty)$

$$\left\| \begin{bmatrix} x(t) \\ \mathbf{x}(t) \end{bmatrix} \right\|_{\mathbb{R}\times L_2} \leq M e^{-\alpha t} \left\| \begin{bmatrix} x(0) \\ \mathbf{x}(0) \end{bmatrix} \right\|_{\mathbb{R}\times L_2}.$$

Formulating LPIs for Exponential Stability of PIEs

Now, PIE representation $\{\mathscr{P}_{\mathrm{f}} \mid w(t) \equiv 0, u(t) \equiv 0\}$, with $\mathscr{P}_{\mathrm{f}}$ defined in (6.19) is used to determine an LPI to test exponential stability.

Theorem 6.4 *Suppose there exist* $\epsilon, \delta > 0$*, a constant real-valued matrix* $P \in \mathbb{R}^{n_x \times n_x}$ *and matrix-valued polynomials* $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$*, and* $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$*, such that*

- $\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^{\top}, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right], \mathcal{P} = \mathcal{P}^* \succcurlyeq \epsilon I$
- *and*

$$\mathcal{A}^* \mathcal{P} \mathcal{T} + \mathcal{T}^* \mathcal{P} \mathcal{A} \preccurlyeq -\delta \mathcal{T}^* \mathcal{T}. \tag{6.24}$$

.

Then, $\{\bar{\mathscr{P}}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ *exponentially stable.*

Proof: The proof is included in the Appendix 6.D. ■

Degree-bounded Tests For Exponential Stability

Searching for a feasible $\mathcal{P} \succ 0$ such that LPI (6.24) is satisfied requires testing positivity of PI operators that is formulated as LMIs. The following feasibility problem enforces the LMI conditions according to the Definition 6.5.

Lemma 6.2 *Suppose there exist* $\epsilon, \delta > 0$, $d_1, d_2 > 0$, $P \in \mathbb{R}^{n_x \times n_x}$, $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$, *and* $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$ *that satisfy*

$$\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \epsilon I, R_1, R_2\}\} \end{smallmatrix}\right] \in \Xi_{d_1}^{d_2}, \tag{6.25}$$

$$-\mathcal{A}^*\mathcal{P}\mathcal{T} - \mathcal{T}^*\mathcal{P}\mathcal{A} - \delta\mathcal{T}^*\mathcal{T} \in \Xi_{d_1}^{d_2}, \tag{6.26}$$

where $\mathcal{T}, \mathcal{A}$ *are defined according to Theorem 6.3. Then,* $\{\mathscr{P}_{\rm p} \mid w(t) \equiv 0, u(t) \equiv 0\}$ *is exponentially stable.*

Academic Illustrations: Exponential Stability of Thermo-Fluidic Processes

Example 6.1 *(Stability of a PDE coupled with ODEs) First, we study the boundary controlled thermo-mechanical process where a lumped mechanical system is driven by converting thermal energy to mechanical work. In [91], such a system is modeled as a finite dimensional ODE with actuator dynamics that are governed by the diffusion equation. A closed loop representation of such a controlled PDE-ODE coupled system is given in [91] as*

$$\begin{aligned} \dot{X}(t) &= -3X(t) + w(0,t), \\ w_t(x,t) &= w_{xx}(x,t), \\ w_x(0,t) &= 0, \\ w(1,t) &= 0. \end{aligned} \tag{6.27}$$

PIETOOLS proves the exponential stability of this system which has also been verified analytically in [91].

6.5.2 Determining the Worst Case Disturbance Amplification

In the previous subsection, the exponential stability of $\bar{\mathscr{P}}_{\rm p}$ in (6.7) is discussed when the system is autonomous; without any inputs and outputs. The focus is now to understanding the effects of inputs $w(t), u(t)$ on the outputs $z(t), y(t)$. In particular, the question is: if the inputs are square integrable (having finite energy), what is the level of amplification or attenuation of such inputs in the outputs? In Definition 6.6, this question is formulated as input-output stability notion; the boundedness of square-integrable outputs in the presence of square-integrable (bounded) inputs.

Definition 6.6 *(Input-Output Stability (IOS): disturbance amplification)*

Let $w \in L_2^{n_w}[0,\infty)$, $u \in L_2^{n_u}[0,\infty)$, $y \in L_2^{n_y}[0,\infty)$, *and* $z \in\in L_2^{n_z}[0,\infty)$. *Then, the behavior* $\mathscr{P}_{\rm p}$ *according to* (6.7) *is defined to be* ***input-to-output stable****, if there exists a constant* $\rho > 0$ *such that for zero initial condition,* $\mathrm{col}\,(x(0), \mathbf{x}(0)) = 0$, *any* $\mathrm{col}\,((w,u,)(x,\mathbf{x}),(y,z)) \in \{\bar{\mathscr{P}}_{\rm p} \mid \mathrm{col}\,(x(0),\mathbf{x}(0)) = 0\}$ *satisfies the following*

inequality

$$\left\| \begin{bmatrix} z \\ y \end{bmatrix} \right\|_{L_2[0,\infty)} \leq \rho \left\| \begin{bmatrix} w \\ u \end{bmatrix} \right\|_{L_2[0,\infty)}.$$

Moreover, ρ is defined as the disturbance amplification factor.

Note that the disturbance amplification factor ρ also indicates the induced L_2 norm of the mapping between the inputs and the outputs. One can also define the smallest value of ρ, ρ_{min}, as follows.

$$\rho_{min} := \sup_{w,u_{\neq 0} \in L_2} \frac{\left\| \begin{bmatrix} z \\ y \end{bmatrix} \right\|_{L_2[0,\infty)}}{\left\| \begin{bmatrix} w \\ u \end{bmatrix} \right\|_{L_2[0,\infty)}}. \tag{6.28}$$

Understanding the behavior of a system in the presence of inputs and outputs is related to dissipation theory of open system; (see [103]). Dissipation theory shows that the amount of energy an open system with inputs and outputs can supply to its environment can not exceed the amount of energy supplied to it. Another striking outcome of dissipation theory is that dissipation properties of linear finite dimensional systems amounts to verifying a feasible sets of Linear Matrix Inequalities (LMIs) (see [85], chapter 2).

In case of $\bar{\mathscr{P}}_{\mathrm{p}}$ in (6.7), its equivalent PIE representation $\mathscr{P}_{\mathrm{f}}$ in (6.19) will used to formulate corresponding LPIs.

Formulating LPIs for Input-Output Stability

Theorem 6.5 *Suppose there exists $\rho > 0$, a constant real-valued matrix $P \in \mathbb{R}^{n_x \times n_x}$ and matrix-valued polynomials $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$, $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$, such that*

- $\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right] \succcurlyeq 0$,
- *and,*

$$\begin{bmatrix} \mathcal{B}_1^* \\ \mathcal{B}_2^* \\ \mathcal{A}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} + \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \preccurlyeq \mathcal{Z} - \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} & \mathcal{C}_1 \\ \mathcal{D}_{21} & \mathcal{D}_{22} & \mathcal{C}_2 \end{bmatrix}^* \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} & \mathcal{C}_1 \\ \mathcal{D}_{21} & \mathcal{D}_{22} & \mathcal{C}_2 \end{bmatrix}. \tag{6.29}$$

with, $\mathcal{P} = \mathcal{P}^*$, $\mathcal{Z} := \mathcal{P}\left[\begin{smallmatrix} \rho I & 0 \\ 0 & 0 \end{smallmatrix}, \begin{smallmatrix} 0 \\ 0, \end{smallmatrix} \begin{smallmatrix} 0 \\ \{0\} \end{smallmatrix}\right]$,

$\mathcal{Z} : \mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b] \to \mathbb{R}^{n_w+n_u+n_x} \times L_2^{n_p}[a,b]$. *Then* $\{\bar{\mathscr{P}}_{\mathrm{p}} \mid \mathrm{col}\,(x(0), \mathbf{x}(0)) = 0\}$ *satisfies the following inequality for any* $w \in L_2^{n_w}[0,\infty)$, $u \in L_2^{n_u}[0,\infty)$, $z \in L_2^{n_z}[0,\infty)$, *and* $y \in L_2^{n_y}[0,\infty)$

$$\left\| \begin{bmatrix} z \\ y \end{bmatrix} \right\|_{L_2[0,\infty)} \leq \sqrt{\rho} \left\| \begin{bmatrix} w \\ u \end{bmatrix} \right\|_{LL_2[0,\infty)}. \tag{6.30}$$

Proof: Proof is given in the Appendix 6.E. ■

Degree-bounded test to determine a bound on the worst-case disturbance amplification

Solving the LPI: $\mathcal{P} \succcurlyeq 0$ such that (6.29) holds true can be formulated as LMIs based on the Definition 6.5. One can also determine a bound on the disturbance amplification by minimizing over all possible $\rho > 0$ that satisfies (6.30). To this end, Lemma 6.3 formulates the corresponding LMI test.

Lemma 6.3 *Suppose there exist,* $d_1, d_2 > 0$, $P \in \mathbb{R}^{n_x \times n_x}$, $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$, $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$ *that satisfy*

$$\hat{\rho} = \arg\min \rho, \tag{6.31}$$

such that

$$\mathcal{P} := \mathcal{P}_{\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0,\, R_1,\, R_2\}\} \end{smallmatrix}\right]} \in \Xi_{d_1}^{d_2}, \tag{6.32}$$

$$\begin{aligned} &- \begin{bmatrix} \mathcal{B}_1^* \\ \mathcal{B}_2^* \\ \mathcal{A}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} - \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \\ &\quad - \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} & \mathcal{C}_1 \\ \mathcal{D}_{21} & \mathcal{D}_{22} & \mathcal{C}_2 \end{bmatrix}^* \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} & \mathcal{C}_1 \\ \mathcal{D}_{21} & \mathcal{D}_{22} & \mathcal{C}_2 \end{bmatrix} + \mathcal{P}_{\left[\begin{smallmatrix} \hat{\rho} I & 0 \\ 0 & 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ \{0\} \end{smallmatrix}\right]} \in \Xi_{d_1}^{d_2} \end{aligned} \tag{6.33}$$

where $\mathcal{T}, \mathcal{A}, \mathcal{B}_1, \mathcal{B}_2, \mathcal{T}_{Bw}, \mathcal{T}_{Bu}, \mathcal{C}_1, \mathcal{C}_2, \mathcal{D}_{11}, \mathcal{D}_{12}, \mathcal{D}_{21}, \mathcal{D}_{22}$ *are defined according to Theorem 6.3. Then* $\{\bar{\mathscr{P}}_{\mathrm{p}} \mid \mathrm{col}\,(x(0), \mathbf{x}(0)) = 0\}$ *satisfies the following inequality for any* $w \in L_2^{n_w}[0,\infty)$, $u \in L_2^{n_u}[0,\infty)$, $z \in L_2^{n_z}[0,\infty)$, *and* $y \in L_2^{n_y}[0,\infty)$

$$\left\| \begin{bmatrix} z \\ y \end{bmatrix} \right\|_{L_2[0,\infty)} \leq \sqrt{\hat{\rho}} \left\| \begin{bmatrix} w \\ u \end{bmatrix} \right\|_{L_2[0,\infty)}.$$

Worst case disturbance amplification: academic example

Example 6.2 *(Stabilizing Boundary Control of PDEs) Consider the example of a PDE stabilized by a backstepping controller at the boundary. The resulting closed-loop system is as follows.*

$$\begin{aligned}\dot{x}(t) &= -3x + w(0,t) + d(t),\\ \dot{w}(s,t) &= w_{ss}(s,t) + d(t), \quad w_s(0,t) = 0,\ w(L,t) = 0,\\ y(t) &= \int_0^L w(s,t)ds\end{aligned}$$

A bound on L_2 gain of this closed-loop system in the presence of disturbance $d(t)$, using PIETOOLS, was found to be 0.4269 for a relatively low order monomial basis (d_1=2, d_2 = 2). On the other hand, the norm bound obtained through a finite difference method (approximately 100 discrete elements) had significant error with a value of 0.5941.

6.5.3 Verifying Input-to-State Stability of Thermo-Fluidic Processes

In the definition of input-output stability, the class of inputs are restricted to be square integrable. In practice, thermo-fluidic disturbances may not always be square integrable. For example, ambient conditions may be constant, sinusoidal disturbances, or square-wave disturbances. These inputs are not square integrable, but are bounded and they are often called persistent disturbances.

To understand a system's behavior under persistence disturbances, the concept of input-to-state stability (ISS) is considered. The ISS is defined as follows

Definition 6.7 *(Input-to-State Stability (ISS); stability to persistent disturbances)*

Let $w \in L_\infty^{n_w}[0,t)$, and $u \in L_\infty^{n_u}[0,t)$. Then, the behavior $\bar{\mathscr{P}}_{\mathrm{p}}$ according to (6.7) *is defined to be* ***input-to-state stable****, if there exist comparison functions $\sigma \in KL$, $\mu \in K$, such that for any $\mathrm{col}\,(w(0),u(0),x(0),\mathbf{x}(0)) \in X_{\mathrm{dom}}$, any $\mathrm{col}\,((w,u),(x,\mathbf{x}),(y,z)) \in \mathscr{P}_{\mathrm{p}}$ satisfies the following inequality for all $t \in [0,\infty)$*

$$\left\|\begin{bmatrix}x(t)\\ \mathbf{x}(t)\end{bmatrix}\right\|_{\mathbb{R}\times L_2} \leq \sigma\left(\left\|\begin{bmatrix}x(0)\\ \mathbf{x}(0)\end{bmatrix}\right\|_{\mathbb{R}\times L_2}, t\right) + \mu\left(\left\|\begin{bmatrix}w\\ u\end{bmatrix}\right\|_{L_\infty([0,t);\mathbb{R})}\right).$$

The theory of ISS for finite dimensional systems is presented in [88]. ISS theory is also used to understand the stability of shear flows under constant perturbation in [2]. Recently, a detailed study is performed for defining and analyzing ISS different classes of PDEs in [48]. Here, based on the type of PDEs, specific treatment is required to show ISS. What follows is that one can analyze a wider class of thermo-fluidic processes in a systematic way using PIE representation and LPIs.

Analysis of ISS for a finite dimensional systems are typically performed by using Lyapunov's theory. Here, one requires to search for a storage functional (also known as ISS Lyapunov functional) and if existence of such ISS storage functional is feasible the system is considered to be input-to-state stable (see [88]). The class of ISS storage functionals is defined as follows

Definition 6.8 *(ISS storage functional)*

Let M, N be normed linear spaces with $M \subseteq N$. A function $V : M \to \mathbb{R}^+$ is defined as ISS storage functional if the following conditions are satisfied

(i). There exists functions $e_1, e_2 \in K$, such that the following inequality holds for all $z \in M$,

$$e_1\left(\|z\|_N\right) \leq V(z) \leq e_1\left(\|z\|_N\right)$$

(ii). The mapping $[0, \infty) \ni t \mapsto V(z(t))$ is absolutely continuous for every $z(t) \in M$.

(iii). There exists a functional $\dot{V} : M \times D \to \mathbb{R}^+$ such that $\frac{d}{dt}V(z(t), d(t)) = \dot{V}(z(t), d(t))$ holds for almost all $t \in [0, \infty)$, for every $z(t) \in M$ and some $d(t) \in D$.

(iv). There exists a semi-norm H of D and positive function $\gamma \in C^0(\mathbb{R}^+; R^+)$ for which the inequality $H(d) \leq \gamma(\|d\|_D)$ holds for all $d \in D$, a positive function $\rho \in C^0(\mathbb{R}^+; R^+)$ and a function $\mu \in K$ such that the inequality $\dot{V}(z, d) \leq -\rho(V(z))$ if $(z, d) \in M \times D$ satisfies $V(z) \geq \mu(H(d))$.

Existence of such storage functional is typically used to prove ISS. In fact, existence of V shows that there exists an upper-bound on the $\|\cdot\|_N$-norm of solutions z in the presence of locally bounded mapping $d : [0, \infty) \to D$ with the following estimate for all $t \geq 0$ (see [47], chapter 2)

$$\|z(t)\|_N \leq \max\left(\sigma(\|z(0)\|_N, t), \sup_{\tau \in [0,t)} e_1^{-1}\Big(\mu\big(\gamma(\|d(\tau)\|_D)\big)\Big)\right), \tag{6.34}$$

for certain $\sigma \in KL$, where $z(t) \in M$ are the trajectories with initial condition $z(0)$ that corresponds to the locally bounded mapping $d : [0, \infty) \to D$.

Formulating LPIs for Input-To-State Stability

Theorem 6.6 *For a constant real-valued matrix $H \in \mathbb{R}^{n_w+n_u \times n_w+n_u}$, define $\mathcal{H} := \mathcal{P}\left[\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}, \begin{matrix} 0 \\ \{0\} \end{matrix}\right]$. Suppose there exist $\tilde{\epsilon}_1, \tilde{\epsilon}_2, \delta > 0$, a constant real-valued matrix $P \in \mathbb{R}^{n_x \times n_x}$ and matrix-valued polynomials $Q : [a, b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a, b] \to \mathbb{R}^{n_p \times n_p}$, $R_1, R_2 : [a, b] \times [a, b] \to \mathbb{R}^{n_p \times n_p}$, such that*

- $\tilde{\epsilon}_2 I \succcurlyeq \mathcal{P}\left[\begin{matrix} P, \\ Q^\top, \end{matrix} \begin{matrix} Q \\ \{\{R_0, R_1, R_2\}\} \end{matrix}\right] \succcurlyeq \tilde{\epsilon}_1 I$

- $H \succ 0$
- *and,*

$$\begin{bmatrix} \mathcal{B}_1^* \\ \mathcal{B}_2^* \\ \mathcal{A}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} + \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} - \mathcal{H} \preccurlyeq -\delta \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix}. \tag{6.35}$$

Then, there exists an ISS storage functional according to the Definition 6.8 and the behavior $\bar{\mathscr{P}}_{\mathrm{p}}$ is input-to-sate stable.

Proof: Proof is given in the Appendix 6.F. ■

Degree-bounded test for input-to-state stability

One can formulate an LMI to test Lemma (6.4).

Lemma 6.4 *Suppose there exist $\tilde{\epsilon}_1, \psi, \delta > 0$, $d_1, d_2 > 0$, $P \in \mathbb{R}^{n_x \times n_x}$, $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$, and $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$ that satisfy*

$$\mathcal{P} := \mathcal{P} \begin{bmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \tilde{\epsilon}_1 I, R_1, R_2\}\} \end{bmatrix} \in \Xi_{d_1}^{d_2}, \tag{6.36}$$

$$\mathcal{P} \begin{bmatrix} H - \psi I, & \varnothing \\ \varnothing, & \{\varnothing\} \end{bmatrix} \in \Xi_{d_1}^{d_2}, \tag{6.37}$$

$$-\begin{bmatrix} \mathcal{B}_1^* \\ \mathcal{B}_2^* \\ \mathcal{A}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} - \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} - \delta \begin{bmatrix} \mathcal{T}_{Bw}^* \\ \mathcal{T}_{Bu}^* \\ \mathcal{T}^* \end{bmatrix} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} + \mathcal{P} \begin{bmatrix} \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}, & 0 \\ 0, & \{0\} \end{bmatrix} \in \Xi_{d_1}^{d_2} \tag{6.38}$$

where $\mathcal{T}, \mathcal{A}, \mathcal{B}_1, \mathcal{B}_2, \mathcal{T}_{Bw}, \mathcal{T}_{Bu}$ are defined in Theorem 6.3. Then, the behavior of $\mathscr{P}_{\mathrm{p}}$ is input-to-state stable for all $t \in [0, \infty)$, and any $w \in L_\infty^{n_w}[0,t)$, and $u \in L_\infty^{n_u}[0,t)$.

Verifying Input-to-state stability: academic example

Example 6.3 *Consider a one dimensional string of length L attached at one end and controlled via damping at the other end. Based on [22, 29], the following hyperbolic PDE model is given in terms of the wave displacement $u(s,t)$ over the domain $[0, L]$*

$$\frac{\partial^2 u(s,t)}{\partial t^2} = \frac{\partial^2 u(s,t)}{\partial s^2} + B_1(s)w(t). \tag{6.39}$$

The boundary conditions are

$$u(0,t) = 0, \qquad \frac{\partial u(s,t)}{\partial s}\,|_{s=L} = -\frac{\partial u(s,t)}{\partial t}\,|_{s=L}\,. \tag{6.40}$$

From the boundary conditions, we choose the states to be $\mathbf{x}_1(s,t) := col\Big(\frac{\partial u(s,t)}{\partial s}, \frac{\partial u(s,t)}{\partial t}\Big)$. *Based on the PIE framework, we obtain*

$$\dot{\mathbf{x}}_1(s,t) = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{:=A_1} \frac{\partial}{\partial s}[\mathbf{x}_1(s,t)] + B_1(s)w(t).$$

Here, $A_0 = 0, A_2$ *is void. The boundary conditions are*

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}}_{:=B} \underbrace{\begin{bmatrix} \mathbf{x}_1(0,t) \\ \mathbf{x}_1(L,t) \end{bmatrix}}_{\mathbf{x}_b(t)} = 0.$$

PIETOOLS proves the ISS of this system which has also been verified analytically in [48].

6.6 Synthesis of $\mathcal{H}_\infty$ Optimal State Estimator

In practice, assets exhibiting thermo-fluidic processes are usually equipped with a finite number of sensors. Therefore, it is a key problem to estimate non-observed or non-measured quantities of these processes on the basis of (noise-corrupted) measurements from the sensors. This section addresses the problem of synthesizing an estimator for systems whose behavior is described by $\bar{\mathscr{P}}_{\mathrm{p}}$ in (6.7). The estimator causally maps sensor information to estimates of the non-observed output (in many applications the non-observed outputs coincide with the state of the system) and achieves an optimal performance such that the effect of disturbances to the estimation error is bounded in $\mathcal{H}_\infty$ sense.

The conventional approach to synthesize a state estimator would be to first find an approximated finite dimensional model using the method developed in Chapter 4. The synthesis of an $\mathcal{H}_\infty$ optimal estimator is then carried out on the basis of the finite dimensional model. Specifically, if the finite dimensional model is represented as $\dot{x} = Ax + Bw, \quad y = Cx + Dw, \quad z = Ex$, with, x as the finite dimensional states, y as the measurements, z as the regulated outputs and w as

the exogenous disturbances. Then the Luenberger estimator has the form $\dot{\hat{x}} = A\hat{x} + L(\hat{y} - y), \quad \hat{y} = C\hat{x}, \quad \hat{z} = E\hat{x}$. Here, $L = P^{-1}Z$ with $P \succ 0$ and Z satisfy the following Linear Matrix Inequalities (LMIs) for a performance gain γ

$$\begin{bmatrix} PA + ZC + (PA + ZC)^\top & -PB - ZD & E^\top \\ -(PB + ZD)^\top & -\gamma I & 0 \\ E & 0 & -\gamma I \end{bmatrix} \preccurlyeq 0. \tag{6.41}$$

In this case, $||\hat{z} - z|| \leq \gamma ||w||$ is achieved.

In the remainder of this section, the objective is to create an analogous LPI formulation for synthesizing $\mathcal{H}_\infty$ optimal state estimator for the behavior $\bar{\mathscr{P}}_{\mathrm{p}}$ in (6.7) using its equivalent PIE representation.

6.6.1 Designing Luenberger-Type State Estimator

State estimation for PDE-ODE coupled systems

A Luenberger state estimator for PDE-ODE coupled systems (6.7) is graphically depicted in Figure 1.

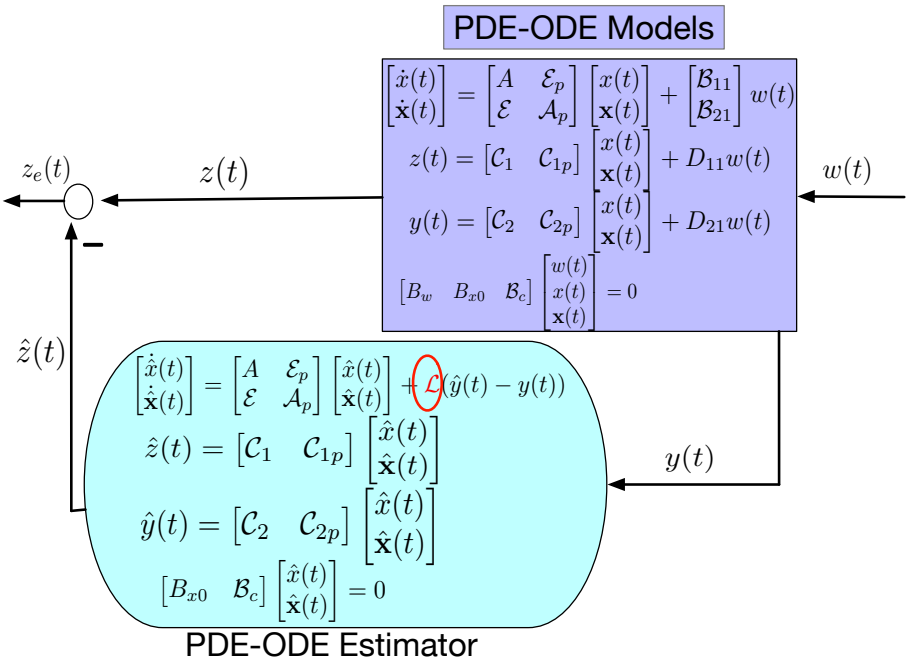


Figure 1: Synthesis of state estimator amounts to finding the operator $\mathcal{L}$ and a minimum value of γ such that $|| z_e ||_{L_2[0,\infty)} \leq \gamma || w ||_{L_2[0,\infty)}$

Finding the estimator gain $\mathcal{L} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_x} \times L_2^{n_0}[a,b] \times H_1^{n_1}[a,b] \times H_2^{n_2}[a,b]$ and the minimum value of γ such that $|| \ z_e \ ||_{L_2[0,\infty)} \leq \gamma \ || \ w \ ||_{L_2[0,\infty)}$ can be done by determining the worst case disturbance amplification at the estimation error $z_e(t)$ in the following PDE-ODE error system:

$$\mathscr{P}_{\mathrm{p}}^{\mathrm{err}} := \left\{ \begin{array}{c} \mathrm{col}\,(w, \mathbf{e}, z_e) \mid \\ \mathbf{e}(0) = 0, \\ \forall t \in [0,\infty), \begin{bmatrix} w(t) \\ z_e(t) \\ \mathbf{e}(t) \end{bmatrix} \in \mathbb{R}^{n_w+n_z+n_x} \times L_2^{n_0}[a,b] \times H_1^{n_1}[a,b] \times H_2^{n_2}[a,b], \\ z_e(t) = \begin{bmatrix} C_1 & \mathcal{C}_{1p} \end{bmatrix} \mathbf{e}(t) - D_{11} w(t) \\ \dot{\mathbf{e}}(t) = \left(\begin{bmatrix} A & \mathcal{E}_p \\ \mathcal{E} & \mathcal{A}_p \end{bmatrix} - \mathcal{L} \begin{bmatrix} C_2 & \mathcal{C}_{2p} \end{bmatrix} \right) \mathbf{e}(t) - \left(\begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} + \mathcal{L} D_{21} \right) w(t) \\ \begin{bmatrix} -B_w & B_{x0} & \mathcal{B}_c \end{bmatrix} \begin{bmatrix} w(t) \\ \mathbf{e}(t) \end{bmatrix} = 0. \end{array} \right\}. \tag{6.42}$$

Here, $\mathbf{e}(t) := \begin{bmatrix} \hat{x}(t) \\ \hat{\mathbf{x}}(t) \end{bmatrix} - \begin{bmatrix} x(t) \\ \mathbf{x}(t) \end{bmatrix}$ and $z_e(t) = \hat{z}(t) - z(t)$.

State estimation for PIEs

A Luenberger state estimator for PIEs (6.19) is graphically depicted in Figure 2.

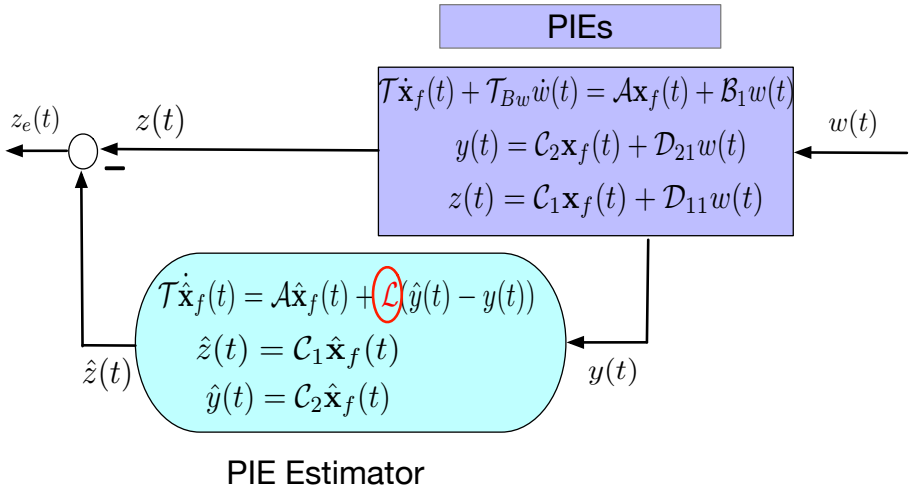


Figure 2: Synthesis of state estimator amounts to finding the operator $\mathcal{L}$ and a minimum value of γ such that $|| \ z_e \ ||_{L_2[0,\infty)} \leq \gamma \ || \ w \ ||_{L_2[0,\infty)}$

Finding an estimator gain $\mathcal{L} : \mathbb{R} \to \mathbb{R}^m \times L_2^{n_p}[a,b]$ and the minimum value of γ

such that $\| z_e \|_{L_2[0,\infty)} \leq \gamma \| w \|_{L_2[0,\infty)}$ can be done by determining the worst case disturbance amplification at the estimation error $z_e(t)$ in the following PIE error system:

$$\mathscr{P}_{\mathrm{f}}^{\mathrm{err}} := \left\{ \begin{array}{c} \mathrm{col}\,(w, \mathbf{e}_f, z_e) \mid \\ \begin{bmatrix} -\mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(0) \\ \mathbf{e}_f(0) \end{bmatrix} = 0, \\ \forall t \in [0,\infty), \begin{bmatrix} w(t) \\ z_e(t) \\ \mathbf{e}_f(t) \end{bmatrix} \in \mathbb{R}^{n_w+n_z+n_x} \times L_2^{n_p}[a,b], \\ \\ \left[\begin{array}{cc|c} I & 0 & 0 \\ 0 & -\mathcal{T}_{Bw} & \mathcal{T} \end{array}\right] \begin{bmatrix} z_e(t) \\ \dot{w}(t) \\ \hline \dot{\mathbf{e}}_f(t) \end{bmatrix} = \left[\begin{array}{c|c} -\mathcal{D}_{11} & \mathcal{C}_1 \\ -\mathcal{B}_1 - \mathcal{L}\mathcal{D}_{21} & \mathcal{A} + \mathcal{L}\mathcal{C}_2 \end{array}\right] \begin{bmatrix} w(t) \\ \hline \mathbf{e}_f(t) \end{bmatrix} \end{array} \right\}, \tag{6.43}$$

Here, $\mathbf{e}(t) := \hat{\mathbf{x}}_f(t) - \mathbf{x}_f(t)$.

Analogous to the treatment of Theorem 6.3, the following Lemma shows the equivalence between of (6.42) and (6.43).

Lemma 6.5 *In* (6.8)*, let* $B \in \mathbb{R}^{(n_1+2n_2)\times 2(n_1+2n_2)}$ *and*

$$B \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix} \text{ is invertible.}$$

Suppose, $\mathrm{col}\,(w, \mathbf{e}, z_e) \in \mathscr{P}_{\mathrm{p}}^{\mathrm{err}}$ *as defined in* (6.42)*. Then* $\mathrm{col}\,(w, \mathbf{e}_f, z_e) \in \mathscr{P}_{\mathrm{f}}^{\mathrm{err}}$ *as defined in* (6.43) *if the PI operators* $\mathcal{D}_{11}, \mathcal{C}_1, \mathcal{D}_{21}, \mathcal{C}_2, \mathcal{B}_1, \mathcal{A}, \mathcal{T}, \mathcal{T}_{Bw}$ *are defined according to the Theorem 6.3.*

Proof: The proof follows identical to the proof of Theorem 6.3. The only exception is that for $\mathrm{col}\,(w, \mathbf{e}, z_e) \in \mathscr{P}_{\mathrm{p}}^{\mathrm{err}}$ and $\mathrm{col}\,(w, \mathbf{e}_f, z_e) \in \mathscr{P}_{\mathrm{f}}^{\mathrm{err}}$, the following identity holds

$$\mathbf{e}(t) = \begin{bmatrix} -\mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ \mathbf{e}_f(t) \end{bmatrix}.$$

■

LPI for $\mathcal{H}_\infty$ Optimal State Estimator Using PIE Representation

Theorem 6.7 *Suppose there exist scalars $\epsilon, \gamma > 0$, constant real-valued matrices $P \in \mathbb{R}^{n_x \times n_x}, Z_1 \in \mathbb{R}^{n_x \times n_y}$ and matrix-valued polynomials $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}, Z_2 : [a,b] \to \mathbb{R}^{n_p \times n_y}$, $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$, such that*

- $\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right] \succcurlyeq \epsilon I$,
- *and*

$$\begin{bmatrix} \mathcal{T}_{Bw}^*(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21}) + (\cdot)^* & 0 & (\cdot)^* \\ 0 & 0 & 0 \\ -(\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T}_{Bw} & 0 & 0 \end{bmatrix} + \begin{bmatrix} -\gamma I & -\mathcal{D}_{11}^\top & -(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})^*\mathcal{T} \\ (\cdot)^* & -\gamma I & \mathcal{C}_1 \\ (\cdot)^* & (\cdot)^* & (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T} + (\cdot)^* \end{bmatrix} \preccurlyeq 0, \tag{6.44}$$

with $\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right]$ and $\mathcal{Z} := \mathcal{P}\left[\begin{smallmatrix} Z_1, & \varnothing \\ Z_2, & \{\varnothing\} \end{smallmatrix}\right]$.

Then $\mathcal{P}^{-1}$ exists and is a bounded linear operator. Moreover, with $\mathcal{L} := \mathcal{P}^{-1}\mathcal{Z}$, any $w \in L_2^{n_w}[0,\infty)$ and $z_e \in L_2^{n_z}[0,\infty)$, the following inequality is satisfied for all $\mathrm{col}\,(w, \mathbf{e}_f, z_e) \in \mathscr{P}_{\mathrm{f}}^{\mathrm{err}}$ as well as for all $\mathrm{col}\,(w, \mathbf{e}, z_e) \in \mathscr{P}_{\mathrm{p}}^{\mathrm{err}}$

$$|| \, z_e \, ||_{L_2[0,\infty)} \leq \gamma \, || \, w \, ||_{L_2[0,\infty)} \, .$$

Proof: The proof can be found in the Appendix 6.G. ■

Degree-bounded synthesis of $\mathcal{H}_\infty$ optimal estimator

One can formulate the following LMI tests according to the definition 6.5.

Lemma 6.6 *Suppose there exist $\epsilon > 0$, $d_1, d_2 > 0$, $P \in \mathbb{R}^{n_x \times n_x}, Z_1 \in \mathbb{R}^{n_x \times n_y}$, $Z_2 : [a,b] \to \mathbb{R}^{n_p \times n_y}$, $Q : [a,b] \to \mathbb{R}^{n_x \times n_p}$, $R_0 : [a,b] \to \mathbb{R}^{n_p \times n_p}$, and $R_1, R_2 : [a,b] \times [a,b] \to \mathbb{R}^{n_p \times n_p}$ which satisfy*

$$\hat{\gamma} = \arg\min \gamma, \tag{6.45}$$

such that

$$\begin{aligned} &\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \epsilon I, R_1, R_2\}\} \end{smallmatrix}\right] \in \Xi_{d_1}^{d_2}, \\ &- \begin{bmatrix} \mathcal{T}_{Bw}^*(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21}) + (\cdot)^* & 0 & (\cdot)^* \\ 0 & 0 & 0 \\ -(\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T}_{Bw} & 0 & 0 \end{bmatrix} - \begin{bmatrix} -\hat{\gamma} I & -\mathcal{D}_{11}^\top & -(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})^*\mathcal{T} \\ (\cdot)^* & -\hat{\gamma} I & \mathcal{C}_1 \\ (\cdot)^* & (\cdot)^* & (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T} + (\cdot)^* \end{bmatrix} \\ &\qquad \in \Xi_{d_1}^{d_2}, \end{aligned} \tag{6.46}$$

with $\mathcal{Z} := \mathcal{P}\left[\begin{smallmatrix} Z_1, & \varnothing \\ Z_2, & \{\varnothing\} \end{smallmatrix}\right]$, $\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \epsilon I, R_1, R_2\}\} \end{smallmatrix}\right]$ *and* $\mathcal{T}, \mathcal{A}, \mathcal{B}_1, \mathcal{T}_{Bw}, \mathcal{C}_1, \mathcal{C}_2, \mathcal{D}_{11}, \mathcal{D}_{21}$ *are defined in Theorem 6.3.*

Then, for $\mathcal{L} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \epsilon I, R_1, R_2\}\} \end{smallmatrix}\right]^{-1} \mathcal{P}\left[\begin{smallmatrix} Z_1, & \varnothing \\ Z_2, & \{\varnothing\} \end{smallmatrix}\right]$, $\hat{\gamma}$ *is the minimum value, such that for any* $w \in L_2^{n_w}[0,\infty)$ *and* $z_e \in L_2^{n_z}[0,\infty)$, *the behavior of* $\mathscr{P}_{\mathrm{f}}^{\mathrm{err}}$ *and* $\mathscr{P}_{\mathrm{p}}^{\mathrm{err}}$ *satisfy the following inequality*

$$\|z_e\|_{L_2[0,\infty)} \leq \hat{\gamma} \|w\|_{L_2[0,\infty)}.$$

6.6.2 Implementation of PIE State Estimator

PIE state estimator is implemented using the following steps:

Step 1: Inversion of PI operator and computing estimator gain

There is no known closed form solution for the inverse in case of a general PI operator. Even in the case of sign definite PI operators, the inverse has not been found analytically. However, if $R_1 = R_2$ in $\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right]$, its inverse is also a PI operator and admits a closed form expression. In such cases, the formula of inversion in given in the following Lemma.

Lemma 6.7 *(c.f. [77]) Suppose that* $Q(s) = HZ(s)$ *and* $R_1(s,\theta) = Z(s)^\top \Gamma Z(\theta)$ *and* $\mathcal{P} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R, R\}\} \end{smallmatrix}\right]$ *is a coercive and self-adjoint operator where* $\mathcal{P} : X \to X$. *If* $\mathcal{P}^{-1} := \mathcal{P}\left[\begin{smallmatrix} \hat{P}, & \hat{Q} \\ \hat{Q}^\top, & \left\{\{\hat{R}_0, \hat{R}, \hat{R}\}\right\} \end{smallmatrix}\right]$ *with*

$$\hat{P} = \left(I - \hat{H} K H^\top\right) P^{-1}, \qquad \hat{Q}(s) = \hat{H} Z(s) R_0(s)^{-1}$$
$$\hat{R}_0(s) = R_0(s)^{-1}, \quad \hat{R}(s,\theta) = \hat{R}_0^\top(s) Z(s)^\top \hat{\Gamma} Z(\theta) \hat{R}_0(\theta),$$

and

$$K = \int_{-1}^{0} Z(s) R_0(s)^{-1} Z(s)^\top ds$$
$$\hat{H} = P^{-1} H \left(K H^\top P^{-1} H - I - K\Gamma\right)^{-1}$$
$$\hat{\Gamma} = -(\hat{H}^\top H + \Gamma)(I + K\Gamma)^{-1},$$

then $\mathcal{P}^{-1}$ *is self-adjoint with* $\mathcal{P}^{-1} : X \to X$, *and* $\mathcal{P}^{-1}\mathcal{P}\mathbf{x} = \mathcal{P}\mathcal{P}^{-1}\mathbf{x} = \mathbf{x}$ *for all* $\mathbf{x} \in X$.

As a result, the estimator gain is computed such a way that,

$$\mathcal{L} := \mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0 - \epsilon I, R, R\}\} \end{smallmatrix}\right]^{-1} \mathcal{P}\left[\begin{smallmatrix} Z_1, & \varnothing \\ Z_2, & \{\varnothing\} \end{smallmatrix}\right].$$

Notice that, $\mathcal{L}$ is an another PI operator and admits the structure $\mathcal{L} := \mathcal{P}\begin{bmatrix}\Phi_1, & \varnothing \\ \Phi_2, & \{\varnothing\}\end{bmatrix}$ for some $\Phi_1 \in \mathbb{R}^{n_x \times n_y}$ and $\Phi_2 : [a,b] \to \mathbb{R}^{n_p \times n_y}$.

Step 2: Simulate the estimator by discretization of PIEs

To implement, the PIE estimator is connected with the original PDE-ODE model as depicted

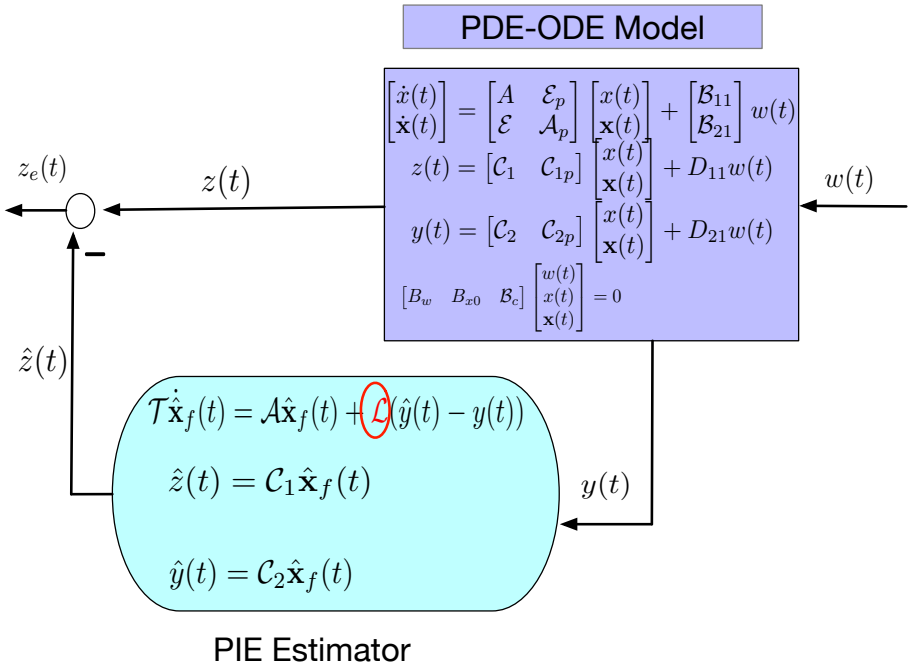


Figure 3: Interconnection of PIE estimator with PDE-ODE model

Here, the PDE-ODE coupled model's measurement $y(t)$ can be obtained from any standard numerical simulator that is able to simulate the specific PDE-ODE coupled model. Considering the $y(t)$ as an external signal, to simulate a estimator, one has to implement a finite dimensional version of the estimator:

$$\begin{aligned}\bar{T}\dot{\hat{\mathbf{x}}}_{f\mathrm{d}} &= \left(\bar{A}_{\mathrm{d}} + \bar{\Phi}_{\mathrm{d}}\bar{C}_{2\mathrm{d}}\right)\hat{\mathbf{x}}_{f\mathrm{d}} - \bar{\Phi}_{\mathrm{d}}\, y_{\mathrm{d}},\\ \hat{z}_{\mathrm{d}} &= \bar{C}_{1\mathrm{d}}\hat{\mathbf{x}}_{f\mathrm{d}}.\end{aligned} \tag{6.47}$$

Here, the functions $\mathbf{x}_f(s)$ is evaluated at the sequence of points $(s_i) \in S_d \subset [a,b] \times [a,b]$, $i \in \{1, \cdots, N+1\}$, $j \in \{1, \cdots, N+1\}$. y_{d} is obtained from PDE-ODE simulator (if they are governed by diffusion-transport-reaction equations, one can also use the approximation method in Chapter 4).

All the matrices in (6.47) are obtained by discretization of respective PIE operators at the sequence of points $(s_i, \theta_j) \in S_d \times S_d \subset [a,b] \times [a,b]$, $i \in \{1, \cdots, N+1\}$, $j \in \{1, \cdots, N+1\}$ using trapezoidal Reimann sum as follows:

$$\int_a^b F(s)\mathrm{d}s \approx \frac{\Delta_s}{2} \sum_{i=1}^{N} (F(s_i) + F(s_{i+1})). \tag{6.48}$$

Note that solving (6.47) may require stiff solvers such as `ode15s`.

Academic Illustration: Estimator Design

Example 6.4 *Consider the same mode in the Example 6.3. The regulated output is chosen as* $z(t) = \int_a^b \frac{\partial u(s,t)}{\partial t} ds$. *The measured output is chosen as* $y(t) = \frac{\partial u(s,t)}{\partial t} \mid_{s=L}$. *In other words,*

$$z(t) = \int_a^b \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{:=C_{a1}} x_2(s,t) ds + D_1 w(t),$$

$$y(t) = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}}_{:=C_{20}} \mathbf{x}_b(t), \mathbf{x}_b(t) := \begin{bmatrix} \mathbf{x}_1(0,t) \\ \mathbf{x}_1(L,t) \end{bmatrix}$$

Using PIETOOLS, we obtain the minimum γ*-value as* $\underline{0.2064}$. *After implementing the discretized estimator on* 50 *grid points, Figure 4 shows the disturbance suppression on the evolution of the discretized* $z_e(t) := \hat{z}(t) - z(t)$ *over time.*

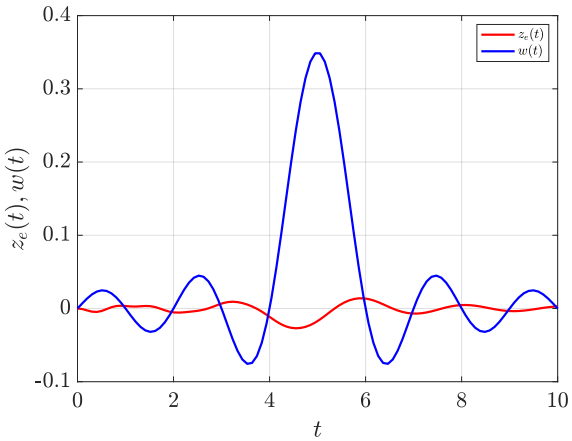


Figure 4: Time evolution of $z_e(t)$ and $w(t)$ where $w(t)$ is the `sinc` function.

6.7 Closing Remarks

This chapter shows that the general class of thermo-fluidic processes admit an equivalent representation in terms of Partial Integral Equations (PIEs) and the systems parameters are Partial Integral (PI) operators. Using the properties of PI operators, the chapter develops Linear PI Inequalities (LPIs) for stability analysis, computing input-ouput properties and designing $\mathcal{H}_\infty$ optimal state estimator for the thermo-fluidic processes. As testing LPIs can be formulated as LMIs, the problems related to analysis and control of thermo-fluidic processes are solved using LMIs that computationally implementable.

Appendices

6.A Proof of Theorem 6.1

Observe the following:

- $\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right]$ is self-adjoint.
- By the definitions in (6.4),

  $$\mathcal{P}\left[\begin{smallmatrix} P, & Q \\ Q^\top, & \{\{R_0, R_1, R_2\}\} \end{smallmatrix}\right] = \mathcal{P}\left[\begin{smallmatrix} I, & 0 \\ 0, & \{\{z_0, z_1, z_2\}\} \end{smallmatrix}\right]^* \mathcal{P}\left[\begin{smallmatrix} P_{11}, & P_{12} \\ P_{21}, & \{\{P_{22}, 0, 0\}\} \end{smallmatrix}\right] \mathcal{P}\left[\begin{smallmatrix} I, & 0 \\ 0, & \{\{z_0, z_1, z_2\}\} \end{smallmatrix}\right]$$

  with

  $$z_0(s) = \begin{bmatrix} \sqrt{g(s)} Z_1(s) \\ 0 \\ 0 \end{bmatrix}, z_1(s,\eta) = \begin{bmatrix} 0 \\ \sqrt{g(s)} Z_2(s,\eta) \\ 0 \end{bmatrix}, z_2(s,\eta) = \begin{bmatrix} 0 \\ 0 \\ \sqrt{g(s)} Z_2(s,\eta) \end{bmatrix}$$ and

$$P_{11} = T_{11}, \qquad P_{12} = \begin{bmatrix} T_{12} & T_{13} & T_{14} \end{bmatrix},$$
$$P_{21} = P_{12}^\top, \qquad P_{22} = \begin{bmatrix} T_{22} & T_{23} & T_{24} \\ T_{32} & T_{33} & T_{34} \\ T_{42} & T_{43} & T_{44} \end{bmatrix}.$$

Since, $\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} := \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} \succcurlyeq 0$, for all $z_1 \in \mathbb{R}^m, z_2 \in \mathbb{R}^{d_1+2d_2}$

$$\left\langle \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\rangle_{\mathbb{R}} \geq 0$$
$$\iff \left\langle \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\rangle_{\mathbb{R}} + \left\langle z_2, P_{22} z_2 \right\rangle_{\mathbb{R}} \geq 0$$

$$\implies \left\langle \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\rangle_{\mathbb{R}} \geq 0 \quad \text{and} \left\langle z_2, P_{22} z_2 \right\rangle_{\mathbb{R}} \geq 0$$

$$\implies \left\langle \begin{bmatrix} y_1 \\ \int_a^b \mathbf{y}_2(s)\mathrm{d}s \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \int_a^b \mathbf{y}_2(s)\mathrm{d}s \end{bmatrix} \right\rangle_{\mathbb{R}} \geq 0$$

$$\text{and} \qquad \left\langle \mathbf{y}_2, P_{22}\mathbf{y}_2 \right\rangle_{L_2} \geq 0 \forall y_1 \in \mathbb{R}^m, \mathbf{y}_2 \in L_2^{d_1+2d_2}[a,b]$$

$$\implies \left\langle \begin{bmatrix} y_1 \\ \int_a^b \mathbf{y}_2(s)\mathrm{d}s \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \int_a^b \mathbf{y}_2(s)\mathrm{d}s \end{bmatrix} \right\rangle_{\mathbb{R}} + \left\langle \mathbf{y}_2, P_{22}\mathbf{y}_2 \right\rangle_{L_2} \geq 0$$

$$\implies y_1^\top P_{11} y_1 + y_1^\top P_{12} \int_a^b \mathbf{y}_2(s)\mathrm{d}s + \int_a^b \mathbf{y}_2^\top(s) P_{21} y_1 ds + \int_a^b \mathbf{y}_2^\top(s) P_{22}\mathbf{y}_2(s) ds \geq 0$$

$$\iff \left\langle \begin{bmatrix} y_1 \\ \mathbf{y}_2 \end{bmatrix}, \mathcal{P}_{\begin{bmatrix} P_{11}, & P_{12} \\ P_{21}, & \{\{P_{22},0,0\}\} \end{bmatrix}} \begin{bmatrix} y_1 \\ \mathbf{y}_2 \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2} \geq 0$$

$$\text{Define } \begin{bmatrix} y_1 \\ \mathbf{y}_2 \end{bmatrix} := \mathcal{Z}x, \text{ with } \mathcal{Z} := \mathcal{P}_{\begin{bmatrix} I, & 0 \\ 0, & \{\{z_0,z_1,z_2\}\} \end{bmatrix}}, \quad \forall x \in \mathbb{R}^m \times L_2^n[a,b]$$

$$\implies \left\langle \mathcal{Z}x, \mathcal{P}_{\begin{bmatrix} P_{11}, & P_{12} \\ P_{21}, & \{\{P_{22},0,0\}\} \end{bmatrix}} \mathcal{Z}x \right\rangle_{\mathbb{R}\times L_2} \geq 0$$

$$\iff \left\langle x, \mathcal{Z}^* \mathcal{P}_{\begin{bmatrix} P_{11}, & P_{12} \\ P_{21}, & \{\{P_{22},0,0\}\} \end{bmatrix}} \mathcal{Z}x \right\rangle_{\mathbb{R}\times L_2} \geq 0$$

$$\iff \left\langle x, \mathcal{P}_{\begin{bmatrix} P, & Q \\ Q^\top, & \{\{R_0,R_1,R_2\}\} \end{bmatrix}} x \right\rangle_{\mathbb{R}\times L_2} \geq 0 \qquad \forall x \in \mathbb{R}^m \times L_2^n[a,b]$$

6.B Proof of Theorem 6.3

For convenience, the following notations are introduced to perform specific operations on the function $\mathbf{x} := \mathrm{col}\,(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$.

- $\mathbf{x}_f(s) := \begin{bmatrix} I & & & 0 \\ & I & & \\ 0 & & \partial_s & \\ & & & \partial_s^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}(s).$
- $\Delta_c \mathbf{x} := \mathbf{x}(c)$ for any function $\mathbf{x}(s), s \in [a,b]$ when evaluated at a fixed point $c \in [a,b]$.

- $\Lambda_1 \mathbf{x} = \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & \Delta_b & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_b \\ 0 & 0 & \Delta_a \partial_s \\ 0 & 0 & \Delta_b \partial_s \end{bmatrix} \mathbf{x}$

Now, using fundamental theorem of calculus and the definition of PI operator, one obtains

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_0(s) \\ \mathbf{x}_1(s) \\ \mathbf{x}_2(s) \end{bmatrix} &= \begin{bmatrix} \mathbf{x}_0(s) \\ \mathbf{x}_1(a) + \int_a^s \partial_s \mathbf{x}_1(\theta) d\theta \\ \mathbf{x}_2(a) + (s-a)\partial_s \mathbf{x}_2(a) + \int_a^s (s-a)\partial_s^2 \mathbf{x}_2(\theta) d\theta \end{bmatrix} \\ &= \mathcal{P} \begin{bmatrix} \overset{\varnothing,}{\varnothing,} & \overset{\varnothing}{\left\{ \left\{ \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}, 0 \right\} \right\}} \end{bmatrix} \mathbf{x}_f(s) \\ &\quad + \mathcal{P} \left[\begin{bmatrix} 0 & 0 & \overset{\varnothing,}{0} \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}, \overset{\varnothing}{\{\varnothing\}} \right] \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s). \end{aligned}$$

Hence,

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_0(s) \\ \mathbf{x}_1(s) \\ \mathbf{x}_2(s) \end{bmatrix} =& \mathcal{P} \left[\begin{bmatrix} 0 & 0 & \overset{\varnothing,}{0} \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}, \left\{ \{ \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \overset{\varnothing}{0} \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}, 0 \} \right\} \right] \\ & \begin{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s) \\ \mathbf{x}_f(s) \end{bmatrix} \end{aligned}$$

Now

$$\begin{aligned} &\Lambda_1 \mathcal{P} \left[\begin{bmatrix} 0 & 0 & \overset{\varnothing,}{0} \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}, \left\{ \{ \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \overset{\varnothing}{0} \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}, 0 \} \right\} \right] \\ &= \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & \Delta_b & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_b \\ 0 & 0 & \Delta_a \partial_s \\ 0 & 0 & \Delta_b \partial_s \end{bmatrix} \mathcal{P} \left[\begin{bmatrix} 0 & 0 & \overset{\varnothing,}{0} \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}, \left\{ \{ \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \overset{\varnothing}{0} \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}, 0 \} \right\} \right] \\ &= \mathcal{P} \left[\begin{bmatrix} \overbrace{\begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix}}^{T} \\ \varnothing, \end{bmatrix}, \begin{bmatrix} \overbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (b-s) \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}}^{Q(s)} \\ \{\varnothing\} \end{bmatrix} \right] = \mathcal{P} \begin{bmatrix} T, Q(s) \\ \varnothing, \{\varnothing\} \end{bmatrix} \end{aligned}$$

where the following identities are used:

$$\partial_s \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ g(s), & \{\{0, R_1(s,\theta), 0\}\} \end{bmatrix} = \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ \partial_s g(s), & \{\{R_1(s,s), \partial_s R_1(s,\theta), 0\}\} \end{bmatrix}$$

and

$$\Delta_b \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ Q_2(s), & \{\{0, R_1(s,\theta), 0\}\} \end{bmatrix} = \mathcal{P}\begin{bmatrix} Q_2(b), & R_1(b,s) \\ \varnothing, & \{\varnothing\} \end{bmatrix}$$

and

$$\Delta_a \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ Q_2(s), & \{\{0, R_1(s,\theta), 0\}\} \end{bmatrix} = \mathcal{P}\begin{bmatrix} Q_2(a), & 0 \\ \varnothing, & \{\varnothing\} \end{bmatrix}$$

Note that, the boundary conditions (6.8) can be written as

$$B\Lambda_1 \mathbf{x} = \mathcal{P}\begin{bmatrix} \begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, B_{xx} \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f \end{bmatrix}$$

As a result, one obtains

$$B\Lambda_1 \mathbf{x}(t,s) = \mathcal{P}\begin{bmatrix} BT, & BQ(s) \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s) \\ \mathbf{x}_f(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} \begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, B_{xx} \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

In other words,

$$\mathcal{P}\begin{bmatrix} BT, & \varnothing \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} \begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, B_{xx} - BQ(s) \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

or

$$\begin{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} (BT)^{-1} \begin{bmatrix} B_w & B_u & B_{xo} \end{bmatrix}, (BT)^{-1}(B_{xx}(s) - BQ(s)) \\ \varnothing, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

In conclusion,

$$\begin{aligned} \mathbf{x}(s) =& \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ \varnothing, & \left\{ \{ \underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{R}_{0f}=L_0}, \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (s-\theta) \end{bmatrix}}_{L_1(s,\theta)}, 0 \} \right\} \end{bmatrix} \mathbf{x}_f(s) \\ &+ \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & (s-a)I \end{bmatrix}}_{K(s)}, & \{\varnothing\} \end{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a \partial_s \end{bmatrix} \mathbf{x}(s) \end{aligned}$$

In other words

$$\begin{aligned}\mathbf{x}(s) =& \mathcal{P}\begin{bmatrix}\varnothing, & \overset{\varnothing}{} \\ \varnothing, & \{\{L_0, L_1(s,\theta), 0\}\}\end{bmatrix}\mathbf{x}_f(s) \\ &+ \mathcal{P}\begin{bmatrix}\varnothing, & \varnothing \\ K(s), & \{\varnothing\}\end{bmatrix}\mathcal{P}\begin{bmatrix}(BT)^{-1}\begin{bmatrix}B_w & B_u & B_{xo}\end{bmatrix}, & (BT)^{-1}(B_{xx}(s) - BQ(s)) \\ \varnothing, & \{\varnothing\}\end{bmatrix}\begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s)\end{bmatrix}\end{aligned}$$

Finally,

$$\mathbf{x}(s) = \mathcal{P}\begin{bmatrix}\varnothing, & \overset{\varnothing}{} \\ \mathbf{Q}_{2f}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\}\end{bmatrix}\begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s)\end{bmatrix},$$

or

$$\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix} = \mathcal{P}\begin{bmatrix}\begin{bmatrix}0 & \overset{0}{} & I\end{bmatrix}, & \overset{0}{} \\ \begin{bmatrix}\mathbf{Q}_{2f}\end{bmatrix}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\}\end{bmatrix}, \tag{6.49}$$

where

$$\mathbf{Q}_{2f}(s) = K(s)(BT)^{-1}\begin{bmatrix}B_w & B_u & B_{xo}\end{bmatrix}, \mathbf{R}_{2f}(s,\theta) = K(s)(BT)^{-1}(B_{xx}(\theta) - BQ(\theta)),$$

$$\mathbf{R}_{1f}(s,\theta) = L_1(s,\theta) + K(s)(BT)^{-1}(B_{xx}(\theta) - BQ(\theta)), \mathbf{R}_{0f}(s) = \begin{bmatrix}I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix},$$

$$L_1(s,\theta) = \begin{bmatrix}0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (s-\theta)\end{bmatrix}, K(s) = \begin{bmatrix}0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & (s-a)I\end{bmatrix},$$

$$T = \begin{bmatrix}I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & (b-a)I \\ 0 & 0 & I \\ 0 & 0 & I\end{bmatrix}, Q(s) = \begin{bmatrix}0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & I & (b-s)I \\ 0 & 0 & 0 \\ 0 & 0 & I\end{bmatrix}.$$

6.C Proof of Theorem 6.3

Notations and derivations related to the previous proof are recalled here.

Reformulation of PDE-ODE Coupled Systems

The original PDE-ODE coupled system is represented as

$$\begin{bmatrix} z \\ y \\ \dot{x} \\ \dot{\mathbf{x}}(s) \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & C_1 & \mathcal{C}_{1p} \\ D_{21} & D_{22} & C_2 & \mathcal{C}_{2p} \\ B_{11} & B_{12} & A & \mathcal{E}_p \\ B_{21}(s) & B_{22}(s) & E(s) & \mathcal{A}_p \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}(s) \end{bmatrix} \tag{6.50}$$

or

$$\begin{bmatrix} z \\ y \\ \dot{x} \\ \dot{\mathbf{x}}(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} \mathbf{P}, & \mathbf{Q}_1 \\ \mathbf{Q}_2, \{\{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2\}\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \Lambda_1\mathbf{x} \\ \Lambda_2\mathbf{x}(s) \end{bmatrix} \tag{6.51}$$

where

$$\Lambda_1\mathbf{x} = \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & \Delta_b & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_b \\ 0 & 0 & \Delta_a\partial_s \\ 0 & 0 & \Delta_b\partial_s \end{bmatrix} \mathbf{x}, \quad \Lambda_2\mathbf{x}(s) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & \partial_s & 0 \\ 0 & 0 & \partial_s \\ 0 & 0 & \partial_{ss} \end{bmatrix} \mathbf{x}(s)$$

The following definition are obtained from the operators' definitions in (6.50).

$$\mathbf{P} = \begin{bmatrix} D_{11} & D_{12} & C_1 & C_{10} \\ D_{21} & D_{22} & C_2 & C_{20} \\ B_{11} & B_{12} & A & E_0 \end{bmatrix}, \qquad \mathbf{Q}_1(s) = \begin{bmatrix} C_{a1}(s) & C_{b1}(s) & C_{c1}(s) \\ C_{a2}(s) & C_{b2}(s) & C_{c2}(s) \\ E_a(s) & E_b(s) & E_c(s) \end{bmatrix},$$

$$\mathbf{Q}_2(s) = \begin{bmatrix} B_{21}(s) & B_{22}(s) & E(s) & E_2(s) \end{bmatrix},$$

$$\mathbf{R}_0(s) = \begin{bmatrix} A_0(s) & A_1(s) & A_2(s) \end{bmatrix}, \quad \mathbf{R}_1(s,\theta) = A_{2l}(s,\theta), \quad \mathbf{R}_2(s,\theta) = A_{2u}(s,\theta).$$

From Theorem 6.15, $\begin{bmatrix} z \\ y \\ \dot{x} \\ \dot{\mathbf{x}}(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}, & 0 \\ \begin{bmatrix} 0 & 0 & \mathbf{Q}_{2f} \end{bmatrix}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\} \end{bmatrix} \begin{bmatrix} z \\ y \\ \dot{w} \\ \dot{u} \\ \dot{x} \\ \dot{\mathbf{x}}_f(s) \end{bmatrix}$

Determining $\Lambda_1\mathbf{x}$

Now recall

$$\begin{bmatrix} \begin{bmatrix} 0 & \Delta_a & 0 \\ 0 & 0 & \Delta_a \\ 0 & 0 & \Delta_a\partial_s \end{bmatrix} \mathbf{x}(s) \end{bmatrix} = \mathcal{P}\begin{bmatrix} (BT)^{-1}\begin{bmatrix} B_w & B_u & B_{xo} \\ \varnothing, & & \end{bmatrix}, (BT)^{-1}(\mathbf{B}_{xx}(s) - BQ(s)) \\ \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

and

$$\Lambda_1\mathcal{P}\left[\begin{bmatrix}0&0&\varnothing,\\I&0&0\\0&I&(s-a)I\end{bmatrix},\left\{\{\begin{bmatrix}I&0&0\\0&0&0\\0&0&0\end{bmatrix},\begin{bmatrix}0&0&\varnothing\\0&I&0\\0&0&(s-\theta)\end{bmatrix},0\}\right\}\right]=\mathcal{P}\begin{bmatrix}T,Q(s)\\\varnothing,\{\varnothing\}\end{bmatrix}$$

$$\begin{aligned}\Lambda_1\mathbf{x}(s)&=\Lambda_1\mathcal{P}\left[\begin{bmatrix}0&0&\varnothing,\\I&0&0\\0&I&(s-a)I\end{bmatrix},\left\{\{\begin{bmatrix}I&0&0\\0&0&0\\0&0&0\end{bmatrix},\begin{bmatrix}0&0&\varnothing\\0&I&0\\0&0&(s-\theta)\end{bmatrix},0\}\right\}\right]\\&\quad\begin{bmatrix}\begin{bmatrix}0&\Delta_a&0\\0&0&\Delta_a\\0&0&\Delta_a\partial_s\end{bmatrix}\mathbf{x}(s)\\\mathbf{x}_f(s)\end{bmatrix}\\&=\mathcal{P}\begin{bmatrix}T,Q(s)\\\varnothing,\{\varnothing\}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}0&\Delta_a&0\\0&0&\Delta_a\\0&0&\Delta_a\partial_s\end{bmatrix}\mathbf{x}(s)\\\mathbf{x}_f(s)\end{bmatrix}\\&=\mathcal{P}\begin{bmatrix}T,Q(s)\\\varnothing,\{\varnothing\}\end{bmatrix}\mathcal{P}\begin{bmatrix}(BT)^{-1}\begin{bmatrix}B_w&B_u&B_{xo}\\0,&&\end{bmatrix},(BT)^{-1}(\mathbf{B}_{xx}(s)-BQ(s))\\&\{\{I,0,0\}\}\end{bmatrix}\begin{bmatrix}w\\u\\x\\\mathbf{x}_f(s)\end{bmatrix}\\&=\mathcal{P}\begin{bmatrix}T,Q(s)\\\varnothing,\{\varnothing\}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}0&\Delta_a&0\\0&0&\Delta_a\\0&0&\Delta_a\partial_s\end{bmatrix}\mathbf{x}(s)\\\mathbf{x}_f(s)\end{bmatrix}\\&=\mathcal{P}\begin{bmatrix}T(BT)^{-1}\begin{bmatrix}B_w&B_u&B_{xo}\\\varnothing,&&\end{bmatrix},Q(s)+T(BT)^{-1}(\mathbf{B}_{xx}(s)-BQ(s))\\\{\varnothing\}\end{bmatrix}\begin{bmatrix}w\\u\\x\\\mathbf{x}_f(s)\end{bmatrix}\end{aligned}$$

Determining $\Lambda_2\mathbf{x}$

Firstly,

$$\Lambda_2\mathbf{x}=\Lambda_2\mathcal{P}\begin{bmatrix}\varnothing,&\varnothing\\\mathbf{Q}_{2f},\{\{\mathbf{R}_{0f},\mathbf{R}_{1f},\mathbf{R}_{2f}\}\}\end{bmatrix}\begin{bmatrix}w\\u\\x\\\mathbf{x}_f\end{bmatrix}$$

and

$$\Lambda_2\mathcal{P}\begin{bmatrix}\varnothing,&\varnothing\\\mathbf{Q}_{2f},\{\{\mathbf{R}_{0f},\mathbf{R}_{1f},\mathbf{R}_{2f}\}\}\end{bmatrix}=\begin{bmatrix}I&0&0\\0&I&0\\0&0&I\\0&\partial_s&0\\0&0&\partial_s\\0&0&\partial_{ss}\end{bmatrix}\mathcal{P}\begin{bmatrix}\varnothing,&\varnothing\\\mathbf{Q}_{2f},\{\{\mathbf{R}_{0f},\mathbf{R}_{1f},\mathbf{R}_{2f}\}\}\end{bmatrix}$$

$$
\begin{aligned}
&= \left(\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{I}_1} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{I}_2} \partial_s + \partial_s \partial_s \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}}_{\mathbf{I}_3} \right) \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{R}_{0f}, \overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{R}_{2f} \}\} \right] \\
&= \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_1\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_1\mathbf{R}_{0f}, \mathbf{I}_1\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_1\mathbf{R}_{2f} \}\} \right] + \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_2\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_2\mathbf{R}_{0f}, \mathbf{I}_2\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_2\mathbf{R}_{2f} \}\} \right] \\
&\qquad + \partial_s \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_3\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_3\mathbf{R}_{0f}, \mathbf{I}_3\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_3\mathbf{R}_{2f} \}\} \right] \\
&= \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_1\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_1 L_0, \mathbf{I}_1\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_1\mathbf{R}_{2f} \}\} \right] + \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_2\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ 0, \mathbf{I}_2\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_2\mathbf{R}_{2f} \}\} \right] \\
&\qquad + \partial_s \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{I}_3\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ 0, \mathbf{I}_3\overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{I}_3\mathbf{R}_{2f} \}\} \right] \\
&= \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_1 L_0 + \mathbf{I}_2 L_1(s,s), (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\overset{\varnothing}{\mathbf{R}}_{1f}, (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\mathbf{R}_{2f} \}\} \right] \\
&\qquad + \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \partial_s\mathbf{I}_3\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ 0, \partial_s\mathbf{I}_3\overset{\varnothing}{\mathbf{R}}_{1f}, \partial_s\mathbf{I}_3\mathbf{R}_{2f} \}\} \right]
\end{aligned}
$$

Where

$$
\mathbf{R}_{1f}(s,s) - \mathbf{R}_{2f}(s,s) = L_1(s,s) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

and hence

$$
\mathbf{I}_3(\mathbf{R}_{1f}(s,s) - \mathbf{R}_{2f}(s,s)) = \mathbf{I}_3 L_1(s,s) = 0
$$

Finally,

$$
\begin{aligned}
&\Lambda_2 \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{R}_{0f}, \overset{\varnothing}{\mathbf{R}}_{1f}, \mathbf{R}_{2f} \}\} \right] \\
&= \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ \mathbf{I}_1 L_0 + \mathbf{I}_2 L_1(s,s), (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\overset{\varnothing}{\mathbf{R}}_{1f}, (\mathbf{I}_1 + \mathbf{I}_2\partial_s)\,\mathbf{R}_{2f} \}\} \right] \\
&\qquad + \partial_s \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \partial_s\mathbf{I}_3\mathbf{Q}_{2f}, \end{smallmatrix} \{\{ 0, \partial_s\mathbf{I}_3\overset{\varnothing}{\mathbf{R}}_{1f}, \partial_s\mathbf{I}_3\mathbf{R}_{2f} \}\} \right] \\
&= \mathcal{P}\left[\begin{smallmatrix} \varnothing, \\ \hat{\mathbf{Q}}_{2f}, \end{smallmatrix} \left\{ \{ \hat{\mathbf{R}}_{0f}, \overset{\varnothing}{\hat{\mathbf{R}}}_{1f}, \hat{\mathbf{R}}_{2f} \} \right\} \right].
\end{aligned}
$$

Here,

$$
\begin{aligned}
\hat{\mathbf{Q}}_{2f} &= \left(\mathbf{I}_1 + \mathbf{I}_2\partial_s + \mathbf{I}_3\partial_s^2\right)\mathbf{Q}_{2f} \\
\hat{\mathbf{R}}_{0f} &= \mathbf{I}_1 \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \mathbf{I}_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} + \mathbf{I}_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \\
\hat{\mathbf{R}}_{1f} &= \left(\mathbf{I}_1 + \mathbf{I}_2\partial_s + \mathbf{I}_3\partial_s^2\right)\mathbf{R}_{1f} \\
\hat{\mathbf{R}}_{2f} &= \left(\mathbf{I}_1 + \mathbf{I}_2\partial_s + \mathbf{I}_3\partial_s^2\right)\mathbf{R}_{2f}
\end{aligned}
$$

where, the following identity is used

$$\partial_s \mathbf{I}_3 \mathbf{R}_{1f}(s,\theta) - \partial_s \mathbf{I}_3 \mathbf{R}_{2f}(s,\theta) = \partial_s \mathbf{I}_3 L_1(s,\theta) = \mathbf{I}_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}$$

Collecting All the relevant terms

So far,

$$\Lambda_1 \mathbf{x}(s) = \mathcal{P}\begin{bmatrix} T(BT)^{-1}\begin{bmatrix} B_w & B_u & B_{xo} \\ \varnothing, & & \end{bmatrix}, Q(s) + T(BT)^{-1}(\mathbf{B}_{xx}(s) - BQ(s)) \\ \{\varnothing\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

and

$$\Lambda_2 \mathbf{x} = \Lambda_2 \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ \mathbf{Q}_{2f}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f \end{bmatrix} = \mathcal{P}\begin{bmatrix} \varnothing, & \varnothing \\ \hat{\mathbf{Q}}_{2f}, & \left\{\{\hat{\mathbf{R}}_{0f}, \hat{\mathbf{R}}_{1f}, \hat{\mathbf{R}}_{2f}\}\right\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f \end{bmatrix}$$

Hence,

$$\begin{bmatrix} w \\ u \\ x \\ \Lambda_1 \mathbf{x} \\ \Lambda_2 \mathbf{x}(s) \end{bmatrix}$$

$$= \mathcal{P}\left[\begin{bmatrix} \overbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ T(BT)^{-1}B_w & T(BT)^{-1}B_u & T(BT)^{-1}B_{xo} \end{bmatrix}}^{\hat{\mathbf{P}}_f}, \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ Q(s) + T(BT)^{-1}(\mathbf{B}_{xx}(s) - BQ(s)) \end{bmatrix}}^{\hat{\mathbf{Q}}_{1f}(s)} \\ \hat{\mathbf{Q}}_{2f}, \qquad \left\{\{\hat{\mathbf{R}}_{0f}, \hat{\mathbf{R}}_{1f}, \hat{\mathbf{R}}_{2f}\}\right\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

$$= \mathcal{P}\begin{bmatrix} \hat{\mathbf{P}}_f, & \hat{\mathbf{Q}}_{1f}(s) \\ \hat{\mathbf{Q}}_{2f}, & \left\{\{\hat{\mathbf{R}}_{0f}, \hat{\mathbf{R}}_{1f}, \hat{\mathbf{R}}_{2f}\}\right\} \end{bmatrix} \begin{bmatrix} w \\ u \\ x \\ \mathbf{x}_f(s) \end{bmatrix}$$

Combining with

$$\begin{bmatrix} z(t) \\ y(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} = \mathcal{P}\left[\begin{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}, & 0 \\ \begin{bmatrix} 0 & 0 & \mathbf{Q}_{2f} \end{bmatrix}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\} \end{bmatrix} \begin{bmatrix} z(t) \\ y(t) \\ \dot{w}(t) \\ \dot{u}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}_f(t) \end{bmatrix}$$

we have

$$\mathcal{P}\begin{bmatrix}\begin{bmatrix}I & 0 & 0 & 0 & 0\\ 0 & I & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & I\end{bmatrix}, & 0\\ \begin{bmatrix}0 & 0 & \mathbf{Q}_{2f}\end{bmatrix}, & \{\{\mathbf{R}_{0f}, \mathbf{R}_{1f}, \mathbf{R}_{2f}\}\}\end{bmatrix}\begin{bmatrix}z(t)\\ y(t)\\ \dot{w}(t)\\ \dot{u}(t)\\ \dot{x}(t)\\ \dot{\mathbf{x}}_f(t)\end{bmatrix}$$
$$= \mathcal{P}\begin{bmatrix}\mathbf{P}, & \mathbf{Q}_1\\ \mathbf{Q}_2, & \{\mathbf{R}_i\}\end{bmatrix}\mathcal{P}\begin{bmatrix}\hat{\mathbf{P}}_f, & \hat{\mathbf{Q}}_{1f}(s)\\ \hat{\mathbf{Q}}_{2f}, & \left\{\{\hat{\mathbf{R}}_{0f}, \hat{\mathbf{R}}_{1f}, \hat{\mathbf{R}}_{2f}\}\right\}\end{bmatrix}\begin{bmatrix}w(t)\\ u(t)\\ x(t)\\ \mathbf{x}_f(t)\end{bmatrix}$$

6.D Proof of Theorem 6.4

Consider the following candidate storage functional:

$$V(x(t), \mathbf{x}_f(t)) = \left\langle \begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}, \mathcal{T}^*\mathcal{P}\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2} \geq \epsilon \left\|\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}$$

Taking time derivative of V along the trajectory $(x(t), \mathbf{x}_f(t))$,

$$\begin{aligned}\dot{V}(x(t), \mathbf{x}_f(t)) &= \left\langle \mathcal{T}\begin{bmatrix}\dot{x}(t)\\ \dot{\mathbf{x}}_f(t)\end{bmatrix}, \mathcal{P}\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2} + \left\langle \mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}, \mathcal{P}\mathcal{T}\begin{bmatrix}\dot{x}(t)\\ \dot{\mathbf{x}}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2}\\ &= \left\langle \mathcal{A}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}, \mathcal{P}\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2} + \left\langle \mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}, \mathcal{P}\mathcal{A}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2}\\ &= \left\langle \begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}, \mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\rangle_{\mathbb{R}\times L_2}\\ &\leq -\delta \left\|\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}\end{aligned}$$

Using Grönwall-Bellman inequality, for any initial condition $\operatorname{col}(x(0), \mathbf{x}_f(0))$, one obtains

$$\left\|\mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2} \leq \frac{\zeta}{\epsilon}\left\|\mathcal{T}\begin{bmatrix}x(0)\\ \mathbf{x}_f(0)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2} e^{\frac{-\delta}{\zeta}t},$$

where, $\zeta := \|\mathcal{P}\|_{\mathcal{L}(\mathbb{R}\times L_2)}$.

Now, using Theorem 6.3, one can show that for any $(x, \mathbf{x}, w, u, y, z) \in \{\mathscr{P}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ and $\operatorname{col}(0, 0, x(0), \mathbf{x}(0)) \in X_{\mathrm{dom}}$, there exists $(x, \mathbf{x}_f, w, u, y, z) \in \{\mathscr{P}_{\mathrm{f}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ where $\begin{bmatrix}x(t)\\ \mathbf{x}(t)\end{bmatrix} = \mathcal{T}\begin{bmatrix}x(t)\\ \mathbf{x}_f(t)\end{bmatrix}$.

Hence,

$$\left\|\begin{bmatrix}x(t)\\ \mathbf{x}(t)\end{bmatrix}\right\|_{\mathbb{R}\times L_2} \leq \sqrt{\frac{\zeta}{\epsilon}}\left\|\begin{bmatrix}x(0)\\ \mathbf{x}(0)\end{bmatrix}\right\|_{\mathbb{R}\times L_2} e^{\frac{-\delta}{2\zeta}t}.$$

Therefore, based on the definition 6.5, $\{\mathscr{P}_{\mathrm{p}} \mid w(t) \equiv 0, u(t) \equiv 0\}$ is exponentially stable with $M = \sqrt{\frac{\zeta}{\epsilon}}, \alpha = \frac{\delta}{2\zeta}$.

6.E Proof of Theorem 6.5

Consider the following candidate ISS-storage functional:

$$V(w(t),u(t),x(t),\mathbf{x}_f(t)) = \left\langle \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix}^* \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2} \geq 0.$$

Taking time derivative of V along the trajectory $(w(t),u(t),x(t),\mathbf{x}_f(t))$,

$$\begin{aligned} \dot{V}(w(t),u(t),x(t),\mathbf{x}_f(t)) &= \left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} \dot{w}(t) \\ \dot{u}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2} \\ &\quad + \left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} \dot{w}(t) \\ \dot{u}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2} \\ &= \left\langle \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2} \\ &\quad + \left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}. \end{aligned}$$

The LPI suggests that

$$\dot{V}(w(t),u(t),x(t),\mathbf{x}_f(t)) \leq \rho \left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}}^2 - \left\| \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} \right\|_{\mathbb{R}}^2.$$

Integrating both sides of the above inequality with respect to time t from 0 to ∞, one obtains

$$\begin{aligned} &V(w(\infty),u(\infty),x(\infty),\mathbf{x}_f(\infty)) - V(w(0),u(0),x(0),\mathbf{x}_f(0)) \\ &\qquad \leq \rho \int_0^\infty \left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}}^2 \mathrm{d}t - \int_0^\infty \left\| \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} \right\|_{\mathbb{R}}^2 \mathrm{d}t. \end{aligned}$$

Now, using Theorem 6.3, one can show that for any $(x, \mathbf{x}, w, u, y, z) \in \mathscr{P}_{\mathrm{p}}$ and $X_{\mathrm{dom}} \ni 0 = \mathrm{col}\,(w(0), u(0), x(0), \mathbf{x}(0))$, there exists $(x, \mathbf{x}_f, w, u, y, z) \in \mathscr{P}_{\mathrm{f}}$ where

$$\begin{bmatrix} x(0) \\ \mathbf{x}(0) \end{bmatrix} = \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(0) \\ u(0) \\ x(0) \\ \mathbf{x}_f(0) \end{bmatrix}.$$

Hence, $-V(w(0), u(0), x(0), \mathbf{x}_f(0)) = 0$. Moreover, $V(w(\infty), u(\infty), x(\infty), \mathbf{x}_f(\infty)) \geq 0$.

Hence, one obtains:

$$\begin{aligned} &\sqrt{\int_0^\infty \left\| \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} \right\|_{\mathbb{R}}^2 \mathrm{d}t} \leq \sqrt{\rho} \sqrt{\int_0^\infty \left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}}^2 \mathrm{d}t} \\ \implies &\left\| \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} \right\|_{L_2[0,\infty)} \leq \sqrt{\rho} \left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{L_2[0,\infty)}. \end{aligned}$$

6.F Proof Of Theorem 6.6

Consider the following candidate ISS-storage functional:

$$V(w(t), u(t), x(t), \mathbf{x}_f(t)) = \left\langle \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix}^* \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R} \times L_2}.$$

Moreover, for some $\tilde{\epsilon}_1, \tilde{\epsilon}_2 > 0$, let

$$\tilde{\epsilon}_1 \left\| \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bw} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\|_{\mathbb{R} \times L_2}^2 \leq V(w(t), u(t), x(t), \mathbf{x}_f(t)) \leq \tilde{\epsilon}_2 \left\| \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\|_{\mathbb{R} \times L_2}^2$$

Taking time derivative of V along the trajectory $(w(t), u(t), x(t), \mathbf{x}_f(t))$,

$$\begin{aligned} \dot{V}(w(t), u(t), x(t), \mathbf{x}_f(t)) = &\left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} \dot{w}(t) \\ \dot{u}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R} \times L_2} \\ &+ \left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} \dot{w}(t) \\ \dot{u}(t) \\ \dot{x}(t) \\ \dot{\mathbf{x}}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R} \times L_2} \end{aligned}$$

$$= \left\langle \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}$$
$$+ \left\langle \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix}, \mathcal{P} \begin{bmatrix} \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{A} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}.$$

The LPI suggests that

$$\dot{V}(w(t),u(t),x(t),\mathbf{x}_f(t)) \leq -\tilde{\delta} \left\| \begin{bmatrix} \mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \\ x(t) \\ \mathbf{x}_f(t) \end{bmatrix} \right\|^2_{\mathbb{R}\times L_2} + H\left(\left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}} \right)$$
$$\leq -\frac{\tilde{\delta}}{\tilde{\epsilon}_2} V(w(t),u(t),x(t),\mathbf{x}_f(t)) + H\left(\left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}} \right),$$

Where, $H\left(\left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}} \right) := \begin{bmatrix} w(t) \\ u(t) \end{bmatrix}^\top H \begin{bmatrix} w(t) \\ u(t) \end{bmatrix}$. Pre-multiplying both side of the last inequality with $e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}$, one obtains

$$\frac{d}{dt}\left(e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} V(w(t),u(t),x(t),\mathbf{x}_f(t)) \right) \leq e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} H\left(\left\| \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \right\|_{\mathbb{R}} \right). \tag{6.52}$$

Integrating both side of the last inequality for the interval $[0,t)$, one obtains

$$e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} V(w(t),u(t),x(t),\mathbf{x}_f(t)) - V(w(0),u(0),x(0),\mathbf{x}_f(0)) \leq \int_0^t e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}\tau} \left(H\Big(\left\| \begin{bmatrix} w(\tau) \\ u(\tau) \end{bmatrix} \right\|_{\mathbb{R}} \Big) \right) \mathrm{d}\tau$$
$$\leq \left(\int_0^t e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}\tau} \mathrm{d}\tau \right) \sup_{\tau\in[0,t)} \left(H\Big(\left\| \begin{bmatrix} w(\tau) \\ u(\tau) \end{bmatrix} \right\|_{\mathbb{R}} \Big) \right).$$

In the last inequality, Hölder's inequality is used. Now,

$$e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} V(w(t),u(t),x(t),\mathbf{x}_f(t)) - V(w(0),u(0),x(0),\mathbf{x}_f(0))$$
$$\leq \frac{e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} - 1}{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}} \sup_{\tau\in[0,t)} \left(H\Big(\left\| \begin{bmatrix} w(\tau) \\ u(\tau) \end{bmatrix} \right\|_{\mathbb{R}} \Big) \right).$$
$$\leq \frac{e^{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}}{\frac{\tilde{\delta}}{\tilde{\epsilon}_2}} \sup_{\tau\in[0,t)} \left(H\Big(\left\| \begin{bmatrix} w(\tau) \\ u(\tau) \end{bmatrix} \right\|_{\mathbb{R}} \Big) \right).$$

Multiplying all the terms by $e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}$ and after rearrangements of the terms, one obtain

$$V(w(t),u(t),x(t),\mathbf{x}_f(t)) \leq e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t} V(w(0),u(0),x(0),\mathbf{x}_f(0)) + \frac{\tilde{\epsilon}_2}{\tilde{\delta}} \sup_{\tau\in[0,t)} \left(H\Big(\left\| \begin{bmatrix} w(\tau) \\ u(\tau) \end{bmatrix} \right\|_{\mathbb{R}} \Big) \right).$$

Using the bounds on $V(w(t),u(t),x(t),\mathbf{x}_f(t))$

$$\tilde{\epsilon}_1\left\|\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}_f(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}$$
$$\leq e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}\tilde{\epsilon}_2\left\|\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(0)\\u(0)\\x(0)\\\mathbf{x}_f(0)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}+\frac{\tilde{\epsilon}_2}{\tilde{\delta}}\sup_{\tau\in[0,t)}\left(H\Big(\left\|\begin{bmatrix}w(\tau)\\u(\tau)\end{bmatrix}\right\|_{\mathbb{R}}\Big)\right).$$

Since $H\succ 0$,

$$\tilde{\epsilon}_1\left\|\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}_f(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}$$
$$\leq e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}\tilde{\epsilon}_2\left\|\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(0)\\u(0)\\x(0)\\\mathbf{x}_f(0)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}+\frac{\tilde{\epsilon}_2}{\tilde{\delta}}H\left(\sup_{\tau\in[0,t)}\left\|\begin{bmatrix}w(\tau)\\u(\tau)\end{bmatrix}\right\|_{R}\right)$$
$$\leq e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}\tilde{\epsilon}_2\left\|\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(0)\\u(0)\\x(0)\\\mathbf{x}_f(0)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}+\frac{\tilde{\epsilon}_2}{\tilde{\delta}}H\left(\left\|\begin{bmatrix}w(\tau)\\u(\tau)\end{bmatrix}\right\|_{L_\infty[0,t)}\right)$$

Now, using Theorem 6.3, one has

$$\begin{bmatrix}x(t)\\\mathbf{x}(t)\end{bmatrix}=\begin{bmatrix}\mathcal{T}_{Bw} & \mathcal{T}_{Bu} & \mathcal{T}\end{bmatrix}\begin{bmatrix}w(t)\\u(t)\\x(t)\\\mathbf{x}_f(t)\end{bmatrix}.$$

Hence, we obtain

$$\left\|\begin{bmatrix}x(t)\\\mathbf{x}(t)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}\leq\frac{\tilde{\epsilon}_2}{\tilde{\epsilon}_1}e^{-\frac{\tilde{\delta}}{\tilde{\epsilon}_2}t}\left\|\begin{bmatrix}x(0)\\\mathbf{x}(0)\end{bmatrix}\right\|^2_{\mathbb{R}\times L_2}+\frac{\tilde{\epsilon}_2}{\tilde{\delta}\tilde{\epsilon}_1}H\left(\left\|\begin{bmatrix}w(\tau)\\u(\tau)\end{bmatrix}\right\|_{L_\infty[0,t)}\right)\tag{6.53}$$

This can be modified to

$$\left\|\begin{bmatrix}x(t)\\\mathbf{x}(t)\end{bmatrix}\right\|_{\mathbb{R}\times L_2}\leq\underbrace{\sqrt{\frac{2\tilde{\epsilon}_2}{\tilde{\epsilon}_1}}e^{-\frac{\tilde{\delta}}{2\tilde{\epsilon}_2}t}\left\|\begin{bmatrix}x(0)\\\mathbf{x}(0)\end{bmatrix}\right\|_{\mathbb{R}\times L_2}}_{\in KL}+\underbrace{\sqrt{\frac{2\tilde{\epsilon}_2}{\tilde{\delta}\tilde{\epsilon}_1}H\left(\left\|\begin{bmatrix}w(\tau)\\u(\tau)\end{bmatrix}\right\|_{L_\infty[0,t)}\right)}}_{\in K}\tag{6.54}$$

This completes the proof.

6.G Proof of Theorem 6.7

Define the storage functional

$$V(\mathbf{e}_f(t), w(t)) = \langle \mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t), \mathcal{P}(\mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t))\rangle_{\mathbb{R}\times L_2}.$$

Since, $V(\mathbf{e}_f(t), w(t)) \geq \epsilon \|\mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t)\|^2_{\mathbb{R}\times L_2}$ holds for some $\epsilon > 0$, $\mathcal{P}$ is bounded, self-adjoint, coercive. Hence, $\mathcal{P}^{-1}$ exists and is a bounded linear operator. Then, using $\mathcal{L} = \mathcal{P}^{-1}\mathcal{Z}$

$$\begin{aligned}
&\dot{V}(\mathbf{e}_f(t), w(t)) - \gamma\|w(t)\|^2 - \gamma\|\upsilon_e(t)\|^2 + 2\langle \upsilon_e(t), z_e(t)\rangle_{\mathbb{R}\times L_2} \\
&= \langle \mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t), (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)\mathbf{e}_f(t)\rangle_{\mathbb{R}\times L_2} + \langle (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)\mathbf{e}_f(t), \mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t)\rangle_{\mathbb{R}\times L_2} \\
&- \langle \mathcal{T}\mathbf{e}_f(t) - \mathcal{T}_{Bw}w(t), (\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})w(t)\rangle_{\mathbb{R}\times L_2} - \langle \mathcal{T}\mathbf{e}(t) - \mathcal{T}_{Bw}w(t), (\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})w(t)\rangle_{\mathbb{R}\times L_2} \\
&- \langle (\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})w(t), \mathcal{T}\mathbf{e}(t) - \mathcal{T}_{Bw}w(t)\rangle_{\mathbb{R}\times L_2} - \gamma\|\omega(t)\|^2 - \gamma\|\upsilon_e(t)\|^2 + \langle \upsilon_e(t), \mathcal{C}_1\mathbf{e}_f(t)\rangle_{\mathbb{R}} \\
&+ \langle \mathcal{C}_1\mathbf{e}_f(t), \upsilon_e(t)\rangle_{\mathbb{R}} - \langle \upsilon_e(t), \mathcal{D}_{11}w(t)\rangle_{\mathbb{R}} - \langle \mathcal{D}_{11}w(t), \upsilon_e(t)\rangle_{\mathbb{R}} \\
&= \left\langle \begin{bmatrix} w(t) \\ \upsilon_e(t) \\ \mathbf{e}_f(t) \end{bmatrix}, \begin{pmatrix} \bar{\bar{\mathcal{P}}} \end{pmatrix} \begin{bmatrix} w(t) \\ \upsilon_e(t) \\ \mathbf{e}_f(t) \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2},
\end{aligned}$$

with

$$\bar{\bar{\mathcal{P}}} = \begin{bmatrix} \mathcal{T}^*_{Bw}(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21}) + (\cdot)^* & 0 & (\cdot)^* \\ 0 & 0 & 0 \\ -(\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T}_{Bw} & 0 & 0 \end{bmatrix} + \begin{bmatrix} -\gamma I & -\mathcal{D}_{11}^\top & -(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})^*\mathcal{T} \\ (\cdot)^* & -\gamma I & \mathcal{C}_1 \\ (\cdot)^* & (\cdot)^* & (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T} + (\cdot)^* \end{bmatrix}.$$

Now, set $\upsilon_e(t) = \frac{1}{\gamma}z_e(t)$. If (6.44) is satisfied, then

$$\dot{V}(\mathbf{e}_f(t), w(t)) - \gamma\|\omega(t)\|^2 + \frac{1}{\gamma}\|z_e(t)\|^2 \leq 0.$$

Integration of this inequality with respect to t yields

$$V(\mathbf{e}_f(t), w(t)) - (\mathbf{e}_f(0), w(0)) - \gamma\int_0^t \|w(\tau)\|^2 \,\mathrm{d}\tau + \frac{1}{\gamma}\int_0^t \|z_e(\tau)\|^2 \,\mathrm{d}\tau \leq 0.$$

$V(\mathbf{e}_f(0), w(0)) = 0$ and $V(\mathbf{e}_f(t), w(t)) \geq 0$ for any $t \geq 0$. Then as $t \to \infty$, one gets $|| \, z_e \, ||_{L_2[0,\infty)} \leq \gamma \, || \, w \, ||_{L_2[0,\infty)}$.

Part V

Applications to Industrial Benchmarks

7 CHAPTER

Industrial Application of Lumped Digital Twin: DoD Inkjet Printhead

This chapter introduces a closed-loop control strategy for maintaining consistency of liquid temperature in commercial Drop on Demand (DoD) inkjet printing. No additional sensors or additional actuators are installed in the printhead while achieving the consistency in liquid temperature. To this end, this chapter presents a novel in situ sensing-actuation policy at every individual liquid-nozzle, where the jetting mechanism has three distinct roles. It is used for jetting liquid droplet onto the print media based on the print-job. It is used as a soft sensor to estimate the real-time liquid temperature of the jetting nozzle. While not jetting liquid, it is used as a heating actuator to minimize the gradient of liquid temperature among nozzles. The soft sensing based in situ controller is implemented in an experimentally validated digital twin that models the thermo-fluidic processes of the printhead. The digital twin is scalable and flexible to incorporate an arbitrary number of liquid-nozzles, making the control strategy applicable for future designs of the printhead.

Outline

7.1 Introduction **170**
7.2 Building a Digital Twin of Inkjet Printhead **170**
7.3 Calibration of Soft Sensor **173**
7.4 Experimental Validation **177**
7.5 In Situ Controller: A Proof of Principle **180**
7.6 Closing Remarks **187**

7.1 Introduction

In the printing industry, model-based control of thermo-fluidic aspects is an unexplored field of research. Most of the literature is on physical modeling and experimental methods of drop formation and relevant fluid dynamics in inkjet printing (c.f. [100, 19, 102]). The applications of control techniques are largely restricted to designing the DoD voltage pulses for generating droplets and compensating problems related to nozzle acoustics (c.f. [51, 53, 49]). In general, a system-theoretic outlook to model and control thermo-fluidic aspects in DoD inkjet printing is still mostly missing.

Contribution of this Chapter

This chapter introduces, for the first time, a synthesis procedure to design a model-based feedback controller for the DoD inkjet printhead that uses no additional sensors and no additional actuators to compensate for the fluctuation of liquid temperature among nozzles. The applicability of the control system is demonstrated with the help of three novel contributing aspects:

1. The modeling framework presents a digital twin of the printhead, which is modular towards an arbitrary number of nozzles. Changes in the number of nozzles do not require redevelopment of the digital twin from the outset.
2. The piezoelectric elements that are already installed in every individual nozzle are used as collocated soft sensors. The self-sensing capability of a piezoelectric element is exploited to develop a data-driven algorithm and estimate the liquid temperature at every individual nozzle.
3. A concept of in situ actuation is introduced to control the fluctuation of liquid temperature. Here, a sensing-actuation policy is developed such that the controller uses the bit-map to anticipate the change in temperature at every nozzle and utilizes only the non-jetting nozzles to compensate for the temperature fluctuation.

7.2 Building a Digital Twin of Inkjet Printhead

Which Method Is Suitable for Inkjet Printhead?

In Chapter 3-5, three distinct methods are presented to develop a digital-twin that is able to describe the thermo-fluidic processes using graph-theoretic framework. The underlying behavior of the thermo-fluidic processes is reflected by these three distinct abstractions with varying complexity and varying accuracy. In order to utilize the digital twin and its functionalities for a commercial peripheral such as inkjet printhead, one needs to make a suitable trade-off between the model's accuracy to capture the thermo-fluidic phenomena

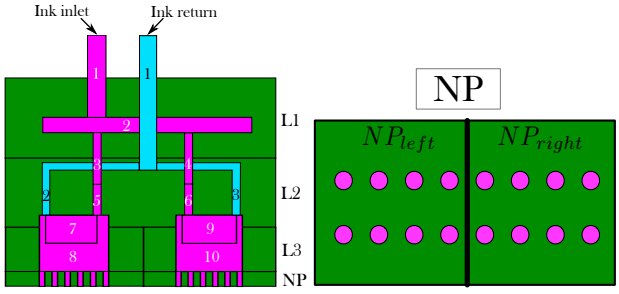
and model's complexity to enable tractable and cheaper implementation. Concerning inkjet printhead, there are two decisive factors to make a choice among three methods:

1. The digital twin may be equipped with an arbitrarily large number of liquid nozzles along with corresponding flow parameter for every individual one of them. This essentially suggests that number of parameter-varying liquid nodes in the digital twin can be significantly large. And it is the requirement to maintain consistent liquid temperature at every individual nozzle.
2. Based on the physical dimensions and constitutive materials, the Biot number of many components in the inkjet printhead justifies to provide sufficient accuracy if their thermo-fluidic processes are modeled based on lumping. However, it is important to point out that not all the component has sufficiently low Biot number. In those cases, further portioning a node may be required.

Based on these two arguments, the method of lumping-then-design is selected for addressing the problem concerning the inkjet printhead. At the same time, flexibility is provided in building the digital twin such at any stage of the design cycle, any individual node can be partitioned into a set of equi-partitioned nodes until required accuracy is met.

Setting Up the Digital Twin for Commercial Printhead

A Drop on Demand (DoD) inkjet printhead jets droplets of liquid based on the specific image that the user demands to print. A typical architecture of a commercial DoD printhead is shown in figure 7.1(a)-7.1(b) (for instance, see XAAR printhead [106]).



(a) Cross-sectional view of a printhead.

(b) Bottom-view of an NP with nozzles.

Figure 1: The printhead consists of solid components (green), and channels for liquid (inlet in purple and return in blue).

Here, the bottom stage of the printhead is called the nozzle-platform (NP), which is divided into two mutually insulated parts NP_{left} and NP_{right}. Each part of NP

consists of an array (divided into rows and columns) of nozzles. An example of the nozzles' placement with respect to an NP is shown in figure 7.1(b). The NP can be removed from the printhead and replaced with a new design. This feature allows for accommodating arbitrary numbers of nozzles without re-designing the entire printhead. In order to build the digital-twin based on lumping, the method presented in Section 3.2.2 is followed. For the printhead shown in figure 7.1(a), individual nodes (components) and the interconnection topology, is depicted in figure 2.

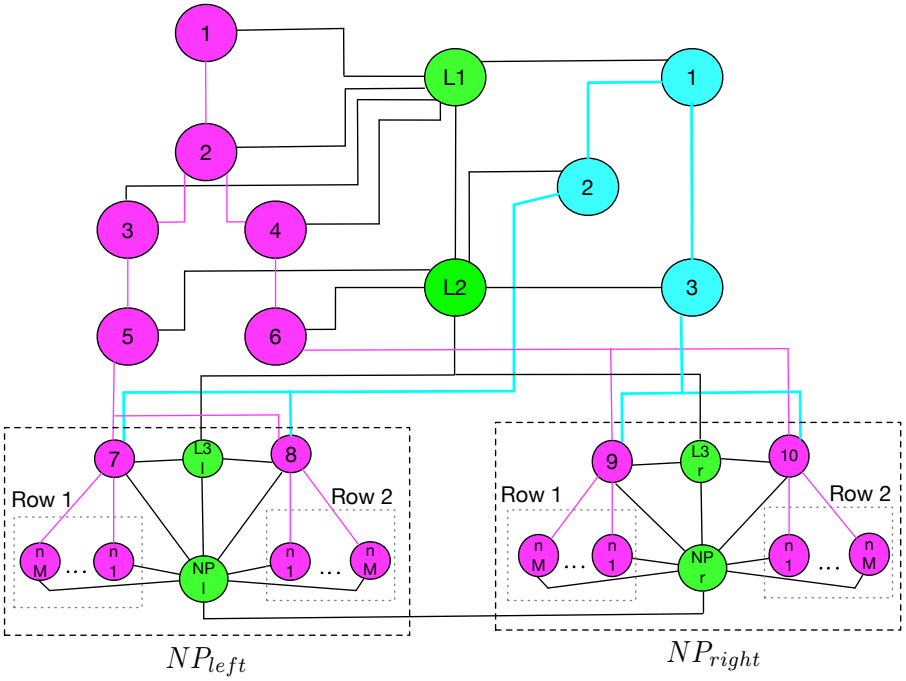


Figure 2: Topology of the printhead with 4M nozzles (denoted by $n1, \cdots, nM$). ● represents a solid component, ● represents a liquid component for inlet, and ● represents a liquid component for re-circulation. ▬ is an edge describing conductive exchange of thermal energy between two solid components or a solid and liquid component. ▬ is an edge describing convective thermal energy due to the inflow of liquid along the inlet channels. ▬ is an edge describing convective thermal energy due to the re-circulation of liquid along the return channels.

In particular,

1. The adjacency matrix A is defined based on the architecture of the printhead. Except for the adjustable NPs, the remaining architecture of the printhead is typically kept identical. Then, specifying the arrays of nozzles is sufficient for defining the entire topology of a newly designed printhead.

2. The job related signals always depend on the user-defined image that is

captured in a bit-map. Given a specific bit-map, the job signals are related according to $p_i(t) = \Theta_i(t)q_i(t)$, where $\Theta_i(t)$ stores the flow parameters in an individual node.

3. Based on the length of individual liquid channel, there is a provision of divide them into a set of equi-partitioned nodes to improve accuracy of the model. Depending on the specific configuration, large volumes of liquid and material properties, the liquid channel L1 and L2 do not always satisfy the Biot number criterion. In such cases, partitioning nodes is essential to achieve desired accuracy.

7.3 Calibration of Soft Sensor

In Chapter 3, Section 3.3.1, a novel method is proposed that allows to use every individual nozzle's piezoelectric element as a soft sensor of liquid temperature. The basic principle behind the soft-sensing is to measure the data samples of the acoustic signal and establish an empirical acoustic energy-temperature relation. In the inkjet printhead, the characteristic relation between acoustic energy and liquid temperature for an individual nozzle is established by performing the following three experiments consecutively.

Experiment 1

In the first experiment, three operating points are chosen for the liquid temperature. At a fixed operating point of temperature, the liquid droplets are jetted using the actuation mode of the piezoelectric element. After that, it is switched to sensing mode 20 times consecutively, and each time the acoustic sensing signal is measured. Every individual signal contains 100 samples of measured data. Using the measured data, the acoustic sensing signal is reconstructed using Algorithm IV.1, and its energy is computed. Based on this experiment, the energy-temperature curve is shown in figure 3. Despite the expected trend in change of acoustic energy with temperature, the variation of energy estimate is significantly large. Therefore, estimating liquid temperature using energy-temperature curve is still unreliable.

Experiment 2

The evaporation of liquid changes its viscosity and temperature [72], and therefore also its acoustic energy. In order to present the consequence of evaporation, an experiment, similar to Experiment 1, is performed where four operating points of temperature are chosen. Liquid droplets are jetted at these operating temperature by using 2000 consecutive jetting pulses. The last jetting pulse is, thereafter, followed by switching the piezoelectric element in sensing mode and measuring a sequence of acoustic sensing signals. Here, the aim is to investigate the evolution of acoustic energy over time. In figure 4, the estimated

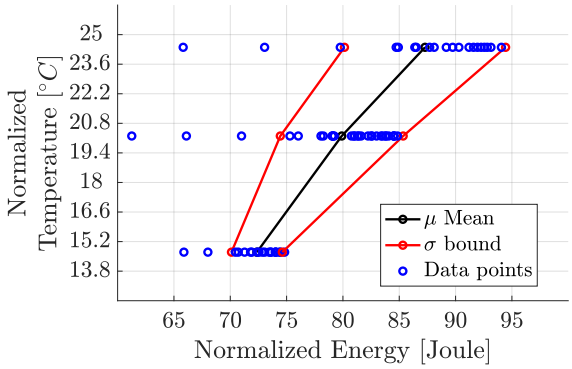


Figure 3: Mean and variance of the energy-temperature curve at three operating points of liquid temperatures in Experiment 1.

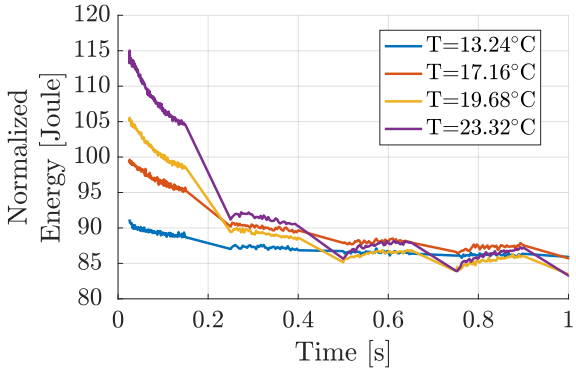


Figure 4: Effect of evaporation on the estimated acoustic energy over time in Experiment 2.

acoustic energy of the acoustic signal is shown over one second. Over time, the effect of evaporation results in an initial decay of energy and the decay rate is higher if the temperature of the liquid droplet is higher. This experiment implies that, due to evaporation, estimating the relation between temperature and acoustic energy depends on the timed sequence between jetting and sensing. In other words, the larger the difference between the time of jetting pulses and the time of measurements, the lower the acoustic energy. As a result, the estimation method based on this experiment has a poor signal-to-noise ratio.

Experiment 3

To circumvent the effect of evaporation from the energy-temperature relation, only those acoustic sensing signals should be considered that are measured immediately after the jetting pulse. To this end, another experiment is performed

where the liquid temperature is varied over four operating points. In contrast to the previous experiments where a series of acoustic signals are measured one after another, this time, jetting of the liquid droplet is directly followed by the sensing of the acoustic signal. This sequence of consecutive jetting and sensing is repeated 20 times. Using the measured data, the acoustic energy is estimated. figure 5 shows that the adapted approach of consecutive jetting-sensing significantly improves the variance in the estimate of energy-temperature relation.

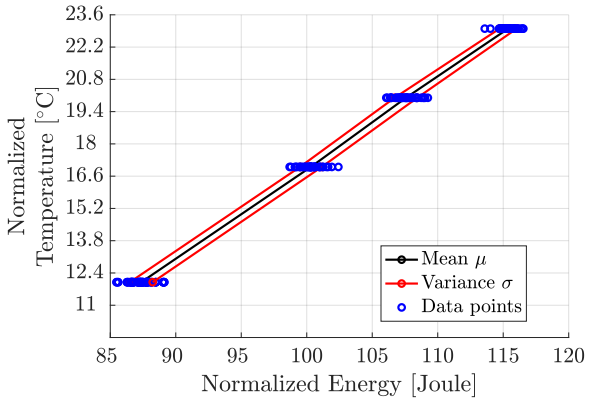


Figure 5: Relation between the liquid temperature of a nozzle and estimated acoustic energy in Experiment 3.

In figure 6-7, the designs of Experiment 1 and Experiment 3 are compared, respectively. The key difference between them is the sequence in which jetting actuation and acoustic sensing are performed. Based on the sequential jetting-sensing mechanism devised in Experiment 3, the energy-temperature characteristic curves are estimated and calibrated for every individual nozzle. In figure 8, energy-temperature curves are shown for four different nozzles in the printhead.

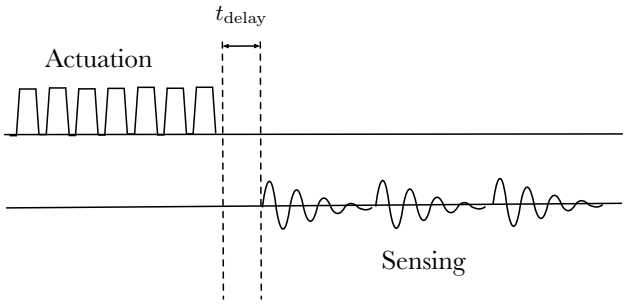


Figure 6: Actuation-sensing sequence in Experiment 1.

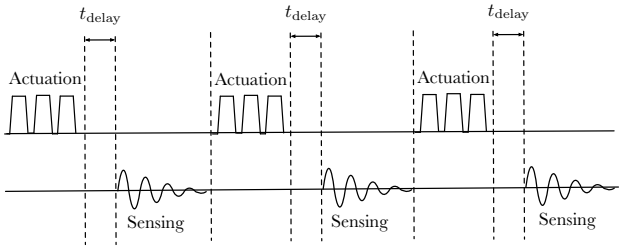


Figure 7: Actuation-sensing sequence in Experiment 3.

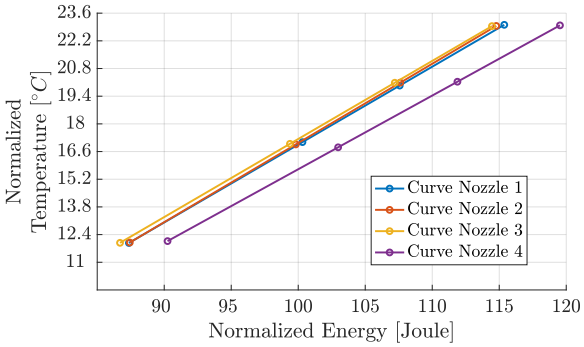


Figure 8: Estimation of energy-temperature curve for four nozzles.

Estimation of Liquid Temperature Using Energy-Temperature Curve

Based on the energy-temperature curves, one can determine a parametric linear model relating the acoustic energy and the liquid temperature. For an individual nozzle, let the energy-temperature characteristic curve be modeled as:

$$x_i = m_i \phi_i + c_i. \tag{7.1}$$

Here, for i^{th} nozzle, x_i is the liquid temperature and ϕ_i is the acoustic energy. The unknown parameters $m_i, c_i \in \mathbb{R}$ are obtained by fitting the respective energy-temperature curve. Every time a liquid droplet is jetted from a nozzle, thereafter, it can be followed by measuring the acoustic signal. The acoustic signal is modeled as (3.1) and its energy ϕ_i is determined using Algorithm IV.1. Subsequently, the corresponding liquid temperature x_i is obtained from (7.1).

Estimation of Liquid Viscosity

One can also use the temperature estimate to infer the change in viscosity of the liquid. Specifically, using the viscosity-temperature curve for a particular ink material, variation of viscosity can be estimated with respect to the acoustic energy. In figure 9, energy-viscosity curves are estimated for four nozzles.

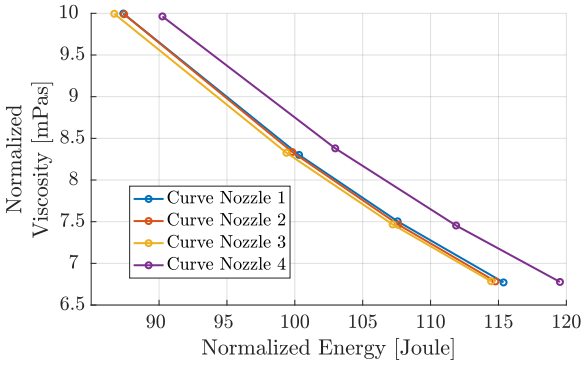


Figure 9: Estimation of energy-viscosity curve for four nozzles.

7.4 Experimental Validation

7.4.1 Experimental Setup

To validate the digital twin as well as the soft sensor, an experimental set-up is built, as shown in figure 10. The set-up consists of two liquid vessels that are connected with a printhead that has 160 nozzles. By keeping two vessels at a fixed level, a constant re-circulation flow through the printhead is established. To raise the temperature of the liquid inlet in the printhead, the upper vessel is equipped with a heater. For each side of the NP, two locations are selected to

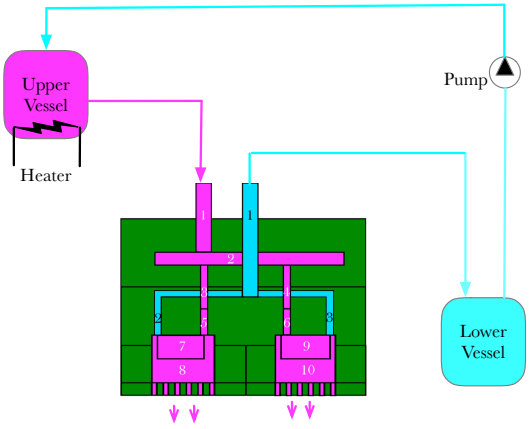


Figure 10: Experimental Setup.

place thermo-couples (in figure 11 they are called $T_{l,1}, T_{l,2}$ for NP_{left} and $T_{r,1}, T_{r,2}$ for NP_{right}) to receive real-time information of its temperature. These thermo-couples are used solely for validation purpose and are not allowed in the final product. Moreover, at each side of NP, one nozzle is selected whose temperature

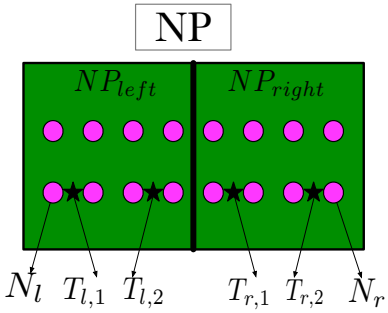


Figure 11: Bottom configuration of NP with thermo-couples ($\star$).

is monitored using the developed soft sensor (in figure 11 they are called N_l for NP_{left} and N_r for NP_{right}). In this way, temperature evolution of both solid and liquid are measured on the both side of the printhead.

7.4.2 Experiment Design

The printhead is operated to print 3500 A-4 sheets of chapter. Only the nozzles in NP_{left} are actuated for jetting while keeping the nozzles in NP_{right} idle. The bit-map is designed such that

1. For 100 seconds, first 500 pages are left unprinted.
2. In the next 100 seconds, another 500 pages are entirely printed by using all the nozzles in NP_{left}.
3. In the next 100 seconds, another 500 pages are left unprinted.
4. In the next 100 seconds, another 500 pages are entirely printed by using half of the nozzles that are located on the left side of NP_{left}.
5. In the next 100 seconds, 500 pages are left unprinted.
6. In the next 100 seconds, another 500 pages are entirely printed by using half of the nozzles that are located on the right side of NP_{left}.
7. In the next 100 seconds, the last 500 pages are left unprinted.

7.4.3 Setting Up the Digital Twin for Simulation

Based on the bit-map, corresponding flow parameters are assigned with the print-job signals. A graph theoretical digital twin of the printhead is built by following the nodal structure that is depicted in figure 2. Every individual node is associated with its temperature (in °C) as its internal state. Based on the topology of the printhead, the interconnection signals and their relations are built. The print-job

also assigns the flow parameters to every individual node through the print-job signals and their relations. Once the model is built using the definitions (**??**)-(**??**), any one of the representations $\mathcal{P}_{\text{I}}$, $\mathcal{P}_{\text{III}}$, $\mathcal{P}_{\text{III}}$ can be used for simulating the digital twin.

7.4.4 Results

In figure 12, at location $T_{l,1}, T_{l,2}$ and $T_{r,1}, T_{r,2}$, temperature data measured by the thermo-couples are compared against the temperatures of the corresponding nodes simulated by the digital twin. Using the developed soft sensor, in figure 13, the estimated liquid temperature at N_l and N_r are compared against the temperatures of the corresponding nodes simulated by the digital twin. The

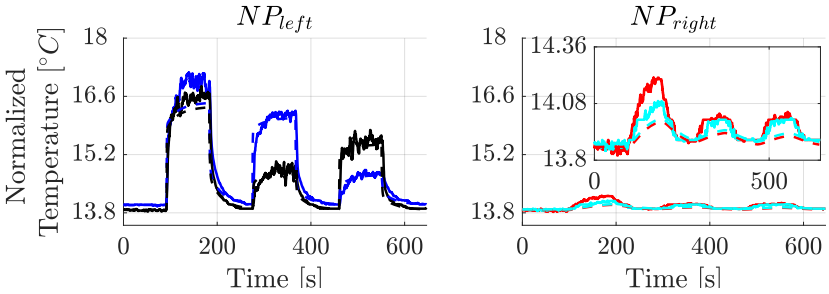


Figure 12: Comparisons among the temperature measurements from the thermo-couple and the digital twin. The legends are as follows: temperature at $T_{l,1}$; temperature at the corresponding node in the digital twin; temperature at $T_{l,2}$; temperature at the corresponding node in the digital twin; temperature at $T_{r,1}$; temperature at the corresponding node in the digital twin; temperature at $T_{r,2}$; temperature at the corresponding node in the digital twin.

developed model captures the thermo-fluidic behaviour of the printhead by indicating the offset in temperature among jetting and non-jetting nozzles. For example, as the nozzles in NP_{right} are idle during the entire print-job, its temperature variation is significantly smaller. Nevertheless, there is still a small temperature variation for nozzles in NP_{right} due to thermo-fluidic cross-talk among nozzles. Using only the left half of the nozzles or the right-half of the nozzles in NP_{left}, among nozzles, there is a significant temperature difference over time (around $2°C$ difference between $T_{l,1}$ and $T_{l,2}$ during $300 - 400$ seconds). The role of the liquid re-circulation is also evident as the temperatures of the NP always go to a steady-state value while being idle.

Similarly, in figure 13, the soft sensor's estimation of the liquid temperature follows the same trend that of the digital twin. The sudden oscillations that appear in the soft sensor's estimation are due to sporadic bubble entrapment or drying of liquid inside the nozzle during jetting, which may fluctuate the liquid

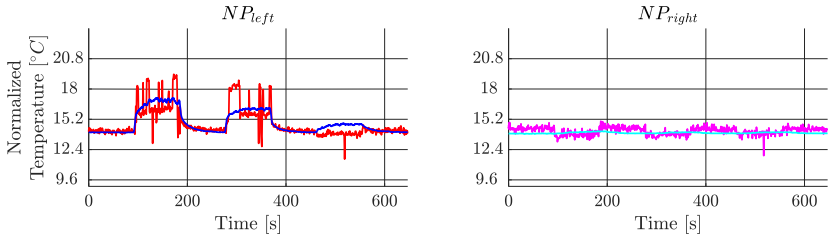


Figure 13: Comparisons of nozzle temperature among the estimates of the developed soft sensor and the digital twin. The legends are as follows: estimated temperature of N_l by soft sensor; temperature at the corresponding node in the digital twin; estimated temperature of N_r by soft sensor; temperature at the corresponding node in the digital twin.

temperature. Such erratic phenomena are not captured in the digital twin. On the other hand, the non-jetting nozzles' temperature estimation is almost identical to the model.

7.5 In Situ Controller: A Proof of Principle

The performance limiting aspects of thermo-fluidic processes is the fluctuation in liquid temperature among adjacent nozzles in a printhead and among adjacent printheads. To maintain temperature consistency, a controller is synthesized that does not require incorporating any additional sensor or actuator. The only resources the controller uses are a) the bit-map as prior knowledge about the flow pattern and jetting sequence of nozzles in the printhead, b) the model derived in Section III to anticipate and predict the evolution of thermo-fluidic behaviour in the printhead, and c) the soft sensor as feedback information on temperature and viscosity at every nozzle.

7.5.1 Concept of In Situ Sensing-Actuation

Every individual nozzle is equipped with a piezoelectric element for jetting droplets of liquid. By applying a voltage, its resistive property allows the piezoelectric element to dissipate heat. In this controller design problem, the resistive property of a jetting actuator is used as a source of heating. Thus, the piezoelectric element makes every individual nozzle equipped with a local, independent, *in situ* self-sensing heating actuator.

However, the piezoelectric elements can only be used as heaters for control input when there is no need for jetting liquid. To implement the in situ sensing-actuation scheme, at all time $t \in \mathbb{T}$, every individual nozzle is in either jetting, heating, sensing or idle mode. To allocate the nozzles in these four modes of operations, following time-varying set-valued maps are defined:

1. $\mathbb{S}_p(t)$: The set of all nozzles that are used for jetting liquid droplets at time t. The cardinality of the set is $n_p(t)$.
2. $\mathbb{S}_h(t)$: The set of all nozzles that are non-jetting and are used as heaters at time t. The cardinality of the set is $n_h(t)$.
3. $\mathbb{S}_s(t)$: The set of all nozzles whose temperatures are sensed at time t. The cardinality of the set is $n_s(t)$.
4. $\mathbb{S}_u(t)$: The set of all nozzles that are neither in any of the above modes at time t. The cardinality of the set is $n_u(t)$.

The cardinality of each set (as a function of time) depends on a specific bit-map. Intersections of $\mathbb{S}_p(t)$ and $\mathbb{S}_h(t)$ must be empty for all time $t \in \mathbb{T}$. This requirement on functional exclusion is a serious engineering challenge as a piezoelectric element in heating mode is not supposed to jet. As described in Section IV, there is always a delay between applying the jetting pulse and sensing the temperature using a soft sensor. Therefore, the jetting nozzles are chosen not to be used for sensing (i.e. the intersection between $\mathbb{S}_p(t)$ and $\mathbb{S}_s(t)$ is chosen to be empty).

7.5.2 Performance Specifications

The controller must satisfy the following performance specifications:

1. Irrespective of the bit-map, the differences in temperature among adjacent nozzles must be below ± 0.3°C.
2. The usage of thermal actuation must be limited to sustain operational lifetime of piezoelectric elements.
3. Cooling of the liquid is not possible using the thermal actuation.
4. Thermal actuation of piezoelectric material should be non-jetting, i.e. it should not form new droplets of liquid.

Intuitively, using the in situ sensing-actuation scheme, the controller is expected to actuate only on a few heating nozzles (from the set $\mathbb{S}_h(t)$) adjacent to the jetting ones to limit the temperature difference among nozzles while satisfying the performance specifications. At the same time, the nozzles' temperature can be monitored using the implemented soft sensor at the sensing nozzles (from the set $\mathbb{S}_s(t)$).

7.5.3 Generating Voltage Pulse for Heating Actuators

To apply the required amount of thermal power as control inputs, one needs to design voltage pulses that are to be applied on the piezoelectric elements in heating mode. Such voltage pulse must not cause ejection of droplets while

satisfying the power requirement. The principle behind generating a non-jetting and heating voltage pulse is to modulate the amplitude and frequency of the trapezoidal pulses (c.f. [93]). Recently, in [73], a mechanism is developed so as to generate specific thermal energy while not interfering with the bandwidth that defines the jetting mode. Using band-pass modulation, the width and the height of the trapezoidal voltage pulses are tuned. Further details on these signal implementations are omitted for brevity.

7.5.4 Implementation and Illustration of MPC Scheme

As proof of principle, the presented MPC scheme is implemented to minimize the difference in liquid temperature among nozzles. To this end, the following specifications are considered:

Printhead Configuration

The printhead chosen for implementing MPC has the architecture shown in figure 7.1(a). There are 48 nozzles that are equally divided over two-sides of the NP, and at each side, 24 nozzles are equally distributed over two rows.

Bit-map Specification

For generating a test bit-map, the following scenarios are considered:

- Scenario 1: All 24 nozzles that located in NP_{left} are used as a jetting nozzle.
- Scenario 2: 12 nozzles in NP_{right} that are located closest to the NP_{left} are used as jetting nozzle.
- Scenario 3: 12 nozzles in NP_{right} that are located farthest from the NP_{left} are used as jetting nozzle.

Setting Up the Digital Twin of the Model

Based on the above scenarios, the flow parameters are assigned to every individual node. At the same time, the thermal power dissipated by the individual jetting nozzle is used as known disturbances.

The model of every individual node is considered in discrete time by choosing the $\mathbb{T} = \{kt_d \mid k \in \mathbb{N} \cup \{0\}\}$ with $t_d = 0.01$ seconds. Euler's approach is used for time discretization due to its sparse and structure preserving implementation ([92], page 4).

The thermo-fluidic model is defined as a graph following the definitions (**??**)-(**??**) in Section III. The equivalent representations $\mathcal{P}_{\text{II}}$ and $\mathcal{P}_{\text{III}}$ are determined by eliminating the interconnection and print-job signals.

The in situ sensing-actuation scheme, as discussed in Section VI.A is implemented by allocating the nozzles based on their four modes operations. The set $\mathbb{S}_p(t)$ determines the set of all jetting nozzles. The rest of nozzles, except the three nozzles that are located at the furthest distance from the jetting nozzles, are used for heating. This determines $\mathbb{S}_h(t)$. All the nozzles that do not belong to $\mathbb{S}_p(t)$ are used as soft sensors to estimate the change in liquid temperature. With the allocated nozzles, the matrices $S^h(t), S^p(t)$, $S^s(t), S^u(t)$ are built. The model for control (3.5) is built subsequently. The unmeasured states are automatically replaced by the corresponding state-updates from the model.

Specifying Control Criterion

The purpose of the controller is to control the temperature difference among nozzles. They are defined by a vector of to-be-controlled variables

$$z(t) = Hx(t), \tag{7.2}$$

where, z denotes the vector of all temperature differences among individual adjacent nozzles. In particular, $H \in \mathbb{R}^{n_x \times n_x}$ is a sparse matrix with $H_{i,i} = 1$ and $H_{i,j} = -1$ whenever the i^{th} nozzle is adjacent to the j^{th} nozzle. At every time instant $t = kt_d$ with $k \in \mathbb{N} \cup \{0\}$, the requirement is to control the liquid temperature gradient among nozzles over a finite horizon of future time instants $t \in \mathbb{T}_N^k$ where $\mathbb{T}_N^k := \{kt_d \mid k \in \mathbb{N}_{[k,N+k]}\}$ for $N > 0$. To this end, the reference tracking problem is formulated where the reference trajectories $(x^r, u^r(t))$ are pre-determined from a steady state model that equates the temperature difference of nozzles to zero. In other words, for $t \in \mathbb{T}_N^k$, $(x^r, u^r(t))$ is the solution of the following linear equations:

$$\begin{aligned} x^r =& \tilde{A}(t)x^r + \tilde{B}(t)S^h(t)u^r(t) + G(t) + W(t)S^p(t)d^p(t), \\ 0 =& Hx^r. \end{aligned} \tag{7.3}$$

Solving MPC Scheme

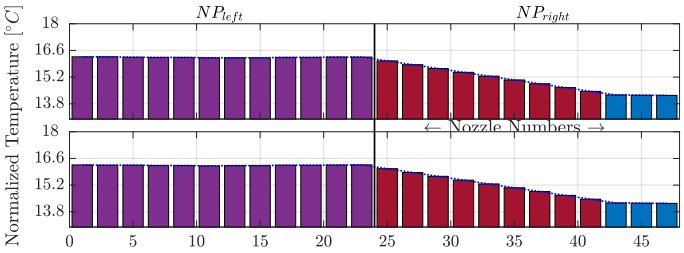
Using (3.5), The prediction model is built over the time horizon $t \in \mathbb{T}_N^k$ where $\mathbb{T}_N^k := \{kt_d \mid k \in \mathbb{N}_{[k,N+k]}\}$. Here, $N = 8$. In (3.10), the weights Q and $R_{i|k}$ are chosen as diagonal matrices. Here, the diagonal entries in $R_{i|k}$ are chosen significantly higher than that of Q to strictly penalize the deviation of inputs from its reference values. Moreover, entries of $R_{i|k}$ are chosen such that the heating nozzles adjacent to the jetting nozzles have more input power.

The optimization problem in (3.12) is solved by the freely available `mpcqpsolver` using the interior point method. Once the optimization yields the optimal control inputs, they are applied to the heating piezoelectric actuators by means of non-jetting voltage pulses. The MPC then repeats the same procedure over the entire bit-map iteratively.

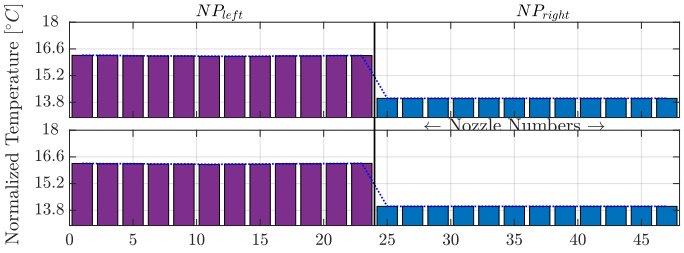
Results

Once the MPC is applied, the liquid temperatures of all 48 nozzles are shown in figure 14. It demonstrates that the performance specifications are met while satisfying the constraints. Moreover, only the adjacent heating nozzles are used to compensate for the temperature inconsistencies among jetting and non-jetting nozzles.

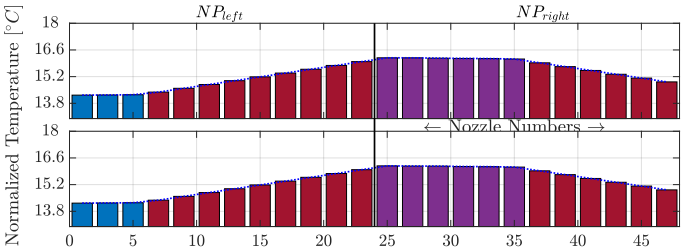
Due to the modularity of the digital twin, up-scaling (or changing) the number of nozzles does not require rebuilding the entire model. The digital twin and the control software are automated to build an up-scaled model with a user-defined number of nozzles, implement in situ sensing-actuation scheme using the bit-map and visualize results of the closed-loop system once the MPC is applied. To demonstrate that, the same printhead, shown in figure 7.1(a), is equipped with 160 nozzles that are equally divided over NP_{left} and NP_{right}. In each side, there are 80 nozzles that are equally distributed over four rows. In figure 15, the result of MPC applied configuration is shown when half the nozzles in NP_{left} are used for jetting.



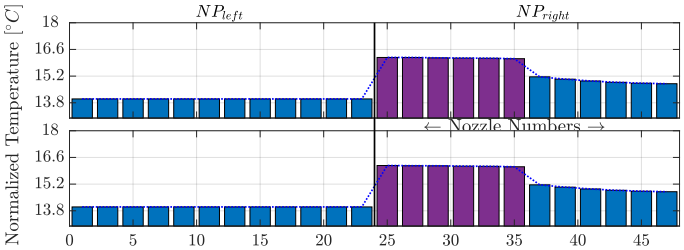
(a) Controlled Scenario 1: NP_{left} is fully used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{0.1988}^{\circ}C$.



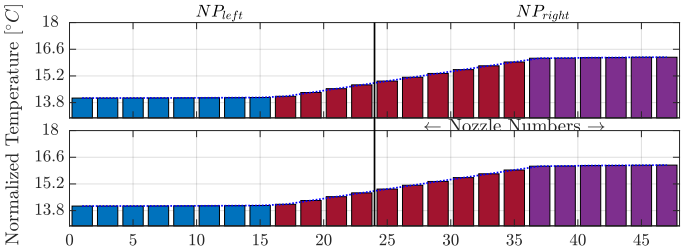
(b) Uncontrolled Scenario 1: NP_{left} is fully used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{2.2551}^{\circ}C$.



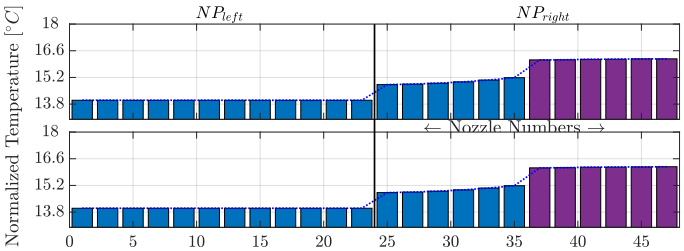
(c) Controlled Scenario 2: Left half of NP_{right} is used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{0.2}°C$.



(d) Uncontrolled Scenario 2: Left half of NP_{right} is used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{2.1726}°C$.

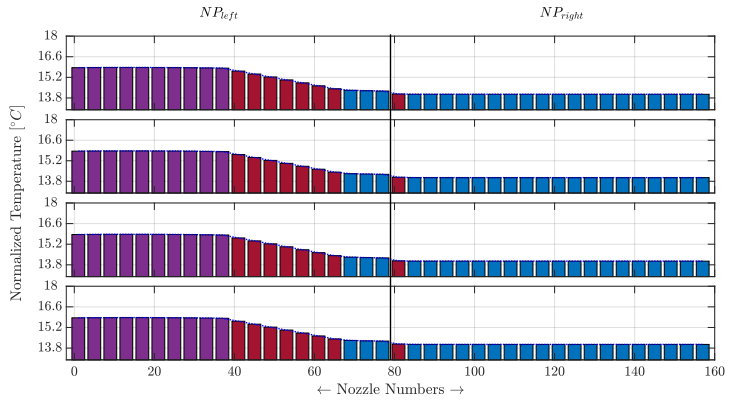


(e) Controlled Scenario 3: Right half of NP_{right} is used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{0.2}°C$.

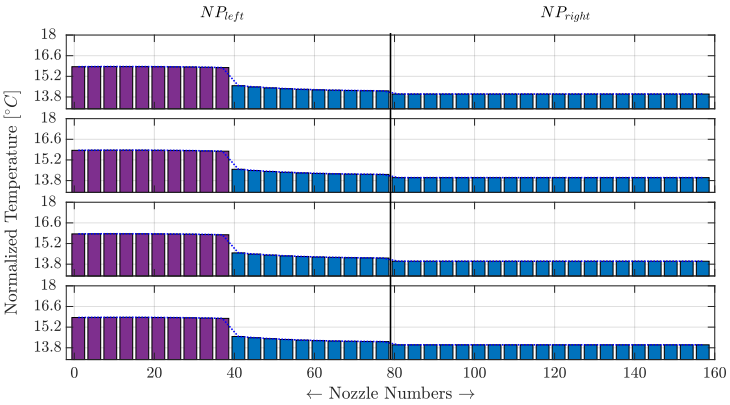


(f) Uncontrolled Scenario 3: Right half of NP_{right} is used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{0.9368}°C$.

Figure 14: Liquid temperature of 48 nozzles for three scenarios in controlled and uncontrolled case. ■ denotes nozzles that are jetting. ■ denotes the adjacent nozzles that are used as thermal actuators. ■ denotes the nozzles that are not used for heating.



(a) Controlled Scenario: NP_{left} is fully used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{0.1988}°C$.



(b) Uncontrolled Scenario: NP_{left} is fully used for jetting. Maximum absolute difference of temperature between two adjacent nozzles is $\mathbf{1.2292}°C$.

Figure 15: Up-scaled controller for 160 nozzles. ■ denotes nozzles that are jetting. ■ denotes the adjacent nozzles that are used as thermal actuators. ■ denotes the nozzles that are not used for heating.

Remark 7.1 *(On computational complexity, c.f. [107]) With an increasing number of the nozzles, evidently, the computational complexity of solving MPC increases. At a specific iteration of k, the number of decision variables is related to the number of states, the number of control inputs and the number of horizons. Let these numbers be n_x , n_h^k and N respectively. In contrast to solving* (3.10)-(3.11) *that has a complexity of $\mathcal{O}(N^3(n_x + n_h^k)^3)$, the dense LCQP* (3.12) *has a lower complexity $\mathcal{O}(N^3 {n_h^k}^3)$ [62]. This is achieved by eliminating the states from the decision variables with substitution of equality constraints. However, in this more condensed formulation, the sparsity of matrices is partially lost [44]. Yet the dense LCQP* (3.12) *is to be preferred from a computational point of view as the dimension of decision variables involves only the number of non-jetting heating actuators. This is typically small in number.*

7.6 Closing Remarks

In this chapter, a modular and flexible digital twin is presented for modeling and control of thermo-fluidic processes in a DoD inkjet printhead. In particular, no additional sensors or actuators are incorporated by developing an in situ sensing-actuation based control strategy that minimizes the liquid temperature fluctuations among individual nozzle. To this end, an experimentally validated graph-theoretic modeling framework is developed that is modular up to an arbitrary number of nozzles. It is demonstrated that this model is flexible, scalable and versatile and that a number of equivalent input-state-output representations can be derived in a straightforward and explicit manner from the model, depending on the intended application.

A control strategy is implemented without using additional sensors and without using additional actuators. Specifically, to circumvent this limitation, the piezoelectric elements at every individual nozzle serves three roles: a) it is a jetting actuator for depositing liquid, b) it is a soft sensor for estimating liquid temperature, and c) it is a control actuator to diminish gradient in liquid temperature among nozzles. Once a voltage pulse is applied to a piezoelectric material, the soft sensor uses its self-sensing mechanism to measure the acoustic signal. An algorithm is presented that uses the energy of this acoustic signal to estimate the liquid temperature at every individual nozzle. An MPC controller is developed to maintain the fluctuation of liquid temperature among nozzles well below a range of $\pm 0.3^{\circ}$C while using only non-jetting piezoelectric elements as thermal actuators.

8 CHAPTER

Estimation of Moisture in Composite Materials in Fixation Process

This chapter discusses a proof of concept on how to solve practically inspired thermo-fluidic process related problem by using the computational tool of approximation and the computational method of PIEs. To this end, this chapter demonstrates how to estimate the moisture content of a newly printed composite material during the fixation process by only using the temperature measurements of the solid support.

Outline

8.1 Introduction 190
8.2 Overview of the Fixation Process 190
8.3 Problem 1: Modeling and Simulation of Diffusive Thermo-Fluidic Processes in Composite Material 191
8.4 Scaling of Spatial Domain for Individual Spatially Distributed Node 194
8.5 Simulation of Heat and Moisture Diffusion in a Composite Material 195
8.6 Application of PIEs and PIETOOLS for Estimating Moisture Content 197
8.7 Closing Remarks 201

8.1 Introduction

After the jetting process, freshly printed products are passed on to the fixation unit where an individual printed product is dried. There is a set of heaters and air impingement units that provide heat and dry air to evaporate excess moisture content from the individual printed product. To achieve an optimal dryness of a printed product, heating and air actuators must obtain the real-time information about the temperature and moisture content of every individual printed product during the fixation process. Currently, the fixation unit is equipped with thermocouple to measure the temperature of the solid support, which is placed underneath the printed product.

Contribution of this Chapter

The main contributions of this chapter are on the following aspects:

1. Modeling and simulation of coupled heat and moisture diffusion through the thickness of a printed product using the modeling framework presented in Chapter 2 and the approximation method presented in Chapter 4.
2. $\mathcal{H}_\infty$ optimal estimation of average temperature and moisture content of printed composite material in the presence of L_2-bounded disturbance when only the temperature of solid support is available as measured output.

8.2 Overview of the Fixation Process

For drying a single printed product, the fixation process is depicted in Figure 1.

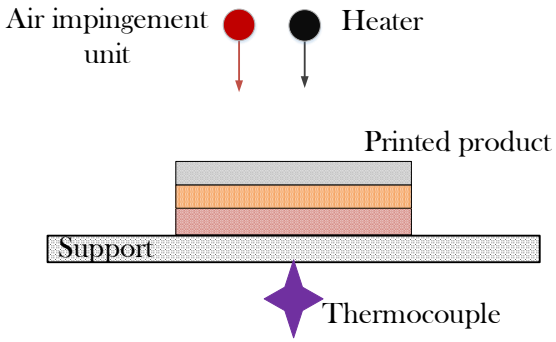


Figure 1: Simplified front view of the fixation process.

Regarding, the fixation setup, the following specifications are taken into account:

- The printed product is considered to be a three-layered composite material over its thickness. At every layer of composite material, the diffusive thermo-fluidic process takes place by means of coupled heat and mass transfer.
- The support is considered to be a lumped solid block that conducts heat.
- External inputs are mass-flux from the air impingement unit and heat flux from the heater. The measured output is the temperature of solid support, measured by the thermocouple placed at the bottom of the support.

8.3 Problem 1: Modeling and Simulation of Diffusive Thermo-Fluidic Processes in Composite Material

In this section, the printed product is modeled as a multi-layered composite material and the thermo-fluidic process is governed by coupled heat and mass diffusion process.

8.3.1 Graph Theoretic Modeling of Fixation Process

The graph theoretic model of the fixation process is depicted in Figure 2

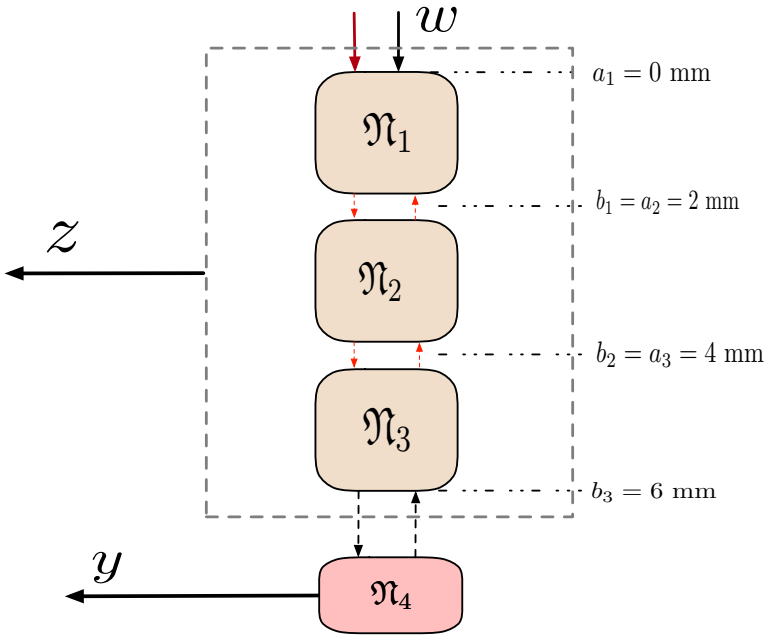


Figure 2: Graph theoretic definition of fixation process.

Following the definitions (D.1)-(D.3) (on page 23-24), the graph consists of four nodes and three edges that are defined below:

Topology

The fixation process is finite and connected graph defined as

$$\mathcal{G} = (\mathfrak{N},\ \mathfrak{E},\ A, \mathbb{T}).$$

Here, $\mathbb{T} = [0, \infty)$, with $\mathfrak{N} = \{\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3, \mathfrak{N}_4\}$ and $\mathfrak{E} = \{\mathfrak{E}_{1,2}, \mathfrak{E}_{2,1}, \mathfrak{E}_{2,3}, \mathfrak{E}_{3,2}, \mathfrak{E}_{3,4}, \mathfrak{E}_{4,3}\}$. The adjacency matrix $A \in \mathbb{R}^{4\times4}$ is symmetric and defined as following:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Spatially Distributed Nodes

Every individual node $\mathfrak{N}_i \in \mathfrak{N}$, $i \in \mathbb{N}_{[1,4]}$ is specified in following items

- $\mathbb{X}_1 = [0, 2]$ (in mm), $\mathbb{X}_2 = [2, 4]$ (in mm), $\mathbb{X}_1 = [4, 6]$ (in mm), $\mathbb{X}_4 = \emptyset$.
- $\mathbb{X}_1^{\text{bc}} = \{0\}$ (in mm), $\mathbb{X}_2^{\text{bc}} = \emptyset$, $\mathbb{X}_3^{\text{bc}} = \emptyset$, $\mathbb{X}_4^{\text{bc}} = \emptyset$.
- Every individual spatially distributed node (i.e. $\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3$) is governed by coupled diffusion equations as given below:

$$\begin{bmatrix} \dot{\mathbf{x}}_{2i}^1(s_i, t) \\ \dot{\mathbf{x}}_{2i}^2(s_i, t) \end{bmatrix} = D_i \begin{bmatrix} \partial_{s_i}^2 \mathbf{x}_{2i}^1(s_i, t) \\ \partial_{s_i}^2 \mathbf{x}_{2i}^2(s_i, t) \end{bmatrix}. \tag{8.1}$$

  Here, for $s_i \in \mathbb{X}_i$, $D_i \in \mathbb{R}^{2\times2}$ is the diffusion coefficient for the node $\mathfrak{N}_i$, when $i \in \mathbb{N}_{[1,3]}$. Moreover, $\mathbf{x}_{2i}^1$ is the spatio-temporal temperature (°C) and $\mathbf{x}_{2i}^2$ is the spatio-temporal moisture (g/m^3) for every individual node $\mathfrak{N}_i$, $i \in \mathbb{N}_{[1,3]}$.

- The boundary condition at $\mathbb{X}_1^{\text{bc}} = \{0\}$ is given by:

$$-D_1 \begin{bmatrix} \partial_{s_1} \mathbf{x}_{21}^1(0, t) \\ \partial_{s_1} \mathbf{x}_{21}^2(0, t) \end{bmatrix} + \begin{bmatrix} \mathbf{x}_{21}^1(0, t) \\ \mathbf{x}_{21}^2(0, t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}. \tag{8.2}$$

  Here, w_1 and w_2 are external input related to the heat flux from heater and mass flux from air impingement unit respectively.

Interconnections Among Spatially Distributed Nodes

Every edge, $\mathfrak{E}_{i,j} \in \mathfrak{E}$, when both $\mathbb{X}_i$ and $\mathbb{X}_j$ are nonempty, is specified in the following items

- $\mathbb{X}^I_{1,2} = \{2\}$ (in mm), $\mathbb{X}^I_{2,3} = \{4\}$ (in mm), $\mathbb{X}^I_{3,4} = \{6\}$ (in mm)
- Among All the edges $\mathfrak{E}_{1,2}$-$\mathfrak{E}_{2,1}$, and $\mathfrak{E}_{2,3}$-$\mathfrak{E}_{3,3}$ are mutual interconnection between two adjacent spatially distributed node. On the other hand, $\mathfrak{E}_{3,4}$-$\mathfrak{E}_{4,3}$ is the interconnection between the support (lumped node) and the $\mathfrak{N}_3$ (spatially distributed node).
- At $\mathbb{X}^I_{1,2} = \{2\}$ and $\mathbb{X}^I_{2,3} = \{4\}$, the interconnection boundary conditions are

$$\begin{aligned}
\begin{bmatrix} \mathbf{x}^1_{21}(2,t) \\ \mathbf{x}^2_{21}(2,t) \end{bmatrix} &= \begin{bmatrix} \mathbf{x}^1_{22}(2,t) \\ \mathbf{x}^2_{22}(2,t) \end{bmatrix}, \\
D_1 \begin{bmatrix} \partial_{s_1}\mathbf{x}^1_{21}(2,t) \\ \partial_{s_1}\mathbf{x}^2_{21}(2,t) \end{bmatrix} &= D_2 \begin{bmatrix} \partial_{s_2}\mathbf{x}^1_{21}(2,t) \\ \partial_{s_2}\mathbf{x}^2_{21}(2,t) \end{bmatrix} \\
\begin{bmatrix} \mathbf{x}^1_{22}(4,t) \\ \mathbf{x}^2_{22}(4,t) \end{bmatrix} &= \begin{bmatrix} \mathbf{x}^1_{23}(4,t) \\ \mathbf{x}^2_{23}(4,t) \end{bmatrix} \\
D_2 \begin{bmatrix} \partial_{s_2}\mathbf{x}^1_{22}(4,t) \\ \partial_{s_2}\mathbf{x}^2_{22}(4,t) \end{bmatrix} &= D_3 \begin{bmatrix} \partial_{s_3}\mathbf{x}^1_{23}(4,t) \\ \partial_{s_3}\mathbf{x}^2_{23}(4,t) \end{bmatrix}.
\end{aligned} \tag{8.3}$$

Interconnections between Spatially Distributed Node and Lumped Node

- At $\mathbb{X}^I_{3,4} = \{6\}$ (in mm), the interconnection between $\mathfrak{N}_3$ and $\mathfrak{N}_4$ is given by

$$D_3 \begin{bmatrix} \partial_{s_3}\mathbf{x}^1_{23}(6,t) \\ \partial_{s_3}\mathbf{x}^2_{23}(6,t) \end{bmatrix} + \begin{bmatrix} \mathbf{x}^1_{23}(6,t) \\ \mathbf{x}^2_{23}(6,t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x(t). \tag{8.4}$$

  where $x(t)$ is the temperature of the solid support (°C).

Dynamics of the Lumped Node

The dynamics of lumped node is a linear ODE of the following form:

$$\dot{x}(t) = -x(t) + \mathbf{x}^1_{23}(6,t), \tag{8.5}$$

where, the boundary value of $\mathfrak{N}_3$ node's temperature acts as an interconnection input to the lumped model.

8.4 Scaling of Spatial Domain for Individual Spatially Distributed Node

Now, the three nodes $\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3$ are scaled by shifting their individual spatial domain $\mathbb{X}_i$ within the interval $[0, 1]$ (in mm). To this end, for every $s_i \in \mathbb{X}_i$, $i \in \mathbb{N}_{[1,3]}$, there exists $s \in [0, 1]$ such that $s_i = m_i s + c_i$. Precisely

$$s_1 := 2s, s_2 := 2s + 2, s_3 := 2s + 4. \tag{8.6}$$

As a result, the complete dynamics of $\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3$ can be coupled together on the same domain $[0, 1]$ and the PDE model takes the following form:

$$\dot{\mathbf{x}}_2(s,t) = D\left[\partial_s^2 \mathbf{x}_2(s,t)\right], \tag{8.7}$$

Here, $\mathbf{x}_2 := \mathrm{col}\,(\tilde{\mathbf{x}}^1_{2i}, \tilde{\mathbf{x}}^2_{2i})_{i\in\mathbb{N}_{[1,3]}}$, where $\tilde{\mathbf{x}}^j_{21}(s,t) = \mathbf{x}^j_{21}(2s,t)$, $\tilde{\mathbf{x}}^j_{22}(s,t) = \mathbf{x}^j_{22}(2s + 2,t)$, $\tilde{\mathbf{x}}^j_{23}(s,t) = \mathbf{x}^j_{23}(2s + 4,t)$ with $j \in 1, 2$. Moreover, $D = 0.25\mathrm{diag}(D_1, D_2, D_3)$.

The boundary conditions are

$$\begin{aligned}
-0.5D_1 \begin{bmatrix}\partial_s\tilde{\mathbf{x}}^1_{21}(0,t)\\ \partial_s\tilde{\mathbf{x}}^2_{21}(0,t)\end{bmatrix} + \begin{bmatrix}\tilde{\mathbf{x}}^1_{21}(0,t)\\ \tilde{\mathbf{x}}^2_{21}(0,t)\end{bmatrix} &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}w_1(t)\\ w_2(t)\end{bmatrix},\\
\begin{bmatrix}\tilde{\mathbf{x}}^1_{21}(1,t)\\ \mathbf{x}^2_{21}(1,t)\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{x}}^1_{22}(0,t)\\ \tilde{\mathbf{x}}^2_{22}(0,t)\end{bmatrix} &= 0,\\
D_1 \begin{bmatrix}\partial_s\tilde{\mathbf{x}}^1_{21}(1,t)\\ \partial_s\tilde{\mathbf{x}}^2_{21}(1,t)\end{bmatrix} - D_2 \begin{bmatrix}\partial_s\mathbf{x}^1_{21}(0,t)\\ \partial_s\mathbf{x}^2_{21}(0,t)\end{bmatrix} &= 0,\\
\begin{bmatrix}\tilde{\mathbf{x}}^1_{22}(1,t)\\ \mathbf{x}^2_{22}(1,t)\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{x}}^1_{23}(0,t)\\ \tilde{\mathbf{x}}^2_{23}(0,t)\end{bmatrix} &= 0,\\
D_2 \begin{bmatrix}\partial_s\tilde{\mathbf{x}}^1_{22}(1,t)\\ \partial_s\tilde{\mathbf{x}}^2_{22}(1,t)\end{bmatrix} - D_3 \begin{bmatrix}\partial_s\tilde{\mathbf{x}}^1_{23}(0,t)\\ \partial_s\tilde{\mathbf{x}}^2_{23}(0,t)\end{bmatrix} &= 0,\\
0.5D_3 \begin{bmatrix}\partial_s\mathbf{x}^1_{23}(1,t)\\ \partial_s\mathbf{x}^2_{23}(1,t)\end{bmatrix} + \begin{bmatrix}\mathbf{x}^1_{23}(1,t)\\ \mathbf{x}^2_{23}(1,t)\end{bmatrix} &= \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} x(t).
\end{aligned} \tag{8.8}$$

Rearranging the states in (8.8), one can obtain,

$$B\begin{bmatrix}\mathbf{x}_2(0,t)\\ \mathbf{x}_2(1,t)\\ \partial_s\mathbf{x}_2(0,t)\\ \partial_s\mathbf{x}_2(1,t)\end{bmatrix} = B_w\begin{bmatrix}w_1(t)\\ w_2(t)\end{bmatrix} + B_x x(t), \tag{8.9}$$

for suitably defined constant matrices B, Bw and B_x.

As a result of such spatial scaling, the lumped model of the support has the following form:

$$\dot{x}(t) = -x(t) + \tilde{\mathbf{x}}^1_{23}(1,t). \tag{8.10}$$

8.5 Simulation of Heat and Moisture Diffusion in a Composite Material

The first task is to simulate the composite diffusive process based on the approximation method developed in Chapter 4. To this end, (8.7) with the boundary conditions (8.8) are considered with setting $x(t) = 0$. To implement the three stage approximation technique, first, $\mathbf{x}_2(s,t)$ decomposed as a direct sum of two functions $\mathbf{z}(s,t)$ and $\mathbf{f}(s,t)$. And $\mathbf{z}(s,t)$ and $\mathbf{f}(s,t)$ can be solved separately.

Solving for $\mathbf{f}(s,t)$

- $\mathbf{f}(s,t) := (F_1 s + F_2)\begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$ for some unknown constant matrix F_1, F_2.
- $F_1, F_2 \in \mathbb{R}^{6\times 2}$ solved by substituting $\mathbf{x}_2 = \mathbf{f}$ in (8.9) that leads to the following matrix equations:

$$F_2 \begin{bmatrix} F_2 \\ F_1 + F_2 \\ F_1 \\ F_1 \end{bmatrix} = B_w,$$

  which can be solved uniquely.

Solving for $\mathbf{z}(s,t)$

- $\mathbf{z}(s,t) := \sum_{n=1}^{\infty} \theta_n(t)\Phi_n(\omega_n, s)$, where $\theta_n(t) \in \mathbb{R}$, $\Phi_n(\omega_n, s) \in \mathbb{R}^6$.
- $\Phi_n(s) = P\Psi_n(\omega_n, s)$,
  where $\Psi_n(s) = \operatorname{col}\left(\psi_{ni}(\omega_n, s)\right)_{i\in\mathbb{N}_{[1,6]}}$. $P \in \mathbb{R}^{6\times 6}$ is obtained by diagonalizing composite diffusion coefficient D in (8.7), i.e. $D = P\Lambda P^{-1}$, where $\Lambda = \operatorname{diag}(\lambda_i)_{i\in\mathbb{N}_{[1,6]}}$ with λ_i being one of the positive eigen values of D.
- $\psi_{ni}(\omega_n, s) := a_{ni}\sin(\frac{\omega_n}{\sqrt{\lambda_i}}s) + b_{ni}\cos(\frac{\omega_n}{\sqrt{\lambda_i}}s)$, where $\omega_n, a_{ni}, b_{ni} \in \mathbb{R}$ are unknown constants. Note that, $\psi_{ni}(\omega_n, s)$ are orthogonal functions on the space $L_2[0,1]$.
- For $i \in \mathbb{N}_{[1,6]}$, a_{ni}, b_{ni} as well as ω_n for $n \in \mathbb{N}$ can be obtained by solving for a_{ni} and b_{ni} from the following set of equations

$$B \begin{bmatrix} \Phi_n(\omega_n, 0) \\ \Phi_n(\omega_n, 1) \\ \partial_s\Phi_n(\omega_n, 0) \\ \partial_s\Phi_n(\omega_n, 1) \end{bmatrix} = 0,$$

that can be rewritten as $\Gamma(\omega_n)\Xi_n = 0$ as defined in (4.16) on page 90. For every individual $n \in \mathbb{N}$, ω_n is obtained by solving $\det(\Gamma(\omega_n)) = 0$, and for a value of ω_n, finding Ξ_n amounts to a set of linear equations. There are 12 unknowns in the vector Ξ_n, $\Xi_n := \operatorname{col}(a_{ni}, b_{ni})_{i\in\mathbb{N}_{[1,6]}}$, which can be solved uniquely by 12 equations resulting from $\Gamma(\omega_n)\Xi_n = 0$.

Finding Finite-Dimensional Approximation Model

By choosing the finite number of basis, i.e. $\{\Phi_n(s) \mid n = 1, \cdots, H\}$, the finite dimensional approximation of $\mathbf{z}(s,t)$ is given by solution $\hat{\mathbf{z}}(s,t)$

$$\hat{\mathbf{z}}(s,t) = \sum_{n=1}^{H} \theta_n(t)\Phi_n(s). \tag{8.11}$$

that satisfies

$$\left\langle \Phi_m, \dot{\hat{\mathbf{z}}}(s) - D\left[\partial_s^2 \hat{\mathbf{z}}\right](s) - (F_1 s + F_2)\begin{bmatrix}\dot{w}_1 \\ \dot{w}_2\end{bmatrix} \right\rangle_{L_2^6[0,1]} = 0; \quad \forall m \in \{1, \cdots, H\}. \tag{8.12}$$

This can be rewritten as a linear H-dimensional state space model of the following form:

$$\dot{\theta}(t) = E\theta(t) + G\begin{bmatrix}\dot{w}_1(t) \\ \dot{w}_2(t)\end{bmatrix},$$

where $\theta := \operatorname{col}(\theta_n)_{n\in\mathbb{N}_{[1,H]}}$. Moreover, $E \in \mathbb{R}^{H\times H}$, and $G \in \mathbb{R}^{H\times 2}$ can be found by substituting the expansion of $\hat{\mathbf{z}}$ in the left hand side of (8.12).

8.5.1 Simulation of Coupled Heat and Moisture Transport in Three-Layered Composite

Now, one can recover the solution to original PDEs of the spatially interconnected nodes $\mathfrak{N}_1, \mathfrak{N}_2, \mathfrak{N}_3$ by rescale the spatial domain back to its original values, i.e. substituting $s = \frac{s_i - c_i}{m_i}$ where m_i, c_i are obtained from (8.6).

For presenting the simulation results, the following specifications are taken into account:

- $w_1(t) = Q_t^* \sin(10t) + Q_t^*$, $w_2(t) = 0$, where $Q_t^* = 50$ watt, $Q_m^* = 0$ are constant values.
- Initial conditions for every node's temperature is kept constant; $\mathbf{x}_{2i}^1(s_i, 0) = 25°\text{C}$, for all $i \in \mathbb{N}_{[1,3]}$.
- Initial conditions for every node's moisture is kept constant; $\mathbf{x}_{2i}^2(s_i, 0) = 4.5 \times 10^{-2}\text{g/m}^3$, for all $i \in \mathbb{N}_{[1,3]}$.
- In (8.11), the number of basis functions are chosen to be 50, $H = 50$.
- The values of D_1, D_2 and D_3 is given below:

Table 1: Values of D_1, D_2, D_3

Parameters	Values
D_1	$\begin{bmatrix} 10 & -0.5 \\ -0.5 \times 10^{-2} & 7 \end{bmatrix}$
D_2	$\begin{bmatrix} 5 & -0.25 \\ -0.25 \times 10^{-2} & 3 \end{bmatrix}$
D_3	$\begin{bmatrix} 7 & -0.25 \\ -0.25 \times 10^{-2} & 4 \end{bmatrix}$

Simulation Results

Using $H = 50$, i.e. 50 basis functions, the simulation is run with above specifications for 100 seconds. The results are shown in

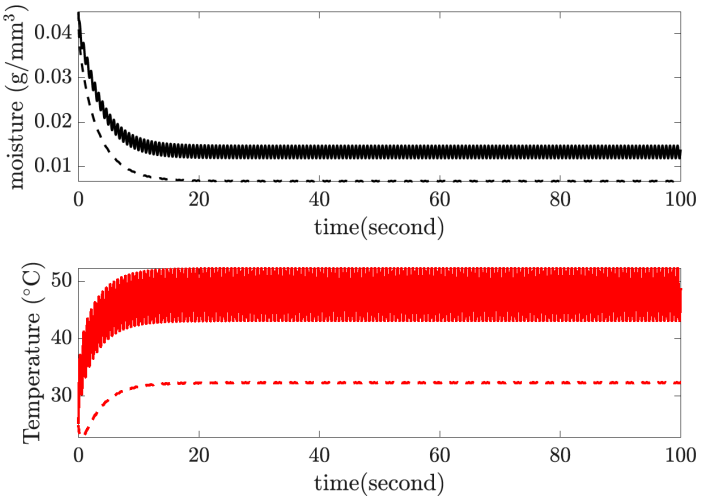


Figure 3: Comparisons of time evolution of moisture and temperature at $s = 0$mm and $s = 6$mm. The legends are as follows: ▬ moisture at $s = 0$mm; ▬ moisture at $s = 6$mm; ▬ temperature at $s = 0$mm; ▬ temperature at $s = 6$mm. As expected, the temperature of the composite material increases while the moisture level decreases over time as fixation process progresses.

8.6 Application of PIEs and PIETOOLS for Estimating Moisture Content

Recall that the fixation process is modeled as a PDE-ODE coupled system with

- PDE Model:

$$\dot{\mathbf{x}}_2(s,t) = D\left[\partial_s^2 \mathbf{x}_2(s,t)\right]$$

- ODE Model:

$$\dot{x}(t) = -x(t) + \tilde{\mathbf{x}}_{23}^1(1,t).$$

- Boundary Condition:

$$B\begin{bmatrix}\mathbf{x}_2(0,t)\\ \mathbf{x}_2(1,t)\\ \partial_s\mathbf{x}_2(0,t)\\ \partial_s\mathbf{x}_2(1,t)\end{bmatrix} = B_w\begin{bmatrix}w_1(t)\\ w_2(t)\end{bmatrix} + B_x x(t).$$

- Measured Output:

$$y(t) = x(t) + \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}w_1(t)\\ w_2(t)\end{bmatrix}$$

- Regulated Output:

$$z(t) = \int_a^b C_{a1}\mathbf{x}_2(s)\mathrm{d}s$$

$\mathcal{H}_\infty$ State Estimator Design

To apply PIE framework, this model can be rewritten in the form of (6.7) on page 126. Using the results from PIE framework, the $\mathcal{H}_\infty$ estimator amounts to taking the following step:

- *Step 1: Determine th equivalent PIE representation*:

In PIETOOLS, this is performed by simply an one-line function

```
convert_PIETOOLS_PDE;
```

- *Step 2: Set up the LPIs*

Finding estimator gain requires, determining $\mathcal{P}$ and $\mathcal{Z}$ by solving the following optimization problem

$$\hat{\gamma} = \arg\min \gamma, \tag{8.13}$$

such that

$$\mathcal{P} := \mathcal{P}\begin{bmatrix} P_2 & Q \\ Q^\top, \{\{R_0 - \epsilon I, R, R\}\} \end{bmatrix} \in \Xi_{d_1}^{d_2},$$
$$-\begin{bmatrix} \mathcal{T}_{Bw}^*(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21}) + (\cdot)^* & 0 & (\cdot)^* \\ 0 & 0 & 0 \\ -(\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T}_{Bw} & 0 & 0 \end{bmatrix} - \begin{bmatrix} -\hat{\gamma}I & -\mathcal{D}_{11}^\top & -(\mathcal{P}\mathcal{B}_1 + \mathcal{Z}\mathcal{D}_{21})^*\mathcal{T} \\ (\cdot)^* & -\hat{\gamma}I & \mathcal{C}_1 \\ (\cdot)^* & (\cdot)^* & (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2)^*\mathcal{T} + (\cdot)^* \end{bmatrix} \in \Xi_{d_1}^{d_2} \tag{8.14}$$

Here, in the definition of $\mathcal{P}$ R_1, R_2 are set to be equal. In this way, $\mathcal{P}$ can be invertible.

- *Step 3: Solving in PIETOOLS and implementation*

For solving the above problem in PETOOLS, one has to

1. declare $\mathcal{P}$ as unknown positive PI operator (using `poslpivar`).
2. declare $\mathcal{Z}$ to be a indefinite PI operator (using `lpivar`).
3. set the inequality constraint can be declared by using the command `lpi_ineq`.
4. set the objective function using `sosdecvar`.
5. set the objective function using `sossetobj`.
6. solve the optimization problem using `sosssolve`.

The implementation of estimator is kept identical t0 the discussion in subsection 6.6.2 (from page 146 onwards).

Simulation Results

The purpose of simulating the $\mathcal{H}_\infty$ estimator for fixation of one composite material is to verify whether it is possible to estimate its moisture content in the presence of boundary inputs from heaters and air impingement units. There are many ways to define what would be physically meaningful regulated variable (given by $z(t)$) from the estimator such that, the framework serves the towards optimum dryness of the printed product. In Table 2, three choices of regulated output are chosen and three different $\mathcal{H}_\infty$ optimal estimator is synthesized. The corresponding level of the worst-case amplification of disturbance to the estimator error is given as γ-value.

Table 2: Values of γ for different regulated outputs

γ-value	Regulated output operator
$\mathbf{0.6557}$	$C_{a1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
0.8	$C_{a1} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$
0.9275	$C_{a1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

1. In the first row, only the average moisture of the first layer of the composite is set to be regulated outputs.

2. In the second row, the entire paper's average moisture is set to be regulated output.

3. In the third row, every individual layer's average moisture are taken as regulated output.

It can be concluded that *in the presence of square integrable disturbances, if one wishes to estimate the average moisture of the composite's top layer, the quality of estimation is better than the average moisture content of all layers individually*.

For simulating the estimator, the applied disturbance is chosen to be same for $w_1(t), w_2(t)$ as damped sinusoidal functions. Moreover, in this simulation, only the average moisture of the first layer is considered to be the regulated output, i.e.

$$z(t) = \int_a^b \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{:=C_{a1}} \mathbf{x}_2(s) \mathrm{d}s.$$

Figure 4 and 5 show the estimation error with respect to the applied disturbance and the estimated level of average moisture content on the first node.

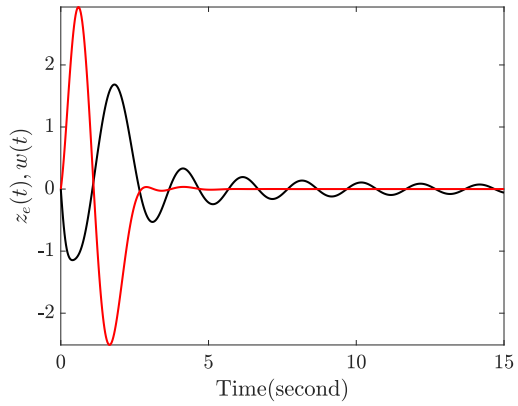


Figure 4: Comparisons of time evolution of applied disturbance and the estimation error. The legends are as follows: ▬ estimation error; ▬ disturbance at the top of $\mathfrak{N}_1$ for both heat flux and mass flux.

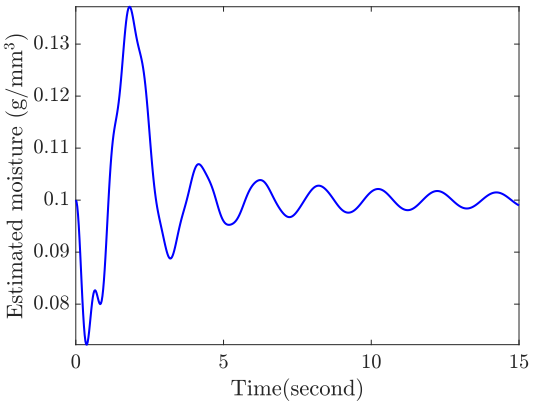


Figure 5: Estimated average moisture along the thickness of $\mathfrak{N}_1$.

8.7 Closing Remarks

In this chapter, a proof of concept is present on how to estimate the moisture content in a composite material despite the presence of square integrable disturbances. To this end, computational framework of PIEs and PIETOOLS are used. It has been shown that estimation of average moisture in the composite's top layer is better than the case when all the layer's combined moisture is inferred from the estimator.

Part VI

Epilogue

9 CHAPTER

Conclusions and Future Outlook

To aid in the digitization of the industrial sector, this thesis develops a digital twin for the assets that are governed by thermo-fluidic processes. A finite graph builds the model of the digital twin, and three distinct computational methods enable its functionalities. The developed tools are applied to construct a digital twin that represents the jetting and fixation processes in a DoD inkjet printer. The digital twin is used to synthesize controllers that guarantee a desired performance from the DoD inkjet printer and improve print quality without adding new sensors and actuators.

Outline

9.1 Viewpoint of the Thesis 206

9.2 Part I: A Framework to Build Thermo-Fluidic Models for the Digital Twin 206

9.3 Part II: Computation with Lumping for Controlling Thermo-Fluidic Process 208

9.4 Part III: Computation with Approximation for Controlling Diffusive Thermo-Fluidic Process 209

9.5 Part IV: Computation with PIEs for Controlling Thermo-Fluidic Process 211

9.6 Part V: Applications to Industrial Benchmarks 213

9.7 Answer to the Research Question 214

9.1 Viewpoint of the Thesis

The viewpoint of this thesis is to develop a generic framework for building a flexible, versatile, modular and integrable digital twin that represents industrial assets governed by thermo-fluidic processes and synthesizes controllers to achieve desirable performance. There are two criteria that the digital twin must facilitate:

1. The digital twin has to demonstrate how to control thermo-fluidic process without adding new sensors and actuators.
2. The digital twin must guarantee that the controlled thermo-fluidic process satisfies a predefined performance that is quantifiable.

To this end, a graph-theoretic modeling framework is presented that allows users to represent the thermo-fluidic processes by specifying a vast class of coupled PDEs and ODEs that are mutually interconnected in an arbitrary spatial topology. Based on the presented model, the digital twin is built to serve various functionalities, including

- The analysis of stability, input-output properties of thermo-fluidic processes,
- Synthesis of a soft sensor allowing real-time updates on thermo-fluidic processes,
- Development of an optimal estimator to determine unmeasured physical quantities in a thermo-fluidic process,
- Predictive control and maintenance of thermo-fluidic processes that are customized towards user-specific demands.

The remainder of this chapter is dedicated to discussing how various aspects of building a digital twin are addressed in this thesis and what are the future recommendations for its further progress.

9.2 Part I: A Framework to Build Thermo-Fluidic Models for the Digital Twin

A mathematical model that represents the thermo-fluidic processes is the core of a digital twin. The model that describes thermo-fluidic processes has the following properties.

- There are more than one physical components that are topologically interconnected among each other.
- The dynamics of some physical components can vary spatio-temporally while other component's dynamics have negligible variation over space.

Subquestion

To address these specifications while deriving the digital twin's model, Chapter 2 finds an answer to the following question:

How to develop a modeling framework that allows users to describe and represent a spatially interconnected thermo-fluidic process such that the model is integrable to the core functionalities of the digital twin?

Answer

Answer to this question is a graph-theoretic framework to represent spatially interconnected components in a thermo-fluidic process. Every individual node in the graph represents a specific component's thermo-fluidic process, and every individual edge describes the physical interconnection between two adjacent components. In particular, every node is either spatially distributed, modeled by linear coupled PDEs or lumped, modeled by linear coupled ODEs. The spatially distributed nodes are represented such a way that both parabolic (e.g. diffusion-transport reaction models of heat and mass transfer) and hyperbolic PDEs (e.g. wave equations, transport equations) can be included in the model. Every lumped node, on the other hand, is equipped with parametric functions that may vary over time based on the user's demand. Every node is equipped with control signals, disturbance signals and sensing signals, as well as the possibilities to add inputs and outputs at the boundary of spatially distributed nodes. By using a suitable algebraic manipulation and scaling of every individual spatial domain, the digital twin's model facilitates various alternative representations of the thermo-fluidic model that are behaviorally equivalent.

Implication

Due to the graph structure, the modeling framework is modular towards any design changes; a thermo-fluidic process may face during its entire life cycle. As far as the implementation is concerned, the modeling framework allows to add or remove a specific component from the graph without causing any significant change in the entire graph. Moreover, arbitrarily many identical components (in terms of node dynamics and edge relations) can be included in the implementation of the model by simply specifying their number. Moreover, the facility of alternative representations allows the model to be utilized for diverse functionalities of the digital twin, including design, estimation, control, performance analysis, diagnosis and monitoring.

Future Outlook

In the current modeling framework, spatially distributed nodes are restricted to be defined in one spatial dimension. Although the mathematical background does not impose any restriction on the spatial dimension, in the future, the model

needs to be extended for spatially distributed nodes that are defined in higher spatial dimensions. Moreover, the spatio-temporal quantities are restricted to admit maximally twice differentiability with respect to the spatial variable. The extension to the PDEs that include more than two spatial differentiations is kept as future work. Additionally, in the future, the PDE models may need a further generalization towards models from the area of fluid dynamics (e.g. Navier Stokes equation).

9.3 Part II: Computation with Lumping for Controlling Thermo-Fluidic Process

Due to the presence of PDEs, there are three computational methods that allow one to perform design, prediction, design of estimator based controller, performance analysis, diagnosis and monitoring of the thermo-fluidic processes using the digital twin.

Subquestion

In Chapter 3, the computational method of lumping is discussed, and an answer is found to the following question:

How to utilize a lumping technique to represent, monitor and control spatially interconnected thermo-fluidic processes without adding new sensors and actuators while meeting user-specific demand?

Answer

Based on the dimensions and physical properties (criterion is based on Biot number, often known or experimentally determined), the method of lumping allows to neglect the spatial distribution of a physical quantity. As a result, the spatially distributed nodes are replaced by lumped models that are typically derived by using lumped (electrical) analogies to thermo-fluidic process and are governed by ODEs. Hence, the PDE models are replaced by newly derived lumped ODE models. However, if the condition on the Biot number is not met, the lumping technique may lead to a thermo-fluidic model with poor accuracy. In order to circumvent this issue, an individual lumped node is allowed to be partitioned in an arbitrary number of mutually interconnected (localized) lumped nodes. The method of node partitioning is kept entirely flexible so that the user can arbitrarily increase the number of partitions until the desired model accuracy is achieved. The utility of a lumped model for building functionalities on the digital twin is addressed with two specific problems In a thermo-fluidic process, often, piezoelectric elements are used to drive a specific volume of liquid flow. Once the piezoelectric elements are activated to drive a liquid droplet, it produces electrical charges that can be measured as output (in Volt). By using the

measured voltage, a data-driven algorithm is presented in this thesis that can provide real-time information about the temperature of the liquid. This algorithm puts the piezoelectric element in (self-)sensing mode and produces a model-based temperature estimate of liquid. In this way, without adding new sensors, the temperature of the liquid can be monitored where the flow-driving piezoelectric elements act as soft-sensor.

Using the model and the soft sensor, a reference-tracking model predictive controller is developed that makes the controlled process to obey a specific performance criterion. Here, the possibility of adding parametric variation on every individual node allows the controller to predict the future evolution of thermo-fluidic process and compensate for any change in the user's demand over time. On the other hand, the developed soft-sensor offers real-time information about liquid temperature. While synthesizing the controller, the user has the flexibility to change the locations of the control inputs and the measured outputs, leading to a flexible and user-friendly design tool.

Implication

The flexible lumping technique set a computational basis for the digital twin without severely compromising the model accuracy. Moreover, a plethora of existing control theories and practices are directly applicable for simulation, design, estimation, control, performance analysis, diagnosis and monitoring of the thermo-fluidic processes. The soft sensor provides a data-driven update on the model, and the controller guarantees a predefined performance from the controlled thermo-fluidic process.

Future Outlook

As the viability of the control strategy is tested in the digital twin, minimizing computational complexity is not the primary focus of this thesis. Especially, the trade-off studies related to different control architectures, e.g. decentralized or distributed architectures for reducing signal overload and faster computation of the controller are not addressed in this thesis. In the case of model mismatch or parametric uncertainty, the robust performance of the controller is also not addressed in this thesis.

9.4 Part III: Computation with Approximation for Controlling Diffusive Thermo-Fluidic Process

In contrast to finding an equivalent lumped model, the second computational method focuses on working with a class of PDE models that describe diffusive thermo-fluidic processes. However, this method numerically approximates them into finite-dimensional ODEs.

Subquestion

In Chapter 4 and 5, the computational method of approximation is discussed and an answer is found to the following question:

How to numerically approximate a spatially distributed diffusive thermo-fluidic process while satisfying the boundary conditions in the presence of boundary inputs and make the approximated model useful for performing various functionalities of the digital twin?

Answer

Diffusive thermo-fluidic processes involve a set of spatially distributed nodes that are governed by coupled diffusion-transport-reaction type PDEs. The developed method of approximating such models is presented in two parts.

- Chapter 4 is dedicated to developing a technique that approximates diffusive thermo-fluidic processes into a finite-dimensional model. To this end, a three-stage procedure is presented where the effect of boundary inputs is first separated from the spatio-temporal solution to the PDEs. Next, the solution to the PDEs is represented as a spectral expansion of basis functions where every element of the sum is written as a product of space-dependent basis functions and time-dependent coefficient functions. At the third stage, only a finite number of space-dependent basis functions are chosen to project the infinite-dimensional solution on to a finite-dimensional subspace. The boundary conditions are treated as explicit constraints in parameterizing the basis functions, making the approximated solutions explicitly consistent with the boundary conditions.
- The developed approximation technique assumes that the physical parameters of the diffusive thermo-fluidic processes are known. In case they are not known, in Chapter 5, a grey-box identification technique is presented to estimate spatially varying physical parameters of diffusion-transport-reaction type PDEs using measurements from a finite number of point locations. The identification technique is performed in the frequency domain using a sparse numerical technique without requiring the boundary conditions to be known in advance. A key feature of the technique is that it allows to simultaneously identify the spatial distribution of diffusive, advective, reactive and input-dependent functions in the underlying PDE, based on measured data.

Implication

The approximation method that is developed for diffusion-transport-reaction type of PDE models defines a second computational method for the digital twin. In this method, the model accuracy directly depends on the number of basis functions chosen for projecting the PDE model on a finite-dimensional subspace. Once the model is approximated into a finite-dimensional ODE model, it can be

interconnected to the remaining lumped model and existing control theories and practices are directly applicable for simulation, design, estimation, control, performance analysis, diagnosis and monitoring of such approximated models. However, the performance of a controlled system on the basis of approximation is not valid for the original spatially distributed diffusive thermo-fluidic processes.

Future outlook

The discussion of Chapter 4 and 5 is restricted to the class of diffusive class of thermo-fluidic processes for avoiding numerical instability and inaccuracy in the approximation method. In the future, the approximation method can be generalized to address the entire class of PDEs presented in the modeling framework of thermo-fluidic processes, including models of higher spatial dimension. The research domain of finite element techniques for numerically approximating the PDEs is extremely wide and vast. In the future, the presented approximation technique needs to be compared to the state of the art finite element methods. Furthermore, the presented approximation technique typically yields a model which is very large in dimension. Standard model order reduction techniques are applicable to the numerical model, however, not discussed in this thesis.

9.5 Part IV: Computation with PIEs for Controlling Thermo-Fluidic Process

In lumping and in approximation method of PDEs, the central conservatism is that the general class of PDE-ODE coupled models is not directly used for design, estimation, control, performance analysis, diagnosis and monitoring. By lumping, the PDE models are lumped, and their spatial dynamical features are ignored. On the other hand, by approximation, the PDEs are projected onto a finite-dimensional subspace, and typically many of their physical properties are lost. In contrast, Chapter 6 directly utilizes the general class of PDE-ODE coupled model of thermo-fluidic processes for design, estimation, control, performance analysis, diagnosis and monitoring.

Subquestion

The question for which Chapter 6 seeks an answer is

How to develop a computational framework that allows to analyze and synthesize estimator based optimal controllers directly on the spatially interconnected thermo-fluidic processes without depending on lumping or approximation techniques while providing a quantifiable performance guarantee?

Answer

There are two critical issues in developing a computational method for analysis and control of thermo-fluidic processes that directly uses the class of PDE-ODE coupled systems. The first key issue is including boundary conditions (that may involve external inputs) as explicit constraints in the evolution of PDEs' state variables. The second key issue is the lack of an algebraic structure on the unbounded differential operators. Chapter 6 resolves these two issues by using Partial Integral (PI) operators to equivalently represent PDE-ODE coupled systems as Partial Integral Equations (PIEs). A class of PI operators forms a *-subalgebra and boundary conditions (including boundary inputs) are directly invoked in the dynamics of PIEs. It is shown that the positivity of PI operators can be tested using Linear Matrix Inequalities (LMIs) that are based on computationally effective algorithms to test feasibility (based on the theory of convex functions). It is proved that for the class of thermo-fluidic processes, analysis of exponential stability, input-to-state stability, and determining worst-case disturbance amplification among inputs-output operators as well as synthesizing optimal state estimators amount to verifying the feasibility of Linear PI inequalities (LPIs) that can be tested using LMIs. A software tool, PIETOOLS, accompanies the PIE framework that allows converting PDE-ODE models to PIEs, declare LPIs and solves them using LMIs and provides quantified guarantee of performance. These results have a profound impact on setting up a novel computational paradigm for wide classes of infinite-dimensional systems.

Implication

Introduction of PI operators and PIEs offer a new computational basis for the digital twin that does not explicitly depend on approximating or lumping the thermo-fluidic model. Moreover, LMI-based computational methods have been the cornerstone in developing modern control theory for finite-dimensional systems. Now, the PIE framework and PIETOOLS offer a unified and systematic foundation in bringing LMI-based computation towards design, estimation, control, performance analysis, diagnosis and monitoring of infinite-dimensional thermo-fluidic processes with no apparent conservatism.

Future Outlook

Extension of the PIE framework in higher spatial dimension is kept as future work. Extending the PIE framework for nonlinear PDE-ODE coupled models such as models from the area of fluid dynamics (e.g. Navier Stokes equations) are also kept as future work. Moreover, $\mathcal{H}_2$ optimal estimation and control framework is not addressed in this thesis. PIETOOLS is still a `Matlab`® based software package. In future, PIETOOLS will be made available for open source language platform such as `Python` or `C`.

9.6 Part V: Applications to Industrial Benchmarks

Subquestion

How to apply the developed methodologies in building a digital twin for the jetting and fixation processes in a DoD inkjet printer such that the print quality can be improved without adding new sensors and actuators?

Answer

In Chapter 6 and 8, the developed methodologies are applied to building a digital twin of a DoD inkjet printer that describes the jetting and the fixation processes.

- Chapter 6 discusses a digital twin that synthesizes a control strategy to maintain a consistent liquid temperature at every individual jetting nozzle in a DoD inkjet printhead without adding new sensors and actuators while coping with varying print-demands from the user. To this end, the method of lumping is used to build a graph-theoretic model describing the thermo-fluidic process in a printhead during jetting. It is demonstrated that this model remains modular when the number of nozzles is increased arbitrarily. The model is validated with an experimental set-up equipped with a commercial printhead. The control strategy is implemented without adding sensors and actuators. Specifically, to circumvent this limitation, the piezoelectric elements at every individual nozzle serves three roles: a) it is a jetting actuator for depositing liquid, b) it is a soft sensor for estimating liquid temperature (as developed in Chapter 3), and c) it is a control actuator to diminish gradients in liquid temperature among nozzles. Once a voltage pulse is applied to a piezoelectric material, the soft sensor uses its self-sensing mechanism to estimate the liquid temperature at every individual nozzle. Then a controller anticipates the future change in print demand and is compensate the fluctuation of liquid temperature among nozzles by keeping it well below a range of $\pm\mathbf{0.3}^{\circ}$C while using only non-jetting piezoelectric elements as thermal actuators.

- Chapter 8 discusses a proof of concept on how to estimate the average moisture content of a newly printed composite material during the fixation process by only using the temperature measurements of the solid support. A three-layered composite material is considered where coupled heat, and moisture diffusion takes place. The solid support is considered to be a lumped node. As a result, the fixation of composite material is modeled as PDE-ODE coupled thermo-fluidic process. By using PIE framework, a state estimator is designed that uses the support's temperature (state of the lumped model) as measured outputs. The synthesized state estimator is $\mathcal{H}_\infty$ optimal. It can be concluded that in the presence of square-integrable disturbances if one wishes to estimate the average moisture of the composite's top layer, the quality of estimation is better than the average

moisture content of all layers individually. Furthermore, with the case studies, it has been found that the worst-case disturbances are attenuated at the estimation error by a factor $\mathbf{0.6557}$.

Implication

Using the digital twin, in both the applications, it has been shown that controlling thermo-fluidic process leads to improved performance of the inkjet printer without the expense of adding new sensors and actuators. By using the modeling framework and three computational methods, the merit of the digital twin is evident in building a systematic and control-oriented approach that remains modular, flexible and versatile towards the future design of printers. Moreover, the combination of model, optimal control and careful utilization of the available resources show a great promise for improving print quality in a DoD inkjet printer.

Future Outlook

The multi-purpose usage of the piezoelectric materials in the printhead paves a path to more extensive functionality, including active fault detection and fault isolation of specific printheads. Some initial research in this direction has already been performed. Moreover, in both the applications, controllers and estimators are not implemented in a real-time environment. Addressing implementation related aspects are kept as future work. Moreover, to control the liquid temperature using non-jetting piezoelectric actuators, designing and scheduling non-jetting voltage pulses that are required to apply control input without forming droplets of liquid are not discussed in this thesis.

9.7 Answer to the Research Question

Question

How to develop a digital twin for an industrial asset that is governed by thermo-fluidic processes and guarantee that the asset achieves a predefined performance without adding new sensors or actuators?

Answer

To build a digital twin on the basis of thermo-fluidic processes, the answer to the main research question is the following top-to-bottom design procedure, depicted in Figure 1.

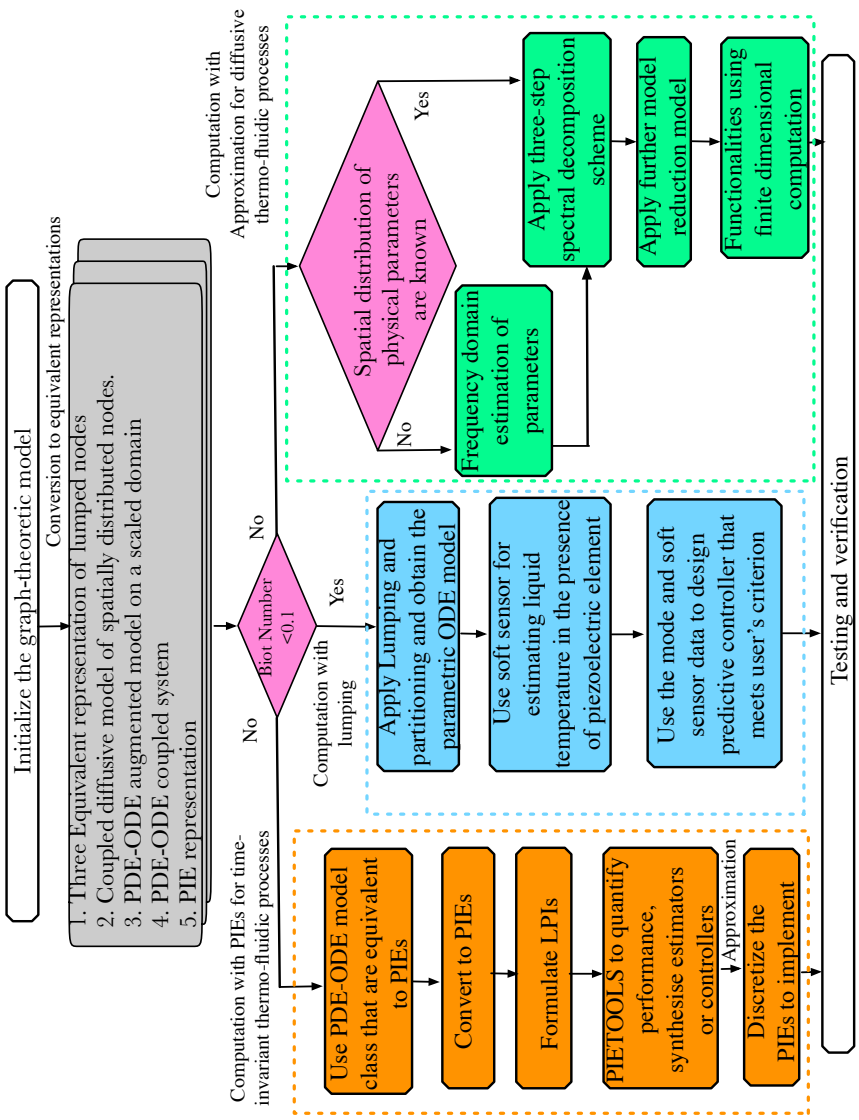


Figure 1: Building digital twin starts from the top by modeling the thermo-fluidic processes and terminates at the bottom by testing and verification.

The given flow-chart results in a modular, flexible, versatile and integrable digital twin that describes any industrial asset that are governed by thermo-fluidic processes and guarantees the asset's performance by synthesizing control strategies that do not require new sensors or actuators.

Bibliography

[1] O. M. Aamo. Disturbance rejection in 2 x 2 linear hyperbolic systems. *IEEE Transactions on Automatic Control*, 58(5):1095–1106, 2013.

[2] M. Ahmadi, D. Valmorbida, G.and Gayme, and A. Papachristodoulou. A framework for input–output analysis of wall-bounded shear flows. *Journal of Fluid Mechanics*, 873:742–785, 2019.

[3] Mohamadreza Ahmadi, Giorgio Valmorbida, and Antonis Papachristodoulou. Dissipation inequalities for the analysis of a class of pdes. *Automatica*, 66:163 – 171, 2016.

[4] M. E. Alpay and M. H. Shor. Model-based solution techniques for the source localization problem. *IEEE Transactions on Control Systems Technology*, 8(6):895–904, 2000.

[5] R. Ambur and S. Rinderknecht. Self-sensing techniques of piezoelectric actuators in detecting unbalance faults in a rotating machine. *Procedia Engineering*, 144:833 – 840, 2016. International Conference on Vibration Problems 2015.

[6] C. Anibas, J. Fleckenstein, N. Volze, K. Buis, R. Verhoeven, P. Meire, and O. Batelaan. Transient or steady-state? using vertical temperature profiles to quantify groundwater–surface water exchange. *Hydrological Processes*, 23(2165-2177), 2009.

[7] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control) (Advances in Design and Control)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[8] A.C. Antoulas. Approximation of large-scale dynamical systems: An overview. *IFAC Proceedings Volumes*, 37(11):19 – 28, 2004. 10th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems 2004: Theory and Applications, Osaka, Japan, 26-28 July, 2004.

[9] Werner J. Auhorn. *Handbook of Paper and Board*, pages 62–149. Wiley-VCH Verlag GmbH & Co. KGaA, 2006.

[10] H. D. Baehr. *Heat conduction and mass diffusion*, pages 105–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[11] K. Baehr, H. D.and Stephan. Heat conduction and mass diffusion. In *Heat and Mass Transfer*, pages 107–273. Springer, 2011.

[12] M. J. Balas. Active control of flexible systems. *Journal of Optimization Theory and Applications*, 25(3):415–436, Jul 1978.

[13] H. T. Banks, J. Crowley, and K. Kunisch. Cubic spline approximation techniques for parameter estimation in distributed systems. *IEEE Transaction of Automatic Control*, 28(7):773–786, 1983.

[14] H. T. Banks and K. Kunisch. Estimation techniques for distributed parameter system. *Birkhaäuser, Boston*, 1989.

[15] A. Bensoussan, G. Prato, M. Delfour, and S. Mitter. *Representation and Control of Infinite Dimensional Systems*, pages 130–450. Birkhäuser Boston, Boston, MA, 2007.

[16] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. Transport phenomena. *AIChE Journal*, 7(2):5J–6J, 1961.

[17] Å. Björck. *Numerical Methods for Least Squares Problems*, chapter 9, pages 339–358. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1995.

[18] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.

[19] D. B. Bogy and F. E. Talke. Experimental and theoretical study of wave propagation phenomena in drop-on-demand ink jet devices. *IBM J. Res. Dev.*, 28(3):314–321, May 1984.

[20] I. J. Busch-Vishniac. *Electromechanical Sensors and Actuators*, pages 17–53. Springer New York, New York, NY, 1999.

[21] E. J. Carr, I. W. Turner, and P. Perré. A dual-scale modeling approach for drying hygroscopic porous media. *Multiscale Modeling & Simulation*, 11(1):362–384, 2013.

[22] G. Chen. Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain. *J. Math. Pures Appl. (9)*, 58:249–273, 1979.

[23] P. D. Christofides. *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*, pages 71–97. Birkhäuser Boston, Boston, MA, 2001.

[24] R. F. Curtain and K. Morris. Transfer functions of distributed parameter systems: A tutorial. *Automatica*, 45(5):1101–1116, 2009.

[25] R. F. Curtain and Hans Zwart. *An Introduction to Infinite-dimensional Linear Systems Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[26] Ruth F. Curtain and Hans Zwart. *An Introduction to Infinite-dimensional Linear Systems Theory*. Springer-Verlag New York, Inc., 1995.

[27] R. D'Andrea and G. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on automatic control*, 48(9):1478–1495, 2003.

[28] A. Das, M. Princen, M. Shokrpour, A. Khalate, and S. Weiland. Soft Sensing Based In Situ Control of Thermo-Fluidic Processes in DoD Inkjet Printing. *IEEE Transactions on Control Systems Technology (Accepted)*, March 2020.

[29] R. Datko. Extending a theorem of A.M. Liapunov to Hilbert space. *Journal of Mathematical analysis and applications*, 1970.

[30] C. De Boor. *A practical guide to splines; rev. ed.* Applied mathematical sciences. Springer, Berlin, 2001.

[31] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[32] R. D'Andrea, C. Langbort, and R. Chandra. A state space approach to control of interconnected systems. In *Mathematical Systems Theory in Biology, Communications, Computation, and Finance*, pages 157–182. Springer, 2003.

[33] E. Evers, B. de Jager, and T. Oomen. Improved local rational method by incorporating system knowledge: with application to mechanical and thermal dynamical systems. *IFAC-PapersOnLine*, 51(15):808–813, 1 2018.

[34] J. P. Freidberg. *Plasma Physics and Fusion Energy*. Cambridge University Press, Cambridge, 2007.

[35] E. Fridman and Y. Orlov. Exponential stability of linear distributed parameter systems with time-varying delays. *Automatica*, 2009.

[36] Avner Friedman. *Partial differential equations of parabolic type*. Dover Books on Mathematics. Dover, Mineola, NY, 2008.

[37] A. Gahlawat and M. Peet. Optimal state feedback boundary control of parabolic pdes using sos polynomials. In *2016 American Control Conference (ACC)*, pages 4350–4355, July 2016.

[38] D. Gelernter. *Mirror Worlds or the Day Software Puts the Universe in a Shoebox: How Will It Happen and What It Will Mean*. Oxford University Press, Inc., USA, 1991.

[39] M. Gevers, P. Högg, H. Hjalmarsson, R. Pintelon, and J. Schoukens. The transient impulse response modeling method and the local polynomial method for nonparametric system identification. *IFAC Proceedings Volumes*, 45(16):55 – 60, 2012.

[40] M. A. Grant and P. F. Bixley. Geothermal reservoir engineering (second edition). pages 249 – 267. Academic Press, Boston, 2011.

[41] C. Hoffmann and H. Werner. Control of heterogeneous lpv subsystems interconnected through arbitrary and switching directed topologies. In *2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, pages 3633–3638. IEEE, 2015.

[42] L. Iapichino. Reduced basis methods for the solution of parametrized pdes in repetitive and complex networks with application to cfd. 2012.

[43] F. P. Incropera. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2006.

[44] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides. A sparse and condensed qp formulation for predictive control of lti systems. *Automatica*, 48(5):999 – 1002, 2012.

[45] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36 – 52, 2020.

[46] Rick K., J. van Schijndel, and H. Schellen. Simplified thermal and hygric building models: A literature review. *Frontiers of Architectural Research*, 1(4):318 – 325, 2012.

[47] I. Karafyllis and Z. Jiang. *Stability and Stabilization of Nonlinear Systems*, pages 55–122. Springer London, London, 2011.

[48] I. Karafyllis and M. Krstic. *Input-to-State Stability for PDEs*, pages 93–140. Springer International Publishing, Cham, 2019.

[49] A. Khalate. *Model-based Feedforward Control for Inkjet Printheads*. PhD thesis, Technische Universiteit Delft, Netherlands, December 2013.

[50] A. Khalate, X. Bombois, G. Scorletti, R. Babuska, S. Koekebakker, and W. de Zeeuw. A waveform design method for a piezo inkjet printhead based on robust feedforward control. *Journal of Microelectromechanical Systems*, 21(6):1365–1374, 2012.

[51] A. Khalate, X. Bombois, S. Ye, R. Babuška, and S. Koekebakker. Minimization of cross-talk in a piezo inkjet printhead based on system identification and feedforward control. *Journal of Micromechanics and Microengineering*, 22(11), 2012.

[52] K. J Kircher and K M. Zhang. On the lumped capacitance approximation accuracy in RC network building models. *Energy and Buildings*, 108:454–462, 2015.

[53] S. Koekebakker, M. Ezzeldin, A. Khalate, R. Babuška, X. Bombois, P. van den Bosch, G. Scorletti, S. Weiland, H. Wijshoff, R. Waarsing, and W. de Zeeuw. *Piezo Printhead Control: Jetting Any Drop at Any Time*, pages 41–85. Springer New York, New York, NY, 2013.

[54] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016 – 1022, 2018. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[55] M. Krstic. *Boundary Control of Pdes: A Course on Backstepping Designs*. Society for Industrial and Applied Mathematics, USA, 2008.

[56] N. Kumar and S. K. Basu. *Diffusive Processes and Modelling: An Introduction*, pages 1–25. Springer International Publishing, Cham, 2014.

[57] K. Kwon. Waveform design methods for piezo inkjet dispensers based on measured meniscus motion. *Journal of Microelectromechanical Systems*, 18(5):1118–1125, Oct 2009.

[58] C. Langbort, R. Chandra, and R. D'Andrea. Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519, 2004.

[59] I. Lasiecka and R. Triggiani. *Control Theory for Partial Differential Equations: Continuous and Approximation Theories*, volume 1 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2000.

[60] J. Lunze. *Feedback control of large-scale systems*. Prentice Hall, New York, 1992.

[61] C. G. Machado, M. Winroth, D. Carlsson, P. Almström, V. Centerholt, and M. Hallin. Industry 4.0 readiness in manufacturing companies: challenges and enablers towards increased digitalization. *Procedia CIRP*, 81:1113 – 1118, 2019. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.

[62] J.M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, New York, 2002.

[63] U. Maeder, F. Borrelli, and M. Morari. Linear offset-free model predictive control. *Automatica*, 45(10):2214 – 2222, 2009.

[64] D. W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, pages 431–441, 1963.

[65] P. Massioni and M. Verhaegen. Distributed control for identical dynamically coupled systems: A decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, 2009.

[66] M. Massoud. *Engineering Thermofluids: Thermodynamics, Fluid Mechanics, and Heat Transfer*, pages 431–600. Springer Berlin Heidelberg, 2005.

[67] Matlab®. Benchmarking A\b. http://nl.mathworks.com/help/distcomp/examples/benchmarking-a-b.html, 2018. [Online; accessed 10-Jan-2018].

[68] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.

[69] T. Meurer. Some perspectives in PDE control. In *Proc. 20th IFAC World Congress*, Toulouse (F), July 2017.

[70] E. Meyer and M. Peet. Stability analysis of parabolic linear pdes with two spatial dimensions using lyapunov method and sos. In *2015 IEEE 54th Annual Conference on Decision and Control (CDC)*. IEEE, 2015.

[71] S. Mokhatab, W. Poe, and J. Mak. Handbook of natural gas transmission and processing (fourth edition). pages 579 – 614. Gulf Professional Publishing, 2019.

[72] U. Muschelknautz. *VDI Heat Atlas*. Springer Berlin Heidelberg, 2010.

[73] H. Nishimura. Temperature uniformity across an inkjet head using piezoelectric actuation, 2017. Patent no. EP3213918A1.

[74] J. Nocedal and S. J. Wright. *Numerical Optimization*, pages 245–269. Springer Science, New York, 2006.

[75] V. V. N. Obreja and A. C. Obreja. Breakdown of semiconductor devices and influence of the interface from passivated termination. In *Proceedings of International Semiconductor Conference*, volume 02, pages 495–498, 2010.

[76] A. Papachristodoulou and M.. Peet. On the analysis of systems described by classes of partial differential equations. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 747–752, Dec 2006.

[77] M. Peet and K. Gu. Sos for systems with multiple delays: Part 2hinfinity optimal estimation. In *2019 American Control Conference (ACC)*, pages 3870–3876, 2019.

[78] Matthew M. Peet. A partial integral equation (pie) representation of coupled linear pdes and scalable stability analysis using lmis, 2018.

[79] J. R. Philip and D. A. De Vries. Moisture movement in porous materials under temperature gradients. *Transactions, American Geophysical Union*, 38:222–232, 1957.

[80] Jan Willem Polderman and Jan C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[81] Y. Privat, Emmanuel Trélat, and E. Zuazua. Optimal observation of the one-dimensional wave equation. *Journal of Fourier Analysis and Applications*, 19(3):514–544, Jun 2013.

[82] A. M. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer Publishing Company, Incorporated, 1st ed. 1994. 2nd printing edition, 2008.

[83] P. Radecki and B. Hencey. Online thermal estimation, control, and self-excitation of buildings. In *52nd IEEE Conference on Decision and Control, Florence*, pages 4802–4807, 2013.

[84] D. Saba, F. Argomedo, J. Auriol, M. Loreto, and F. Meglio. Stability analysis for a class of linear 2x2 hyperbolic pdes using a backstepping transform. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[85] C. Scherer and S. Weiland. Lecture notes on linear matrix inequalities in control. Dutch Intsitute of Systems and Control, TU Delft, 1999.

[86] U. Schneidewind, M. van Berkel, C. Anibas, G. Vandersteen, C. Schmidt, I. Joris, P. Seuntjens, O. Batelaan, and H. J. Zwart. LPMLE3: A novel 1-D approach to study water flow in streambeds using heat as a tracer. *Water Resour. Res.*, 52(8):6596–6610, 2016.

[87] C. H. Séquin. Rapid prototyping: A 3D visualization tool takes on sculpture and mathematical forms. *Communications in ACM*, 48(6):66–73, June 2005.

[88] E. Sontag. *Input to State Stability: Basic Concepts and Results*, pages 163–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[89] G. A Susto and M. Krstic. Control of pde–ode cascades with neumann interconnections. *Journal of the Franklin Institute*, 2010.

[90] M. Sznaier and M. J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *26th IEEE Conference on Decision and Control*, volume 26, pages 761–762, Dec 1987.

[91] S. Tang and C. Xie. Stabilization of a coupled pde-ode system by boundary control. In *49th IEEE Conference on Decision and Control (CDC)*, pages 4042–4047, Dec 2010.

[92] R. Tóth, M. Lovera, P. S. C. Heuberger, M. Corno, and P. M. J. Van den Hof. On the discretization of linear fractional representations of lpv systems. *IEEE Transactions on Control Systems Technology*, 20(6):1473–1489, 2012.

[93] H. Usuda. Droplet discharging apparatus and method, 2004. Patent no. US2004/0135832A1.

[94] Giorgio Valmorbida, Mohamadreza Ahmadi, and Antonis Papachristodoulou. Stability analysis for a class of partial differential equations via semidefinite programming. *IEEE Transactions on Automatic Control*, 2016.

[95] M. van Berkel. *Estimation of heat transport coefficients in fusion plasmas*. PhD thesis, Eindhoven University of Technology, 2015.

[96] M. van Berkel, G. Vandersteen, H. J. Zwart, G. M. D. Hogeweij, and M. R. de Baar. Maximum likelihood estimation of diffusion and convection in tokamaks using infinite domains. *IEEE International Conference on Control Applications (CCA)*, pages 1230–1234, 2013.

[97] A. van der Bos, R. Jeurissen, M. Versluis, D. Lohse, H. Reinten, H. Wijshoff, M. van den Berg, and J. de Jong. Acoustic measurement of bubble size and position in an ink jet printhead. *The Journal of the Acoustical Society of America*, 2013.

[98] Eelco P van Horssen and Siep Weiland. Synthesis of distributed robust h-infinity controllers for interconnected discrete time systems. *IEEE Transactions on Control of Network Systems*, 3(3):286–295, 2016.

[99] J.A. Villegas. *A Port-Hamiltonian Approach to Distributed Parameter Systems*. PhD thesis, University of Twente, The Netherlands, 5 2007.

[100] M. G. Wassink. *Inkjet printhead performance enhancement by feedforward input design based on two-port modeling*. PhD thesis, Technische Universiteit Delft, The Netherlands, February 2006.

[101] R. E. White and V. R. Subramanian. *Method of Lines for Parabolic Partial Differential Equations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[102] H. Wijshoff. *Structure and fluid dynamics in piezo inkjet printheads*. PhD thesis, Technische Universiteit Delft, The Netherlands, January 2008.

[103] J. C. Willems. Dissipative dynamical systems part I: General theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, January 1972.

[104] F.W. Williams, S. Yuan, K. Ye, D. Kennedy, and M.S. Djoudi. Towards deep and simple understanding of the transcendental eigenproblem of structural vibrations. *Journal of Sound and Vibration*, 256(4):681 – 693, 2002.

[105] S. Wollnack, C. Kloock, A. K. Schug, and H. Werner. Well-posedness stability analysis for spatially interconnected systems with finite extent. In *2017 American Control Conference (ACC)*, pages 1803–1808, May 2017.

[106] XAAR. Product Description. https://www.xaar.com/en/products/xaar-printheads/xaar-1003-c/, 2020.

[107] J. Yang, T. J. Meijer, V. S. Dolk, B. d. Jager, and W. P. M. H. Heemels. A system-theoretic approach to construct a banded null basis to efficiently solve mpc-based qp problems*. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1410–1415, 2019.

[108] X. Zhang and E. Zuazua. Control, observation and polynomial decay for a coupled heat-wave system. *Comptes Rendus Mathematique*, 336(10):823 – 828, 2003.

[109] E. Zuazua. Propagation, observation, and control of waves approximated by finite difference methods. *SIAM Review*, 47:197–243, 2005.

List of symbols

$\to$	going to
$\mapsto$	maps to
$[a, b]$	closed interval for any $a, b \in \mathbb{R}, a \leq b$
$:=$	is defined as
$\emptyset$	empty set
$\varnothing$	empty variable
$\partial_s^q f$	q^{th}-order partial derivative of the function f respect to s, i. e. $\frac{\partial^q f}{\partial s^q}$
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{R}^+$	the interval $[0, +\infty)$
$\left\langle x, y \right\rangle_{\mathbb{R}}$	$:= x^\top y$, defines the standard inner product on R^n when $x, y \in \mathbb{R}^n$
$\|x\|_{\mathbb{R}}$	$:= \sqrt{x^\top x}$, for $x \in \mathbb{R}^n$
$L_2^m[a, b]$	the space of square integrable $\mathbb{R}^m$-valued functions on $[a, b]$
$\left\langle x, y \right\rangle_{L_2}$	$:= \int_a^b x^\top(s) y(s) \mathrm{d}s$, the standard inner product on $L_2^m[a, b]$ when $x, y \in L_2^m[a, b]$.
$\|x\|_{L_2}$	$:= \sqrt{\int_a^b x^\top(s) x(s) \mathrm{d}s}$, when $x \in L_2^m[a, b]$.
$\left\langle \begin{bmatrix} x \\ \mathbf{x} \end{bmatrix}, \begin{bmatrix} y \\ \mathbf{y} \end{bmatrix} \right\rangle_{\mathbb{R} \times L_2}$	$:= \left\langle x, y \right\rangle_{\mathbb{R}} + \left\langle \mathbf{x}, \mathbf{y} \right\rangle_{L_2}$, defines the standard inner product on $\mathbb{R}^m \times L_2^n[a, b]$ when $x, y \in \mathbb{R}^m$ and $\mathbf{x}, \mathbf{y} \in L_2^n[a, b]$.
$\left\| \begin{bmatrix} x \\ y \end{bmatrix} \right\|_{\mathbb{R} \times L_2}$	$:= \sqrt{\left\langle x, x \right\rangle_{\mathbb{R}} + \left\langle y, \mathbf{y} \right\rangle_{L_2}}$, when $x \in \mathbb{R}^m \times$ and $y \in L_2^n[a, b]$.
$H_q^m[a, b]$	Sobolev space; $H_q^m[a, b] := \{x \in L_2^m([a, b]) \mid \partial_s^k(x) \in L_2^m[a, b]$ for all $k \leq q, q > 0\}$
$\mathrm{col}\,(x_i)_{i \in \mathbb{A}}$	an ordered stacking of all the x_i in a column for all i in the set $\mathbb{A}$. For example: $\mathrm{col}\,(x_i)_{i \in \{m,n\}} := \begin{bmatrix} x_m \\ x_n \end{bmatrix}$
$\mathrm{row}(y_i)_{i \in \mathbb{B}}$	an ordered stacking of all the y_i in a row for all i in the set $\mathbb{B}$. For example: $\mathrm{row}(y_i)_{i \in \{m,n\}} := \begin{bmatrix} y_m & y_n \end{bmatrix}$

$\mathrm{diag}(M_i)_{i\in\mathbb{D}}$	an ordered stacking of all the matrices M_i block-diagonally for all i in the set $\mathbb{D}$. For example: $\mathrm{diag}(M_i)_{i\in\{m,n\}} := \begin{bmatrix} M_m & 0 \\ 0 & M_n \end{bmatrix}$
$L_\infty^m[a,b]$	defines the space of all essentially bounded $\mathbb{R}^m$-valued functions on $[a,b]$
$\|x\|_{L_\infty[a,b]}$	$\sup_{\tau\in[a,b]} \|x(\tau)\|_{\mathbb{R}}$ for $x \in L_\infty^m[a,b]$.
$\mathbb{N} = \{1,2,\ldots\}$	positive natural numbers
$\mathbb{N}_0 = \mathbb{N} \cup \{0\}$	non-negative natural numbers
$\mathbb{N}_{[m,n]} = \{m, m+1, \ldots, n\}$	finite set of non-negative natural numbers for $0 \leq m \leq n$
$\mathbb{S}^n_{\succcurlyeq 0}$	the set of $n \times n$ positive semi-definite matrices
$\mathbb{S}^n_{\succ 0}$	the set of $n \times n$ positive semi-definite matrices
$\mathrm{Re}(a)$	denotes the real-part of a when a is complex number
$\mathrm{Im}(a)$	denotes the imaginary-part of a when a is complex number
$\mathcal{A}^*$	Adjoint of operator $\mathcal{A}$
$A^\dagger$	denotes the pseudo-inverse of a matrix A
A^{H}	denotes the Hermitian transpose of a matrix A
$\wedge$	logical And operation
$(\mathscr{A})^{\mathbb{L}}$	defines the space of all the mappings $x : \mathbb{L} \to \mathscr{A}$
K	the class of strictly increasing, continuous functions $a : \mathbb{R}^+ \to \mathbb{R}^+$ with $a(0) = 0$
K_∞	the class of strictly increasing, continuous functions $a : \mathbb{R}^+ \to \mathbb{R}^+$ with $a(0) = 0$ and $\lim_{s\to+\infty} a(s) = +\infty$
KL	the class of strictly increasing, continuous functions $\sigma : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ with with the following properties: (a) for every $s > 0$, the mapping $\mathbb{R}^+ \ni t \to \sigma(s,t)$ is non-increasing with $\lim_{t\to+\infty} \sigma(s,t) = 0$, b) for every $t \geq 0$, the mapping $\mathbb{R}^+ \ni s \to \sigma(s,t)$ is of class K

List of Abbreviations

ODE	Ordinary Differential Equations
PDE	Partial Differential Equation
FFT	Fast Fourier Transform
DoD	Drop on Demand
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MIMO	Multiple-Input Multiple-Output
PI	Partial Integral
PIE	Partial Integral Equation
LPI	Linear PI (Partial Integral) Inequality

Acknowledgments

The last four years' journey and where I am today would not be possible without the gracious support of many people in my life. I am grateful to call them my mentors, colleagues, friends and family.

First and foremost, I would like to extend my deepest gratitude to my mentor and promotor, prof. dr. Siep Weiland. In February 2016, in the middle of my graduation project, I made a naive appeal to Siep that I want to pursue a PhD under his supervision. Despite having very little to show for my academic capabilities, Siep was very supportive of my decision and encouraged me to pursue it. From the very first day of my PhD life, he has been the ideal mentor. He has always nurtured me and patiently guided me through the academic life. On an uncountable number of occasions, his depth of knowledge and the ability to interpret mathematics have helped me see the beauty of fundamental theories. His ability to connect theories with practices has not only helped me to improve my research but also made me fall in love with the subjects. His enthusiasm about scientific and application-oriented research is quite contagious, and there were many industrial Q-meetings to prove it. I highly appreciate the freedom he has given me to explore novel ideas, even if the risks were often relatively high. Because of his efforts, I was able to travel abroad for four months and pursue a completely different topic of research. Moreover, it is his encouragement that has inspired me to continue an academic career after my PhD. It has been an absolute pleasure and privilege to work with him, and I thank him for his teaching, advice, guidance and inspiration.

At the same time, I am incredibly grateful to my co-promotor Dr. Matthew Peet. In 2018, when I contacted him for research collaboration, I was not expecting any reply; let alone an invitation to visit his group and work with him. With his support and steady pursuit, I was able to visit him at Arizona State University for four months and became part of a strong research collaboration. From the very first meeting, his constant guidance has been a catalyst in my research. Despite a considerable time difference, he has been genuinely accommodating to all of my meeting requests. He has always pushed me to become scientifically more rigour and precise. During the writing phase of this thesis, his eyes on the details have helped me to better shape my narrative, for example, how to precisely write a theorem statement or define a mathematical concept. I will carry these lessons with me for the rest of my academic career.

I am deeply indebted to Canon Production Printing for financially supporting my PhD research under the IMPULSE-II program. My heartfelt appreciation goes to dr. Amol Khalate, my daily supervisor. His invaluable advice, constructive criticism, willingness to brainstorm ideas, and incisive eye for finding practical relevance out of an idea have tremendously helped to shape my research. I would also like to acknowledge the guidance and advice I got during my collaboration with Bert van Beek, Mahnaz Shokpour Roudbari and Sjirk Koekebakker. I thank them all for keeping my project relevant to the industrial applications while giving me a complete freedom to pursue my own areas of interest.

I would like to express my most profound appreciation to my committee members prof. dr. Antonis Papachristodoulou, prof. dr. ir. Tamas Keviczky, prof.dr. ir. Herman Wijshoff, prof dr. ir. Maurice Heemels and dr. Sjirk Koekebakker. I thank them for making time to be in my thesis committee as well as read my thesis and approve it. Their comments and suggestions are highly appreciated.

Someone whose help cannot be overestimated is Sachin Shivakumar. I am quite fortunate to have such a talented researcher as a collaborator and dear friend of mine. I will dearly cherish our late night discussions, technical debates as well as sporadic jovial chit-chat. I wish him all the very best for his PhD. I am also grateful to Matthijs van Berkel for opening the door of nuclear fusion, a field of research that always fascinates me. He is a very close friend of mine, and I will treasure our dinner parties and after-work drinks in Van Moll. I would also like to offer my sincere thanks to my students Martijn Princen, Pradheep Shakthivel and Dat Hoang. They have been instrumental in conceptualizing theoretical ideas and translating them to exact specifications of the industrial partner. I would also like to thank Ricky van Kampen for starting a new collaboration on such a fantastic concept of parameter identification.

I also had the great pleasure of working with some of the leading scientists in my domain. I thank Dr. Chayan Bhawal from IIT Guwahati, Dr. Jan Heiland from Max Planck Institute, Dr. Laura Iapichino from CASA group, Dr. Felix Miranda from Inria, and prof. Rodolphe Sepulchre from the University of Cambridge for their guidance, constructive discussions and incurring new ideas.

It has been a great pleasure to be a part of CS group for the past four years. I would like to thank Paul and Mircea for providing crucial feedback and suggestions during the annual evaluations. Their formative assessment on my research themes has helped me to sharpen my thesis. I would also like to thank Paul for his professional advice that was helpful to find the next step in my academic career. I much appreciate the occasional feedback from Sofie on my presentations, technical writings and grant proposals. Discussions with Roland, Maarten and Lyla have always been illuminating. I also received generous technical support from Will from time to time.

My time in CS group would not be so enjoyable without the friendship of David. Since the very first day, I have appreciated our discussion on software design, (intense) critical arguments about our ideas and, more importantly, the regular coffee breaks near the fifth-floor balcony at Flux. I would also like to thank Karthik, Paul (Padilla) and Tom for a long-standing friendship, scientific debates,

brainstorming, afternoon coffee, after-work drinks and very delightful conferences in Cyprus and Nice. A special thanks go to Karthik's wife Adithi for her kind hospitality. Among others, Tomas and Fatima always have a special place in my heart and I am pleased to receive their assistance, kindness and care. I also have to mention Yanin, Bahadir and Mohsin for their advice and assistance during my first year as a PhD student. My office-mates were the best! Thanks to Zuan, Beppe, Daming, Feye, Henrik, and Tuan for making the environment fun and casual. The credit should also goes to all the other CS members, past and present, for keeping up such a high degree of professionalism in the group. The CS group would not be complete without Diana, Hiltje, and Lucia (and also late Barbara). I am grateful for their care and kindness and, of course, tolerating too many of my administrative requests which always come at the very last minute.

My friends, Suraj and Shivam, have brightened my life through laughter, funny jokes and childish nuisance. As a founder of our salon, *the Chicken Curry*, I take great joy in the moments we shared during the last six years. Thanks to them, weekends were never boring with good food and better company. I will dearly cherish our thrilling adventures during the trips in Italy, USA, Germany and Poland (I would not disclose the exact details, I promise). I cannot leave Eindhoven without mentioning my dear friends Vivian, Lu, Ashish, Parth, Ugo, Mike, Shah, Sunder, Chen, Bee, Wouter, Avinash, and so many others. During my visits to Paris, I very much appreciate the warm welcome and hospitality of Sophie, Florian, Frank, Juan and Chotu uncle. Brendon, Morgan, Shuangshuang, Sofía, Ian, Dharani have made my visit to the USA very memorable. A very special shout out to my friends from India; Apurva, Hithesh, Arun, Ramana. Amrith, Archishman, Sattwaki, Ayan, Arnab, Ritam, Madhurima, Rituparna, Ratul, Debashi-da, Gabla-da and Prometheus-da.

Finally, sincere gratitude goes to my family for their unparalleled love, support and encouragement. Mam and Babai, I thank you for sacrificing so much to make me who I am today. I am deeply indebted to Dadu and late Dadi for taking care of me from the very childhood. Your sense of values, life lessons and endless encouragements have helped me in every walk of my life. I am grateful to my four uncles and momoni (Boro, Sejo, Na and Choto) for their optimism, reassurance and kind hospitality. My sincere love goes to my grandparents and every family members in Shrirampur. I would also acknowledge my lovely cousins Bunu, Tunu, Binti, Muna, Guddu, Tattu, as well as boudi Rituparna and my nephew Gublu. And last but not least, special love goes to Gaurie, who will soon be a part of our big family.

Amritam
Valkenswaard, 02-10-2020

About the Author

Amritam Das was born in Shrirampur, India on 31st July 1992. After finishing his high school, he pursued his bachelor's degree in India in Mechatronics Engineering from 2010 till 2014. During his bachelor study, he was the valedictorian of his class and was awarded the gold medal. In 2014, with a full scholarship, he was admitted to the Eindhoven University of Technology for the master's program in Systems and Control. During his master's study, he was also awarded the Océ Merit Scholarship (awarded to 5 students across Europe). Since he finished the master's education in 2016, he has been working towards a Ph.D. degree at the Control Systems Group of the Electrical Engineering Department at the same university. His Ph.D. project was a part of the IMPULSE -II program, a joint consortium among the high-tech industries and the university, and is funded by Canon Production Printing.

In his doctoral research, Amritam has been developing computational tools to model and control thermo-fluidic processes. At the same time, he has been applying and experimentally validating these tools on a commercial inkjet printer in close collaboration with Canon Production Printing. During his Ph.D., he has been a visiting researcher at Arizona State University in the United States of America. Here, he has been collaborating with Dr. Matthew Peet to co-develop a computational framework and software tools for Partial Integral Equations, a novel representation of the thermo-fluidic processes. His doctoral research has led to a digital twin that describes the thermo-fluidic process and designs controller to improve its performance.

Since October 2020, he has been appointed as a research associate at the Engineering Department and the Sidney Sussex College in the University of Cambridge. His research interests are spatio-temporal systems, dissipation theory, convex optimization and nonlinear control.