

Exploring Microfluidic Design Automation: Thin-wall Membrane Regulator

by

Atulan Zaman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Sciences
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2017

© Atulan Zaman 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Microfluidics and lab-on-a-chip are a growing technology, influential in many areas of engineering. This project focuses on the necessity for better computer aided design tools for this area. Specifically, it focuses on the automated synthesis of T-junction components with thin-walled membranes for stability. A T-junction is a passive droplet generation component, common in microfluidics which suffers from behavior instability in highly integrated circuits with many components. One way of improving stability is using flexible membranes to mitigate pressure perturbations. This thesis describes the design process of such membranes so that a model can be used to synthesize stable T-junctions. The thesis also discusses Manifold, a software framework for automated synthesis of microfluidic circuits. This is the framework where the design process fits in. To compare the result of the software framework with the analytic model described, physical circuits were fabricated to validate the accuracy of the analytic model and the software. Besides the T-junction, another microfluidics component that was investigated was a “Capillary Electrophoresis Channel”. This component was also investigated with respect to automated synthesis and verification using the Manifold framework, and the details are discussed.

Acknowledgements

I would like to acknowledge the invaluable support and encouragement of my supervisor Derek Rayside without whom this thesis would not be possible.

I would like to especially thank Cody Chen for his continuous insight and expertise in area of microfluidics during the length of my Master's.

I would like to thank the collaborating researchers for this project Derek Wright, Carolyn Ren and Chris Backhouse for sharing their technical expertise in the areas of microfluidics and systems modelling.

I would like to thank Murphy Berzish for his contributions in the Manifold Software Framework, which were fundamental to the automated synthesis software. I acknowledge Asif Khan for being one the founding members of this project and making much contribution throughout his Master's degree. I would also like to acknowledge fellow co-op students Natascha Van Lieshout, Thomas Kennedy, Ming-Cee Yee, Chris Willar & Alex Willerth for their various contributions to the project.

Table of Contents

List of Figures	vii
1 Introduction	1
1.1 Contributions	3
1.1.1 Collective Contributions	3
1.1.2 Unpublished collective contributions	4
1.1.3 Independent Contributions	4
2 Motivation	7
2.1 Introducing Lab-on-a-chip	7
2.2 Related Works	9
2.3 Requirement for scale-up in complexity	11
3 Manifold Software Stack	13
3.1 Manifold toolchain overview	13
3.2 Architecture of the back-end	14
3.3 Input and output of the stack	15
4 T-Junction and Membrane	17
4.1 Parallel T-junction and its challenges	17
4.2 Solution for flow fluctuations	20
4.2.1 Regulator channels	20
4.2.2 Flexible membranes	21

5	Design Process	22
5.1	Hydraulic and electrical analogy	22
5.2	Lumped parameter model	24
5.3	Membrane design using capacitor	24
5.3.1	Basic schematic	25
5.4	Micro-regulating Membrane Design with Capacitor model	27
5.5	Design simulation using MapleSim	28
5.5.1	Simulation Results	31
5.6	Physical characterization of membranes	33
5.6.1	Hydraulic capacitance	34
5.7	Fabrication and Experiments	37
5.7.1	Soft Lithography	37
5.7.2	Fabricated Designs	38
6	Result and analysis	39
6.1	Reliability of membrane design	39
6.1.1	Comparison of simulation and measurements	39
6.2	Droplet data from experiments	41
7	Design limitations and caveats	43
7.1	Limitations in Software Framework	43
7.2	Limitations in Membrane Design	45
8	Other Related Work	46
8.1	Capillary electrophoresis	46
9	Future Work	49
10	Conclusion	51
	References	53

List of Figures

3.1	Manifold toolchain overview	14
3.2	Manifold back-end data abstractions	15
4.1	Formation of a bubble at a planar T-junction with height h . Liquid and gas, respectively, enter the main channel (width: w) and side channel (width: w_{in}) at rates of flow q_c and q_d . A formation cycle comprises two periods: a filling period in which the bubble fills-up the junction and grows to a size V_{fill} and a squeezing period in which the forming bubble is squeezed by the liquid until it pinches-off. [37].	18
4.2	Schematic of a 4-junction multi-way droplet generator circuit [6].	19
4.3	Schematic for placement of a regulator channel to stabilize flow.	21
5.1	Hagen-Poiseuille Equation	23
5.2	Hydraulic resistance in rectangular channels	23
5.3	Basic circuit schematic representation of a 2 way T-junction circuit	25
5.4	Concentration distribution of a droplet across channel cross sections, from finite element analysis	27
5.5	Optimal circuit schematic with voltage regulating capacitor	28
5.6	Current response with capacitor values for 10% variance in resistance.	32
5.7	Current response with capacitor values for 5% variance in resistance.	33
5.8	Current response with capacitor values for 2% variance in resistance.	33
6.1	Deformation of membrane with pressure	40

6.2	Young's modulus from experiments	41
6.3	A figure with two subfigures	42
6.4	Droplet velocity profile in 4 seconds	42
8.1	Schematic of an electrophoretic cross	47

Chapter 1

Introduction

Microfluidics and Lab-on-a-chip is a fast growing field that is influenced by contemporary design processes in mechanical, chemical and biomedical engineering. It involves micro-scale miniaturization of processes and assays involving fluids. Especially in bio-medicine, this technology has catalyzed some of the recent developments in point-of-care diagnostics, stem cells research, single cell analysis and genome sequencing. The earliest applications of microfluidics involved printing technologies such as the ink-jet printers.

Recent reviews [3][7] have discussed the scale-up in the increasing complexity and integration that microfluidic technologies need to achieve and some of its primary challenges. One of the challenges identified is the lack of computer-aided design automation tools for rapid prototyping, by reducing the number of design iterations. Like most highly integrated systems, a lab-on-a-chip device contains several coupling components that interact with each other in real-time to process an assay. The common design process involves making a model for each component in COMSOL or MATLAB and simulating each model separately with the *best guess* environmental parameters and limits. Therefore designing the integration between each component requires a high skill level, experience and many rounds of prototype fabrication. Usually, each round of prototyping is a time-consuming process, usually few weeks to a month long and is also expensive. The dream is that this process of design can be automated using CAD tools so that rapid-prototyping and hardware synthesis using software is possible.

This project addresses the computer-assisted engineering of a very specific microfluidic circuit. A T-junction is a droplet generating component for two-phase microfluidic circuits. While the design of a single T-junction is a well-understood design process versus that of a complex circuit with multiple T-junctions is a difficult task without automated design

tools. The source of instability in a multi-way T-junction is pressure perturbations caused by each individual droplet generator and how it influences the rest of the circuit because of their coupling. More on the exact nature of this is discussed in later sections. One of the viable design choices to minimize perturbations is using thin-walled membranes along the walls of the fluid channels to compensate for the temporary change in pressure. This happens because of the flexible nature of the membranes, where they expand and act as temporary reservoirs to counteract the pressure fluctuation during droplet formation.

This project primarily focuses on the design of circuits using these membranes to mitigate pressure perturbations. This both involves discussion of the analytical representation of its behavior mathematically, and also its characterization in the software framework, followed by how the two are combined to arrive at a viable solution. This is discussed primarily in chapters 4 and 5.

As mentioned above, the design process involved two separate components: *Analytical* and *Software Representation*. Formation of the analytical representation involved literature survey for appropriate analytical solutions to deformable solids that are pertinent to the context. This is necessary for obtaining the hardware fabrication parameters for the membrane and circuit. In traditional numerical simulation solutions, generally a Finite Element Analysis technique is applied using higher order integrals. Derivation of simpler analytical solutions involves making certain assumptions about the system and the exploration of such boundary conditions is also discussed in section 5. It is this mathematical simplification that allows the design process to be scaled up for the Manifold software framework.

The interface to the Manifold software framework [6] has a hardware description language that provides a library to describe microfluidic circuits using certain components. One of the components that are available in the microfluidics library is the T-junction. The back-end for the Manifold framework contains the analytical relationships describing the components and uses the hardware description provided by the designer to synthesize design parameters for the circuit using SMT-based solving techniques. The specific SMT solver used in the framework is called dReal [11]. SMT stands for Satisfiability Modulo Theories, which is an extension of the theories described by SAT solvers. After a design is synthesized by the SMT solver, a time domain simulation of the design is run using MapleSim to verify the behavior of the circuit. The architecture of the Manifold framework is described in detail in chapter 3.

In order to validate the designs synthesized by the CAD tool MapleSim, a set of experiments was conducted to validate certain claims. The first set of experiments was designed to check whether the analytical models used to represent the T-junction and thin-walled

membranes were following similar relationships as measurements made from prototype fabrications. In order to demonstrate this, a stress/strain curve for the thin-walled membrane was plotted for these three data sets and their correlation was presented. Secondly, experiments were performed to verify the validation of designs synthesized by the software. Physical fabrications were created using the soft lithography technique for designs produced by the synthesis software. For the T-junction circuit, measurements were made to ensure the proposed membranes increased droplet size stability.

Finally, some work related to the broader manifold project, but unrelated to a T-Junction, is presented in chapter 8. This chapter provides details for experiments to automate the design and verification for a capillary electrophoresis channel. Also, in chapter 7 the design limitations and caveats for the design process involving the Manifold framework are provided.

1.1 Contributions

There are three distinct sets of contributions worth highlighting in this document. Manifold is a fairly large project with many aspects in development concurrently. Two main distinct topics being the software framework itself and the microfluidic component designs, which my thesis is primarily focused on.

1.1.1 Collective Contributions

In the first aspect, while the software framework itself was not the focus of my thesis, I was part of the collective contribution from the group, during my graduate studies. That work has been published in conference proceedings cited in [19], [26]. The relation between the software and the component design process lies in aspects such as:

- How can the components be characterized in the Front-End Language?
- Integrating the process of simulation and physics models into the software in a collaborative manner.
- Canvassing the evolution of the software framework to support features and components of microfluidic circuits.

I have also been the presenter for [19] in CASCON '16 as one of the authors on behalf of the group.

1.1.2 Unpublished collective contributions

Some of these novel ideas appear here for the first time, as my work is about analyzing them.

Parallel T-junction topology One of the prime objectives of the near future is to facilitate microfluidic designers to build parallel T-junctions using the Manifold framework, which is currently intractable using traditional design process. As part of this exploration, we find multiple topologies in which the parallel T-junctions can be constructed in a scalable manner. While this is not discussed in much detail in my thesis, this is one of the contributions of my research achieved collaboratively in the process of building analytical models.

A new way of using membranes Using flexible membranes to stabilize microfluidics circuits is not a new concept. As discussed in chapter 5 there are multiple literature studies exploring this concept. This novelty in our work is that we explore a discrete, scalable way of using multiple membranes in adjacency to achieve a certain level of capacitance produced by our analytical models. Another novelty of our design approach is that we explore placing membranes at the side of the channel rather than on the floor or ceiling of the channels. This facilitates measurements during experiments using commonly available non-confocal microscopes and also designs that are easier to fabricate using soft-lithography.

Designing the capillary electrophoresis cross In collaboration with Dr. Chris Backhouse we worked on design verification for the capillary electrophoresis cross. This was performed as an exploration for how design automation can be achieved in the single phase microfluidics domain which is different from the multi-phase domain involving droplets. This is discussed in detail in chapter 8. I was part of the manifold team in designing how the physics of the capillary electrophoresis cross can be characterized in the SMT language as well as constructing models in COMSOL to better understand its functional behavior.

1.1.3 Independent Contributions

The specific contributions I can independently claim credits for are the following:

Refuting regulator channels as a design solution for multi-way T-junctions This involves validating intuitive solutions in the fluidic domain using the electrical domain, to support or refute hypotheses. For example, one of the initial ideas for stabilizing the T-junctions was the usage of regulator channels between the channels carrying droplets to buffer pressure perturbations. We characterized this in the electrical domain by having parallel resistors between the channels. From our simulations, we observed that this solution does not work in electrical theory. Since this proposal did not work in the electrical theory it is reasonable to believe it would definitely not work in the fluidic theory. This is discussed in greater detail in chapter 4.

Constructing Analytical Models in MapleSim As discussed earlier, it was important for the automation framework to have analytical models of component behavior in order to be able to synthesize designs. Numerical solutions are too computationally expensive for the software to support scalability. Creating analytical models of microfluidic components is a challenging opportunity. One of the biggest challenges is the translation of domain abstraction. It is not quite obvious how the specific designs translate from the electrical domain to the fluidic domain, although there is much research proving their analogy. For example, how can capacitance in the electrical domain be represented by flexible membranes of specific dimensions? This has been explored in this thesis.

Another challenge in analytical model construction is creating the models in simulation software. Once the mapping for concepts between domains has been established, for example, channels in fluidics maps to electrical resistance in the electrical domain, the models are constructed in MapleSim to verify the accuracy of this representation. Much of this was done in collaboration with Derek Wright, one of the Faculty collaborators for this project.

Constructing analytical models essentially involves making reasonable assumptions to simplify a more complex and accurate model, that is a good representation of a system within certain limits. Formalizing these assumptions is an iterative process. This thesis discusses the assumptions used for representing a T-junction and membrane, why some of those assumptions are reasonable, and why some are naive in a reflection of the experimental results.

This work also explores the analytical representation of a thin-walled membrane, relating concepts of electrical capacitance, hydraulic capacitance and plate theory of mechanical deformation. The significance of these analytical models is that once adequately defined, they can be used as recipes by other researchers in their own applications.

Representing components in SMT solver for synthesis Another step in the translation of domains is in representing the analytical models for SMT solvers. Once the physical concepts have been represented in analytical models, they have to be translated to an abstraction that is comprehensible for the SMT solver for synthesis. This is often tricky because the MapleSim model captures some aspects of the design that the solver does not. Fundamentally, modeling software represents a design as a graph of nodes, related to boundary conditions that define their relationships. SMT solvers represent models as a set of constraints using variable declarations, that characterize their relationships. This work explores the representation of a T-junction circuit in the SMT solver domain. Much of this was done in collaboration with Murphy Berzish, one of the Ph.D. collaborators for this project and a leading member of the software framework.

Experimental result analysis Based on the analytical models, physical experiments are performed followed by fabrication. In this thesis, I discuss the analysis of the results to validate the completeness of our analytical models. Much of this was done in collaboration with Alex Willerth, one of the undergraduate research assistants for this project in his co-op.

Chapter 2

Motivation

This chapter discusses some motivations for this project with respect to existing work and some sample applications. It starts by providing some introduction into the area of *Lab-on-a-chip* design. This is primarily written for readers not familiar with the technology, and provides a very high level, slightly motivating representation of that space. Next, it discusses aspects of design automation in the LOAC design space. It discusses common existing practices, what is missing from those and proposes a better design methodology, that makes comparisons with more “mature” design spaces. Finally, it arrives at the focus of this project and discusses why the requirement for scale-up is necessary for this domain, and how this project relates to this objective.

2.1 Introducing Lab-on-a-chip

Microfluidics, Lab-on-a-Chip, MEMS are different names for a similar technology in different fields. The core element of this technology revolves around using fluid dynamics at a micro-scale to achieve different engineering functions. A more intuitive understanding of the scale of this technology can be perceived by imagining the size of a speck of dust. Technology smaller than a speck of dust can be considered in the *nano* scale of technology. Most micro-channels in these chips range in thickness from 50 to 100 μm , which is the same average thickness for a strand of human hair. Fluid is pumped through these micro-channels into chambers/components that have functions and usually some output is measured experimentally. One fitting analogy is the one it draws with electrical circuits. They are similar to electrical circuits in the way that electrical circuits carry electrons

where they carry liquid molecules. Instead of conducting wires, it has micro-channels. It even follows similar laws of Physics as *Ohm's law* but more on that is discussed later.

Currently, the fields of engineering and science most involved with this technology are chemical engineering, biomedical engineering, pharmaceutical research, mechanical engineering and printing devices, genome research etc. In mechanical engineering it is most often called MEMS, while in other fields, microfluidics and LOAC are coined interchangeably. These two journal entries from nature illustrate some of the early motivations of this technology [31, 39]. Some areas of this technologies are Droplet microfluidics, Drug discovery, personalized drug design, pharmaceuticals, genome sequencing, organ-on-a-chip. Organ-on-a-chip technology is a revolutionary new technology that has been accelerating in the last 5 years or so that promises to replace drug testing in animals. This happens by creating microfluidic components that simulates the environments of human organs in-vitro, and more accurately resembles the reactions of human organs to different drug environments.

The growing popularity of LOAC technology has naturally attracted professionals from different fields to become interested in its design because of its versatility. Along with popularity, the demand for more complex and integrated devices has emerged. Due to the heterogeneity of its stakeholders the need for better CAD tools has emerged, to perform simulation, prototyping and others functions in the design stage.

The physics involved in some of these designs are quite complex. Unlike electrical circuits, where the internal coupling effect of circuit elements is negligible, in microfluidic circuits the mechanics of one element in a circuit might influence the behavior of an element in a different part of the circuit. This is mostly governed by the law of fluid dynamics, which is very expensive to simulate using computational fluid dynamics software available. Without simulation and support of CAD tools, it difficult to design these devices with confidence, even after prototyping.

One such device is the parallel T-junction network [13, 14]. It is used to create a multitude of droplets simultaneously across parallel channels in a network. However even a relatively simple design by intuition becomes quite complex with increasing number of parallel channels and the uniformity of its behavior is compromised due to internal coupling effects. It is possible to regulate these effects, but it is a challenging task to design these without the proper tool, and sometimes even impossible. A similar difficulty of scale-up is prevalent.

2.2 Related Works

The related works here mention some of the design automation projects that has been done in the context of microfluidics.

Chakrabarty et al. describes the overall process of modern microfluidics design and underscores the need for better design automation tools for the microfluidic design space [8]. The papers also present some existing work on design automation of microfluidic devices. Araci & Brisk [3] do a similar survey of design automation tools for the microfluidic design spaces. Their survey was more centered towards design synthesis solutions for placement and routing of microfluidics software, analogous to how placing and routing are done for digital electronics. The difference between the design space mentioned in their works and this work is that for placing and routing, there is no consideration of the physics involved in the component design. Therefore a combinatorial approach can be taken into the placement of components and channels based on a discrete grid. However, the software framework involved in this report is more centered on individual components, the topology, and functional behavior. There is potential for both forms of synthesis to be complementary to each other for comprehensive solutions for microfluidics design synthesis.

The works of Amin et al. [2] describes a well-cited work in microfluidics, which introduces a solution named AquaCore. AquaCore provides a platform for general purpose, programmable microfluidic devices that can be programmed to achieve a wide array of similar tasks. With the design and hardware already created, the use case for designers is that they can program a device compatible with AquaCore to create an assay based on the written script. It is analogous to an FPGA in the digital world, where there is a programmable chip that is feasible for an array of use cases. The synthesis framework that this project is a subject of provides a more general solution of hardware description. Fundamentally, AquaCore does not do hardware synthesis, which the manifold framework aims to achieve.

McDaniel et al. [22] introduce a hardware synthesis framework and domain specific language called MHDL. As indicated by the name of the language, MHDL is actually similar to the popular hardware description language called VHDL. MHDL accepts hardware specification of microfluidic devices and produces design and layout of the circuit. However there is a key difference between MHDL and the Manifold framework. The MHDL framework accepts a more detail specification of the circuit and does not consider the physical properties relevant to a design. It only solves layout and placement problems with respect to behavioral properties specified for the design. Manifold accepts a more functional specification from the user and solves for physical parameters of the system by characterizing

governing equations defining the system. Since it uses an SMT solver to find a solution, designs generated by the Manifold framework are guaranteed to respect the user requirements which are part of the synthesis process.

What’s missing?

Other related works for design automation also exist [1, 5, 15, 24, 32, 34]. However, as investigated by Asif Khan [17] in his Master’s thesis, there aren’t too many design automation tools that characterize the true challenges in LOAC design with respect to complex fluid dynamics phenomenon. Most of these tools address a very niche area of microfluidic design where the coupling effects of fluid dynamics are either minimum or simply ignored for the sake of engineering simplicity with a compromise for accuracy. This project attempts to fill that void by creating a framework using basic fluidic phenomena using analytical equations, for a limited and discretized space but guaranteeing “safety”. More on safety parameters is discussed in the chapter discussing the software framework.

“Synthesis, Simulation & Verification”

Besides design synthesis, there are other steps in the design process that are also of importance. One of them is design simulation. In conventional practice, simulations were done through numeric computation models such as COMSOL run a model simulation using Finite Element Analysis. However, that doesn’t allow the designer to functionally simulate, and monitor specific parameters that are of interest. For example, when a designer is simulating the design for a droplet generator the parameter that is of interest is droplet volume for each droplet created in a time-step simulation, not so much the concentration of different liquids across the cross sections. However only the latter is available using numeric simulation, and the desired parameter must be computationally derived from the available data which often requires skills in a different specialization that the designer might not possess. Therefore, a better simulation engine for LOAC design is highly necessary. This project does not yet aspire to provide a simulation engine, however it does utilize the strength of MapleSim’s simulation engine to run functional simulations to synthesize a final design.

Another crucial feature that is missing among the LOAC design space is support for design verification. This in concept, would be something similar to formal methods and model checking in the context of software engineering. This would involve a design specification that the designer can create and a separate description of the operational and safety parameters that the system must satisfy. The function of a verification tool would be to analyze and predict whether the design satisfies all the constraints of design requirements. Obviously, the reliability of this exercise will depend on the comprehensiveness of the constraints provided. This is in fact, a major challenge in the microfluidics design space. For example, there could be operational parameters for environment variables such as humid-

ity. During the design process, the design might naively not consider all the corner cases, and create a design that operates only within certain humidity ranges. The function of the verification tool would be to provide a *counter-example* of a scenario, where the design would not work. In this case, this would be to provide a humidity parameter, which is very much possible, where the design might not work. This is one of the main challenges behind the design of capillary electrophoresis, discussed in chapter 8.

From VLSI to Microfluidics

The aspiration is to create a toolchain that can have a similar influence and workflow as the ones that VHDL and Verilog provided for VLSI design, along with their comprehensive set of simulation software [23]. Development of mature CAD software has allowed the semiconductor industry to reach its peak of production in the last few decades. However, for LOAC design any such influential design tool is yet to arrive, it is reasonable to believe that such functionalities will have a profound impact on the growth of this technology space.

2.3 Requirement for scale-up in complexity

In this section, some example design and their challenges are discussed. These specific designs were inspired by collaborators for this project, Carolyn Ren, and Chris Backhouse and were considered in depth during the research and exploration phase.

The parallel T-junction

The T-junction and the parallel T-junction is discussed in much detail in chapter 4. The T-junction is a widely used component in microfluidic circuits for passively generating droplets. A single T-junction is relatively easy to design. Sometimes designers require a high-frequency droplet generator, where one T-junction is inadequate. This requires multiple T-junctions to be connected to a single input so that multiple channels can simultaneously create droplets. One of the important design specification of droplet creation is the droplet specifications. The droplet specifications such as volume, geometry, and frequency must be kept constant during operation of the T-junction. However, when multiple T-junctions are connected in parallel, internal coupling effects influence the T-junctions, and uniformity of operation is broken. In order to solve this problem, regulatory mechanisms can be used, but it is difficult for a human to design this. It is however, easier for a Computer Aided Design tool that uses, for example, an optimization technique or a combinatorial technique (like the one proposed in here) to find a solution fairly easily. This is where a design tool would be useful in the design of a T-junction network.

Capillary Electrophoresis

The capillary electrophoresis design is discussed in greater detail in chapter 8. A capillary electrophoresis technique is used to separate different components of a liquid assay, by applying a strong electromagnetic field across the channel. A subject component in the assay can be tagged using a fluorescent element, and after separation the concentration profile is extrapolated using a fluorescence detection microscope. The challenge with the capillary electrophoresis chamber is design validation and verification. Since it is a standalone system, there is not much challenge in synthesis. Since the system is very sensitive to environment parameters and discrepancies of any sort from an ideal scenario, it is very hard to predict how a designed chip is going to behave under certain operating parameters. The desired use case here would be to have a separate design description and safety parameters. After a designer creates a design for a certain assay, it can be validated using the verification tool to ensure it can operate safely across a range of environmental and external parameters. Such a tool is yet to exist. How this problem was explored for this project is discussed in more detail in chapter 8

Chapter 3

Manifold Software Stack

This chapter discussed the software framework of *Manifold*. Manifold is a hardware description language with a back-end for hardware design synthesis. The first section discusses the overall toolchain for the compiler framework. This describes the different language abstractions in the process of compilation. Section 3.2 discusses the software backend in greater detail. In particular, the backend itself has multiple abstractions that are relevant during the process of synthesizing designs. The rationale behind those abstractions and the tools that use those abstractions are illustrated. It both provides some background on the two contexts and also the relationship between the two. Finally, in section 3.3 an example input and output for the software is presented, provide an understanding of current state of the software and its purpose.

3.1 Manifold toolchain overview

The Manifold framework is a domain agnostic hardware design synthesis language. The figure describing the toolchain overview of Manifold is shown in figure 3.1. The interface for the designer is the Manifold hardware description language. The hardware description language supports multiple domains such as “microfluidics”, and “digital hardware”. The domain specific knowledge of the components that the designer wants to use is encapsulated in domain-specific libraries. These libraries are imported by the designer, in the hardware design description written in Manifold. The language syntax, semantics and features are discussed in [6]. Further details are also discussed in conference proceedings [19], [26].

After the hardware description is written in the Manifold language, the “front-end” module parses and compiles the user input file. The front-end parses the design and

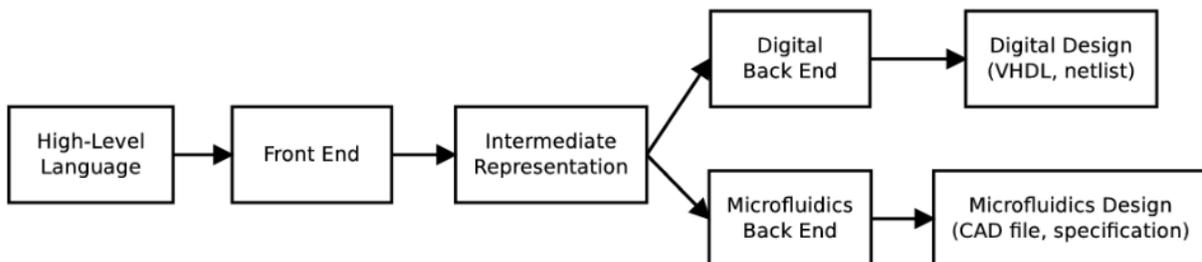


Figure 3.1: Manifold toolchain overview

creates a schematic in an intermediate representation. This intermediate representation has a well-formed structure that is recognizable by the back-end of the framework. The back-end of the framework has mappings of the components used in the design, with the design details, and mathematical relationships of the specific components. For example, for a T-junction component, the back-end would have the mechanics for periodic droplet formation for two-phase flow. The back-end uses the schematic created by the front-end to drive the steps for synthesis and simulation, using the component descriptions defined in the domain-specific libraries.

3.2 Architecture of the back-end

The back-end has three main modules: The compiler, dReal, and MapleSim simulator. The three modules and the abstractions they represent are shown in figure 3.2.

Part of what the back-end accomplishes using the schematic produced by the front-end is to infer which parameter of certain components are provided by the user, and which ones need to be inferred by the synthesis engine. For example, while defining a droplet generating T-junction system, the designer might specify certain requirements such as required droplet volume, input fluid pressure, and the number of T-junctions but not such details as channel dimensions and T-junction geometry. Such parameters need to be *inferred* by the synthesis framework. That is primarily the function of dReal.

A complete schematic is created by the compiler, and a dReal model is created in a *smt2* format in order to solve the equations to find values for the parameters that need to be inferred. The dReal representation of the model is somewhat minimal. It contains the declaration of the required variable, physical constraints characterizing some of the parameters and as well the model itself. It also has the analytical equations that characterize the

physical behavior of the components. These relationships are used to solve for the unknown variables in the model, and dReal returns an assignment for all the variables that satisfy the model. dReal is an SMT solver, that uses boolean satisfiability to find a satisfying assignment of a variable for a model described in analytical equations and ODEs. It is to be noted that in the description for dReal, design details such as layout and physical topology are not necessary unless it is relevant to solving an unknown parameter.

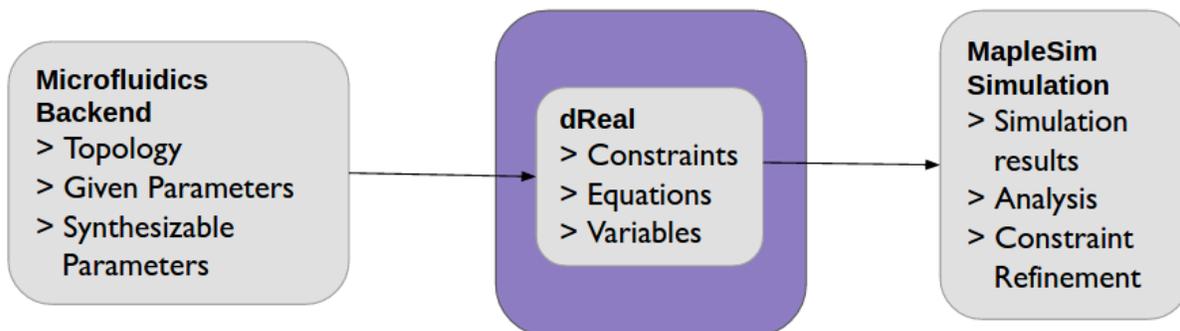


Figure 3.2: Manifold back-end data abstractions

The output for dReal is used to create a complete description of the design with all relevant parameters defined. This complete description is used to run a time domain simulation in MapleSim to verify that certain design requirements are met. The corresponding MapleSim models are also part of the description of the library of components. The MapleSim description of a model is similar in topology to the front-end abstraction of the model in the notion that they both follow a graph structure, consisting of components, connections, and ports that match the boundary condition between components. The MapleSim model is used to run a time-domain simulation of the system to check that output of dReal indeed produces a design that meets the requirement criteria. Figure 3.2 shows the data abstractions in the different modules on the backend.

3.3 Input and output of the stack

In summary, the input to the framework is a hardware description written in the Manifold language. This description is written in a functional approach, and at a very high level, without describing the behavior of certain components. After processing by the front-end, the back-end runs a synthesis step using dReal, and a simulation step using MapleSim to

verify the synthesized design. The final output is a list of parameters with explicit values that are inferred through synthesis and these parameters can be used by the designer to fabricate a physical design.

Chapter 4

T-Junction and Membrane

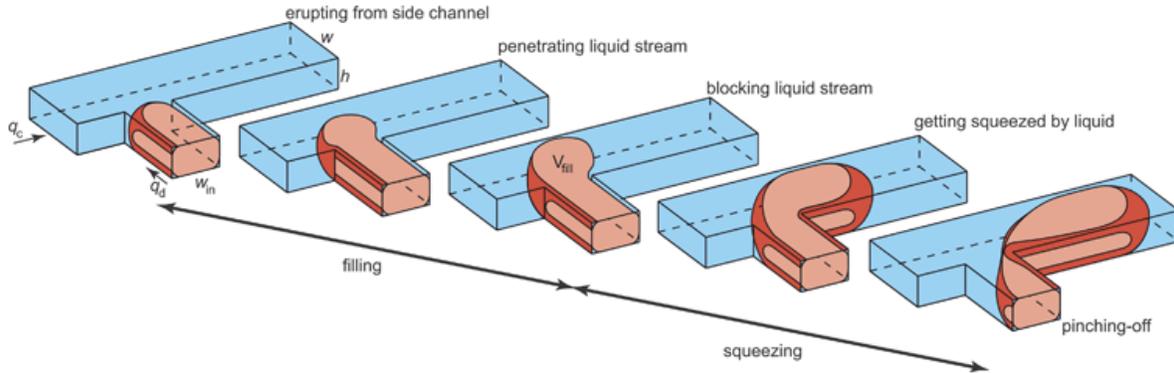
4.1 Parallel T-junction and its challenges

One of the fundamental elements in droplet microfluidic circuit is a droplet generator. There are multiple mechanisms of generating droplets in microfluidic circuits, which can be broadly categorized as either passive systems, such as T-junctions, or active systems that use sensors and actuators to regulate flow [43]. The advantage of active systems to regulate droplet generation is that it allows very precise control of parameters influencing the mechanism of droplet formation, such as volumetric flow-rate and pressure. This is done through pressure sensing in the channels or image processing to analyze droplet frequency at the junctions and influencing an actuator in a feedback loop. One drawback of the active systems is that they are expensive to maintain and operate and more importantly, they limit the scalability and size of microfluidic circuits for production and fabrication.

Passive droplet generation techniques, on the other hand, are much easier to fabricate and integrate into a scalable microfluidic circuit, however, they are imperfect with respect to design requirements in a higher level of integration such as microfluidic circuits and networks. The source of variance in desired behavior is broadly from different categories of concerns. One of the categories is from manufacturing defects in fabrication, for example where not all channels are created in equivalent dimensions or slight variation in pressure from a pressure flow source such as a pump. Another category of variance is the behavioral influence of events as the circuit starts operating, for example, the formation of droplets resulting in pressure perturbations that influence the behavior of the rest of the circuit [13]. Our work is concerned with passive droplet generation techniques, particularly the T-junctions.

A T-junction is a junction of two separate flow channels of mutually immiscible fluids that form droplets at the junction when the two fluids collide. The primary requirements of a T-junction are droplet volume and frequency which can be defined by predictive models based on device geometry parameters as well as functional parameters such as flow rate, pressure and fluid properties[10, 12, 37]. A two-phase drop generating T-junction is shown in figure 4.1

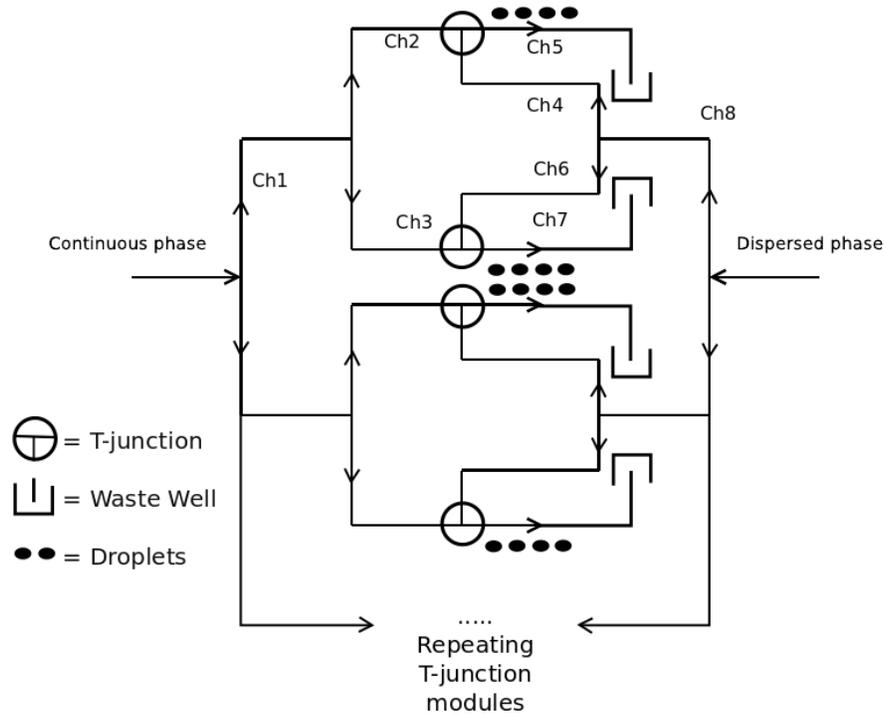
Figure 4.1: Formation of a bubble at a planar T-junction with height h . Liquid and gas, respectively, enter the main channel (width: w) and side channel (width: w_{in}) at rates of flow q_c and q_d . A formation cycle comprises two periods: a filling period in which the bubble fills-up the junction and grows to a size V_{fill} and a squeezing period in which the forming bubble is squeezed by the liquid until it pinches-off. [37].



It is desirable to create a network of parallel droplet generators for various applications. For example, production of droplets in multiple channels from a single source allow testing of multiple of drug designs with respect to reagents in parallel. It is also possible to have a network of droplet generators with each channel being of different dimensions to create droplets of different sizes in each channel. This is a more challenging design requirement, and our work only focuses on creating uniform droplets in all channels of similar dimensions. There are many possible topologies for creating parallel T-junctions with limited flow sources. For this work, we assume the requirement of having a $2 : n$ network where 2 distinct flow sources of immiscible fluids create a network of n droplet channels. [6]

The topology of a parallel T-junction circuit can be generalized in multiple ways. The topology explored in this work is shown in figure 4.2. The topology of the circuit is analogous to an overlapping balanced binary tree, with respect to a single input source for each of the continuous phase and the disperse phase.

Figure 4.2: Schematic of a 4-junction multi-way droplet generator circuit [6].



The primary requirements for a T-junction network are uniform droplet size and volume, and droplet frequency across all droplet channels. Under the assumption of acceptable tolerance of manufacturing error, the regulation of this requirement can be achieved by a constant flow rate from the two flow channels entering each T-junction. However, this is difficult to achieve. When droplets start forming at the T-junction, the resistance across droplet channel changes due to the formation of droplets which creates additional hydraulic resistance. Moreover, the rate at which droplets are lost at the waste wells are often not synchronized with the rate of formation of droplets at the junction, therefore, resulting in a variation of the number of droplets present inside the channel with respect to time. This change in resistance due to the droplet formation in each droplet channel results in a pressure perturbation which influences the input flow rate for each junction, resulting in droplets of non-uniform dimensions forming in each droplet channel, which breaks design requirements [13, 36].

Even in state of the art design and modeling practices in microfluidic device design, it is intractable to create reliable designs with multiple parallel T-junctions using mostly

numerical simulations methods and best practices from empirical experimentation. In a numeric simulation, only limited modules can be simulated at a time, and not the full system because of the complexity and computation time. The reason empirical experimentation falls short is because of, what computer scientists would say, lack test coverage. With experimentation, only certain scenarios of a design can be tested and guarantees cannot be made of robustness across a variety of operating environments. Our work aims to facilitate this process of making parallel T-junctions of arbitrary branches, by using simplified models to simulate designs and using a software synthesis framework to synthesize hardware designs based on requirement specification [6]. This report is focused on modeling of the parallel T-junction system using simplified lumped parameter models using MapleSim, and discussing the simplifications made and their limitations.

4.2 Solution for flow fluctuations

There has been many proposed solution in research and practice for stabilizing flow in a microfluidic circuit. In broad consideration, they fall into two categories: *passive regulation* and *active regulation*. Active regulation includes sensors and actuators to actively instrument relevant flow parameters such as flow rate, pressure, droplet volume etc, and influence stability by using intricate actuators to controlling those parameters [16, 44]. Passive techniques include mechanisms of controlling without additional active circuitry. This often results in simpler solutions, however, provide less control to the designer. Passive solutions are also favored in large-scale production of microfluidic devices when the components need to be used ubiquitously in devices of various applications.

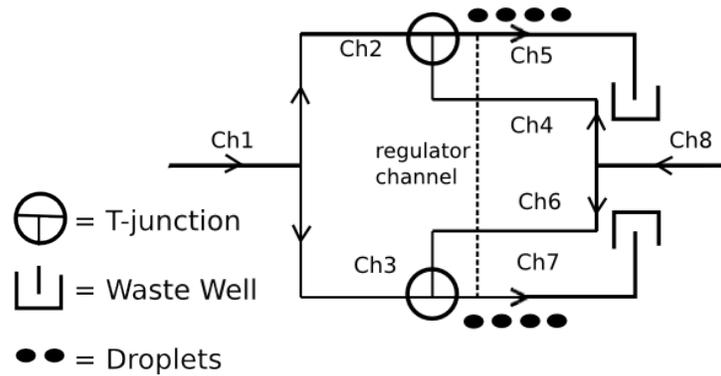
For design synthesis passive solution are favorable to active solutions, because they are easier to design and hence easier for the automated design tool to synthesize. Therefore with synthesis in mind, two different passive regulation techniques were explored: regulator channels and flexible membranes.

4.2.1 Regulator channels

One of the considerations for regulation was to add secondary regulator channels connecting the channels that carry the droplets in a T-junction circuit. The intuition behind this consideration is that since the irregularities in droplet generation are created due to pressure perturbations that create an imbalance in pressure, the regulator channels would work as load balancing channels by transferring some of the continuous phase liquid from one

droplet carrying channel to another whenever there is a perturbation causing an imbalance. In the regulator channels, only the continuous phase liquid would be allowed to flow, and droplets would not be allowed. This can be achievable by fabricating micro-gratings at the junctions that would not allow a droplet to pass through, because they are too large, and only allow liquid to pass through.

Figure 4.3: Schematic for placement of a regulator channel to stabilize flow.



4.2.2 Flexible membranes

One method of passively regulating the pressure perturbations is by using a thin PDMS diaphragm that can deform according to the changing pressure to buffer the extra volume of fluid due to the perturbation in pressure, resulting from the changing resistance in the droplet channel from the formation of droplets [18, 28, 33]. This solution is analogous to the behavior of a capacitor in the electrical circuit domain, which fits perfectly to our model of simulation. The symmetry in this domain is expressed in the fact that a capacitor buffers potential difference to resist alternating voltage when connected in parallel, similar to the pressure regulating diaphragm that buffers for changes in fluid pressure, by resisting changes in pressure. The design of these diaphragms is discussed in further detail in section 5.4.

Chapter 5

Design Process

5.1 Hydraulic and electrical analogy

Mathematically, the most precise characterization of computational fluid dynamics is expressed by a set of Navier-Stokes equations. The Navier-Stokes equations are derived from the Cauchy momentum equation describing non-relativistic momentum transport in any medium. This is expressed in a series of partial differential equations which are functions of physical properties of fluids. In order to accurately model computational fluid dynamics, appropriate Navier-Stokes equations need to be solved. This is an intractable problem even for computer modeling software, and more importantly, it is not proven that in the three-dimensional space, a solution always exists to the equations. Most numerical computation tools, such as COMSOL, use some simplification of these equations to run simulations over time.

In the traditional methodology of designing a network of microfluidic circuits, it is common to use numerical computation tools, that use physics of computational fluid dynamics to perform finite element method analysis to generate exact solutions. This method is accurate for modeling purposes with respect to the physics characterization. However, computational fluid dynamics in the context of finite element analysis is a very expensive computational task to perform by tools such as COMSOL, which takes very long to simulate when modeling network of fluid components which are temporally dependent on each other. This makes that approach unscalable. Our approach uses lumped parameter modeling techniques, based on the analogous behavior of microfluidic system compared to electrical systems to create a scalable model for simulating a microfluidic circuit.

The works of Kwang et.al and Quero et. al explore in detail, design dimensions of modeling a fluidics network as an electrical circuit [27, 29], in particular, its limitations and benefits. This form of modeling can only be applied to laminar, incompressible and viscous flow, which satisfies the context of T-junction modeling. The basis of this translation of physical domains is found in the conceptual symmetry between Hagen-Poiseuille equation of fluidic mechanics and Ohm’s law of electrical circuits. Hagen-Poiseuille equation is a general derivation from the Navier-Stokes equation, describing pressure flow in laminar incompressible fluids while Ohm’s law defines the relationship between Voltage, Current and Resistance in electrical circuits. The Hagen Poiseuille equation is presented in figure 5.1, where ΔP represents pressure drop, L represents the length of a channel, Q represents volumetric flow rate, and r represents radius.

Figure 5.1: Hagen-Poiseuille Equation

$$\Delta P = \frac{8\mu LQ}{\pi r^4} \quad (5.1)$$

The original Hagen-Poiseuille equation describes flow in channels for circular cross sections. A derivation of that is used to characterize channel resistance in rectangular channels, as shown in figure 5.2. In this equation, μ represents viscosity of the fluid, w represents the width of a channel, L represents the length of a channel and h represents the height of a channel.

Figure 5.2: Hydraulic resistance is rectangular channels

$$R_h = \frac{12\mu L}{wh^3(1 - 0.630h/w)} \quad (5.2)$$

In the hydraulic analogy of electrical circuit, *pressure drop* in fluid is analogous to *voltage*, *channel resistance* is analogous to *electrical resistance* and *volumetric flow rate* is analogous to *electrical current*. This intuition follows over to higher order analog circuits as well, where *capacitance* in the electrical circuit domain can be represented with *flexible membranes* in the fluid domain, that follows laws on Young’s modulus. This forms the basis of our analysis. Surely, there are limits to this translation of domain. Most notably, in the electrical domain, there is just one form of flow in the form of electrons, however, in the fluidics domain, it is common to have different fluids interacting with each other in a component which cannot be characterized accurately. Also, in the fluidics domain, it is common to have velocity profile distributions in a channel cross sections, which often

influences physical behavior. This is also not characterized in the electrical domain. The T-junctions, which is the device demonstrated in our study, is not influenced majorly by velocity profiles. The modeling of the interaction of the multiphase immiscible fluids is an important aspect of our lumped parameter modeling technique described in section 4.

5.2 Lumped parameter model

Lumped parameter modeling is a simplification of spatially distributed characterization of physical models, possible using assumptions of uniformity upon parameters and discretization of continuous functions [41]. In general, lumped parameter models express system described in partial differential equations or higher order integrals in a simplified form, using ordinary differential equations based on assumptions. A simple example of a lumped parameter model of a higher order physical system is how Ohm's law describes the behavior of electrical circuits from its distributed generalization in the form of Maxwell-Faraday's equations describing the change in potential as a function of the change in magnetic flux. This is precisely how lumped parameter modeling defers from numerical computation methods such as Finite Element Analysis where the physical behavior is analyzed across a granular mesh of discrete points rather than for the system as a whole, which is the case for lumped parameter modeling. For our experiments, we have used the symbolic computation engines of Maple and MapleSim [20, 21] to model our discrete model of the fluid network using an electrical circuit, with specifically customized components to generalize fluidic behavior such as droplet formation. These custom components and generalizations are described in more detail in the next section. The models for MapleSim are written in the modeling language Modelica [25]. It is a component-oriented modeling language for describing physical systems, with supporting libraries for several design domains such as fluid mechanics, analog or digital electronic circuits. It is a supported front-end language for many popular computer aided design frameworks such as MATLAB, MapleSim, Dymola etc, primarily for describing Lumped parameter models. In this context, MapleSim was the CAD tool that was supported.

5.3 Membrane design using capacitor

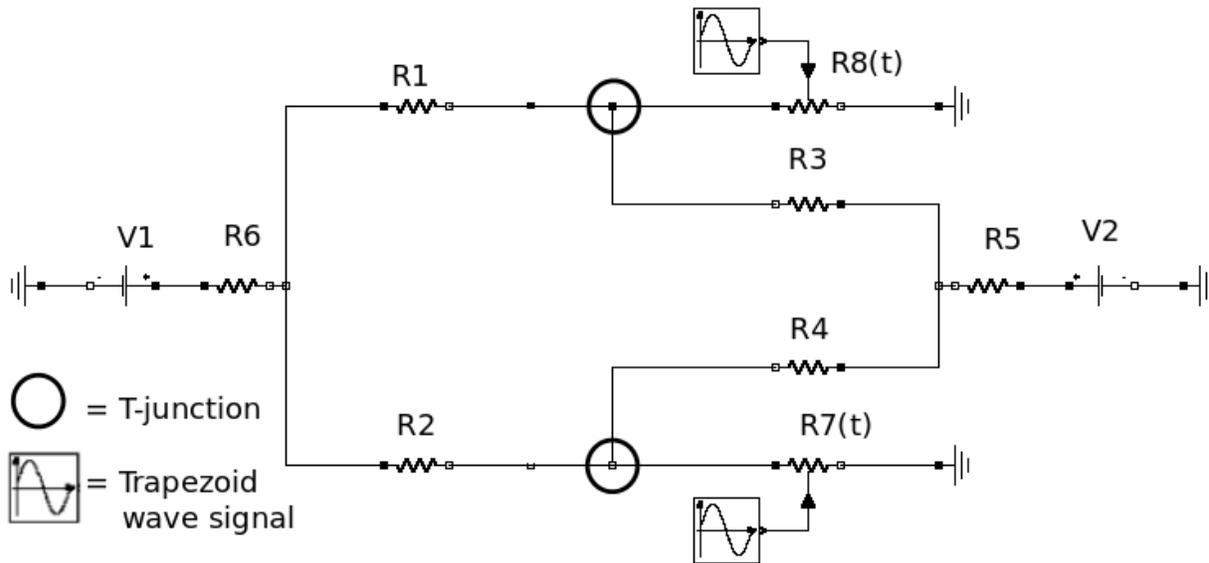
This section describes the domain translation strategies in modeling a microfluidics circuit as an electrical circuit. The basic schematic of the circuit is first introduced along with justification for using some of the basic components such as simulating droplet formation and

behavior. Following that, the idea of using capacitors as analogies for pressure regulating diaphragms is explored, as well as the different placement strategies explored during the research. Finally, it presents design specifications for the circuit used to model a physical T-junction system.

5.3.1 Basic schematic

The analogous circuit schematic for the T-junction network is a design with a very similar topology as the microfluidic circuit. The two separate input sources for the different fluids is expressed using two different voltage sources. The channel resistance of each section of the channels is expressed using discrete resistances connected in series or parallel. This is an example of the discretization aspect of the lumped parameter modeling technique. The different properties of the two fluids are characterized in the discrete resistors for each channel section using equation 5.2 which is a function of both the geometry of the physical channels as well the fluid flowing inside the channels. The waste well for the fluids is characterized as an electrical ground. The circuit schematic of a basic parallel T-junction circuit with 2 inputs and 2 droplet channels is shown in figure 5.3. This basic model of the schematic does not include the regulating capacitors.

Figure 5.3: Basic circuit schematic representation of a 2 way T-junction circuit



One design choice made for the models used for simulations presented in this report is

that for flow input, a voltage source is used, which is a translation from the fluid domain of a pressure source or pump. In the practice of lab-on-chip design, both pressure sources, as well as flow regulating sources, are used as flow inputs to a system. Murphy Berzish previously considered this dual binary tree topology with current sources, which results in different regulator placement[6].

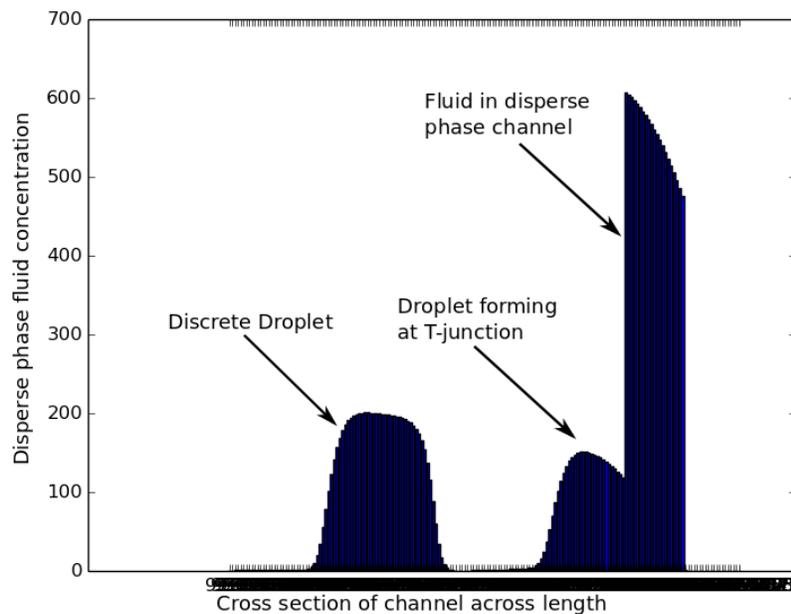
In the translation of domain from microfluidics to electrical circuit, we are only interested in the behavior of the circuit with respect to its expected behavior. In this notion, some fundamental assumptions are made. One of the fundamental assumptions is that the model only simulates steady-state behavior of the circuit and not transient state behavior. In other words, it is assumed that droplet channel is already populated with the maximum number of droplets they can accommodate. The resistance in the droplet channels due to droplets is characterized by $R7(t)$ and $R8(t)$ in figure 5.3 which are characterized by a variable resistor with a trapezoid signal of positive offset of R_dtotal which is the sum of the resistance of the number of droplets the channel can support minus 1. The amplitude of the trapezoid wave is R_d which is the hydraulic resistance expressed by a single droplet. The rationale behind the offset R_dtotal is that at steady state, the channel is expected to have at least n number of droplets at all time. The variation in hydraulic resistance comes from the difference in phase of one droplet being lost at the waste well and a droplet being created at the generator, therefore the amplitude of the trapezoid function is just the resistance from one droplet.

$$R_d = \frac{12\mu\frac{w}{3}}{wh^3(1 - 0.630h/w)} \quad \text{Resistance from a single droplet} \quad (5.3)$$

The justification behind choosing a trapezoid wave function to simulate droplet behavior is because of simulation results from finite element analysis using numerical method computation performed by COMSOL. In order to accurately characterize the behavior of droplet formation we ran a time step simulation of a single T-junction in COMSOL and analyzed the fluid concentration of the dispersed phase fluid across the cross sections. The resulting distribution of the disperse phase fluid looked like the following figure 5.4. As can be seen in the figure, the distribution of fluid concentrations from the droplets look approximately like rounded trapezoid functions, therefore that function was used to express a continuously varying resistance for our simulation model.

It can be noticed that our simulation model actually doesn't characterize the mechanism of droplet formation in any way. The influence of droplets on the entire system is embodied by the varying resistance in the channel, which is the most important aspect of modeling how the entire system is influenced by its perturbations.

Figure 5.4: Concentration distribution of a droplet across channel cross sections, from finite element analysis



5.4 Micro-regulating Membrane Design with Capacitor model

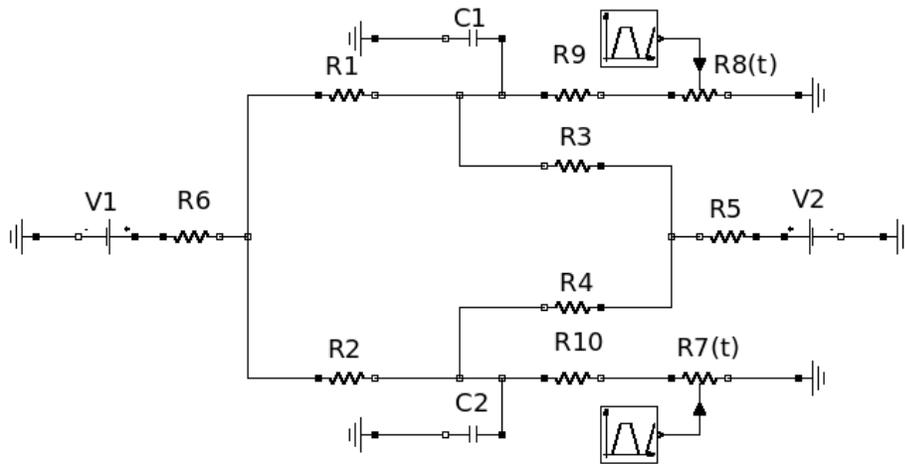
The circuit schematic shown above does not incorporate the regulating mechanism using capacitors discussed previously. It is common to use capacitors to buffer change in voltage in AC circuits, which is the inspiration for this design exploration. Important considerations in the usage of capacitors to regulate pressure are following:

- How many capacitors are needed to stabilize the currents coming into the T-junctions.
- What is the optimal placement strategy for capacitors to achieve desired stability
- Do the capacitor specifications meet physical limits of the diaphragm it characterizes

Several placement strategies were observed in considering the most optimal strategy for capacitor placement. Obviously, the capacitors need to be placed in parallel to the

circuit in some configuration in order to buffer changes in potential difference. Figure 5.5 represents the placement strategy that was optimal from our research. This approach involves having a capacitor right at the T-junction where the droplets are forming. This strategy is sensible because the capacitors buffer the change in the potential difference and minimizes the perturbation from influencing the part of the circuit before the T-junction. In the circuit domain the intersection of the flow channels is one junction, therefore it doesn't matter where the capacitor is placed in the junction, however, in the microfluidics domain this translates to inserting the capacitor right at the T-junction where the perturbations are starting.

Figure 5.5: Optimal circuit schematic with voltage regulating capacitor



Two other strategies explored include inserting a capacitor between the input flow channels instead of to a constant initial pressure. This results in suboptimal results compared to the desired result. Another strategy involved using a shunt capacitor across the flow channels, connecting the part of the channel producing the droplets with the part of the channel where flow is originating. These two strategies are presented in the Appendix. Further details about this comparison are explored in a technical report by Chris Willar in [42].

5.5 Design simulation using MapleSim

For the translation of circuit specification from the fluidic domain to circuit domain, one challenge is the translation of units. While there are analogous physical values these two

Parameter	Electrical Domain	Fluidics Domain	Specification
R1, R2, R3, R4	Resistance (MOhm)	Hydraulic Resistance	2.8×10^6
R6, R9, R10	Resistance (MOhm)	Hydraulic Resistance	2.8×10^6
R5	Resistance (MOhm)	Hydraulic Resistance	2.8×10^5
V1, V2	Voltage (V)	Hydraulic Pressure (Pa)	80×10^3
C1, C2	Capacitance (pF)	Material Capacitance	0.025, 0.01, 0.005
R7(t)	Variable Resistor (MOhm)	Droplet Resistance	
	Amplitude (MOhm)		0.056×10^6
	Frequency (1/s)	Droplets/s	32
	Amplitude Offset (MOhm)		1
	Time offset (s)		0
R8(t)	Variable Resistor (MOhm)	Droplet Resistance	
	Amplitude (MOhm)		0.056×10^6
	Frequency (1/s)	Droplets/s	32
	Amplitude Offset (MOhm)		1
	Time offset (s)		0

Table 5.1: Specification for circuit model in figure 5.5

domains, the order of magnitudes in which the two domains physically operate is very different. However, since we are only creating characteristics models of the analogous circuit in MapleSim and not creating empirical experiments to test our circuit designs, we decided to create the circuit with one to one mapping with respect to the analogous parameters with respect to the units of numerical values. Simply put, and $80,000 Pa$ pressure system in the fluidics domain would translate to a $80,000 V$ DC voltage source in the electrical circuit domain. This argument is consistent with other parameters in the system such as resistance and flow rate as well.

The translation is slightly more complicated in the case of a capacitor, for which there is no direct property in the fluidics domain.

Table 5.1 below shows the circuit specifications for the model circuit from figure 5.5 used in a later section to validate results regarding the stabilizing characteristics of capacitors for the parallel T-junction circuit.

It can be noticed that all of the static resistors have the same value except $R5$ although they carry different fluids in them. This is done on purpose, for ease of design. Since the viscosity of oil is one order of magnitude higher than that of water the channels carrying water are kept one order of magnitude longer in order to balance the resistance. These resistance values are computed using the equation for hydraulic resistance in rectangular

channels defined in equation 5.2. The choice of capacitance values is determined through experimental runs to satisfy requirements of flow rate stability entering the T-junctions. These capacitance values were chosen to exhibit the trend in stability response for a range of capacitance values. The stability response results are exhibited in section 5.5.1 which discusses the simulation results and their motivation in greater depth. It is also worth noting that there is no time offset between the trapezoid wave signals exhibited in table 5.1 in the form of $R7(t)$ and $R8(t)$. This is because in our experiments it was shown that the capacitor configuration that works for variable resistor functions in phase also work for functions that are out of phase. Table 5.2 below show the physical specifications for channel and diaphragm for the circuit specified in table 5.1. The channel labels in figure 4.2 are used to demonstrate the physical dimensions of the channels as well as the fluid they carry. Notice that the circuit schematic in figure 5.5 only represents half the demonstrated fluidic circuit shown in figures 4.2.

Component	Property	Specification
Ch1 - Ch3	Length	200 μm
	Width	100 μm
	Height	50 μm
Ch4 & Ch6	Fluid viscosity (Oil)	0.01 kg/m^3
	Length	2000 μm
	Width	100 μm
Ch8	Height	50 μm
	Fluid viscosity (Water)	0.001 kg/m^3
	Length	200 μm
Ch5 & Ch7	Width	100 μm
	Height	50 μm
	Fluid viscosity (Water)	0.001 kg/m^3
Diaphragms	Length (μm)	75 μm
	Thickness (μm)	25 μm

Table 5.2: Channel specifications of fluidic channel shown in table 5.1

For the analytical model of representing membranes with capacitors, there are more design dimensions that are worth exploring. One of them is how the positioning of the

membrane in the channels, translate to positioning of the capacitors in the analogous circuit. In the design described by figure 5.5 the capacitor is positioned right at the beginning of the T-junction. Since any channel length in the physical domain translates to resistance in the analogous domain, this is expressed by no resistances between the T-junction node and the capacitor in the circuit. Therefore, in order to reposition the membrane to any other point in the channel, this can be achieved by redistributing the total resistance of the channel to before and after the capacitor. For example, to have the membrane being in the middle of the channel, there would equal resistance before and after the capacitor, with the sum of the two resistances summing up to the hydraulic resistance of the channel.

From simulating the analogous circuit, it is possible to find a capacitance value that would allow us to have a circuit where the pressure fluctuations are regulated to an acceptable level. After we have the capacitance, this needs to a membrane that can represent the capacitance. This involves deriving the physical parameters such as length, width and thickness of the membrane to characterize the capacitance. This is the subject of section 5.6.

5.5.1 Simulation Results

The simulation results shown here are created using MapleSim. The equivalent circuit was created using the basic analog circuit components provided by MapleSim. A time domain simulation was run for the model, by probing for the appropriate response variable. In this case the response variable we are interested in tracking is the current at the node right before the T-junction. The objective of the simulations is to show that introducing the capacitors in the proposed strategy reduces the fluctuation in current caused by the fluctuation in pressure created by the changing of resistance during the formation of droplets and their disappearance at the waste well.

As described in the section above, the base resistance of the channel containing the droplets is a sum of the hydraulic resistance of the channel and as well as the total number of droplets in the channel. The variance comes from the formation of droplet and losing of droplets to waste wells, at the end of the channel. This is what the variance in resistance characterizes. Three sets of graph were created for three cases of resistance variance from the baseline: 10%, 5%, 2%.

For each simulation graph, four different plots are drawn for different capacitance used. The capacitance values chosen for the simulations were $0.005pF$, $0.01pF$, $0.025pF$. These values were chosen by running multiple simulations with the specific design and in order to achieve different tolerance requirements for current fluctuations. For other designs of

different physical parameters, the required capacitance for stability might be different. Future versions of the synthesis software would be able to generate these values of capacitance by solving the model using an SMT solver.

Taking a take step back, it is worth realizing that without this lumped parameter simulation technique, a designer would have to go physically fabricate T-junction circuits with different membrane designs to see what works. That would be much more time consuming than simply running simulations based on an analytical model. The simulation results are shown in figure 5.6, 5.7, 5.8.

Figure 5.6: Current response with capacitor values for 10% variance in resistance.

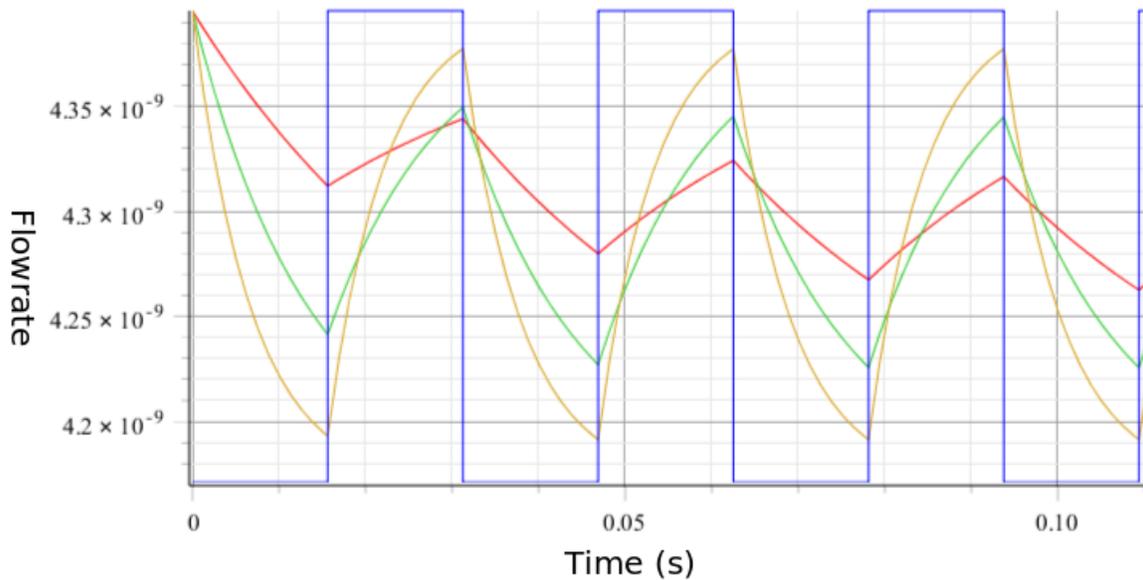


Table 5.3 summarizes the result of the simulations. The columns show the percentage of variation in current, the resistance in variance for different capacitor configurations, including with no capacitance. In all the cases, the variation in current is being reduced by a higher capacitance value, with $0.025pF$, being the most optimal capacitance in all three cases.

From simulating the analogous circuit, it is possible to find a capacitance value that would allow us to have a circuit where the pressure fluctuations are regulated to an acceptable level. After we have the capacitance, we need to fabricate a membrane that can represent the capacitance. This involves deriving the physical parameters such as length,

Figure 5.7: Current response with capacitor values for 5% variance in resistance.

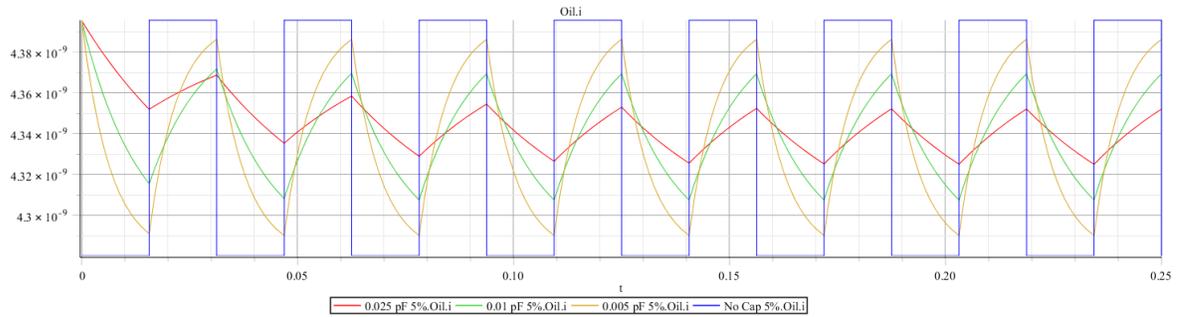
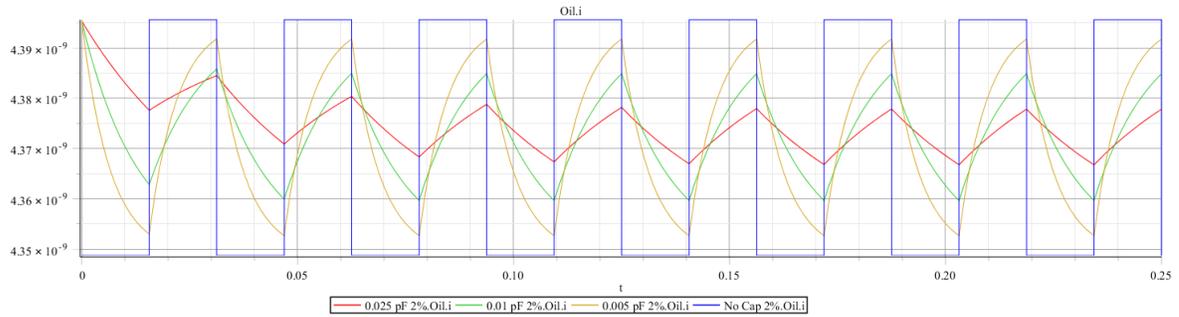


Figure 5.8: Current response with capacitor values for 2% variance in resistance.



width and thickness of the membrane to characterize the capacitance. This is the subject of the next section.

Resistance Variance	0 pF	0.005 pF	0.01 pF	0.025 pF
10%	22.38%	18.54%	11.89%	5.21%
5%	11.48%	9.56%	6.15 %	2.7%
2%	4.66%	3.89 %	2.52 %	1.11 %

Table 5.3: Stability response of current for different capacitors and variance in resistance

5.6 Physical characterization of membranes

Using flexible membranes to stabilize microfluidic circuits is not really a new concept. There are many models in literature for achieving this. However, we wanted to choose

the simplest model, so that it can be represented in our electrical analogous circuit. In [9] for example, they use a pair of membranes accessible through side channels to restrict flow. While this is passive, this is slightly more complicated to fabricate, because for each regulator, two side channels need to be created. In [16], soft membranes are used for flow regulation by actively applying positive or negative pressure to increase or decrease the cross-sectional area available, for liquid to flow through. Zhang et al [44] also use a similar concept of side channels in combination with membranes to regulate flow. The works of [18, 28] is probably the closest to what we have attempted, at least from the perspective of simplicity. In the work of Stone et al, a soft wall is used for one face of the entire channel in order to passively regulate flow. That shows promising results, but structurally, it compromises the robustness of the entire circuit. In this work discrete, shorter sections of membranes are explored. Also, the soft membranes are placed on the side of the channels rather than on the floor or the ceiling of the channel. This makes them easier to measure, without a confocal microscope.

5.6.1 Hydraulic capacitance

Capacitance is expressed electrically by:

$$C = \frac{\Delta q}{V}$$

where q is the electrical charge held by the capacitor, and V is the potential difference across it. In the domain of mechanical physics, capacitance is expressed in the change in volume caused by the pressure applied to an elastic material. In other words:

$$C_{hydro} = \frac{\Delta V}{P} \tag{5.4}$$

Since we want to derive the physical parameters of a membrane from the capacitance calculated from simulations, we want to find the parameters that define the *Volume*. The shape that deformed rectangular membrane exhibits is closer to a *Rectangular Paraboloid*, but not quite. A rectangular paraboloid does not have the four edges of the rectangular fixed on a two-dimensional plane like the rectangular membrane would. That is why that does not describe the shape precisely. It is an over-approximation of the actual volume expressed by a deformed membrane.

Another way to approximate the volume under the 3D shape would be to assume it is similar to a *rectangular pyramid*. This is also not quite precise because the surface

of a pyramid is described by a planar surface, rather than a paraboloid. For the same rectangular base, this gives an under-approximation for the volume under a deformed membrane. For the sake of simplicity of the equation, the experiments were done with the assumption that the volume under the deformed membrane is approximately close to a rectangular pyramid. With all other parameters known in equation 5.4, the only unknown parameter is the height of the pyramid. which is expressed by the maximum deflection of a flexible membrane from applied pressure.

There are many equations for calculating structural deflection under stress, and several strategies were explored for the experiments to evaluate the best strategy.

Membrane deflection

The simplest equation for calculating membrane deflection is using *beam theory*. In structural mechanics, a beam is a two-dimensional object of certain length, and shape (usually rectangular) with certain flexibility properties. It is a simple relationship because it is two dimensional. In the context of the flexible membrane, the beam theory relationship for a *uniform beam clamped on both ends with a uniform load* is relevant. The equation for maximum deflection of such a relationship is given below:

$$W_{max} = \frac{5}{384} \frac{wl^3}{EI} \quad (5.5)$$

where W_{max} is maximum deflection, w is uniform force, l is length of beam, E is Young's modulus and I is second moment of inertia, given by:

$$I = \frac{lb^3}{12} \quad (5.6)$$

The shortcoming of applying beam theory in this context is that beam theory describes a relationship in two dimensions, where the membrane would have forces acting in three dimensions. Since beam theory ignores a dimension of forces, it over-approximates the deflection. Therefore the equation describing maximum deformation was explored in *plate theory*, which is comprehensive enough to describe the relationship in three dimensions.

The relationship from plate theory which is most relevant in this context is *uniform rectangular plate, fixed on four edges with a uniform load* applied across the surface. The equation for maximum deflection defined by *plate theory* is given by:

$$W_{max} = Cl \frac{p\{max(l, w)\}^4}{Eh^3} \quad (5.7)$$

where W_{max} is maximum deflection, p is pressure, l is length, b is breadth, h is thickness, E is Young’s modulus and C is Roark’s coefficient.

The equation for plate theory for this relationship is analytically derived based on some underlying assumptions. The *Roark’s constant* is a coefficient for the equation characterizing properties from the stress-strain curve of material properties. This is retrieved from a table of values derived analytically for a more complex relationship, meant to make calculations simpler. One of the underlying assumptions made about materials in this model is that *Poisson’s ratio* ≤ 0.33 . This is violated in our case because the Poisson’s ratio material property for PDMS = 0.5. From using a simulator, it was deduced that the plate theory equations were approximating the deformation of a PDMS membrane, for a given pressure and known geometric and material parameters. Therefore a more accurate representation would be necessary.

The works of Eswaren et. al used a formula for the deflection of PDMS membranes. The equation for deflection of rectangular PDMS membrane is given by:

$$W_{max} = C \frac{Pb^4}{Eh^3} 12(1 - \nu^2) \quad (5.8)$$

Where, W_{max} is maximum deformation, P is pressure, b is breadth of membrane, C is coefficient as a function of (l/b) , E is Young’s modulus, h is thickness, ν is Poisson’s ratio. C is given by a look-up table, in [35].

b/a	1.0	1.3	1.4	1.5	1.6	1.7	2.0	∞
C	0.00126	0.00191	0.00207	0.00220	0.0230	0.00238	0.00254	0.00260

Table 5.4: Coefficient table for deformation of square membrane [35]

Experiments have shown that using equation 5.8 gives the most reasonable value for membrane deflection in relation to experiments. Therefore, this was used to derive the thickness of the membrane.

It must be noted that all of these formulae are only valid assuming the expansion of the membrane is operating in the linear or small deformation region of the stress-strain curve, where structures follow physics of Hooke’s law of elasticity.

Challenges

One of the challenges for this design process has been choosing the appropriate Young’s modulus for PDMS. Young’s modulus is property of materials that characterizes the elasticity of materials. More specifically, it defines the strain-strain relationship of materials.

A lower Young's modulus means higher elasticity. Since PDMS is not a naturally occurring material and is made through a "baking" process involving multiple stages, Young's modulus of PDMS depends on the process. According to MIT reference for material properties, Young's modulus of PDMS ranges from 360 to 870 kPa. There is, however, a "standard" baking procedure for PDMS in most applications of lab-on-a-chip. According to the reference papers that use PDMS membrane designs, the commonly used parameter for YM is 400 kPa. According to [38], Young's modulus for PDMS for the common baking procedure is 2.63MPa , which is much higher than values suggested in the papers discussing membranes. This was a source of ambiguity. The choice was made to use Young's modulus from the reference papers of 400 kPa, since the baking process used for the designs in this project were same as the ones in the papers.

5.7 Fabrication and Experiments

This section provides a short description of *Soft Lithography*, which is the process used for fabricating the design for experiments. It also describes the designs that were chosen for fabrication and the reasons behind choosing those specific designs. After fabrication, the process of running experiments and instrumentation is also discussed.

5.7.1 Soft Lithography

Soft lithography is a fabrication strategy for micro and nano-scale structures. It is called soft lithography because it relies on hardening of *soft* elastomers such as PDMS during the fabrication process. The process was first proposed by Whitesides et. al for rapid prototyping of micro-structures [30].

For the process, first the design of the mask needs to be created using a CAD tool. After the design is created a mask needs to be created on a silicon chip. This is usually done by 3d printing or photolithography. The PDMS mold is made by combining a silicone agent, usually Sylgard 182 or 184 with a curing agent. The curing ratio influences the material properties such and Young's modulus this is an important specification for the fabrication process. For the experiments in this report, a curing ratio of 10:1 was used. The mixture is poured on the master, which has the design and placed on a petri dish for baking. The baking process usually takes about 2.5 hours at 80°C . After baking is complete, the PDMS mixture forms a sturdy, solid structure. The PDMS is peeled off from the mask and then bound to a glass frame. One of the challenges of the fabrication process is the peeling off

process. Since the membranes are very thin, they are prone to breakage during the peel off activity. Therefore, usually multiple copies of the same design need to be made to hope that at least of them will be fully functional.

5.7.2 Fabricated Designs

For analysis of the results, there were two main set of experiments that were relevant. One set of experiments is to demonstrate that thin-walled membranes are a viable strategy for regulating flow for scalable designs that include many T-junctions. The second set of experiments is to demonstrate that the simplified analytical model that is suggested for automated design synthesis is an accurate model within engineering limits.

The second set of results is not so much *results*, but more in the nature of evidence for credibility of the simplified model. The bigger objective of the project is to be able to create scalable circuits using automated design, so the T-junction circuit is the main highlight. The schematic of the circuit fabricated follows the specifications expressed in table 5.2. The missing element from the schematic in table 5.2 is for the membrane specifications. A certain membrane design was chosen out of many designs for fabrication. Based on the equation for *deflection* described in equation 5.8, a membrane specification is chosen that is closest to the capacitance required by simulations, described in figure 5.6, 5.7, 5.8. For the chosen membrane and the circuit schematic, experiments are conducted with varying pressures, to test stability of the circuit, by measuring poly-dispersity and also measuring the deflection of the membrane. The specification of the membrane, chosen for the fabrication is expressed in table 5.5.

Length(μm)	Thickness(μm)	Height(μm)	Pressure(bar)	Pressure(Pa)	E(kg m/s^2)	Deformation(μm)	Capacitance(pF)
150	20	50	0.1	10000	88200	16.789	0.042
150	20	50	0.2	20000	88200	33.577	0.084
150	20	50	0.3	30000	88200	50.366	0.126
150	20	50	0.4	40000	88200	67.156	0.168
150	20	50	0.5	50000	88200	83.944	0.209

Table 5.5: Membrane specification

Chapter 6

Result and analysis

This section analyzes the output from the design process described earlier and compares them to results obtained from empirical experiments.

6.1 Reliability of membrane design

For the experiments, a T-junction circuit was fabricated using the laboratory process described in section 5.7. It is a T-junction with membranes installed using the parameters produced by the design process using the analytical solutions in the software model.

6.1.1 Comparison of simulation and measurements

The first set of comparisons performed were to compare the mechanical response of the membrane. This involved retrieving the data for deflection of the membrane from physical experiments and comparing them with the analytical equation derived from plate theory. For data analysis, a high-speed camera was used take images of the membrane behavior during experiments. Using a combination of ImageJ and Matlab, data analysis was performed to record the deformation of the membranes for varying pressure. The analysis was done to record deformation while increasing pressure and while decreasing pressure. This was done to ensure that the membrane was still operating in the linear region of the stress-strain curve from the pressure that was applied. In the case of linear deformation, the deformation for increasing pressure is supposed to correlate with the deformation for

decreasing pressure. The data for deformation is presented in figure 6.1. The blue and orange data points in the graph show the deformation for increasing and decreasing pressure, and a direct correlation is observed.

The green data points in the graph show the expected deformation using the analytical relationship derived from plate theory described by equation 5.8. Evidently, the deformation predicted using the equation from plate theory quite accurately predicts the deformation in the experiments. The caveat in this data is the Young's modulus for the device used for the experiments. Since the Young's modulus depends on the fabrication process, it was important to calculate the Young's modulus for the device used in the experiment.

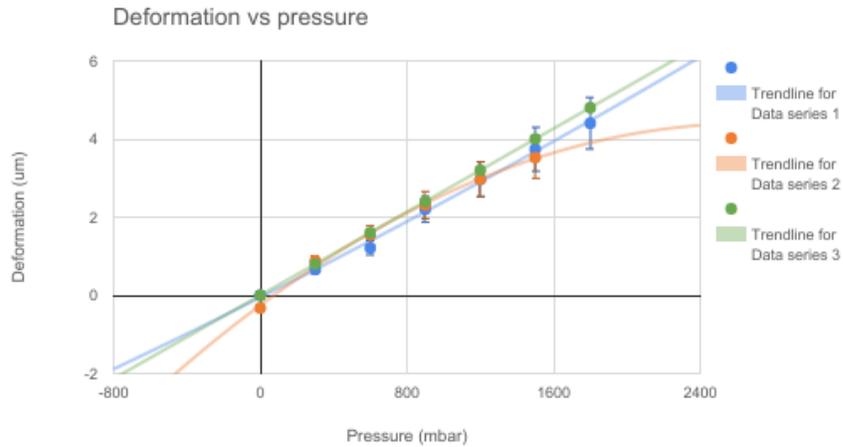


Figure 6.1: Deformation of membrane with pressure

The deformation data from the experiments was used to evaluate the Young's modulus using equation 5.8. The experimental data for the Young's modulus is shown in figure 6.2. As evidenced in the data, the same Young's modulus was not recorded for each of the experimental data points. Some variation is expected because the Young's modulus depends on more factors than maximum vertical deformation with response to pressure. However, with the assumption that some of the points are outlier, an approximate Young's modulus of 1.65 GPa was used for the relationship shown in figure 6.1

This shows that the analytical equation is an approximately accurate model for predicting deformation of a membrane fabricated using the defined parameters.

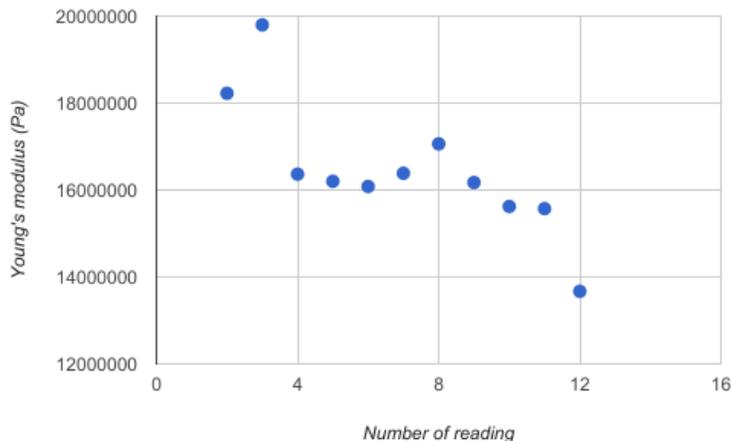


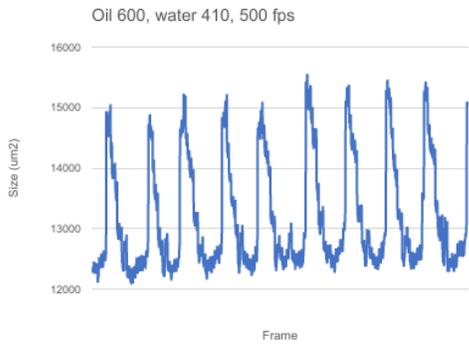
Figure 6.2: Young's modulus from experiments

6.2 Droplet data from experiments

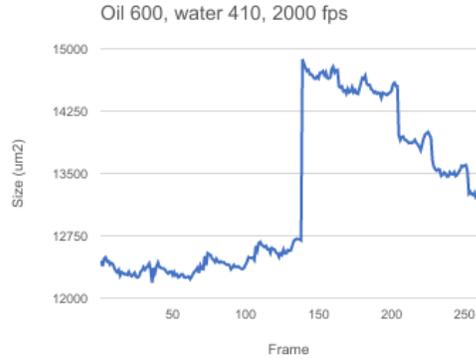
Besides measuring the deformation of a membrane from applied pressure, experiments were performed to record data about droplet sizes. Images were captured for 4 seconds and 0.25 seconds using 500 fps and 1000 fps settings. Image analysis was performed on these images using ImageJ and Matlab to record cross-sectional droplet area across the experiments. Figure 6.3 shows the droplet size distribution across frames.

As evidenced in the figure, there is clearly periodic variation in droplet size. This is the behavior the thin-walled membrane designed to minimize, with respect to designs that do not have the thin-walled membranes. Figure 6.3a shows the periodic variation of droplet sizes across 4 seconds. Figure 6.3b presents a zoomed-in view of the droplet variation in 0.25 seconds but with 4 times the fps. Unfortunately, during the time of this write-up the experiments for measuring droplets in T-junctions without the thin-walled membranes were not conducted, and there a quantitative deduction for how much stability the membranes add could not be derived.

This periodic fluctuation is further evidenced in the velocity profile of droplet, illustrated in figure 6.4. It can be seen that variation in velocity of the droplets coincides with the variation in sizes according to the frame reference. There are some outliers in the experimental data, which are due to limitations in image analysis by the tool ImageJ.



(a) Droplet area across 4 seconds



(b) Droplet area across 0.25 seconds

Figure 6.3: A figure with two subfigures

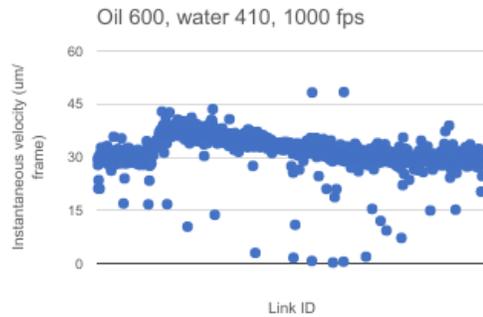


Figure 6.4: Droplet velocity profile in 4 seconds

One of the primary shortcomings of this result is that there is no benchmark data to compare with. Ideally, to analyze the stability of T-junction because of added membranes, it would need to be compared against experiments performed on T-junctions without a membrane. This remains to be one of the future work to be performed. However, even without that data, it is shown in section 6.1 that the analytical model described in the software accurately models the physical membranes.

Chapter 7

Design limitations and caveats

This chapter discusses the caveats of some of the design choices made during the project. In general, the design choices are relevant in two different and inherently independent aspects. The first aspect is the software framework for Manifold. The choices made here and the limitations are imposed by some of the technologies and methodologies used during modeling. The second aspect is in the modeling of the membranes for hydraulic capacitance.

7.1 Limitations in Software Framework

One of the fundamental assumptions that must hold for any of the simulations is that the fluidic environment has to be *laminar flow* and under a *low Reynold's number* range. Low Reynold's number in this context, it refers to values ≤ 1 . This is the most stable range of laminar flow and produces the most predictable fluid behavior. There are applications of microfluidics, that operate beyond this range such as *Centrifugal Microfluidics* and *Inertial Flow Microfluidics*. These types of microfluidic devices cannot be expressed in the current system.

Inertial Forces

The electrical and hydraulic analogy is the basis of much of the lumped parameter modeling that allowed the simplification of the hydraulic systems. One of the ways in which fluid dynamics is different from the physics of electricity is that fluid dynamics innately exhibit inertial flow, whereas no such physical phenomenon happens for electricity. This is fundamentally true because fluid dynamic concerns movement of *mass* using pressure, however, electricity involves the movement of electrons which do not have mass and therefore no inertia. This makes a huge conceptual difference in the characterization of the two

domains. For some ranges of Reynold’s number, even laminar flow is greatly influenced by inertial forces. This happens when inertial forces contribute significantly to the transport of mass, compared to viscous forces. However, in Reynold’s number range ≤ 1 , the influence of inertial forces is negligible. That is why the electrical analogy proposed here will only work on systems that operate under this specific Reynold’s number range.

Transient Behavior

When systems run physically, there is always a transient period that occurs before the system reaches steady state. Usually, in a fluid flow system it happens at the beginning when the system is started and all the channels are still filling up, or at the end when the pressure pump is deactivated and all the channels are starting to empty out. During these phases, the mechanics of the system work differently from steady state. The behavior of the system in transient stages is not characterized by the software. The hardware synthesis happens on the basis of how the system is expected to behave in steady state behavior and therefore, during the simulation of the system in MapleSim, it also only simulates the steady state behavior.

Limits of Van Steijn Model

For modeling the droplet generation in a T-junction, the droplet volume predictor model suggested by Van Stein et al was used [37]. While it is experimentally a robust model, it has its limitations and by proxy the limitations of the model are relevant in the software model.

One of the conditions of the predictive model is that the ratio of the width of the two incoming channels has to be within a certain range. Specifically $0.33 \leq w_{out}/w_{in} \leq 3$. Another condition involving channel geometries is that the ratio between the height and width of the individual channels have to be within a range. Specifically, $0.1 \leq h/w \leq 0.5$. Also, the capillary number for the system has to be smaller than 0.01.

Limitations of dReal

One of the limitations of the SMT Solver dReal is that it cannot solve partial differential equations and integrals. This is a limitation because most equations of fluid dynamics derived from the Navier-Stokes equation are in the form of PDEs. Therefore they need to be simplified to ordinary differential equations before they can be characterized in dReal for the solver. It’s not always intuitive how to transform a behavior described in PDEs in a reasonable way in the form of ODEs. Usually, some simplifying assumptions have to be made to find an engineering solution, from a mathematical expression. This can sometimes be challenging.

7.2 Limitations in Membrane Design

This section discusses the caveats and assumptions involved in the membrane design methodology using electrical capacitors.

Upper limit of Hydraulic Capacitance

Hydraulic capacitance is higher for membranes that are more flexible and is capable of higher displacement with less pressure. Generally, thinner membranes have a higher flexibility than thicker membranes. A limitation appears because at the micrometer scale: PDMS membrane can only be made up to a certain thickness without compromising the structure. For example, if membranes are made too thin, they might break off during the peeling-off process during fabrication and that would compromise the design. Based on experience from designers, the thinnest membrane that can be fabricated using soft-lithography with reliable results *most* times is in the range of 15 to 20 μm . Another important parameter in the fabrication of membranes is the aspect ratio. Also based on experience from designers, a reasonable aspect ratio for which membranes can be fabricated is from 1:1 ratio up to 1:3 ratio. This sets the grounds for the upper limit of hydraulic capacitance using technique used in this project. From analytical experiments, it was determined that within a reliable range, the highest possible hydraulic capacitance for PDMS would be in the range of 10 to 20 pF . If higher capacitance than that is required, more membranes can be attached *in series* with each other, to linear increase capacitance in an additive manner. The circuit analogy to this would be attaching multiple capacitors parallel to the circuit, but in series with each other.

Approximating hydraulic capacitance using volume of a pyramid

One of the analytical assumptions made in modeling hydraulic capacitance of a membrane is that the volume under a deflected membrane can be approximated reliably using the formula for volume under a rectangular pyramid.

Chapter 8

Other Related Work

This section discusses some of the work that was done as part of the master's research project relevant to the automated design framework, but not quite relevant to the membrane design for regulating droplet generation, which forms the bulk of the research involved. In particular, significant effort was given in investigating a design called an *electrophoretic cross* to perform capillary electrophoresis using pull-back voltages. This was performed in collaboration with Chris Backhouse, from the department of Nanotechnology who specializes in different techniques in Nano and Micro-biological systems engineering.

8.1 Capillary electrophoresis

This section describes the capillary electrophoresis using an electrophoretic cross.

Background & purpose

Capillary electrophoresis is a profiling technique for determining the concentration of substances or analytes in a fluid sample. This is important in the analysis of DNA, in molecular biology and many other disciplines of bio-medicine and biotechnology. One of the most primitive techniques to this objective is chromatography, where a fluid sample is presented to a chromatography paper and after diffusion, the distance from the starting point indicates the type of analytes present in the sample. An advancement of that is *gel electrophoresis*, where the separation of analytes happens in an agarose gel matrix, by passing a very high potential difference across it, using an electric field [40]. Capillary electrophoresis is an evolution of that technique, where the separation happens at a much smaller scale, and hence faster and more reliably with higher resolution.

The specific design that was investigated as part of this project is an *electrophoretic cross*, performing capillary electrophoresis. It is introduced by Chris Backhouse in his book [4], who was the main collaborator for this specific investigation. An illustration of the electrophoretic cross is given in figure 8.1.

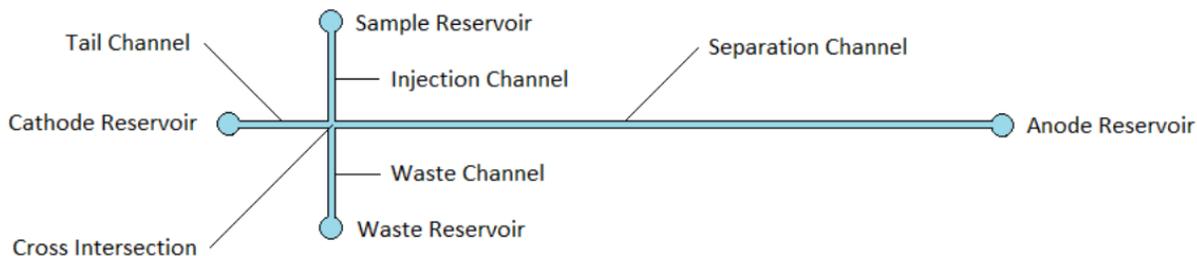


Figure 8.1: Schematic of an electrophoretic cross

Physics and mechanics

The electrophoretic cross has two primary channels: The injection channel and the separation channel. At either end of the injection channel, there is sample reservoir and a waste reservoir. The sample is loaded into the sample reservoir at the start of the experiment. On either side of the separation channel, there is a cathode reservoir and an anode reservoir. These two reservoirs are connected to opposite electromagnetic polarities at a very high voltage to induce a large potential difference across the separation channel. There are three separate steps involved in running an experiment in an electrophoretic cross: loading phase, injection phase, and separation phase.

In the loading phase, the sample is introduced into the sample reservoir, and a potential difference is applied across the injection channel to *load* the sample into the channel. The ideal place for the sample to be loaded is in the middle of the channel, at the intersection. After loading is complete and the sample is in the intersection, the system is ready for the injection phase.

For the injection phase, a potential difference is applied across the separation channel using the cathode and anode reservoirs. This causes a *slice* of the sample at the intersection to start moving towards the anode reservoir. As the sample moves through the channel, the different analytes in the sample move at different velocities due to the potential difference. This is because of the mass, density and electrical charge of the molecules involved. The motion of the particles is primarily due to two separate forces involved: electro-osmotic flow and electro-phoretic flow.

In the separation phase, the potential difference makes the analytes separate across the channel. The measurement involved in the experiment is done in this phase. Usually, the experiment is performed for a specific sample, and a specific analyte in the sample is targeted. For example, one might be interested in measuring the concentration of a specific strand of DNA, which describes a pathogen, in a blood sample. To measure the concentration of a specific analyte a fluorescent compound is added to the sample, that binds to the specific analyte. Usually, the fluorescent compounds are sensitive to a very specific wavelength of electromagnetic waves, that is used to detect the presence of that analyte, using a light source and an imaging device. The sensors are positioned at a certain distance along the separation channel, using prior knowledge to ensure the subject analyte will have separated from the sample at that point. At the end of the experiment an *electropherogram* is produced which illustrates the concentration of the specific analyte passing through that point in the separation channel over time. A good experiment would provide concentration distribution, that has a shape similar to a Gaussian distribution.

Challenges and pullback voltage

There are many challenges in achieving satisfactory results in a capillary electrophoresis experiment. The system described above is a basic electrophoretic cross design. There are advanced designs that attempt to address some of the challenges in running these experiments. One of these designs is a cross with a mechanism called *pullback voltage*.

During the separation phase of the experiment, the ideal scenario would be to get an isolated slice of the sample in the separation channel, with no contamination. However, in reality what happens is after some of the samples move into the separation channel from the intersection, more of the sample in the injection channel also leaks into the separation channel. This happens due to diffusion and after the sample diffuses into the separation channel, it gets pulled in by the electric field. This causes the results to be obscured.

The solution to this problem is using a technique called pullback voltage. The idea behind it is that as soon as the potential difference is activated across the separation channel, a set of pullback voltage is applied to the ends of the injection channel to “pull back” the samples in this channel from diffusing into the separation channel.

The challenge behind the pull back voltage is knowing what potential difference to apply across the channels to ensure the system is stable.

Experiments

Stephen Chou describes in detail the design procedure for characterizing this problem in the Manifold framework.

Chapter 9

Future Work

In spite of much progress, there is much to be done both on the aspect of design process of the microfluidic circuit as well as the Manifold software framework, before this research makes significant impact. This section highlights some of those action items.

Experiments with T-junction without membranes

As mentioned above, one of the shortcomings of the results presented to support the suggested design process is reference to compare results with. Therefore in order to quantitatively verify the influence of the thin-walled membranes, it is important to repeat the experiments with a T-junction without membranes, and observe how much difference the membranes make for stabilizing the droplet size.

Fabricating a multi-way T-junction

One of the main concrete objectives of this project was to use automated design tools to create a multi-way T-junction device, which is currently intractable using traditional methods. Before the challenge of a multi-way T-junction can be tackled the mechanism for stabilizing circuits using the thin-walled membranes need to be well established. This is because, the difficulty with multiple T-junctions as described in chapter 4 is pressure perturbations due to internal coupling. Once the automated design process for thin-walled membranes is validated, explorations can be made on how place and characterize membranes in a multi-way T-junction circuit.

Completing the Manifold Framework

The Manifold framework still remains a work in progress. Recently, there has been much progress in getting the program to work end-to-end, with a design being synthesized and simulated based on a design description in the front-end language. However works still

remains to be done on the framework before the framework is accessible to microfluidic designers.

One of the main action items for the manifold framework is making a more robust synthesis mechanism. This is necessary because often the constraints provided by the designer produce a model for dReal that is under-constrained, and the output is a system that is not a viable design because the parameter ranges are too broad. This can easily be detected when a simulation is run using MapleSim and the simulation output is compared with the design requirements. One solution to this is a *Counter Example Guided Abstraction Refinement* loop. This is a concept used in Software Engineering to iteratively refine a certain model based on counter examples created by a second entity. In the case of manifold, the “CEGAR” loop would be used to add constraints to the model based on simulation results in MapleSim, until a satisfying model is found.

Ongoing improvements to the Manifold framework would involve adding more components to the microfluidics library. Currently, only simple components are available such as a T-junction and simple pipes. A broader library of related components needs to be added, based on collaboration with microfluidic designers.

Sometimes synthesis is not what is desired by designers and instead a verification tool is desired. This is evidenced in section 8 in the discussion about capillary electrophoresis channels. Features in the framework need to be added where the designers simply request for a verification counter-example to the model description instead of a synthesis. This is surely possible using the SMT solver dReal and in fact is what it is meant to be used for, since SMT solvers excel at verification of constraints rather than synthesis.

Chapter 10

Conclusion

In conclusion, this thesis describes the exploration of the design process of a specific microfluidic component using novel techniques using software. This project fits into the bigger picture of creating automated design tools for microfluidic circuits which in this specific case is the hardware design synthesis framework, Manifold.

The novelty of the design process for the thin-walled membranes lies in a few areas:

- Use of system modeling software such as MapleSim describe a complete circuit, and its viability is scaled-up circuits.
- Usage of lumped parameter modeling to raise the abstraction of design, so that simpler relationships for components can be used to describe more complex circuits.
- Significance of using hydraulic/electrical analogy in simplifying fluid mechanics relationships in complex circuits.

The design process in chapter 5 discusses how a high-level model of a system can be created with analogous domains such as electrical circuits using MapleSim. This involved description of a functional model for a T-junction, and how a capacitor relates to a membrane in the hydraulic domain. It then discusses how the values obtained in the electrical domain translates to physical values for fabricating the membranes. The fabrication results for this process is shown in chapter 6. A mixed result is shown. The result shows that the simplified analytical equations used for lumped parameter modeling characterize the system fairly accurately within a tolerance, based on the membrane deformation data. However, the analysis for comparing how well the process predicts system stability remains

to be investigated, because the experiments using a T-junction without any membrane under the same environmental parameters remains to be performed.

Main challenges

One of the main challenges in this project has been constructing the simplified analytical relationships for complex mechanics. For example for the relationship between pressure and deformation, formulae from plate theory of mechanical deformation were chosen based on experimental iterations. However, the plate theory formulas themselves are empirically derived and work within certain limiting assumptions, and therefore do not translate universally. The formula for plate theory are primarily appropriate for materials with much higher Young's modulus, such as metals and therefore their translation for PDMS circuits is limiting. Another challenge in deriving analytical equations was in the behavior of a T-junction. This process often requires a high level of domain expertise, therefore a collaboration between engineers of different specializations is required to conclude an analytically correct model.

Future Work

As highlighted in chapter 9, future work to be done in this project can be categorized into two areas. In the microfluidic specific domain, more experiments need to be performed to validate the analytical relationships to ensure that they are characterized accurately enough and that the underlying assumptions behind those model are thoroughly defined. In the software domain, the Manifold framework needs to be made more mature so that the designs produced by synthesis are reliable with respect to the design descriptions.

Design automation of microfluidic devices seem to be a very promising and necessary step towards significant progress in that field, and it has been shown in this thesis that it is very much possible, albeit much work needs to be done. The achievement of this will surely rely on the close collaboration of expertise in software engineering and microfluidics design engineering.

References

- [1] Mirela Alistar, Paul Pop, and Jan Madsen. Operation placement for application-specific digital microfluidic biochips. In *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, apr 2013.
- [2] Ahmed M. Amin, Mithuna Thottethodi, T. N. Vijaykumar, Steven Wereley, and Stephen C. Jacobson. Aquacore: A programmable architecture for microfluidics. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, pages 254–265. ACM, 2007. ISBN 978-1-59593-706-3. doi: 10.1145/1250662.1250694. URL <http://doi.acm.org/10.1145/1250662.1250694>.
- [3] Ismail Emre Araci and Philip Brisk. Recent developments in microfluidic large scale integration. *Current Opinion in Biotechnology*, 25:60–68, 2014.
- [4] C.J. Backhouse. *Engineering of Nanobiotechnological Systems*. Gardland Publishing, 2015.
- [5] Aditya Bedekar, Yi Wang, Sachin S Siddhaye, Siva Krishnamoorthy, and Stephen F. Malin. Design software for application-specific microfluidic devices. *Clinical chemistry*, 53(11):2023–6, November 2007.
- [6] Murphy Berzish. A software toolchain for physical system description and synthesis, and applications to microfluidic design automation. Master’s thesis, University of Waterloo, 200 University Ave. West, Waterloo, ON, Canada, April 2016.
- [7] Krishnendu Chakrabarty and Fei Su. System-level design automation tools for digital microfluidic biochips. In *CODES+ISSS'05*, 2005.
- [8] Krishnendu Chakrabarty and Jun Zeng, editors. *Design Automation Methods and Tools for Microfluidics-Based Biochips*. Springer, 2006.

- [9] Il Doh and Young-Ho Cho. Passive flow-rate regulators using pressure-dependent autonomous deflection of parallel membrane valves. *Lab Chip*, 9:2070–2075, 2009. doi: 10.1039/B821524C. URL <http://dx.doi.org/10.1039/B821524C>.
- [10] Wilfried Engl, Matthieu Roche, Annie Colin, Pascal Panizza, and Armand Ajdari. Droplet traffic at a simple junction at low capillary numbers. *Phys. Rev. Lett.*, 95: 208304, Nov 2005. doi: 10.1103/PhysRevLett.95.208304. URL <http://link.aps.org/doi/10.1103/PhysRevLett.95.208304>.
- [11] Sicun Gao, Soonho Kong, and Edmund Clarke. dReal: An SMT Solver for Nonlinear Theories of the Reals. In *International Joint Conference on Automated Reasoning*, 2013.
- [12] Piotr Garstecki, Michael J. Fuerstman, Howard A. Stone, and George M. Whitesides. Formation of droplets and bubbles in a microfluidic t-junction-scaling and mechanism of break-up. *Lab Chip*, 6:437–446, 2006. doi: 10.1039/B510841A. URL <http://dx.doi.org/10.1039/B510841A>.
- [13] Tomasz Glawdel and Carolyn L. Ren. Global network design for robust operation of microfluidic droplet generators with pressure-driven flow. *Microfluidics and Nanofluidics*, 13(3):469–480, 2012. ISSN 1613-4990. doi: 10.1007/s10404-012-0982-y. URL <http://dx.doi.org/10.1007/s10404-012-0982-y>.
- [14] Tomasz Glawdel, Caglar Elbuken, and Carolyn Ren. Passive droplet trafficking at microfluidic junctions under geometric and flow asymmetries. *Lab Chip*, 11:3774–3784, 2011. doi: 10.1039/C1LC20628A. URL <http://dx.doi.org/10.1039/C1LC20628A>.
- [15] N. Gleichmann, P. Horbert, D. Malsch, and T. Henkel. System simulation for microfluidic design automation of lab-on-a-chip devices. In *15th International Conference on Miniaturized Systems for Chemistry and Life Sciences*, Seattle, oct 2011.
- [16] T. Fukuba H. Kinoshita, T. Atsumi and T. Fujii. Active micro flow-rate regulation technique based on soft membrane deformation using miniaturized electro-osmotic pumps. In *14th International Conference on Miniaturized Systems for Chemistry and Life Sciences*, Groninger, The Netherlands, October 2010. URL http://www.rsc.org/binaries/loc/2010/pdfs/Papers/136_0292.pdf.
- [17] Asif Khan. Towards microfluidic design automation. Master’s thesis, University of Waterloo, 200 University Ave. West, Waterloo, ON, Canada, April 2016.

- [18] Daniel C. Leslie, Christopher J. Easley, Erkin Seker, James M. Karlinsey, Marcel Utz, Matthew R. Begley, and James P. Landers. Frequency-specific flow control in microfluidic circuits with passive elastomeric features. *Nat Phys*, 5:231–235, 2009. URL <http://dx.doi.org/10.1038/nphys1196>.
- [19] A. Zaman V. Ganesh M. Berzish, A. Khan and D. Rayside. Manifold: An smt-based declarative language for electronic and microfluidic design synthesis. In *CASCON 2016*, September 2016.
- [20] Maplesoft. Maplesim - high performance physical modelling and simulation, 2016. URL <http://www.maplesoft.com/products/maplesim/>.
- [21] Maplesoft. Maplesim - high performance physical modelling and simulation. <http://www.maplesoft.com/products/maplesim/>, 2016. Accessed: 2016-04-02.
- [22] J. McDaniel, A. Baez, B. Crites, A. Tammewar, and P. Brisk. Design and verification tools for continuous fluid flow-based microfluidic devices. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 219–224, Jan 2013. doi: 10.1109/ASPDAC.2013.6509599.
- [23] Carver Mead and Lynn Conway. *Introduction to VLSI Systems*. Addison, 1980.
- [24] W.H. Minhass, P. Pop, and J. Madsen. Synthesis of biochemical applications on flow-based microfluidic biochips using constraint programming. In *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, 2012.
- [25] Modelica and the Modelica Association. Modelica, 2016. URL <https://www.modelica.org/>.
- [26] M. Prysiazny P. Roth P. Socha M. Berzish A. Zaman N. Klassen, M. Lyons and D. Rayside. Manifold 2.0: A hardware description language for microfluidic devices. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering*, April 2017.
- [27] Kwang W. Oh, Kangsun Lee, Byungwook Ahn, and Edward P. Furlani. Design of pressure-driven microfluidic networks using electric circuit analogy. *Lab Chip*, 12: 515–545, 2012. doi: 10.1039/C2LC20799K. URL <http://dx.doi.org/10.1039/C2LC20799K>.
- [28] Yan Pang, Hyoungsoo Kim, Zhaomiao Liu, and Howard A. Stone. A soft microchannel decreases polydispersity of droplet generation. *Lab Chip*, 14:4029–4034, 2014. doi: 10.1039/C4LC00871E. URL <http://dx.doi.org/10.1039/C4LC00871E>.

- [29] F. A. Perdigones, A. Luque, and J. M. Quero. Correspondence between electronics and fluids in mems: Designing microfluidic systems using electronics. *IEEE Industrial Electronics Magazine*, 8(4):6–17, Dec 2014. ISSN 1932-4529. doi: 10.1109/MIE.2014.2318062.
- [30] Dong Qin, Younan Xia, and George M Whitesides. Soft lithography for micro- and nanoscale patterning. *Nat Phys*, 5(3):491–502, 03 2010. ISSN 1754-2189. URL <http://dx.doi.org/10.1038/nprot.2009.234>.
- [31] Eric K. Sackmann, Anna L. Fulton, and David J. Beebe. The present and future role of microfluidics in biomedical research. *Nature*, 507(7491):181–189, 03 2014.
- [32] M.F. Schmidt, W.H. Minhass, P. Pop, and J. Madsen. Modeling and simulation framework for flow-based microfluidic biochips. In *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, apr 2013.
- [33] Howard A. Stone. Microfluidics: Tuned-in flow control. *Nat Phys*, 5:178–179, 2009. URL <http://dx.doi.org/10.1038/nphys1213>.
- [34] T. Thorsen, S. J. Maerki, and S. R. Quake. Microfluidic large-scale integration. *Science*, 298(5593):580–584, October 2002.
- [35] S. Timoshenko and S. Woinowsky-Krieger. *Theory of Plates and Shells*. McGraw-Hill, 1959. pp 197 - 198.
- [36] V. van Steijn, M. T. Kreutzer, and C. R. Kleijn. Velocity fluctuations of segmented flow in microchannels. *Chemical Engineering Journal*, 135(1):S159–S165, 2008. ISSN 1385-8947. doi: 10.1016/j.cej.2007.07.037.
- [37] Volkert van Steijn, Chris R. Kleijn, and Michiel T. Kreutzer. Predictive model for the size of bubbles and droplets created in microfluidic T-junctions. *Lab on Chip*, 10(19):2513–8, 2010.
- [38] Zhinxin Wang. *Polydimethylsiloxane Mechanical Properties Measured by Macroscopic Compression and Nanoindentation Techniques*. PhD thesis, University of Southern Florida, 4202 E Fowler Ave, Tampa, FL 33620, USA, December 2011.
- [39] George M. Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–273, July 2006.
- [40] Wikipedia. Gel electrophoresis, 2016. URL https://en.wikipedia.org/wiki/Gel_electrophoresis.

- [41] Wikipedia. Lumped element model, 2016. URL https://en.wikipedia.org/wiki/Lumped_element_model.
- [42] Chris Willar. Comparison of circuit schematics for modelling a multiway t-junction circuit. Master's thesis, University of Waterloo, 200 University Ave. West, Waterloo, ON, Canada, April 2016.
- [43] Wen Zeng, Ian Jacobi, Songjing Li, and Howard A Stone. Variation in polydispersity in pump- and pressure-driven micro-droplet generators. *Journal of Micromechanics and Microengineering*, 25(11):115015, 2015. URL <http://stacks.iop.org/0960-1317/25/i=11/a=115015>.
- [44] Xinjie Zhang, Nan Xiang, Wenlai Tang, Di Huang, Xin Wang, Hong Yi, and Zhonghua Ni. A passive flow regulator with low threshold pressure for high-throughput inertial isolation of microbeads. *Lab Chip*, 15:3473–3480, 2015. doi: 10.1039/C5LC00647C. URL <http://dx.doi.org/10.1039/C5LC00647C>.