Expanding the Microfluidic Design Automation Capabilities of Manifold: Electrophoretic Cross and Time-Domain Simulation

by

Joshua Stephen Reid

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

Waterloo, Ontario, Canada, 2019

Author's Declaration

I hereby declare that I am the 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

Lab-on-a-chip devices are finding applications in several different fields, from point-of-care diagnostics to genome sequencing. However, lab-on-a-chip is a multidimensional field that makes it difficult for designers to have a full understanding of the entire system. There currently lacks a computer aided design (CAD) tool that allows microfluidic designers to express partial designs, only defining the parts of the system that they know and the tool determines the rest of the system while still ensuring the device will operate as expected. This results in devices being tested by physically constructing them and performing multiple design iterations should the prototype fail to operate correctly, increasing the time and cost of microfluidic design. The Manifold language was developed to address this problem by allowing the microfluidic designer to specify the parameters that they know and then Manifold solves for the ranges that the rest of the parameters can take, reducing the cognitive load required to design microfluidic devices. This thesis discusses the improvements that were made to Manifold's design capabilities to create Manifold V3.0: the addition of electrophoretic cross channel simulation and the ability to simulate designs in the time-domain in MapleSim through the use of Modelica. The Modelica design is generated automatically, creating a feedback loop that allows the designer to see their microfluidic device in operation before manufacturing a prototype. Finally, a preliminary validation of the software was performed through the comparison of Manifold's simulations to historical data collected from real microfluidic devices. This validation was structured as seven research questions that are asked of Manifold and they are each worked through using the historical data to determine if Manifold is able to answer these questions.

Acknowledgements

I would like to acknowledge the fantastic support and encouragement of my supervisor Derek Rayside who helped build by software abilities from the ground up to give me the proficiency I now have. I also want to thank my other supervisor Chris Backhouse who helped enlighten me on the intricacies of microfluidics beyond what my undergraduate in Nanotechnology Engineering showed me. Together with his graduate student Kamyar Ghofrani they were able to provide me with historical data of microfluidic devices they have tested which were used to validate the simulations performed by Manifold. In addition, I would like to thank Derek Wright for his insight into simulating microfluidic devices from his extensive knowledge of MapleSim both inside and out.

I would also like to thank the several students who helped with the development of this project, Chris Willar, Yifan Mo, Shubham Verma, Devika Khosla and Ali Abdullah. I also want to thank Stephen Chou for his work in researching the physics behind the electrophoretic cross.

Finally, I also must thank Murphy Berzish for creating the Manifold project and laying the foundation for the creation of an easy to use microfluidic simulation software.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Microfluidics is a relatively new field involving the creating of devices that can manipulate fluids in the range of microliters to attoliters ($10^{-18}L$), in channels in the range of nanometers to micrometers [5]. These have been growing in popularity with the advancement in microfabrication techniques and the reduction in cost of tools to create microfluidic devices making it easier for more people to be able to make these devices. Their popularity has also been growing as some of the different applications for microfluidics devices have been gaining increased attention, such as lab-on-a-chip devices to perform point-of-care diagnostics [6], stem cell research [7] and genome sequencing [8].

Development of microfluidic devices presents similar challenges as other hardware devices in that errors are challenging for humans to detect ahead of time. This is due to the multi-disciplinary knowledge required to fully explain the operation of these systems resulting in models that are too complex for humans to compute. While it is possible to build custom devices to test for these errors, this is time consuming and increases the cost of development. Electronic hardware devices overcame this challenge through the development of Computer-Aided Design (CAD) tools that both reduce the time and cost of designing prototypes. These alert the designer of issues that it detects ahead of time, reducing the number of design iterations required. Microfluidics currently requires the use of multi-domain software like COMSOL [9], MATLAB [10] or MapleSim [11] to design each component and their interaction with each other before simulating. However, this requires knowledge of the fluidic equations and parameters involved in the device being designed before it can be simulated. This restricts the use of these tools to people who have extensive domain knowledge in the physics of microfluidic devices that are challenging to obtain. These challenges lead to designers foregoing simulations in place of trial and error approaches that increase the number of design iterations required to get a working design.

Reviews of large scale microfluidic integration [12] [13] [10] have identified the lack of CAD tools for microfluidics as one of the major challenges facing it's scale-up and advancement.

Any tool built to fill this need must be easy to use for a designer that does not have an extensive background in microfluidics. It should also be able to work with partial designs, only being given the parameters that are known by the designer, and then be able to generate reasonable values for the unknown parameters that still allow the device to function. These assumptions should be based on the underlying physics of the system and will get more precise the more the user knows about the system ahead of time. This reduces the expertise in the underlying physics required to design a microfluidic device, lowering the cognitive load on the designer.

The Manifold project was started to address this need. It was started as a domain specific language for design automation that allows the designer to define the parameters of their device that are known. From the beginning, the (then new) dReal SMT solver [14] was selected as the initial computational backend. SMT stands for Satisfiability Modulo Theories, and an SMT solver always includes a Boolean Satisfiability solver plus some additional theories: for example, some SMT solvers have concepts of integers or arrays or strings. dReal's strength is its focus on real numbers and support for some non-linear functions. dReal is designed to analyze safety properties of cyber-physical systems. Most SMT solvers are designed to analyze computer-only systems that can be described with discrete logic, and do not need equations for physics. Some of the features and limitations of dReal will be discussed later in this document. A critical feature for dReal's usage in the design of microfluidic devices is that it can work with underspecified systems of constraints, in which there might not be enough information to find a specific solution. In these cases, dReal can find ranges of values for the unknown variables in which solutions (if any) must lie — outside of these ranges dReal proves that there are no possible solutions. These unique features of dReal enable the microfluidic designer to explore ideas during preliminary phases of design, and help to refine the design so it is more completely specified and can then be analyzed by other mathematical tools such as MapleSim or COMSOL.

Another important part of modeling a microfluidic device is determining how it will perform in the time domain in order to provide feedback on how the modelled device will perform. This is something that dReal is not suited to do, so a 3D Modelica model of the device is generated based on the parameter specifications from dReal. This model allows for simulation performed in the time domain using MapleSim. This simulation allows for the designer to see how the fluid will flow through the device over time and have an opportunity to adjust the design specified in Manifold so that it meets their needs.

The analytical representation of the physics of the devices simulated by Manifold were found from a literature search of other papers that analyzed the physics of these devices. Previous work [2] contributed the physical equations behind simulating T-junction droplet generators. In this work the equations for determining how to optimally position the detector for an electrophoretic cross were added. These are used in practise to separate analytes contained within a sample plug through a gel in order to determine what analytes are present, and the position of the detector is critical to be able to distinguish the individual analytes.

In order to perform a preliminary validation of the simulations performed by Manifold, historical data was received from Professor Backhouse of past tests run on physical microfluidic devices containing a T-junction droplet generator and ones containing an electrophoretic cross. The known parameters of the devices used to collect this data were then input into Manifold and the output was compared with the actual measured output parameters of the devices to determine Manifold's accuracy. This is presented as seven research questions that are asked of Manifold and they are each worked through using the historical data to determine if Manifold is able to answer these questions. Finally, some of the design limitations that are imposed on Manifold due to the equations used are also presented.

1.1 History

The V1.0 architecture of the Manifold project was as its own hardware description language. This had its own parser and interpreter created in Java and the syntax of the language was specialized for the design of microfluidic devices. This language acts as the front-end of Manifold and is translated into an intermediate representation which outlines all of the components in the device, the connections between these components, and their parameters. This was then read by different back-end tools which generate information that is useful for the microfluidic designer. The major back-end tool created in that work was the implementation of the dReal SMT solver to produce ranges of values for every parameter in the system.

Manifold was improved by Nik Klassen et al. [15] where the syntax of the front-end language was expanded with several useful features, including a type system, a module system, and making tuples first-class values. Also, the initial framework for producing the Modelica design through the use of MapleSim's Java API called OpenMaple was created. This was called Manifold V2.0.

Atulan Zaman [16] worked on strengthening the simulations of Manifold in the time domain in MapleSim. This was done by drawing parallels between microfluidic and electronic devices, such as a channels acting like a resistor by dropping the pressure of the fluid from one side to the other. These devices were then manually simulated in MapleSim and the results compared to physical devices. This will be named Manifold 2.5.

1.2 Contributions

In this section I will outline the contributions to Manifold to create Manifold V3.0 that will be discussed in this work.

1.2.1 Extensions on Previous Work

This work is an extension on the work performed by Murphy Berzish, Atulan Zaman, Nik Klassen, and others as described in the previous section. The analytical equations implemented to simulate the electrophoretic cross in this work were an extension of work performed by Stephen Chou in the appendix: Stephen Chou Electrophoretic Cross Reports. In this report he researched the underlying physics behind capillary electrophoresis and the equations required to determine the position of a detector in an electrophoretic cross in order to detect every analyte.

1.2.2 Supervisory Contributions

Throughout the course of this work I oversaw the work of four undergraduate research assistants and one full time undergraduate intern. Together they made significant contributions to the Manifold Python library. Ali Abdullah implemented the method to generate the Modelica output. Chris Willar implemented the intermediate representation for Manifold. Devika Khosla contributed to the SMT translation of the electrophoretic cross. Shubham Verma implemented a way of more easily handling the fluid constants and Yifan Mo worked on improving the website for Manifold.

1.2.3 Collaborative Contributions

We collaborated with Prof. Backhouse and his student Kamyar Ghofrani to provide us the data from the physical devices they have built and tested in the past to use for validating the output of Manifold. We also collaborated with Derek Wright who provided the MapleSim model of the T-junction droplet generator that was used to produce the Modelica output.

1.2.4 Independent Contributions

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

Creation of Python Library

Previously Manifold was written in Java and was its own external domain specific language with its own syntax, intermediate representation and back-end computation. This additional complexity presented a barrier to entry for microfluidic designers wanting to use the software. In order to reduce this, it was decided that Manifold should have a Python implementation as a library that can be imported as used within Python. It is our belief that Python is an easier language to learn for microfluidic designers due to its use in research for data analysis [17] and lab automation [18]. This complexity also made it harder for developers to contribute to the project as before this project was a total of 20,000 lines in Java. This has now been reduced to 1,500 lines of Python code, and it no longer requires compilation, making it easier for developers and researchers to contribute back to this project.

Addition of Electrophoretic Simulation

Previously Manifold was able to simulate microfluidic channels and a T-junction droplet generator. With this work the ability for Manifold to simulate an electrophoretic cross was added. This allows for the simulation of a device which generates its own droplets and then analyze the droplets on the same device.

Preliminary Validation of Manifold Output

In collaboration with my co-supervisor Professor Chris Backhouse we were able to perform a preliminary validation of the output of Manifold on his historical data from both T-junction droplet generators and electrophoretic cross devices. This preliminary validation is structured as seven research questions that are asked of Manifold and they are each worked through using the historical data to determine if Manifold is able to answer these questions. The T-junction case study is closer to a specific historical device [1] than is the electrophoretic cross study, which is inspired by historical devices but does not describe any one specific historical device.

Automated Generation of Model File to Simulate Channels in Time Domain

In the past work on the Manifold project performed by Atulan Zaman [16] he presented the ability to perform time domain simulations for microfluidic systems designed in Manifold using MapleSim. However, these models were created manually within MapleSim based on the output of the SMT solver from Manifold which required additional work from the user. In order to speed up the simulation and reduce user input, a function was created that converts the schematic designed by the user into a Modelica file that can be read by many other multi-domain modeling software like MapleSim. These programs can then run the simulation of these devices automatically, providing the designer with much more diverse information about their devices.

Chapter 2

Materials and Methods

This chapter will discuss how the Manifold software framework v3.0 evolved from previous versions, along with the manufacturing and testing procedures for the physical devices. The first section will provide background on the motivation for creating this software and how it operates. The second section will discuss the structure of the current framework and what changes were made over the previous version. The third section will discuss the creation of the output Modelica file to allow for time domain simulations in MapleSim. Finally, the manufacturing and testing procedures will be described.

2.1 Manifold Overview

Manifold is a design automation tool to help microfluidic designers simulate the behaviour of their designed microfluidic devices before fabrication. The goal of this is to automate microfluidic device design, reducing the number of design iterations that need to be performed by designers before getting a working design. The need for a software such as this was identified as a barrier currently for large scale integration of microfluidic devices [12, 13, 10]. This project started as a its own Domain Specific Language (DSL) for hardware description that drew inspiration from VHDL, a language the simulates the inputs and outputs of digital electrical circuits given the fixed inputs and circuit layout. Manifold is instead a language to simulate the output of a microfluidic device given the inputs and the layout. In this case the inputs would be properties of the input fluids, the pressures and flow rates of the fluid travelling through channels of different sizes. This is analogous to voltage, current and resistance used in electronics simulations.

This software models microfluidic devices using 4 primitive components that are used as the building blocks of a microfluidic device:

1. *Nodes:* can be fluid input/output, or can be logic devices like switches, logic gates or bends
2. *Ports:* which describe a type of node where fluid can input or output
3. *Connections:* channels which connect two nodes together
4. *Constraints:* which describe design rules or goals that are too complex to be described in terms of the other three primitives

The parameters of a physical device can be broken down into two different categories, those that the user can modify and those that are *consequential parameters* the system. Parameters such as the channel length, width or pressure can be modified when they manufacture or operate the device, called *design parameters*. *Design parameters* can also be broken down into those that are static, such as the dimensions of the device, and those that are dynamic that the user controls during testing such as pressure. Consequential parameters, such as the droplet size and the flow rate cannot be changed directly, only by modifying other parameters of the device can these be changed. For each of these parameters, there are ones that the user specifies and ones that they do not specify. It is the goal of Manifold to generate ranges of values for those unspecified values, as well as computing ranges for the consequential parameters that the user cannot directly control in order to solve for the system. The user can also specify desired values for the consequential parameters of the system in order to limit the ranges of the design parameters that the user can modify in order to get that desired value. This is shown in Table 2.1.

Note that this classification of parameter types is conceptual within the design process. From the standpoint of the computations, variables are just variables and this classification is irrelevant. It is in the user's design process that this classification is important.

Parameter Type	Influenced By	Example
Static Design	Manufacturing	Channel Width
Dynamic Design	Operations	Pressure
Consequential	Output	Droplet Size

Table 2.1: Examples of the parameter types

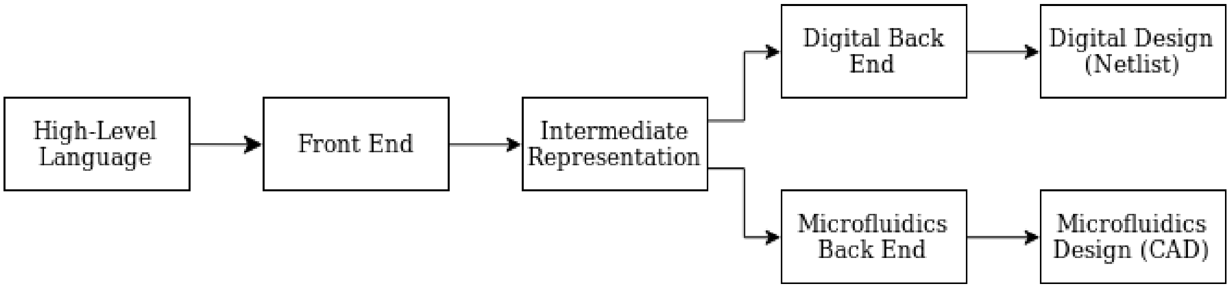

Figure 2.1: The architecture of the Manifold V1.0 toolchain [2].

The front-end of Manifold was the programming language itself which has its own syntax with a parser and interpreter written in Java. This interpreter converts the Manifold language into an intermediate representation (IR) that can be read by different algorithms used to simulate the microfluidic device. The IR has a syntax similar to JSON, using key-value pairs to store all of the parameters of each primitive component. This architecture allows for multiple simulation components to be added, only requiring them to each read this IR that is easy for computers to understand, unlike the Manifold language is instead easy for humans to understand.

The back-end of Manifold performs interprets the IR into output that is beneficial for designers. The main back-end tool uses the Satisfiability Modulo Theories (SMT) solver dReal which takes the components of the microfluidic device, applies all of the constraints along with addition constraints from fluid dynamics. Through the use of δ-satisfiability this returns a range of values that each parameter can take while maintaining that all of the constraints are true [14]. The flow of information between each part of Manifold is shown in Figure 2.1

2.2 Manifold v3.0 Architecture

With the end users of Manifold intended to be microfluidic designers, it is ideal for any simulation software to be easy to use and integrate with as many of their current systems as possible. Since Python is currently used in research for data analysis [17] and lab automation [18] it is assumed in this work that it is an easier language to learn and will integrate better with the designer's current systems, resulting in them being more likely to use it. As such, it was decided that a new front-end for Manifold should be created as a Python library since this will have the potential of integrating better with the designer's current systems.

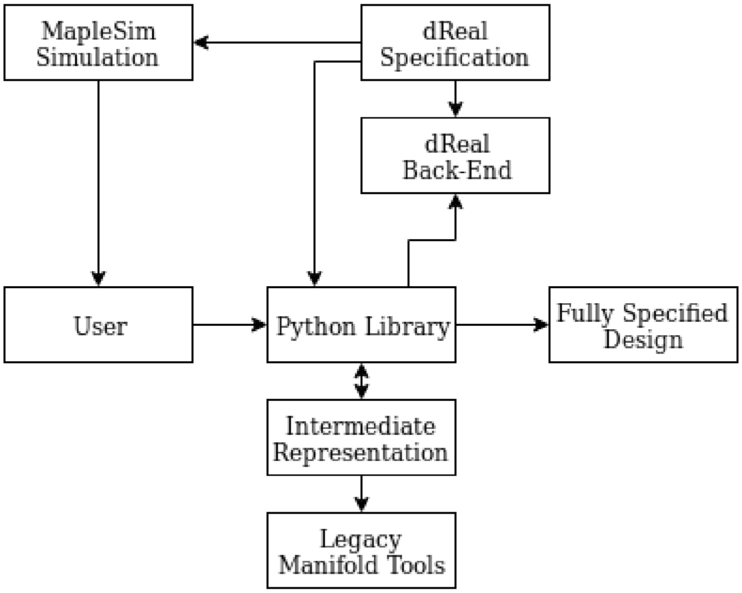

Figure 2.2: The architecture for V3.0 of the Manifold toolchain.

As a consequence of this change, the simulation back-end was also rewritten in Python to allow for easier testing and sharing of the code. The IR code is still output from the Python code, so it can still integrate with the existing Manifold Java back-end tools. However, due to the dReal requiring being built from source this caused additional complexity. In order to mitigate this, a Docker image has been created for Manifold that contains the Python library and all of its dependencies which can be pulled from online and used by anyone. See Appendix §B Replicating Results.

The way that the simulation is computed was also changed, with the design of the device being stored using a different data structure. Previously, in order to translate every component created by the designer a visitor design pattern was used. This would visit all of the primitive type objects and call a translate method on them. This method would translate each component into equations asserted in the SMT solver based on the type of component and its user defined parameters. In the new design, the layout of the microfluidic device is stored as a directed graph where the nodes of the graph are nodes or ports within the device, edges in the graph are the channels within the device and the constraints are stored within each of these. This graph is then traversed starting at each input port and ending at an output port, translating each primitive type into constraints based on the parameters of that component and the underlying fluidic equations.

The ability to simulate additional microfluidic components was also added. Originally, Manifold focused on simulating fluid flow through channels and the physics surrounding a T-junction droplet generator. This is a popular component to microfluidic devices, as it is used to create many droplets from a fluid sample and each droplet can be an individual test for an experiment. The new component simulation that was added was an electrophoretic cross, which enables controlled movement of droplets and analysis of the contents of that droplet. This analysis determines the size of the analytes in the droplet to characterize what is present. The user can specify different parameters of the analytes of interest present in the sample being analyzed, such as the diffusivity, initial concentration, hydrodynamic radius and charge of the analytes. The underlying physics of this simulation is explained in more detail in Chapter 3. The new structure of Manifold is shown in Figure 2.2.

2.3 Manifold Output to Modelica

Modelica is described on their association's website[19] as a "multi-domain modeling language for component-oriented modeling of complex systems", which means that it is a language used to model the various physics occurring in real world systems. For microfluidics it can be used to simulate the flow of fluids in channels to determine the pressure, flow rate and concentration over time. However, designing the channels in these software tools along with having to enter the fluidic equations of the system manually can be time-consuming, increasing the likelihood that this simulation is foregone in favour of trial and error testing.

Manifold's architecture allows for partial design of devices from the designer to reduce the required cognitive load, but it lacks the ability to perform simulations in the time domain. Modelica can bridge this gap, by having Manifold generate a Modelica file that replicates the designed device allows for tools like MapleSim to load this Modelica file and perform time domain simulations. This also provides the user an interactive interface that allows them to make small tweaks to the model as needed, and also provides time domain simulations of the device.

Although there are Modelica libraries for Python, unfortunately none of them allow for generation of Modelica code, only modification of existing files through OpenModelica's OMPython. As a result of this, a sample Modelica file was created that simulates a T-junction droplet simulator in MapleSim and calculated values from the SMT solver are added when Manifold is ran. These calculated values are added by writing lines of text to a file due to the use of some libraries that are proprietary to Maple and are not supported in OpenModelica.

2.3.1 Feedback to Manifold

The Modelica output from Manifold is able to be opened in MapleSim to provide the user with time-domain simulations of their microfluidic device. This provides the user with a better idea of how their system will operate, and also presents an opportunity to provide feedback into Manifold in order to produce more precise simulations of what the user is looking for.

MapleSim is a system-level modeling application, meaning that it can calculate the physics of systems across the entire system. Other multi-domain simulation software used to simulate microfluidic systems [20], such as COMSOL, use finite element analysis in order to simulate the physics of the device. These speed up the computation of complex equations by dividing up the space into multiple two or three-dimensional sections. These sections, or finite elements, only require the equations of the systems to be solved at their boundaries, producing a discrete result that, depending on the density of these sections, can come close to being continuous.

MapleSim was chosen to perform the simulations of the devices because of its support for Modelica, making it possible to generate files that describe the designed system because of its open source syntax. A model of the T-junction droplet generator was created in MapleSim and this was exported as a Modelica file. Using this file, Manifold takes the ranges of parameters that are generated by dReal to modify the parameters of this model and create a Modelica file for the device specified within Manifold. Opening this file in MapleSim allows for simulations to be ran over time and space.

This simulation provides a visualization of how the device looks and how it will operate. The user interface of MapleSim also allows for the user to interact with the model directly in its current form, instead of requiring the user to design it all from the ground up. This allows the user to modify the parameters and see the resulting effect to refine the design until it produces the desired result. If they are unsure of what to modify, they can update some of the parameters within Manifold with new values based on the range provided by the initial simulation to get a more refined simulation.

2.4 Design of Microfluidic Devices

A preliminary validation of the simulation output was performed by comparing the simulation results with historical data collected from physical microfluidic devices. Data was used from microfluidic devices that perform both droplet generation at a T-junction and analyte separation in a electrophoretic cross. The following section will outline how the devices that the historical data was collected from were manufactured and tested.

2.4.1 Manufacturing Process

This section will outline the manufacturing process for making the T-junction droplet generator and the electrophoretic cross devices. The techniques used to manufacture each involve laser etching of PMMA (acrylic) to create the channels and then bonding layers of acrylic together to create the device. The T-junction droplet generator device has additional active components on the device that require additional layers which will be outlined in the following section. For both devices, the laser engraving machine used was the Universal Laser system model VLS2.3, which is a bench-top CO_2 laser engraving machine. A 30 W 10.6μm CO_2 laser module is used to generate the high power laser beam and is directed though to the computer-controlled stage. According to the manufacturers specification [21], the maximum speed is 32 inches/second. The laser operates in pulse mode with a maximum pulse density of 1000 pulses per inch. Through a 2 inch focusing lens, the laser beam is focused into a 25μm spot as per the manual [22].

T-junction Droplet Generator

These devices were manufactured using a laser cutter to etch the channel designs into acrylic. This manufacturing method allows for the rapid creation of microfluidic devices at a low-cost. This is also part of the motivation to create this software, as the manufacture of these devices has higher variability compared to devices manufactured using polydimethylsiloxane (PDMS) soft lithography [23, 24]. Through the use of this simulation software together with the expected variability in the parameters of the channel it will tell the manufacture within what range each parameter can be within while still being certain that the device will work.

After channels are etched into a sheet of acrylic it is thermally bonded to another piece of acrylic. The layers are pressed together through the use of a vacuum sealed pouch. Vacuum sealing accomplishes two tasks, first it applies uniform pressure across the entire

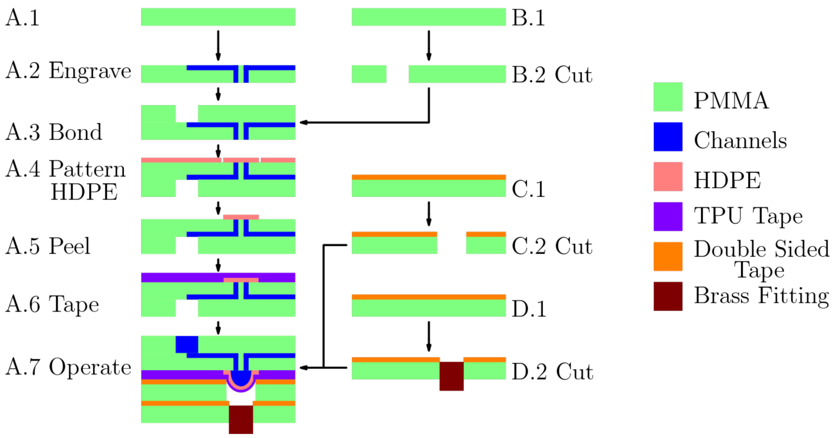

Figure 2.3: (a) The fabrication process cuts and pattern pieces of PMMA: one with channels and vias (A.1), and another with cut-outs alone (B.1). The pieces are engraved, cut and thermally bonded at $120°C$ for 30 minutes (Steps 2, 3), and then, a layer of HDPE is applied and laser patterned (A.4, A.5). TPU tape is then applied (A.6), and two capping layers are then added using double-sided tape. The outermost capping layer contains a brass fitting that is directly connected for pneumatic actuation [1].

device and second it evacuates any gas produced by outgassing from the heating of the plastic potentially causing bubbles in the bonding.

The TPU layer is taped on top of the acrylic to bond to another etched piece of acrylic. This creates a flexible layer that can be used over large etched areas to create a capacitor. Air is pumped into the etched cavity through a brass barbed fitting attached on the other side in order to control the pressure to create a capacitor. This also allows for it to be used as a valve by blocking the channel when the pressure is greatly increased. A cross section of the steps performed to manufacture the microfluidic devices are shown in Figure 2.3

This manufacturing technique also allows for the creation of multi-layer microfluidic devices by laminating multiple etched layers of acrylic together and connecting them with through holes in the acrylic, similar to a via in printed circuit board manufacturing. Multi-layer microfluidic devices are capable of doing complicated processes in a smaller sized device, making it possible to make more portable lab-on-a-chip devices.

Electrophoretic Cross

The manufacturing process for these devices is outlined in detail by Tianchi in his work with Professor Backhouse [22]. First, a sheet of acrylic is etched using the laser cutter to create the 4 channels and wells of the electrophoretic cross. Next, this piece along with a smooth acrylic piece the same size are surface treated by heating them for 10 minutes at 130^oC in a forced-air oven (Model No. 6916, Fisher Sci.) and then cooled at 80^oC for 1 hour. These two pieces of acrylic are then stacked and clamped together with large paper clips and placed back in the oven at 110^oC for 1 hour to anneal them, and again cooled down to 80^oC for another hour. Brass fittings are then attached to each of the wells in order to allow for connecting tubes for pumping fluids and to allow for electrical connection to the channels.

2.4.2 Fluid control

Conventionally, fluids are pumped through microfluidic devices using either a peristaltic pump or a syringe pump. Peristaltic pumps use a piece of tubing with one end in a reservoir of the liquid to be pumped and the other end connected to the input port on the microfluidic device. Fluid is pumped by squeezing the tube in the direction that the fluid should travel. This is done through the use of arms on a motor pressing against the tube in a circular motion. A syringe pump uses a different mechanism, instead of squeezing the tube it mechanically actuates the syringe in a more precise way than a human. This is done through the use of a linear actuator, usually a threaded rod, that closes the plunger at a controlled speed. This then requires attaching the tubing to the syringe and the other end to the microfluidic device.

A new design of pump was implemented for this microfluidic device that uses the TPU valves to actuate the flow of the fluid directly on the device. This works with three valves in series to pull the liquid in, capture it and then squeeze it out, similarly to a peristaltic pump. This provides a compact mechanism to pump fluids that only requires a source of pressurized air. The benefit of this mechanism is that is has very little dead volume, the volume of liquid left behind in the tubing for both the peristaltic and syringe pumps. This dead volume results in wasted fluid, which when the sample or solvent is very valuable can result in a large additional cost. However, this design does generate pulsatility in the fluid input into the device similar to a peristaltic pump. To compensate for this, a series of fluidic capacitors and resistors are used to remove pulsatility. The mechanism to do this is similar to an electric low-pass noise filter in electrical circuits that removes high frequency electromagnetic noise.

Valve	A1	A2	A3	B1	B2	B3
Step 1	X	O	O	O	O	X
Step 2	O	X	O	O	O	X
Step 3	O	O	X	O	O	X
Step 4	O	O	X	X	O	O
Step 5	O	O	X	O	X	O
Step 6	O	O	X	O	O	X

Table 2.2: The two pumps (A and B) each follow the same sequence of steps, but are configured so that each operates its set of three valves only when the other set is not changing (where O represents the valve open and passing fluid, and X represents the valve being closed and sealed). The cycles are completely independent here, but could operate with any degree of overlap or phase shift.

2.4.3 Testing Procedure

T-Junction Droplet Generator

The following procedure is from Strike et al. [1] where they outlined the work that was performed to manufacture and test the T-junction droplet generator devices.

The computer-controlled pneumatic system was controlled with an integrated microcontroller that takes commands from a computer connected via a universal serial bus (USB) link. The pumps can be operated independently and are configured so that only one is operating at a time. A pause time after each step can be specified, and values of 0.06, 0.15 or 0.3 seconds were used, corresponding to pump frequencies of 2.5, 1.0 and 0.5 Hz, respectively. The frequency is the inverse of the time taken to run through the six steps required to complete a pumping sequence for each of the two pumps shown in Table 2.2.

Before forming droplets with the device, the aqueous pump was cycled (with the third valve of the oil pump closed) until the water reached 2.5 mm past the T-Junction. The oil pump was then cycled, with the third valve of the aqueous pump closed, until the entire channel past the T-junction was filled with oil. After those steps were performed, the device was operated with the pumping sequence in Table 2.2 for three min, such that droplets could be formed and fill the device. During this time, the pressure settings on the regulators were adjusted to achieve the optimal pressure settings. Then, the device would be tested with the pumping sequence in Table 2.2 and the pressure settings adjusted for optimal droplet formation. The device was then operated for one minute before any videos or pictures were taken of it. On occasion, some valves were not immediately usable. In

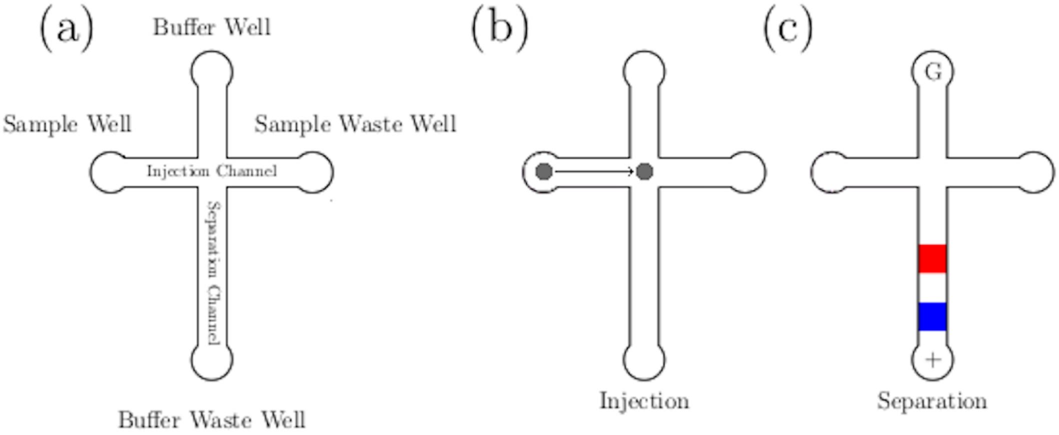

Figure 2.4: Demonstration of an electrophoretic cross. a) Annotated drawing of the electrophoretic cross device: Sample well is loaded with DNA sample, while the rest of wells are loaded with electrophoresis running buffer. b) Injection step: the sample plug is pumped towards the waste well, c) Separation step: After injection, the electric field is applied across the separation channel. Analytes within the sample plug are pulled towards the buffer waste well based on their properties.

this case, a pressure of 20kPa was applied for as much as 5 min to release the membranes, after which the valves operated normally.

Electrophoretic Cross

The operation of this device is shown in 2.4 which is a modified demonstration of the operation of an electrophoretic cross from previous work done by Backhouse and Ma [22]. Droplets containing a sample of the DNA to be tested are first injected into the channel. The droplet injection into the sample channel is controlled electrophoretically so that the droplet is positioned in front of the separation channel when injected. After injection the voltage is turned on and the droplet begins to be dragged through the separation channel.

Chapter 3

Microfluidics Background

This chapter will cover the physical equations that were used to apply constraints within Manifold for the T-junction droplet generator and the electrophoretic cross. Lab-on-a-chip devices are a large field with many different applications, so currently these two applications are provided by Manifold as representative examples to demonstrate how Manifold can benefit designers.

For all devices, the physics of the pressure, flow rate and resistance in each channel are constrained by the equation $\Delta P = QR$. It is also enforced that the flow rate into and out of every node is equal as this must be true to ensure conservation of mass assuming the fluid is incompressible. The resistance of each channel is determined from the dimensions of that channel (length L, width w and height h) and the viscosity (μ) of the fluid travelling through that channel according to Equation 3.1 [25].

$$R = \frac{12 * \mu * L}{w * h^3 * (1 - 0.630h/w)} \tag{3.1}$$

Some of the *design parameters* of the device that are not specified by the user can potentially take any numeric value within dReal. This is due to the system being under-specified, when not enough parameters are specified then the ranges of feasible values that the parameters can take becomes quite large. This provides a poor output to the user as these parameters are too broad for a designer to reliably pick a single value and will require them to specify parameters that they may not know. As a result, all parameters when not specified by the user are constrained to be within a physically reasonable range of values. These values are shown in Table 3.1.

Parameter	Symbol	Units	Lower Limit	Upper Limit
Channel Length	L	m	1E-09	1
Channel Height	h	m	1E-09	1
Channel Width	w	m	1E-09	1
Channel Flow Rate	Q	m^3/s	1E-15	1
Channel Droplet Volume	V	m^3	1E-15	1
Channel Resistance	R	m	0	1E+09
Port Pressure	P	Pa	1E-06	1E+06
Port Flow Rate	Q	m^3/s	1E-12	1E-03
Port Viscosity	μ	$Pa * s$	1E+02	1E-04
Port Density	ρ	kg/m^3	5E+02	2E+03
Port X	x	m	1E-09	1

Table 3.1: The physically reasonable upper and lower limits that each unspecified parameter is constrained within. In devices with multiple channels and ports these variables would be instantiated individually for each.

3.1 T-junction Droplet Generator

The T-junction droplet generator is the first microfluidic device that Manifold was originally created to simulate. The underlying equations used to assert constraints on the parameters for the SMT solver were translated into Python in this new implementation. These equations have been explained in the previous theses written on Manifold [2] [16] and will be summarized in this section.

A T-junction droplet generator consists of 2 channels meeting in the shape of a T. The top channel in the T contains the fluid of the continuous phase, which for droplet generation is generally an oil. The bottom channel of the T-junction contains the dispersed phase which is the fluid of interest that droplets will be made of. This fluid contains the sample of interest and is usually an aqueous solution due to the samples being biological in nature such as bacteria, proteins or DNA [26]. These types of samples require many tests to be ran due to the nature of biological samples producing stochastic results. In order to combat this the results are usually looked at statistically, requiring large sample sizes to gain greater confidence in the results.

In order to produce droplets, the parameters of the device must be set such that it is in the squeezing regime, which is when the dispersed solution will flow into the continuous phase and droplets will break off periodically, producing droplets. This process is visually shown in Figure 3.1 along with the names of the different channels.

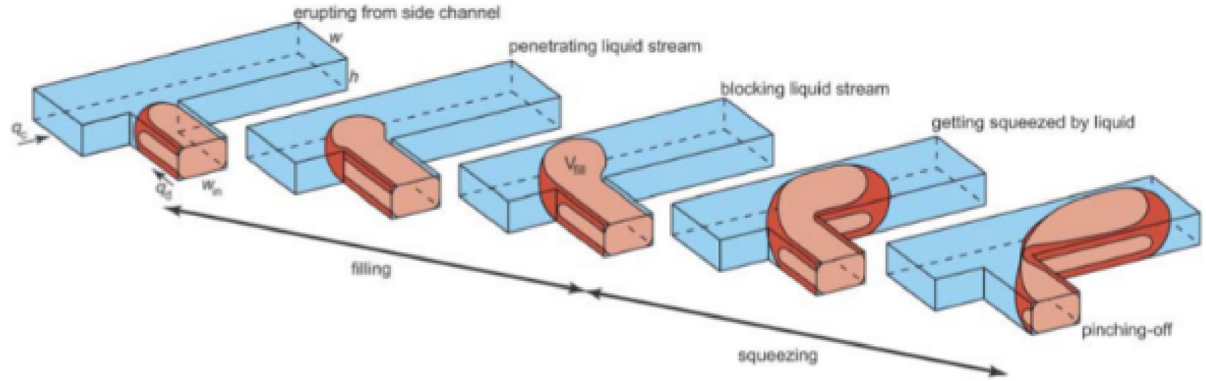

Figure 3.1: Formation of a droplet at a T-junction droplet generator. The design parameters are h, the height of the junction; w and w_{in}, the width of the continuous and dispersed channels respectively; and q_c and q_d, the continuous and dispersed flow rates [3].

3.1.1 Underlying Physics

The physics at play in determining the size of droplets that will be created involves the application of fluidic and geometric equations. Theoretical and experimental analysis of this has been performed by van Steijn et al. [3] and Garstecki et al. [27] which was used as the basis of the equations used to assert constraints on the parameters on these junctions. The equations derived in these works specify the physics of a T-junction droplet generator producing droplets in the squeezing regime and these equations will be summarized in this section. This regime requires the capillary number (Ca) is less than 0.01 and that the ratio of the width of the dispersed channel to that of the continuous channel is not small. In these equations, this regime is specified by the ratio of the flow rate in the dispersed channel to that of the continuous channel, however the designer will generally care about the pressure because that is what they can control. The flow rate and pressure will be related within Manifold based on the equation $\Delta P = QR$ specified earlier, using the resistance of the channel the input port is flowing into. For reference, the names of each of the channels are shown in 3.2 which shows an example of a T-junction droplet generator microfluidic device in the squeezing regime, when droplets of the dispersed phase are being created periodically.

The final volume of a droplet produced by a T-junction droplet generator (V) is broken into two parts in the derivation by van Steijn et al. [3]. The first part (V_{fill}) is the volume that fills the continuous channel as the dispersed fluid is pushed into it, normalized by hw^2. The second part $(\alpha \frac{q_d}{q_c})$ is the volume that is injected into the channel as the droplet

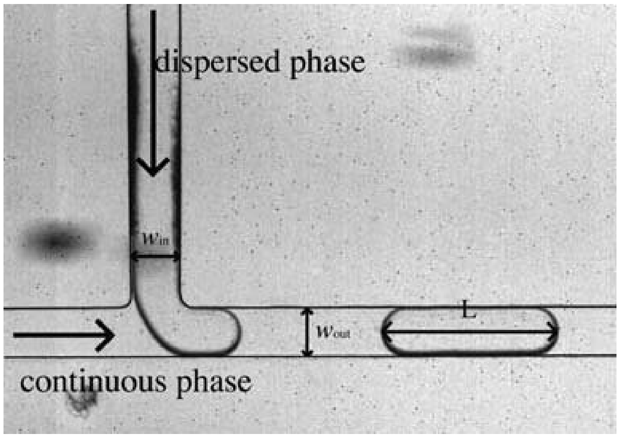

Figure 3.2: Magnified view of a T-junction droplet generator with the continuous and dispersed channels labeled, with the output droplets flowing to the right [4].

is pinched off. The overall equations for the volume can be calculated using the following equation [3],

$$\frac{V}{hw^2} = \frac{V_{fill}}{hw^2} + \alpha \frac{q_d}{q_c} \tag{3.2}$$

where V/hw^2 is the dimensionless volume of the droplet normalized with hw^2. V is the volume of the droplet, h is the height of the channels and w is the width of the continuous channel. The volume that initially fills the continuous channel is V_{fill}, which is also normalized by be dimensionless by hw^2 to become V_{fill}/hw^2. The volume that is injected while the droplet being pinched or squeezed off is $\alpha q_d/q_c$. q_d is the flow rate of the dispersed channel and q_c is the flow rate in the continuous channel. α is a constant that can be determined from the following equation [3],

$$\alpha = \left(1 - \frac{\pi}{4}\right)\left(1 - \frac{q_{gutter}}{q_c}\right)^{-1}\left(\left(\frac{R_{pinch}}{w}\right)^2 - \left(\frac{R_{fill}}{w}\right)^2 + \frac{\pi}{4}\left(\frac{R_{pinch}}{w} - \frac{R_{fill}}{w}\right)\frac{h}{w}\right) \tag{3.3}$$

Where,

$$R_{fill} = max(w, w_{in})$$

$$R_{pinch} = w + w_{in} - \left(\tfrac{hw}{h+w} - \epsilon\right) + \left(2\left(w_{in} - \left(\tfrac{hw}{h+w} - \epsilon\right)\right)\left(w - \left(\tfrac{hw}{h+w} - \epsilon\right)\right)\right)^{\frac{1}{2}}$$

$\frac{q_{gutter}}{q_c} = 0.1$

Where q_{gutter}/q_c is the fraction of the flow rate of the continuous phase that instead of pushing the bubble or droplet downstream bypasses the bubble or droplet and flows along the corners of the main channel. Due to the constraint that w is always larger than w_{in}, then R_{fill} is always equal to w, meaning that V_{fill} is a quarter circle of radius R_{fill}. It can be seen that through these equations the volume of the droplet can be characterized by the dimensions of the channels along with the flow rate of the dispersed and continuous channels. In order to simplify the equation, Manifold enforces that the width of the dispersed channel must be less than or equal to the width of the continuous channel. This simplifies the filling volume of the droplet to,

$$\frac{V_{fill}}{hw^2} = \frac{3\pi}{8} - \frac{\pi}{2}\left(1 - \frac{\pi}{4}\right)\frac{h}{w} \tag{3.4}$$

Applying all of these formulas as constraints within dReal will provide a range of values for the volume of a droplet along with the other parameters of the channels used in these equations for a T-junction droplet generator.

Looking at this system of equations it appears that they do not consider pressure, which is a *dynamic design parameter* that microfluidic designers generally control to operate the device. However, since dReal handles all of the equations described so far simultaneously, the pressure is considered for the T-junction droplet generator device indirectly. The equations being applied in this case are $\alpha\frac{q_d}{q_c}$ and $\Delta P = QR$ where Q is q_d and q_c when that equation is applied as a constraint to the dispersed and continuous channels respectively. This means that increasing pressure will in turn increase the flow rate, which will then impact the droplet volume. This shows how more complex interacting systems can be handled by a tool such as dReal.

Microfluidic designers may also be interested in knowing what the monodispersity of the droplets produced is, which is the consistency of the size of the droplets. For most applications a high monodispersity is desired, producing droplets that are consistently the same size. Unfortunately, the theory used here does not support determining the monodispersity as the analytical solution only determines a single solution for the droplet volume given the input parameters. However, Manifold can indirectly provide this information as the designer refines their design they will observe how changing different parameters of the device result in the output droplet volume changing. This can help to determine which parameters could be impacting the monodispersity the most in the designer's experimental setup.

3.1.2 Manifold Implementation

The conditions outlined in the derivation of the above equations are enforced as constraints in the SMT solver. The constraints are summarized as follows:

- The height of each of the three channels in the T-junction are constrained to be the same height
- The width of the continuous and output channels were constrained to be the same width since they should connected together in a straight line
- The width of the dispersed channel was constrained to be equal to or less than the width of the continuous channel
- The sharpness of the T-junction ϵ is constrained to be equal to 0, so this assumes the channel is a perfect T with no rounding
- Viscosity ν in the continuous channel is constrained to be equal to the output channel since the bulk of these are the same fluid
- The flow rate Q of the output channel is constrained to equal the sum of the flow rate in the dispersed and continuous channels to follow conservation of mass
- The continuous and output channels are constrained to be in a straight line
- The droplet volume is constrained to equal $V/hw^2 = V_{fill}/hw^2 + \alpha q_d/q_c$

3.2 Electrophoretic Cross

One of the new contributions in this work is the addition of the ability to simulate an electrophoretic cross. These are microfluidic devices created using two intersecting channels, one where fluid containing the samples and the other with electrodes at both ends containing an gel. Droplets are positioned at the intersection using electrophoresis, and then the electrodes are powered to a high voltage which then drags the analytes through the gel to separate the analytes within based on their size. This allows for the analytes within the sample to be individually characterized, which is useful in biology to determine if samples of bacteria contain a certain gene or protein.

The gel filled channel in the electrophoretic cross is called the separation channel which is where the sample will be dragged through the gel and separated into its analytes. The other end of this channel contains the cathode for electrophoresis and is much shorter

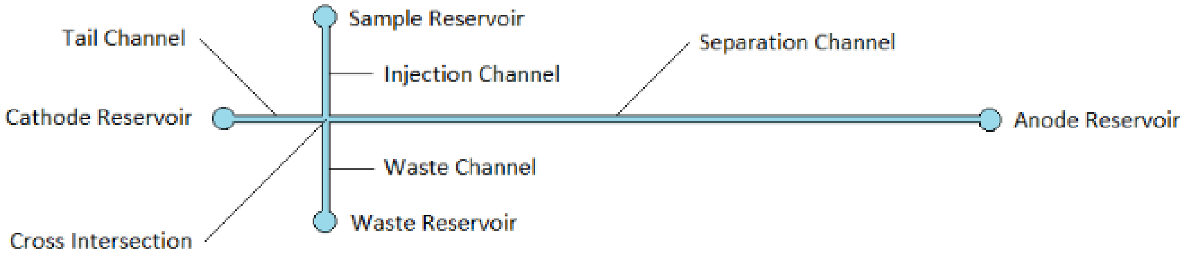

Figure 3.3: The design of electrophoretic cross used in this work with the names of each of the ports and channels.

than the separation channel. The channel where the samples are pumped into the cross is another short channel called the injection channel, and the other end is the waste channel where used samples and fluid leave the electrophoretic cross. Samples are aligned with the separation channel by calibrating the pumping time and pressure required move a sample from the reservoir into the electrophoretic cross. The design of the electrophoretic cross is shown in Figure 3.3.

3.2.1 Underlying Physics

The following equations are summarized from work done by Stephen Chou for the Manifold project, his full report can be found in Appendix Stephen Chou Electrophoretic Cross Reports. To begin, the equation governing the movement of particles in a bulk fluid is,

$$\overrightarrow{v} = \mu \overrightarrow{E} \tag{3.5}$$

where $\overrightarrow{v}$ denotes the velocity of the molecule, $\overrightarrow{E}$ denotes the electric field in which the molecule is placed and μ represents the mobility of the molecule [28]. It should be noted that this formula assumes that the electric field is small, so situations where this is not the case may find results that are less accurate. The electric field is able to move particle through two different flows: electrophoresis and electroosmosis. Electrophoretic movement is the movement of the particles themselves, where electroosmotic flow is due to the movement of the bulk fluid surrounding the particles. Charged particles move due to the interaction of their electric field with the surrounding electric field, with positive particles being attracted to the negative electrode and vice versa. This will also cause the bulk fluid to move due to the molecules in it also having a charge. The total mobility considering both of these movements is [29],

$$\mu = \mu EP + \mu EOF \tag{3.6}$$

where μEP and μEOF denote electrophoretic mobility and electroosmotic mobility respectively. This shows that the total mobility is linear with respect to the strength of the electric field. The electrophoretic mobility of a particle is dependent upon its charge, size and shape as well as the viscosity, pH and density of the bulk fluid. For a spherical particle with charge q and radius r in a bulk fluid with viscosity ν, the electrophoretic mobility can be approximated as [29],

$$\mu EP = \frac{q}{6\pi\nu r} \tag{3.7}$$

This equation shows that a particle with a smaller radius r, a higher charge q or in a fluids with a lower viscosity ν would have a larger mobility.

Electroosmostic mobility is also dependent on several parameters of the fluid, as well as being dependent upon parameters of the material surrounding the bulk fluid [28]. This includes the viscosity, density, zeta potential, and electrical permittivity. The electroosmotic mobility will generally be a parameter that is known by the designer as it will be specified by the manufacturer of the material that they are making their device out of. For example, glass is approximately $1.1E - 7m^2/Vs$ where for PMMA it is approximately $4.0E - 8m^2/Vs$ [30]. In order to simplify the following calculations, an approximation of the electroosmotic mobility is used that is in the correct order of magnitude for most materials [31],

$$\mu EOF \approx 1.0E - 8\frac{m^2}{VsW} \tag{3.8}$$

It is important to note that electroosmotic flow is not completely uniform across the entire cross-section of a channel. At distances that are relatively close to the channel wall, those on the order of the Debye length, the velocity of the bulk fluid due to electroosmotic flow approaches zero. However, since the cross-sectional radius of a channel on microfluidic devices is usually several orders of magnitude greater than the Debye length, it is reasonable to treat electroosmotic flow as being uniform across the cross-section of a channel in the context of microfluidics [28].

3.2.2 Analyzing a Sample Plug

The electrophoretic cross uses these electrophoretic and electroosmostic forces for the analysis of a sample containing multiple analytes. These analytes can be different proteins or different strands of DNA in a biological sample. This is done to separate each analyte based on their sizes or chemical properties. This allows for different characteristics of the sample to be determined, such as whether a certain protein is being produced or a certain gene is present inside of a biological sample.

This concentration is measured by a detector located somewhere along the length of the separation channel. This detector uses laser light and a photosensor to determine the concentration based on the amount of light absorbed. Since the amount of light absorbed by the sample is linearly related to the concentration, this measurement can be used to determine what the concentration of the sample is at that location in the channel at any given time [22]. The data from this detector can be plotted to show the concentration in the channel over time to produce what is called an electropherogram. the concentration is determined based on the response of the detector, which is then multiplied by some factor to find the concentration. An example of what this looks like is shown in Figure 3.4 from work performed previously by Professor Backhouse.

The movement of the sample plug can be visualized as the concentration profile of the analyte across the length of the channel. This profile is a function of both distance along the channel and the time that has passed since the sample plug started moving. If only the electrophoretic and electroosmotic flow is considered and assuming the acceleration is instantaneous, then the concentration profile at any time t can be expressed in terms of the initial concentration profile $(C(x, t = 0))$ as,

$$C(x, t) = C(x - vt, 0) \tag{3.9}$$

where v denotes the velocity of the sample due electrophoretic and electroosmotic flow. This implies that if sample material does not diffuse within the separation channel, then the electropherogram output would retain the general shape of the initial concentration profile within the separation channel. Each separate analyte within the sample would separate from each other as they travel down the channel, creating multiple peaks with a similar shape offset from each other.

In the real-world the concentration profiles are not rectangular due to diffusion which causes the sample plug to spread out along the length of the channel over time. This diffusion results in the concentration profile to become more Gaussian over time. This concentration profile is described as [32],

$$C(x,t) = \frac{C_0}{2\sqrt{D\pi t}} exp\left(\frac{-x^2}{4Dt}\right) \tag{3.10}$$

where C_0 is related to the concentration of sample material that was initially injected into the separation channel. This solution assumes that the sample starts out at a high concentration of negligible dimensions at $t = 0$. Situations which deviate from this assumption could result in less accurate simulations. From the above equations, one can observe that the peak concentration of the sample plug decreases at the same time that the length of the sample plug is increasing. In general, if the effects of diffusion are relatively significant (i.e. if the diffusion coefficient is large), then the electropherogram output can have a shape that is drastically different from the concentration profile within the separation channel, as shown in Figure 3.5

This model approximates the concentration in an electrophoretic cross that has channels with circular cross-sections. However, if the channels have rectangular cross-sections, then the concentration profile of the sample plug at the time of injection can be better modelled by [33],

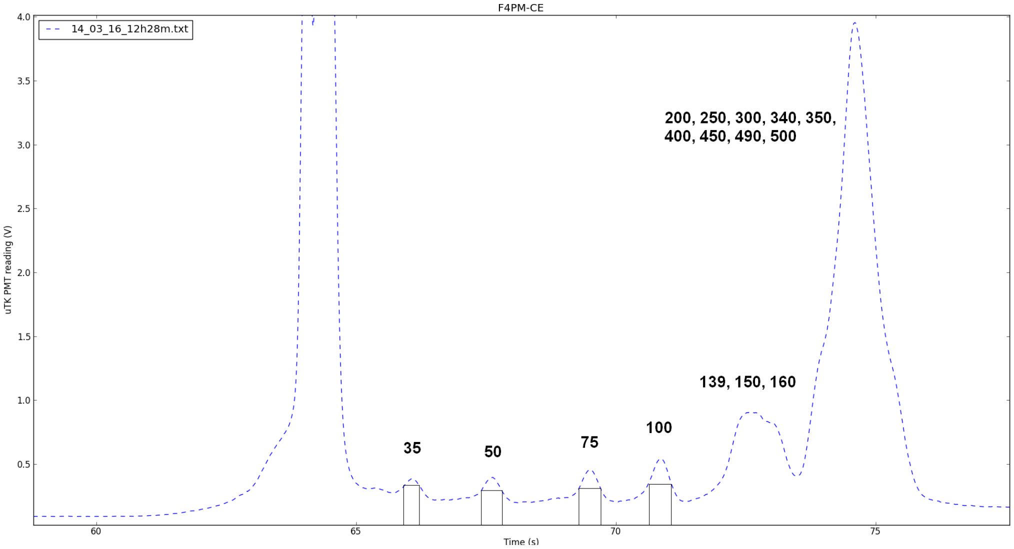

Figure 3.4: An example of an electropherogram of a sample with multiple analytes travelling through a channel. The y-axis is the detector response in volts which is proportionally related to concentration. The x-axis is the time that has passed since the measurement began (seconds) and the sample began to pass by the detector.

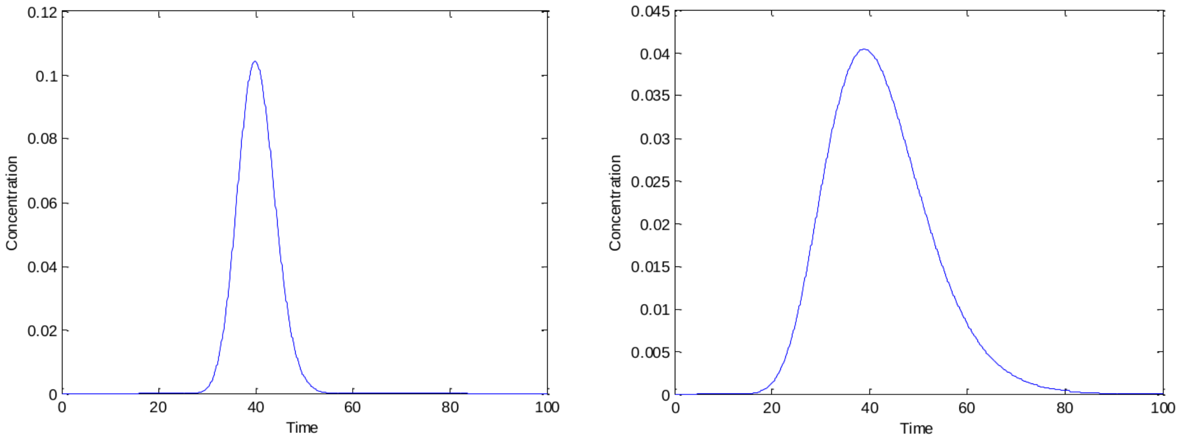

Figure 3.5: Left) Ideal electropherogram for a hypothetical separation with $x_{detector} = 0.04m, v = 0.001m/s$ without diffusion, time is in seconds and concentration is arbitrary units. Right) Electropherogram for a hypothetical separation with $D = 0.1m^2/s, x_{detector} = 0.04m, v = 0.001m/s$ with diffusion, time is in seconds and concentration is arbitrary units.

$$C(x,t) = \frac{C_0}{2}\left[erf\left(\frac{\frac{w_{channel}}{2} - x + vt}{2\sqrt{Dt}}\right) + erf\left(\frac{\frac{w_{channel}}{2} + x - vt}{2\sqrt{Dt}}\right)\right] \tag{3.11}$$

where $w_{channel}$ denotes the cross-sectional width of the injection and waste channels [33]. It should be noted that as the concentration profile specified by Equation 3.11 begins to approximate a Gaussian function after separation has elapsed for a significant amount of time. This can be observed in Figure 3.4.

3.2.3 Position of the Detector

Determining the optimal position of the detector along the separation channel is an important task for a microfluidic designer, if it is positioned too close to the intersection then the sample plug will not have a chance to separate each of the analytes. This will result in there being only one peak which does not provide any information about what the different analytes are. If the detector is placed too far from the intersection then the sample will be too dispersed. This will produce peaks that are too small to be accurately detected above the background noise.

The position of the detector can be analytically determined by finding the value x where

$C(x,t)$ is above the detection limit for the detector $C_{detectable}$ which will be named $x_{detector}$. Finding the maximum of the concentration profile provides an easy way to determine if it exceeds $C_{detectable}$, meaning that the detector can be placed there. For samples with multiple (n) analytes this problem now requires the detection of n local maxima and $n-1$ local minima in order to find distinct peaks. The condition for each maximum is as follows,

$$\left.\frac{dC}{dt}\right|_{t-t_{max,i}} = 0 \ and \ \left.\frac{d^2C}{dt^2}\right|_{t-t_{max,i}} < 0 \tag{3.12}$$

And the difference between adjacent maxima and minima must be sufficiently large that it is discernible by the detector. The constant $C_{detectable}$ is used as the minimum difference in concentration that is discernible by the detector as follows,

$$\frac{C(t_{min,i})}{C(t_{max,i})} \leq C_{detectable} \tag{3.13}$$

$t_{samplerate}$ is the minimum time that the detector can resolve, so any time separation between peaks in constrained to be larger than this parameter. The time until the i-th local maximum is reached is defined as $t_{peak,i}$. These can be approximated as the following by assuming that at time $t_{min,i}$, the only analytes that can be detected by the detector in any significant quantity would be the i-th analyte and the (i+1)-th analyte. In order to get the peak concentration at the position of the detector, the peak concentration can be set to the approximation,

$$t_{peak,i} \approx \frac{x_{detector}}{v_i} \tag{3.14}$$

Which holds true when there is not too much diffusion. It should be noted that approximating $t_{max,i}$ as $t_{peak,i}$ defined above reduces the amount of computation required by dReal by removing the need to repeatedly evaluate $C'(t)$ and $C''(t)$ which is expensive to calculate. This process is also performed for the minima, but in this case is the minimum of all peaks, called C_{floor}, and is constrained to be greater than $C_{detectable}$ to ensure that it can be detected. The derivation of this equation can be found in Appendix Stephen Chou Electrophoretic Cross Reports, but in summary it ensures that the minimum concentration is able to be detected at position $x_{detector}$,

$$C_{floor} = \frac{\min_i C_{0,i}}{\sigma_0 + \sqrt{\frac{2 * \max_i D_i x_{detector}}{v}}} \tag{3.15}$$

However, this still cannot be solved in a reasonable amount of time by dReal when there are greater than 6 analytes since a Gaussian curve has to be calculated for each extremum. Rather than attempting to actually estimate the values of the extrema then, reasonable bounds for the extrema and will be imposed as design constraints as follows,

$$C(t_{peak,i}) = C_i(t_{peak,i}) + p \min_i C_i(t_{peak,i}) \tag{3.16}$$

Where $C(t_{peak,i})$ is the concentration of the sample plug, $C_i(t_{peak,i})$ is the peak from the most concentrated analyte in the sample plug and $p \min C_i(t_{peak,i})$ is the concentration of the lowest concentration analyte multiplied by a positive constant p. This lowest concentration is C_{floor}.

The ratio of the two adjacent peaks heights can be approximated by the following difference between the C_{peak} and C_{floor} of the analyte concentration,

$$diff = \frac{C_{0,i}}{C_{0,i+1}} \sqrt{\frac{D_{i+1}\mu_i}{D_i\mu_{i+1}}} \tag{3.17}$$

This diff is used to enforce that the difference between peak heights is close enough to be detectable based on $C_{detectable}$. In order for this model to be solvable for a nontrivial number of analytes in a reasonable amount of time by the SMT solver, the model must have reasonable approximations applied to it. These approximations are pessimistic, which is to say that they will predict peak concentrations lower than the actual maxima. Justification for each of the following approximations can be found in appendix Stephen Chou Electrophoretic Cross Reports:

1. For all values of t_1 and t_2 , $t_1 < t_2$, $C(t)$ must have at least one local maximum in the open interval (t_1, t_2) if there exists some $t_3 \in (t_1, t_2)$ that satisfies the constraints $C(t_1) < C(t_3)$ and $C(t_2) < C(t_3)$
2. For all values of t_1 and t_2 , $t_1 < t_2$, $C(t)$ must have at least one local minimum in the open interval (t_1, t_2) if there exists some $t_3 \in (t_1, t_2)$ that satisfies the constraints $C(t_1) > C(t_3)$ and $C(t_2) > C(t_3)$
3. For all i between 1 and $n - 1$ and all $t \in [t_{max,i}, t_{max,i+1}]$, $C(t) \geq C(t_{min,i})$
4. For all i between 1 and n and $t \in [t_{min,i1}, t_{min,i}]$, where $t_{min,0}$ and $t_{min,n}$ are defined to be 0 and some arbitrarily large value respectively in this context, $C(t) \leq C(t_{max,i})$
5. If the approximate model is satisfied, then $C(t)$ must have at least n local maxima

3.2.4 Manifold Implementation

The conditions outlined in the derivation of the above equations are enforced as constraints in the SMT solver. The constraints are as follows:

- The height and width of the separation, tail, droplet input, and droplet output channels are constrained to be equal
- The electric field E is constrained based on Equation 3.5
- For every analyte in the sample plug:
  - The electric mobility μ is constrained to be equal to equations 3.6 3.7 and 3.8
  - The velocity of the particles are constrained to be equal to $\mu\overrightarrow{E}$
  - The time until peak intensity is constrained to equal the distance to the detector $x_{detector}$ divided by velocity v
  - Initialize the variable for the minimum time separation needed for the peak to be detectable
  - The position of the detector is constrained to be within the length of the separation channel
  - The minimum concentration for each analyte is constrained to be equal to C_{floor}
  - The negligible concentration of each analyte is set to be some positive multiple of C_{floor}
- The time difference between the peaks is constrained to be large enough to be detected according to Equation 3.17
- The discernible peaks are defined by Equation 3.13 which is constrained by Equation 3.15

Chapter 4

T-Junction Droplet Generator Results and Analysis

This chapter will compare the output of the new Python implementation of Manifold to data from physical microfluidic devices in order to perform a preliminary validation of Manifold's output. The data from T-junction droplet generator microfluidic device was provided by Professor Backhouse's lab in work performed by Strike et al [1]. This was used to perform a preliminary validation of the output of Manifold. The known *design parameters* of the devices will be defined within Manifold, constraining those parameters to their measured value in the devices the data was obtained from. Next, Manifold will generate ranges of values that the output parameters of the device can take. These ranges are based on the parameters specified from the historical data and the physical equations defining the system outlined in Chapter 3: Microfluidics Background.

Multiple validation tests will be performed, both to test that the output is logically sound and that it reflects the observed values from the historical data. First, the ranges returned by Manifold for the output parameters will be compared to the value for those parameters measured in the physical device to determine if the values lie within the range. If they do, this will provide a preliminary validation that the ranges of values generated by Manifold have the potential to specify a working device. Next, the value of the droplet volume will be fixed to the observed value and the ranges of the other parameters will be compared to their original ranges. Because the system is now more constrained, the ranges should be reduced. Next, Manifold will have the continuous channel width halved to if the final droplet volume is reduced. With this reduced channel width, the observed droplet volume will be added back in as a constraint to see if Manifold will not be able to find a solution due to the channel width being too small now. Finally, the impact of a representative selection of the *design parameters* has on the droplet volume will be assessed, presented both as the analytical relationship and the variance of the droplet volume due to manufacturing and experimental variability.

4.1 Research Questions

The following seven research questions are intended to establish that the Manifold tool properly embodies the mathematical modelling of the previous chapter and functions as expected. This is important for any computational tool, but is especially important for Manifold when it uses the dReal SMT solver, because Manifold uses dReal in a manner different from what it was originally designed for.

dReal was developed as a verification tool for cyber-physical systems. In its intended usage, the designer is aiming for the result "no solution could be found," because the designer is asking questions like "will this robot hurt the human?" If dReal says there is no solution, then one can be confident that the human is safe (assuming the mathematical model corresponds to physical reality). On the other hand, if dReal says that perhaps there is a solution in the given ranges, it is not certain that the human is safe: it's just that dReal cannot prove the human is safe, and maybe danger lies in the indicated ranges. Outside of the indicated ranges, one can be confident that the human is safe.

There are three reasons why dReal might be applicable to the kinds of microfluidic design tasks considered here:

1. **Underconstraint:** Like the kinds of systems that dReal is designed to analyze, the preliminary stages of microfluidic design are often highly underconstrained. There are many equations and variables for the whole system that it is difficult for the microfluidic designer to create a complete design with a single pass of the pen. One of Manifold's goals is that the designer can provide a partial specification and then Manifold will computationally flesh out the design.

2. **Formulas:** The kinds of formulas dReal is designed to solve are mathematically similar to those needed for preliminary microfluidic design: equalities and inequalities over the reals, with ordinary arithmetic operators. For example, trigonometric functions, booleans or ordinary differential equations. Further steps of the microfluidic design process might use more sophisticated mathematical models, but by then the design will be less underconstrained.

3. **Existence of Solutions:** dReal was originally designed to be applied to safety questions, where it's goal is to determine that there is no circumstance that is unsafe. By contrast in microfluidic design, the designer expects that there is a plausible design and Manifold uses dReal to help the designer narrow down the feasible range for the unspecified parameters.

The research questions (RQ) considered here are:

- Consistency with empirical data:

  1. Does fixing known design parameters compute ranges for consequential parameters that have empirically observed values?
  2. Does fixing known design parameters and empirically observed consequential parameter values refine computed values for unknown design parameters *only* by narrowing their ranges?
  3. Do the computed values correspond to physical intuitions about the device?

- Pathological perturbations result in expected errors:

  4. Do pathological perturbations of the known design parameters result in ranges for consequential parameters that exclude the desired values? For example, cutting the channel width in half results in a computed range for droplet size that does not include the desired droplet size.
  5. Do these perturbations of known design parameters in combination with constraining consequential parameters to desired values result in no computed solution?

- Consistency with expected design behaviour:

  6. Does the computation show that select design parameters do indeed have the expected linear relationship to the consequential design parameters? For example, demonstrate that the computations show that channel width has a linear relationship with droplet size, as expected.
  7. Does the manufacturing variability of static design parameters cause more or less variance in the droplet size than the operating variability of dynamic design parameters?

4.1.1 Understanding the Ranges Returned by dReal

If dReal cannot prove that there is no solution, then it will return ranges for each variable indicating where solutions might lie. These are *not stochastic predictions* of real-world performance. dReal is not a stochastic solver, and the equations described in previous chapters are not stochastic. dReal's strength is in handling problems that involve a wide range of nonlinear real functions.

What dReal's ranges tell is that there are definitely no solutions outside of these ranges. So if dReal returns that the droplet size will be between 1μm and 4μm, then it is known that there cannot be larger or smaller droplets (assuming the mathematical models correctly represent physical reality, and that the manufactured and operated device is properly characterized by the constants).

These ranges can still be useful to the microfluidic designer at the early stages of design, in three ways:

- Identifying which parts of the design are more and less constrained.
- Searching for appropriate values for currently unspecified design parameters. The designer can fix the consequential parameters to desired values, and see which ranges are returned for these unspecified design parameters. This can be especially helpful when there is a complex non-linear relationship between the consequential parameters and the unspecified design parameters.
- Helping to characterize the influence of specific design parameters on the consequential parameters. The designer can run multiple computations, changing different design parameters, to see how they influence the consequential parameter ranges.

4.2 Preliminary Validation With Backhouse Lab Chip

In this section a preliminary validation of the research questions above is performed through various computations with empirical data from a microfluidic chip produced by the Backhouse Lab and documented by Strike et al [1], and pictured here in Figure 4.1. The blue liquid is the dispersed phase which is water with blue food colouring added to it, the continuous phase used is mineral oil. After the T-junction it can be seen that small droplets of blue liquid are created which travel along the channel to a circular reservoir to store the droplets for later analysis. The three circles on the left are values that are pneumatically controlled to pump the fluid from the reservoir on the far left.

Strike et al [1] report that the dimensions of the T-junction droplet generator channels were $210\mu m$ ($2.100E-04m$) wide and with a height of $200\mu m$ ($2.100E-04m$). Strike et al [1] measured the size of the generated droplets using a camera to capture video of the output channel after the junction. Video analysis was performed to determine the average length of the droplets produced (0.40mm), which in this work was multiplied by the measured channel width and height gives the volume of 16.8nL (1.680E-11m^3).

As described in the previous chapter, van Steijn et al [3] gives two key criteria to be in the regime described by their equations: (1) low capillary number (<0.01); and (2) that the ratio of the sizes of the disperse and continuous channel is "not small." The device designed in the Backhouse Lab and described by Strike et al [1] appears to meet these criteria: (1) there is high interfacial tension between oil and water, which results in a low capillary number; and (2) the ratio of the sizes of the disperse and continuous channel in this device is close to 1, which is "not small." The analyses that follow therefore apply the van Steijn et al equations to the Backhouse Lab device.

For each of the research questions discussed above, it was found that Manifold/dReal computes values/ranges that are generally consistent with expectations. There are two cases showing limitations of the technology: one in the numerical computations [RQ3], and another in attempting an analysis of non-uniform variability (which one might expect to occur in the real world) [RQ7]. In the first case, a potential remedy is discussed in future work. In the second case, a different computational tool, such as MapleSim, might be better equipped to represent and reason about designs at this fidelity. Manifold's intended connection to MapleSim was discussed above in chapter 2, and is revisited at the end of this chapter.

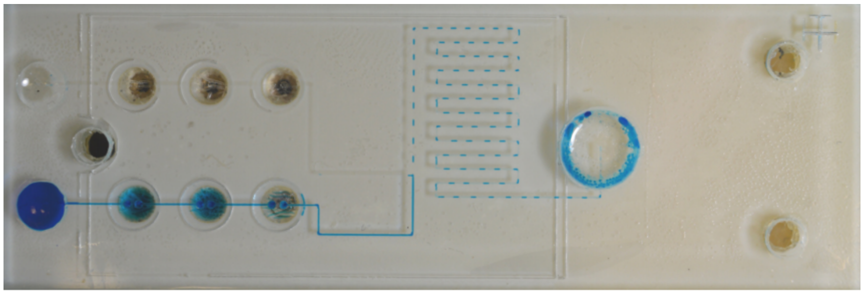

Figure 4.1: Formation of droplets in the T-junction droplet generator microfluidic device that the data was collected from. Designed by the Backhouse Lab [1].

4.2.1 Measured Droplet Size is Within Predicted Range [RQ1]

Table 4.1 shows a subset of the results returned from Manifold/dReal when used to predict the droplet size produced by the microfluidic chip produced by the Backhouse Lab [1]. (The full output is in appendix Full Manifold Output.) The first section of the table lists design parameters that are given in the paper [1] that describes the chip. The second section of the table lists computed ranges for some of the design parameters that are not specified in the paper. The third section of the table gives the *consequential design parameters* that are determined by Manifold based on the defined parameters and the applied constraints. This also includes the predicted droplet size which is the output parameter of interest, which has a range of: $[1.055E-11m^3, 4.061E-11m^3]$. The Backhouse Lab experimentally observed a droplet size of $1.680E-11m^3$, which is within the predicted range.

Parameter	User	Parameter	Units	Lower	Upper
Type	Specified	Name		Limit	Limit
Design	Yes	output_channel_width	m	2.100E-04	Same
	[1]	output_channel_height	m	2.000E-04	Same
		continuous_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		continuous_port_density	$kg/(m^3)$	8.000E+02	Same
		dispersed_port_viscosity	$kg/(m*s)$	1.000E-03	Same
		dispersed_port_density	$kg/(m^3)$	9.999E+02	Same
		output_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		output_port_density	$kg/(m^3)$	9.999E+02	Same
		continuous_channel_width	m	2.100E-04	Same
		continuous_channel_height	m	2.000E-04	Same
		dispersed_channel_width	m	2.100E-04	Same
		dispersed_channel_height	m	2.000E-04	Same
Design	No	t_j_node_x	m	0	7.620E-02
		t_j_node_y	m	0	2.540E-02
		dispersed_pressure	$kg/(m*s^2)$	9.844E+05	9.851E+05
		continuous_pressure	$kg/(m*s^2)$	5.002E+05	5.018E+05
		out_pressure ‘	$kg/(m*s^2)$	9.837E+05	9.843E+05
		output_channel_length	m	1.000E-09	7.620E-02
Consequential	No	output_channel_flow_rate	m^3/s	1.911E-09	1.911E-09
		output_channel_viscosity	$kg/(m*s)$	6.562E+01	6.875E+01
		output_channel_resistance	$kg/(m^4*s)$	2.705E+11	2.705E+11
		t_j_node_pressure	$kg/(m*s^2)$	9.847E+05	9.863E+05
		droplet_volume	m^3	**1.055E-11**	**4.061E-11**

Table 4.1: Table of the data from the T-junction droplet generator device along with the corresponding range of values output by Manifold/dReal. The parameters specified values have exact values, with the upper limit labelled as *Same*. The parameter name t_j refers to the node at the intersection of the T-junction.

4.2.2 Measured Droplet Size is Analytically Viable [RQ2]

Table 4.2 shows the results from a computation in which the design parameters are fixed as in Table 4.1, and additionally the droplet volume is set to the empirically observed value of 16.8nL ($1.680E - 11m^3$) [1]. In the second section of Table 4.2 shows that Manifold/dReal finds viable solution ranges for the unspecified design parameters, as expected. Additionally, the ranges computed for these unspecified design parameters in Table 4.2 are equivalent to, or narrow than, the ranges computed in Table 4.1; as expected.

Parameter Type	User Specified	Parameter Name	Units	Lower Limit	Upper Limit
Design	Yes [1]	t_j_out_width	m	2.100E-04	Same
		output_channel_height	m	2.000E-04	Same
		continuous_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		continuous_port_density	$kg/(m^3)$	8.000E+02	Same
		dispersed_port_viscosity	$kg/(m*s)$	1.000E-03	Same
		dispersed_port_density	$kg/(m^3)$	9.999E+02	Same
		output_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		output_port_density	$kg/(m^3)$	9.999E+02	Same
		continuous_channel_width	m	2.100E-04	Same
		continuous_channel_height	m	2.000E-04	Same
		dispersed_channel_width	m	2.100E-04	Same
		dispersed_channel_height	m	2.000E-04	Same
Design	No	t_j_node_x	m	0	7.620E-02
		t_j_node_y	m	0	2.540E-02
		dispersed_port_pressure	$kg/(m*s^2)$	**9.846E+05**	**9.850E+05**
		continuous_pressure	$kg/(m*s^2)$	**5.005E+05**	**5.014E+05**
		output_port_pressure	$kg/(m*s^2)$	**9.839E+05**	**9.841E+05**
		output_channel_length	m	1.000E-09	7.620E-02
Consequential	No	output_channel_flow_rate	m^3/s	1.911E-09	1.911E-09
		output_channel_viscosity	$kg/(m*s)$	**6.632E+01**	**6.801E+01**
		output_channel_resistance	$kg/(m^4*s)$	2.705E+11	2.705E+11
		t_j_node_pressure	$kg/(m*s^2)$	9.847E+05	9.863E+05
Consequential	**Observed** [1]	**droplet_volume**	m^3	**1.680E-11**	**Same**

Table 4.2: Manifold/dReal results for a computation like that of Table 4.1, except the droplet volume has now been fixed to the empirically observed value of 1.680E-11m^3 [1]. The parameters specified values have exact values, with the upper limit labelled as *Same*. Rows that have changed from Table 4.1 are in bold. The parameter name t_j refers to the node at the intersection of the T-junction.

4.2.3 Physical Interpretation of Results [RQ3]

The previous two subsections have discussed interpretations of the droplet size, both as an output of and as an input to computation. Now explanations of how to interpret the physical meaning of the output of some of the other variables, such as the unspecified design parameters, will be provided.

T-Junction Placement: In both Tables 4.1 and 4.2 it can be seen that the variables t_j_node_x and t_j_node_y may range from zero to their maximum allowable value based on the size of the chip. These variables represent the Cartesian coordinates of the T-junction on the chip. There is nothing in the design that constrains this physical placement, so the range of these coordinates is the entire chip, as expected.

Output Channel Length: The maximum length for the output_channel_length in both tables is the length of the chip, as expected. The minimum value is 1.000E-09m, which is certainly long enough to hold a droplet in the range of 1.055E-11m^3 to 4.061E-11m^3, and is the minimum size that could be reasonably manufactured.

Channel Pressures: The pressure of the continuous phase is about 5 atmospheres (around 500kPa). The pressure of the dispersed phase is almost double that (around 985kPa). It is expected that the pressure of the dispersed phase should be higher than the pressure of the continuous phase in order for the T-junction to function, since the dispersed phase needs to be squeezed into the continuous phase to form the droplets.

The pressure at the output is just slightly lower than the pressure of the dispersed phase, as expected, because the output includes both dispersed phase droplets and continuous phase media. The lower limit drops $984.4kPa - 983.7kPa = 0.7kPa$, and the upper limit drops $985.1kPa - 984.3kPa = 0.8kPa$.

These ranges and values might be larger than one would physically expect, but their relationships appear reasonable.

Flow Rate: The range of values for the flow rate is highly constrained, which is a consequence of the δ value for dReal not being able to handle the large value for the resistance. As such, it constrains the resistance to be relatively close to only having one value, which has resulted in both the pressure and the flow rate to be very small ranges, potentially smaller than they should be in practise. This is due to them being related by Equation 3.1. This problem is discussed in Chapter Design Limitations and presented as future work to solve.

4.2.4 Pathological Inputs Produce Expected Errors [RQ4&5]

A consistency check was performed for the T-junction device where the continuous and output channel widths were reduced by half with the droplet volume unspecified. The results of this are shown in Table 4.3 showing that the range of values for the droplet volume now excludes the original observed size of 16.8nL ($1.680E-11m^3$).

Next, the volume of the droplet was specified to be 16.8nL ($1.680E-11m^3$) with this halved channel width as this was outside of the specified value from Manifold. It was found that Manifold found that there was no solution, which was expected as this droplet size should be impossible with such a channel width.

Parameter	User	Parameter	Units	Lower	Upper
Type	Specified	Name		Limit	Limit
Design	Yes	output_channel_width	m	2.100E-04	Same
	[1]	output_channel_height	m	2.000E-04	Same
		continuous_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		continuous_port_density	$kg/(m^3)$	8.000E+02	Same
		dispersed_port_viscosity	$kg/(m*s)$	1.000E-03	Same
		dispersed_port_density	$kg/(m^3)$	9.999E+02	Same
		output_port_viscosity	$kg/(m*s)$	3.050E-04	Same
		output_port_density	$kg/(m^3)$	9.999E+02	Same
		continuous_channel_width	m	1.050E-04	Same
		continuous_channel_height	m	2.000E-04	Same
		dispersed_channel_width	m	2.100E-04	Same
		dispersed_channel_height	m	2.000E-04	Same
Design	No	t_j_node_x	m	0	7.620E-02
		t_j_node_y	m	0	2.540E-02
		dispersed_port_pressure	$kg/(m*s^2)$	9.844E+05	9.851E+05
		continuous_pressure	$kg/(m*s^2)$	5.002E+05	5.018E+05
		output_port_pressure	$kg/(m*s^2)$	9.837E+05	9.843E+05
		output_channel_length	m	1.000E-09	7.620E-02
Consequential	No	output_channel_flow_rate	m^3/s	1.911E-09	1.911E-09
		output_channel_viscosity	$kg/(m*s)$	6.562E+01	6.875E+01
		output_channel_resistance	$kg/(m^4*s)$	1.224E+11	2.705E+11
		t_j_node_pressure	$kg/(m*s^2)$	1.852E+06	9.863E+05
		droplet_volume	m^3	**5.235E-12**	**8.893E-12**

Table 4.3: Table of the data from the T-junction droplet generator device along with the corresponding range of values output by Manifold with the continuous channel width halved. The parameters specified values have exact values, with the upper limit labelled as *Same*. Rows that have changed from Table 4.1 are in bold. The parameter name t_j refers to the node at the intersection of the T-junction.

4.2.5 Expected Parameter Correlations Are Predicted [RQ6&7]

For this device, the consequential parameter of interest to the designer is the droplet volume. In order to determine the correlation between the different design parameters and the droplet volume within Manifold, a subset of the design parameters were swept across different values and the change in droplet volume was measured. Because this is a parameter sweep, the time complexity of the calculation grows exponentially with each parameter added, so only 6 parameters were checked to allow for computation in a reasonable amount of time. Three values were swept across for every parameter, resulting in $3^6 = 729$ different tests. This was run in a single thread and took roughly 10 minutes on a laptop computer with a Intel Core i7-3612QM, 2.10GHz CPU. Each of these computations is independent, and so could be run in parallel.

Table 4.4 shows the correlation between these six selected design parameters and the droplet volume. The correlation is expressed as the slope of a line of best fit of the three computed data points for each design parameter, and the line of best fit was generated based on minimizing the mean squared error.

Table 4.4 shows that the channel dimensions have a linear correlation to the droplet size, as expected. It also shows that the pressures have a non-linear correlation to the droplet size, also as expected.

Design Parameter	Double % Change	Halving % Change	Slope of Line of Best Fit
output_channel_width	127	52	1
continuous_channel_width	104	52	1
channel_height	102	48	1
dispersed_channel_width	85	49	1
dispersed_port_pressure	21	16	0.35
continuous_port_pressure	36	23	0.2

Table 4.4: The percentage change of the droplet volume when each of the listed *design parameters* were individually doubled, halved and then the slope of the trend line.

Table 4.5 appears to show that a 10% variation in the static (manufacturing-related) design parameters has a greater impact on droplet size than does a 10% variation in the dynamic (operating) design parameters. The 10% number was chosen as a reasonable variation to investigate in conversation with the Backhouse Lab.

It is not clear that strong conclusions about the relative influence of the static and dynamic design parameters can be formed based on the data in Table 4.5. A potential methodological flaw in this computation is that these 10% variations that a designer might expect to observe in practice are not uniformly distributed through time and space, whereas the analytical equations of the previous chapter are unable to capture this kind of inconsistent variation.

Nevertheless, a microfluidic designer might still find this kind of analysis useful in making design decisions. For example, getting an understanding of the impact of pressure on droplet size might inform pump design or selection.

Representing and reasoning about non-uniform variation in design parameters could be better done with a simulation engine such as MapleSim, rather than with the kinds of analytical equations that dReal can solve. As described in chapter 2, Manifold is intended to also output the design to MapleSim. The next section will revisit the intended Manifold design methodology introduced in chapter 2. In brief, dReal is intended to be used in the preliminary design stages, when the design is underspecified and underconstrained. As the design is refined, other computational tools such as MapleSim or MapleSim can be used for subsequent analyses.

	Droplet Volume	
Design Parameters	-10% change	+10% change
Static / Manufacturing		
continuous_channel_width	1.528E-11	1.852E-11
dispersed_channel_width	1.517E-11	1.802E-11
output_channel_width	1.501E-11	1.862E-11
channel_height	1.497E-11	1.816E-11
Dynamic / Operating		
continuous_port_pressure	1.656E-11	1.668E-11
dispersed_port_pressure	1.622E-11	1.683E-11

Table 4.5: The mean value predicted for the droplet volume by Manifold/dReal when each of the listed *design parameters* were individually increased and decreased by 10% (upper and lower limit respectively). The original volume is $1.680E - 11m^3$.

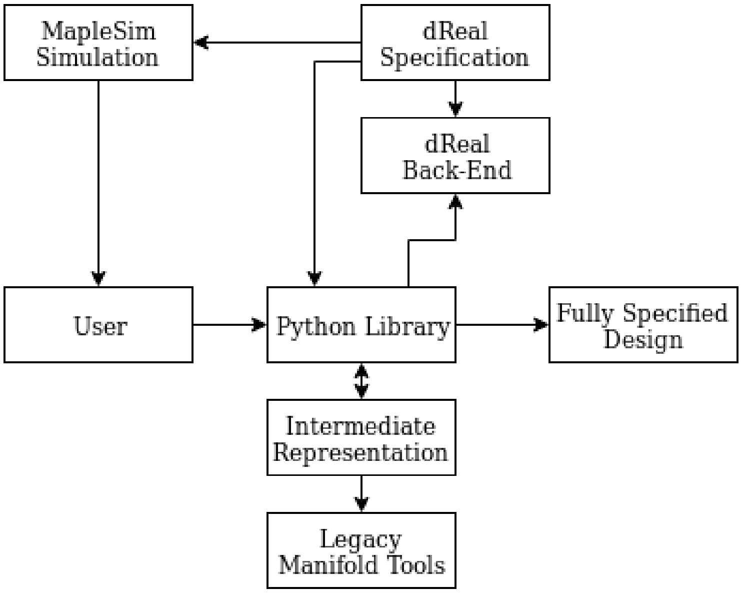

Figure 4.2: The architecture for V3.0 of the Manifold toolchain.

4.3 Design Methodology

The goal for Manifold is for a microfluidic designer to receive some initial feedback on their design before manufacture. The tests performed above represent potential uses of Manifold by a design to get this feedback on their design. Designers would also want to know what *design parameters* have the most impact on the droplet volume so they can focus on refining those parameters. This part of the new workflow developed for Manifold in this work shown in Figure 4.2 is the feedback to the user in both a numerical form from dReal and a visual form from the Modelica model. This visual feedback loop will be outlined in the following section.

4.3.1 Modelica Feedback Loop

In order to demonstrate how Manifold's human-in-the-loop feedback process reduces the cognitive load in designing microfluidic devices, an example will be walked through for a T-junction droplet generator device. The microfluidic designer will first run their simulation in Manifold, specifying the parameters that they know. Manifold can then produce a Modelica file of that design if the correct function is ran. This function will take the ranges of values produced by dReal for all of the unspecified parameters, calculate the middle value in the range and set the parameter equal to the value. These values will then be inserted into the Modelica file in the corresponding location for that parameter.

This Modelica file can then be opened in a simulation software like MapleSim to run the simulation of the fluid flow within the device. An example of this is shown in 4.3, where a T-junction droplet generator is simulated in MapleSim showing the concentration of the sample in the dispersed phase breaking off into droplets over time.

The designer will then look at the behaviour of their device in this simulation and tune the parameters within MapleSim until the desired behaviour is achieved. If they are unsure of how to change all of parameters then they can modify some of the parameters and pass these new parameters back into Manifold to get a new simulation that is closer to what is desired and provides more accurate ranges for each parameter.

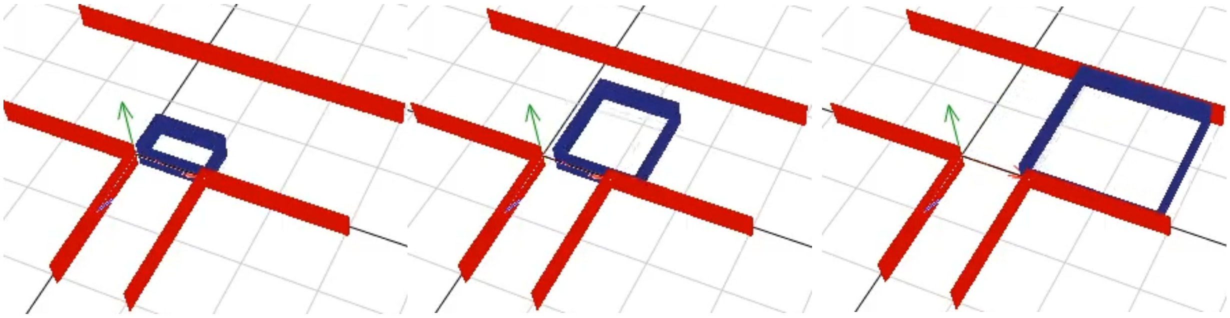

Figure 4.3: A sequence of photos from the MapleSim time-domain simulation of a T-junction droplet generator from a Modelica file.

Chapter 5

Electrophoretic Cross Results and Analysis

This chapter applies Manifold/dReal in a completely different microfluidic domain: electrophoretic cross. The initial intent of this chapter was to validate Manifold/dReal with some electrophoretic devices that were specifically fabricated to support this work. Unfortunately, fabrication was not successful. Consequently, the analysis in this chapter is inspired by historical work in the Backhouse Lab, but does not precisely correlate to any specific individual device or publication. The conclusions of this chapter, therefore, are weaker: this chapter demonstrates that Manifold/dReal might be used to analyze an electrophoretic cross device, and serves as a basis for further work in this area.

In the previous chapter, for a T-junction droplet generator, the designer's primary concern was the droplet volume. In this chapter, for an electrophoretic cross device, the target output parameter is the position of the detector along the separation channel that allows for each of the analyte peaks to be distinguished. This chapter uses a representative value of $8.000E - 3m$ for this position.

This chapter follows the general outline and research questions presented previously.

5.1 Inspiration from The Backhouse Lab

Figures 5.1 and 5.2 show electrophoretograms from two different devices in the Backhouse Lab. Both of these devices operated on mixtures that contained twelve analytes. The first device (Figure 5.1) performs poorly: there are four control peaks, and then the rest of the analytes are merged into two large peaks at the beginning and end. The second device (Figure 5.2) performs well: most of the twelve analytes have their own individual peak. The goal of a design an electrophoretic cross device is to create a device that produces electrophoretograms like Figure 5.2. Characteristics of these devices are in Table 5.1.

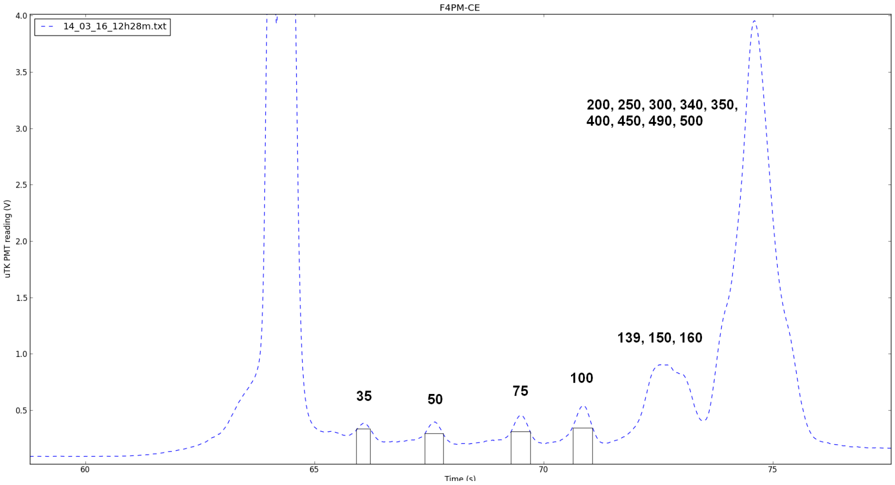

Figure 5.1: An electrophoretogram with four control peaks and all of the sample peaks merged together into two large peaks, the estimated size of the particles in that peak written above.

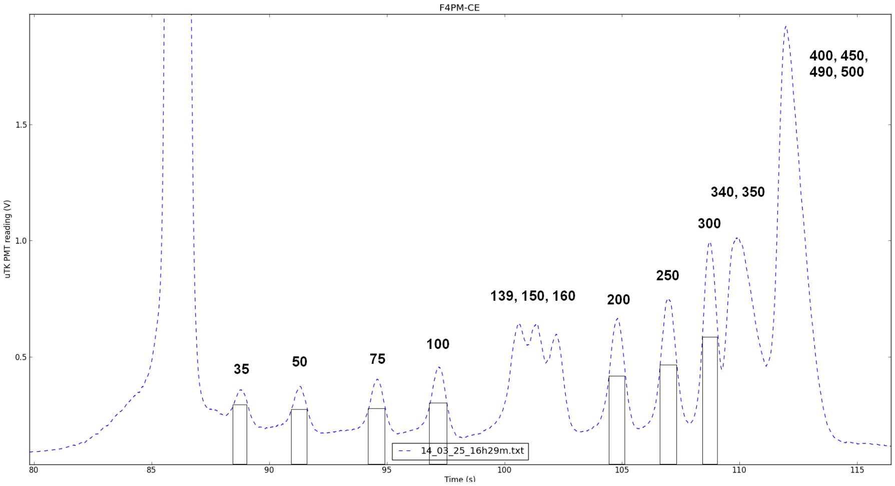

Figure 5.2: An electrophoretogram with four control peaks and several individual sample peaks, the estimated size of the particles in that peak written above.

Characteristic	Device From Figure 5.1	Device From Figure 5.2
anode voltage	**300V**	**200V**
cathode voltage	0V	0V
detector position	0.008m	0.008m
channel width	$210\mu m$	$210\mu m$
channel height	$200\mu m$	$200\mu m$
total # of analytes	12	12
# of control analytes	4	4
# of sample analytes	8	8
peaks of control analytes	35, 50, 75, 100	35, 50, 75, 100

Table 5.1: Characteristics of the devices used to produce the electrophoretograms in Figures 5.1 and 5.2. All characteristics are the same except for the anode voltage (in bold).

The analysis in this chapter is inspired by these devices: the characteristics listed in Table 5.1 were used, and the viscosity of the separation gel was greater than the input continuous media (which is a requirement to use the formulas described in previous chapters). There are some differences, however. While the electrophoretograms in these figures are based on samples with 12 analytes, the analysis done in this chapter is for 4 analytes.

Four Imaginary Analytes. Table 5.2 describes the four imaginary analytes used in this chapter's computations. They are supposed to represent four different bits of DNA. All four have the same initial concentration and diffusivity; they vary in electrical charge and hydrodynamic radii. This variation is supposed to represent a 10 base pair length difference, since each base pair has an overall negative charge of 2. The equations relating the size to the mobility are outlined in §3.2.

Property	Analyte 1	Analyte 2	Analyte 3	Analyte 4
Initial Concentration	$1E-2mol/L$	$1E-2mol/L$	$1E-2mol/L$	$1E-2mol/L$
Diffusivity	$1E-13m^2/s$	$1E-13m^2/s$	$1E-13m^2/s$	$1E-13m^2/s$
Electrical Charge	20e	40e	60e	80e
Hydrodynamic Radii	$1.0\mu m$	$1.1\mu m$	$1.2\mu m$	$1.3\mu m$

Table 5.2: Characteristics of the four imaginary DNA analytes. $e = 1.602E-19$

Glossary of Parameters. There are many new parameters introduced for the electrophoretic cross devices that also have reasonable ranges applied to them. The name and symbol for the parameter used in Chapter 3 Microfluidics Background are shown in 5.3.

			Limits	
Parameter	Symbol	Units	Lower	Upper
Channel Detector Position	$x_{detector}$	m	1E-09	1
Channel Min Sampling Rate	$t_{samplerate}$	s	0	1E+06
Electrophoretic Cross Node Electric Field	E	V/m^2	0	1E+06
Electrophoretic Cross Node Analyte Peak Time	t	s	0	1E+06
Electrophoretic Cross Node Analyte Min Time	t	s	0	1E+06
Electrophoretic Cross Node Analyte Mobility	μ	m^2/Vs	-1	1
Electrophoretic Cross Node Analyte Velocity	v	m/s	0	1
Electrical Port Voltage	V	V	-1E+05	1E+05

Table 5.3: The physically reasonable upper and lower limits that each unspecified parameter is constrained within. In devices with multiple channels and ports these variables would be instantiated individually for each.

5.2 Position is Within Predicted Range [RQ1]

Table 5.4 summarizes the Manifold/dReal results of a preliminary design analysis (full output in appendix §A). The range of possible values for the anode detector position was computed as [0.00627m, 0.00824m], which includes the 0.008m value expected based on the inspirational devices. For this computation, the x and y coordinates of every node (except the anode detector) on the device were defined, so as to reduce the size of the search space.

Parameter Type	User Specified	Parameter Name	Units	Lower Limit	Upper Limit
Design	Yes	anode_channel_width	m	2.100E-04	Same
		anode_channel_height	m	2.000E-04	Same
		anode_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
		cathode_port_voltage	m	2.000E+02	Same
		anode_port_voltage	m	0	Same
		in_port_viscosity	$kg/(m*s)$	1.000E-03	Same
		in_port_density	$kg/(m^3)$	9.999E+02	Same
		ep_c_viscosity	$kg/(m*s)$	1.500E-04	Same
		cathode_channel_width	m	2.100E-04	Same
		cathode_channel_height	m	2.000E-04	Same
		anode_channel_width	m	2.100E-04	Same
		anode_channel_height	m	2.000E-04	Same
		anode_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
		in_channel_width	m	2.100E-04	Same
		in_channel_height	m	2.000E-04	Same
		in_channel_viscosity	$kg/(m*s)$	1.000E-03	Same
		output_channel_width	m	2.100E-04	Same
		output_channel_height	m	2.000E-04	Same
		output_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
Design	No	anode_channel_flow_rate	m^3/s	7.625E-13	9.920E-13
		anode_channel_resistance	$kg/(m^4*s)$	8.307E+11	8.307E+11
		input_channel_length	m	1.000E-03	2.540E-02
		ep_c_node_x	m	0	7.620E-02
		ep_c_node_y	m	0	2.540E-02
		t_peak_0	s	66.735	73.365
		t_min_0	s	63.221	70.291
Consequential	No	anode_channel_x_detector	m	6.270E-03	8.240E-03

Table 5.4: Table of the data from the electrophoretic cross device along with the corresponding range of values output by Manifold. The parameters specified values have exact values, with the upper limit labelled as *Same*. The label *ep_c* refers to the electrophoretic cross node at the intersection of the two channels.

5.3 Detector Position is Analytically Viable [RQ2]

Table 5.5 shows the results of a consistency check similar to the one performed in the previous chapter: the consequential parameter (anode x position) is fixed to the inspired value $8.000E - 03m$. As expected, this additional constraint resulted in the computed ranges of some of the other variables being narrowed — in this case, the anode channel flow rate.

Parameter Type	User Specified	Parameter Name	Units	Lower Limit	Upper Limit
Design	Yes	anode_channel_width	m	2.100E-04	Same
		anode_channel_height	m	2.000E-04	Same
		anode_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
		cathode_port_voltage	m	2.000E+02	Same
		anode_port_voltage	m	0	Same
		in_port_viscosity	$kg/(m*s)$	1.000E-03	Same
		in_port_density	$kg/(m^3)$	9.999E+02	Same
		ep_c_viscosity	$kg/(m*s)$	1.500E-04	Same
		cathode_channel_width	m	2.100E-04	Same
		cathode_channel_height	m	2.000E-04	Same
		anode_channel_width	m	2.100E-04	Same
		anode_channel_height	m	2.000E-04	Same
		anode_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
		in_channel_width	m	2.100E-04	Same
		in_channel_height	m	2.000E-04	Same
		in_channel_viscosity	$kg/(m*s)$	1.000E-03	Same
		output_channel_width	m	2.100E-04	Same
		output_channel_height	m	2.000E-04	Same
		output_channel_viscosity	$kg/(m*s)$	1.500E-04	Same
Design	No	anode_channel_resistance	$kg/(m^4*s)$	8.307E+11	8.307E+11
		in_channel_length	m	1.000E-03	2.540E-02
		ep_c_node_x	m	0	7.620E-02
		ep_c_node_y	m	0	2.540E-02
		t_peak_0	s	**68.237**	**73.155**
		t_min_0	s	**64.123**	**68.921**
Consequential	**Observed**	**anode_channel_x_detector**	m	**8.000E-03**	**Same**

Table 5.5: Table of the data from the electrophoretic cross device along with the corresponding range of values output by Manifold. The parameters specified values have exact values, with the upper limit labelled as *Same*. Rows that have changed from Table 5.4 are in bold. The label *ep_c* refers to the electrophoretic cross node at the intersection of the two channels.

5.4 Physical Interpretation of Results [RQ3]

The previous two sections have discussed interpretations of the $x_{detector}$, both as an output of and as an input to computation. Now explanations of how to interpret the physical meaning of the output of some of the other variables, such as the unspecified design parameters, will be provided.

Electrophorectic Cross Placement: In both Tables 5.4 and 5.5 it can be seen that the variables ep_c_node_x and ep_c_node_y are now a fixed value since these were specified within Manifold. This helps to make the system less underspecified, reducing the ranges of values for the other parameters of the system compared to the T-junction droplet generator device that was looked at in the previous section.

Output Channel Resistance: The range for the anode_channel_resistance is tightly constrained to 8.307E+11 $kg/(m^4 * s)$. This is due to this value being a large order of magnitude resulting in the δ parameter specified within dReal to be much smaller than it, meaning that it is unable to resolve a difference on that order of magnitude.

Channel Pressures: The pressure and flow rate in this device have no meaning for the electrophoretic cross device as the samples are flowing due to electrophoretic forces, not pressure driven flow. Their existence in the output is due to the structure of the code requiring all nodes and channels to have values computed for these. The flow rate should be 0 in the separation channels since it is assumed that there is no bulk flow of the gel due to pressure, however Manifold still solves for it since the anode is a sub class of the port class, which contains the parameters and constraints for the flow rate and viscosity. However, since there is no pressure at one end of the channel this flow rate should be close to 0.

Time: The time that is predicted for an analyte to flow past the detector is another parameter that a designer could be interested in knowing. The ranges output by Manifold are reasonably close to the measured times shown in Figure 5.2, and the range is narrowed when the position of the detector is set to a fixed value. This makes sense as Manifold would be able to know the time it takes to reach the detector when the distance to the detector is known.

5.5 Expected Parameter Correlations are Predicted [RQ6&7]

Table 5.6 shows the impact on the anode detector position of doubling and halving select design parameters. The electrophoretic cross equations take longer for Manifold/dReal to solve than do the T-junction equations, so this parameter sweep was limited to four design parameters. The anode voltage has the greatest impact on the position of the anode detector, which is consistent with the inspirational devices.

Design Parameter	Double % Change	Halving % Change	Slope of Line of Best Fit
Static/Manufacturing			
ep_c_anode_width	5	8	0.07
ep_c_cathode_width	25	18	0.3
channel_height	19	23	0.2
Dynamic/Operating			
anode_voltage	96	58	1

Table 5.6: The percentage change of $x_{detector}$ when each of the listed *design parameters* were individually doubled, halved and then the slope of the trend line.

Table 5.7 shows the results of perturbing each of these design parameters by 10%, which is used as a representative value of the kind of natural variation that might occur in the manufacturing process or during experimental operation. Similar to above, the anode voltage clearly has the greatest impact on the detector position.

Category	Design Parameter	Lower Position	Upper Position
Static	ep_c_anode_width	7.922E-03	8.052E-03
	ep_c_cathode_width	7.357E-03	8.602E-03
	channel_height	6.863E-03	9.162E-03
Dynamic	anode_voltage	5.126E-03	1.569E-02

Table 5.7: The mean value predicted for the $x_{detector}$ by Manifold when each of the listed *design parameters* were individually increased and decreased by 10% (upper and lower limit respectively). The original position is $8.000E - 03m$.

Chapter 6

Design Limitations

This chapter discusses some of the design choices that were made in developing and verifying this simulation software. Due to the complex nature of the fluid dynamics behind the microfluidic devices, several approximations had to be made in order to create analytical equations to constrain the parameters of the microfluidic channel to.

6.1 Limitations of Mathematical Models

In order to find analytical equations that outline the physics of real-world situations, approximations and constraints had to be imposed that are reasonable for the microfluidics being simulated.

The limitations specific to the T-junction droplet generator were outlined previously by Atulan Zaman [16] and will be summarized below.

6.1.1 Laminar Flow

One of the fundamental assumptions made in deriving these equations was that the fluid flow was laminar since this produces predictable fluid behaviour. However, there are microfluidic devices that operate outside of this regime, such as centrifugal and inertial flow microfluidics. As such in its current form, Manifold is unable to simulate these.

6.1.2 Transient Behavior

When microfluidic devices are ran there is an transient period as the fluid flows before it reaches steady state at the beginning and end of testing the device. During these phases, the fluid dynamics behave differently than those in steady state and are much more complex. As a result, this behaviour is not characterized by Manifold and all simulations assume steady state.

6.1.3 Limits of Van Steijn Model

For modeling the droplet generation in a T-junction, the mathematical model for droplet volume derived by Van Stein et al. was used [3]. Although it is a model that was proven to be experimentally robust, this required putting several limits on the parameters in the equations. First, that the ratio of the width of the two incoming channels have to be within a certain range, $0.33 \leq w_{out}/w_{in} \leq 3$. Second, the ratio between the height and width of the individual channels has to be within $0.1 \leq h/w \leq 0.5$. Third, the capillary number $\mu V/\sigma$ where μ is the dynamic viscosity, V is the velocity and σ is surface tensions, for the system has to be smaller than 0.01.

6.1.4 Limits of Stephen Chou's Electrophoretic Model

One of the assumptions in the model created by Stephen Chou is that there is at least four analytes in the solution. This is not too restricting, as usually control samples are ran with the sample to provide peaks at known analyte sizes which calibrates the reading so that the relative position of the peaks can be determined. Another assumption is that the diffusion of the samples is low relative to the velocity through the channel, which may not be true for every sample, especially those with a high diffusion coefficient. Also, the height and width of the separation, tail, droplet input, and droplet output channels are assumed to be equal, devices that deviate too far from this may result in a less accurate simulation.

6.2 Limitations of dReal

Manifold uses the dReal SMT solver to compute ranges for unknown design parameters. This facilitates the microfluidics designer doing preliminary design exploration and feasibility. As the designer refines and completes their design, they can switch over to other computational tools such as MapleSim or COMSOL. Manifold is intended to provide a unified way to describe the design that can be communicated to these tools. While computing ranges of unknown parameters for partially specified designs is a powerful feature, it does involve some limitations:

- **Formula simplification:** Sometimes formulas need to be simplified for use with dReal. In particular, partial derivatives. dReal is used here to solve analytic equations that do not describe the time or space varying behaviour that actually occurs in the physical world.
- **Orders of Magnitude:** dReal works best with systems of equations in which all of the variables take values from the same order of magnitude. The microfluidic equations used here often have variables of vastly different orders of magnitude. This range might cause dReal to find ranges that are too wide for some variables and too narrow for others. Scaling all of the variables to have the same order of magnitude (see Future Work) might produce better ranges — and it might help the computation complete more quickly as well.
- **No Stochastic Variation:** dReal has no ability to consider stochastic variation in variable values. This functionality would potentially be useful for a microfluidic designer, because real devices are never ideal.

6.3 Limitations in Microfluidic Manufacturing

The devices that were manufactured for verifying Manifold's simulations also had limitations due to the manufacturing process used. The limitations, their reasons and the impact it had will be outlined below:

6.3.1 Electrical Setup

Due to the high voltage (up to 2000V) required for performing electrophoresis, a jig was used to align the tubes and electrical connectors to the electrophoretic cross device. This jig was built in a position that had a limited ability to have it's dimensions adjusted, so this limited the range of sizes for the electrophoretic cross. This was set to hold a detector at a distance of 8mm which was found to perform well in the experiments performed by Tianchi [22].

6.3.2 Laser Etching

In order to perform laser etching, acrylic was used as the substrate. This allows for faster prototyping cycles and reduced the cost of manufacturing these devices. However, acrylic being etched by a laser cutter has more variability in depth than with other microfabrication techniques such as soft lithography of PDMS. The use of a laser etching machine also limits the minimum size of the channel to the focus size of the laser, which in this work was around $200\mu m$. In order to get a channel size that smaller than this would require a different manufacturing technique.

Chapter 7

Conclusion

The ability to automate the design and simulation of devices is potentially useful, allowing designers to create more complex devices in fewer design iterations. This is an area that is currently lacking for microfluidic devices due to the complexity of the physics underlying the mechanisms that these devices use. As the applications for microfluidics continue to grow with lab-on-a-chip devices, the need for design automation will grow as well.

This thesis describes the improvements that were made to the microfluidic design automation language Manifold in order to give it more functionality and improve its usability. The major contributions are in the following areas:

- Addition of the ability to simulate an electrophoretic cross
- Automated output of a Modelica file based on the specified device
- Verification of the output with real devices
- Creation of the Python library

The fabrication and testing procedures outlined in Chapter 2 outline how these physical devices were evaluated and will allow for other researchers to replicate the results found in this work. The underlying physics used as constraints for the evaluation of parameters for T-junction droplet generator and the electrophoretic cross are explained in Chapter 3. The output ranges for each of these parameters from the SMT solver dReal are evaluated in Chapter 4 and Chapter 5 and compared to what values were experienced in physical devices. Chapter 4 analyzes this for the T-junction droplet generator and demonstrates how Manifold is able to come up with reasonable ranges for the parameters of this device in a underspecified system. Chapter 5 provides a representative example of how an electrophoretic cross device can be simulated within Manifold using placeholder values to reproduce results from historical data of a physical device. This is done by proposing seven research questions and demonstrating how Manifold is able to answer them. An example of the design feedback provided by the Modelica design file is provided. In order to produce the constraints for dReal based on the physics of the real-world devices there were some limitations that were imposed on the equations in order to allow for them to be solved in a reasonable amount of time. These software limitations are outlined in Chapter 6 along with some of the limitations of the manufacturing process.

7.1 Future Work

Empirical Validation with Other Historical Devices. It would be worthwhile to conduct more case studies of the kind presented here, with closer correlation to specific historical devices.

Empirical Validation by Design of New Devices. Devices should also be built based on the output of Manifold to test different values for each parameter within the range output by dReal. This will allow for a more extensive validation to be performed. This would determine if the entire range of values for each parameter is valid, and if not to help come up with additional constraints that should be applied to the system.

Normalize Orders of Magnitude when Solving with dReal. The values for all of the parameters used within Manifold should also be normalized so that they are all around the order of magnitude 1. This would allow for δ to be set to 10 and it would be able to determine accurate ranges for every parameter. However, this would require modifying the underlying equations in order to ensure that the normalized values are denormalized before being used in the equations to ensure that they are still accurate.

Enriching the Mathematical Models used with Manifold/dReal. Mathematical modelling for computation is a trade-off between accuracy and richness of the models and computational feasibility. This work demonstrates that the current mathematical models are computationally feasible with dReal. Perhaps, then, there might be some targeted opportunities to enrich the models and improve accuracy while still retaining computational feasibility. For example, in the electrophoretic cross study, the model assumed that diffusivity was a constant property of the media, when in fact it varies with charge and electric field.

Generalizing the Modelica Output. Generalizing the Modelica output to not be MapleSim specific would allow for the use of any Modelica software. Since it is an open source language, it would make sense to allow the use of open source simulation tools like OpenModelica in the future. Adding the ability to generate the Modelica file for more microfluidic components besides the T-junction droplet generator would also be beneficial so that any designed device could be simulated in the time-domain.

GUI. The addition of a drag and drop user interface would make for an intuitive experience for users to design their microfluidic devices. There is a project out of the University of Washington called 3DuF [34] which provides a web interface to design microfluidic devices. This has the option to output a JSON of the device design, which could be read by Manifold to provide the limits on each of the dimensions of the channel. To add onto this, adding the ability to simulate each of the components available in 3DuF would allow for better integration. This would require the addition of the ability to simulate microfluidic mixers and splitters.

References

[1] Z. Strike, K. Ghofrani, and C. Backhouse, “Co2 laser-based rapid prototyping of micropumps,” *Micromachines*, vol. 9, no. 5, 2018.

[2] M. Berzish, “A software toolchain for physical system description and synthesis, and applications to microfluidic design automation,” Master’s thesis, University of Waterloo, 200 University Avenue West Waterloo, ON, Canada N2L 3G1, 4 2016.

[3] V. van Steijn, C. R. Kleijn, and M. T. Kreutzer, “Predictive model for the size of bubbles and droplets created in microfluidic t-junctions,” *Lab Chip*, vol. 10, pp. 2513–2518, 2010.

[4] C. N. Baroud, F. Gallaire, and R. Dangla, “Dynamics of microfluidic droplets,” *Lab Chip*, vol. 10, pp. 2032–2045, 2010.

[5] G. Whitesides, “The origins and the future of microfluidics,” *Nature*, vol. 442, 01 2006.

[6] C. M. Pandey, S. Augustine, S. Kumar, S. Kumar, S. Nara, S. Srivastava, and B. D. Malhotra, “Microfluidics based point-of-care diagnostics,” *Biotechnology Journal*, vol. 13, no. 1, p. 1700047, 2018.

[7] Q. Zhang and R. Austin, “Applications of microfluidics in stem cell biology,” *BioNanoScience*, vol. 2, pp. 277–286, 12 2012.

[8] L. Y. Yeo, H.-C. Chang, P. P. Y. Chan, and J. R. Friend, “Microfluidic devices for bioapplications,” *Small*, vol. 7, no. 1, pp. 12–48, 2011.

[9] R. W Muthard and S. L Diamond, “Side view thrombosis microfluidic device with controllable wall shear rate and transthrombus pressure gradient,” *Lab on a chip*, vol. 13, 04 2013.

[10] P. Pop, W. Hassan Minhass, and J. Madsen, *Microfluidic Very Large Scale Integration (VLSI)*. Springer International Publishing, 01 2016.

[11] D. Poli, G. Lutzemberger, and P. Pelacchi, "Electromechanical dynamics of overhead electric power line conductors analysed through the modelica language models," in *Power Systems Computation Conference*, pp. 1–7, 06 2016.

[12] I. E. Araci and P. Brisk, "Recent developments in microfluidic large scale integration," *Current Opinion in Biotechnology*, vol. 25, pp. 60 – 68, 2014. Analytical biotechnology.

[13] D. M. T. H. N. Gleichmann, P. Horbert, "System simulation for microfluidic design automation of lab-on-a-chip devices," in *International Conference on Miniaturized Systems for Chemistry and Life Sciences* (J. Landers, ed.), vol. 15, pp. 915 – 917, Royal Society of Chemistry, Chemical and Biological Microsystems Society, 2011.

[14] S. Gao, S. Kong, and E. M. Clarke, "dreal: An smt solver for nonlinear theories over the reals," in *Automated Deduction – CADE-24* (M. P. Bonacina, ed.), (Berlin, Heidelberg), pp. 208–214, Springer Berlin Heidelberg, 2013.

[15] N. Klassen, M. Lyons, M. Prysiazny, P. Roth, P. Socha, M. Berzish, A. Zaman, and D. Rayside, "Manifold 2.0: A hardware description language for microfluidic devices," in *Canadian Conference on Electrical and Computer Engineering*, pp. 1–5, 04 2017.

[16] A. Zaman, "Exploring microfluidic design automation: Thin-wall membrane regulator," Master's thesis, University of Waterloo, 200 University Avenue West Waterloo, ON, Canada N2L 3G1, 5 2017.

[17] H. G. Kerkhoff and H. P. Hendriks, "Fault modeling and fault simulation in mixed micro-fluidic microelectronic systems," *Journal of Electronic Testing*, vol. 17, pp. 427–437, Oct 2001.

[18] P. Bienstman, L. Vanholme, W. Booafrts, P. Dumon, and P. Vandersteegen, "Python in nanophotonics research," *Computing in Science & Engineering*, vol. 9, pp. 46–47, 06 2007.

[19] T. M. Association, "Modelica." https://www.modelica.org/, 2000.

[20] V. Mengeaud, J. Josserand, and H. Girault, "Mixing processes in a zigzag microchannel: Finite element simulations and optical study," *Analytical chemistry*, vol. 74, pp. 4279–86, 08 2002.

[21] U. L. Systems, “Vls desktop series.” https://cdn.ulsinc.com/assets/pdf/platforms/592b989576b6741c15762261/vls_desktop_spec_sheet_2017-03.pdf, 2016.

[22] T. Ma, “An integrated polymer based polymerase chain reaction and capillary electrophoresis system for genetic diagnosis,” Master’s thesis, University of Waterloo, 200 University Avenue West Waterloo, ON, Canada N2L 3G1, 2014.

[23] H. Schmid and B. Michel, “Siloxane polymers for high-resolution, high-accuracy soft lithography,” *Macromolecules*, vol. 33, 03 2000.

[24] F. Qian, Z. He, M. P. Thelen, and Y. Li, “A microfluidic microbial fuel cell fabricated by soft lithography,” *Bioresource Technology*, vol. 102, no. 10, pp. 5836 – 5840, 2011.

[25] H. Bruus, *Theoretical Microfluidics*. Oxford University Press, 01 2008.

[26] M. Shojaeian and S. Hardt, “Fast electric control of the droplet size in a microfluidic t-junction droplet generator,” *Applied Physics Letters*, vol. 112, no. 19, p. 194102, 2018.

[27] P. Garstecki, M. J. Fuerstman, H. A. Stone, and G. M. Whitesides, “Formation of droplets and bubbles in a microfluidic t-junctionscaling and mechanism of break-up,” *Lab Chip*, vol. 6, pp. 437–446, 2006.

[28] J. Louis Viovy, T. Duke, and F. Caron, “The physics of dna electrophoresis,” *Contemporary Physics - CONTEMP PHYS*, vol. 33, pp. 25–40, 01 1992.

[29] C. Backhouse, *Engineering of Nanobiotechnological Systems*. Garland Publishing, 2015.

[30] D. Botin, J. Wenzl, R. Niu, and T. Palberg, “Colloidal electro-phoresis in the presence of symmetric and asymmetric electro-osmotic flow,” *Soft Matter*, vol. 14, pp. 8191–8204, 2018.

[31] H. Stone, A. Stroock, and A. Ajdari, “Engineering flows in small devices: Microfluidics toward a lab-on-a-chip,” *Annu. Rev. Fluid Mech.*, vol. 36, pp. 381–411, 01 2004.

[32] J. Landers, “Introduction to capillary electrophoresis in capillary and microchip electrophoresis and associated microtechniques,” *Handbook of Capillary and Microchip Electrophoresis and Associated Microtechniques*, 01 2008.

[33] J. Crank, *The Mathematics of Diffusion*, vol. 44. Clarendon Press, Oxford, Second Ed., 01 1975.

[34] R. Sanka, J. Lippai, D. Samarasekera, S. Nemsick, and D. Densmore, “3df - interactive design environment for continuous flow microfluidic devices,” *Scientific Reports*, vol. 9, 12 2019.

[35] S. Choi, M. Gwon Lee, and J.-K. Park, “Microfluidic parallel circuit for measurement of hydraulic resistance,” *Biomicrofluidics*, vol. 4, 09 2010.

APPENDICES

Appendix A

Full Manifold Output

Parameter	Units	User	Lower	Upper
Name		Specified	Limit	Limit
continuous_viscosity	$kg/(m*s)$	Yes	3.050E-04	3.050E-04
continuous_pressure	$kg/(m*s^2)$	No	5.002E+05	5.018E+05
continuous_flow_rate	m^3/s	No	4.697E-08	4.704E-08
continuous_density	$kg/(m^3)$	Yes	8.000E+02	8.000E+02
continuous_x	m	No	0	7.620E-02
continuous_y	m	No	0	2.540E-02
dispersed_viscosity	$kg/(m*s)$	Yes	1.000E-03	1.000E-03
dispersed_pressure	$kg/(m*s^2)$	No	9.844E+05	9.851E+05
dispersed_flow_rate	m^3/s	No	1.864E-09	1.864E-09
dispersed_density	$kg/(m^3)$	Yes	9.999E+02	9.999E+02
dispersed_x	m	No	0	7.620E-02
dispersed_y	m	No	0	2.540E-02
out_viscosity	$kg/(m*s)$	Yes	3.050E-04	3.050E-04
out_pressure	$kg/(m*s^2)$	No	9.837E+05	9.843E+05
out_flow_rate	m^3/s	No	1.911E-09	1.911E-09
out_density	$kg/(m^3)$	Yes	9.999E+02	9.999E+02
out_x	m	No	0	7.620E-02
out_y	m	No	0	2.540E-02
t_j_pressure	$kg/(m*s^2)$	No	9.847E+05	9.863E+05
t_j_flow_rate	m^3/s	No	1.911E-09	1.911E-09
t_j_viscosity	$kg/(m*s)$	No	6.562E+01	6.875E+01
t_j_density	$kg/(m^3)$	No	9.999E+02	9.999E+02
t_j_x	m	No	0	7.620E-02
t_j_y	m	No	0	2.540E-02
t_j_out_length	m	No	1.000E-09	7.620E-02
t_j_out_width	m	Yes	2.100E-04	2.100E-04
t_j_out_height	m	Yes	2.000E-04	2.000E-04
t_j_out_flow_rate	m^3/s	No	1.911E-09	1.911E-09
t_j_out_droplet_volume	m^3	No	1.055E-11	4.061E-11
t_j_out_viscosity	$kg/(m*s)$	No	6.562E+01	6.875E+01
t_j_out_resistance	$kg/(m^4*s)$	No	2.705E+11	2.705E+11
continuous_t_j_length	m	No	1.000E-09	1.041E-09
continuous_t_j_width	m	Yes	2.100E-04	2.100E-04
continuous_t_j_height	m	Yes	2.000E-04	2.000E-04
continuous_t_j_flow_rate	m^3/s	No	4.697E-08	4.704E-08
continuous_t_j_viscosity	$kg/(m*s)$	No	3.050E-04	3.050E-04
continuous_t_j_resistance	$kg/(m^4*s)$	No	5.888E+10	5.888E+10
dispersed_t_j_length	m	No	1.000E-09	1.041E-09
dispersed_t_j_width	m	Yes	2.100E-04	2.100E-04
dispersed_t_j_height	m	Yes	2.000E-04	2.000E-04
dispersed_t_j_flow_rate	m^3/s	No	1.864E-09	1.864E-09
dispersed_t_j_viscosity	$kg/(m*s)$	No	1.000E-03	1.000E-03
dispersed_t_j_resistance	$kg/(m^4*s)$	No	4.737E+11	4.737E+11

Table A.1: Table of the data from the T-junction droplet generator device along with the corresponding range of values output by Manifold. The conditions were the same as referenced for Table 4.1.

Parameter Name	Units	User Specified	Lower Limit	Upper Limit
epsilon	m	No	2.100E-06	2.100E-06
cathode_viscosity	$kg/(m * s)$	No	4.062E+01	4.375E+01
cathode_pressure	$kg/(m * s^2)$	No	4.553E+02	4.579E+02
cathode_flow_rate	m^3/s	No	3.388E-08	3.401E-08
cathode_density	$kg/(m^3)$	No	1.396E+03	1.399E+03
cathode_x	m	Yes	0	7.620E-02
cathode_y	m	Yes	0	2.540E-02
cathode_voltage	m	Yes	2.000E+02	2.000E+02
anode_viscosity	$kg/(m * s)$	No	4.500E+01	5.500E+01
anode_pressure	$kg/(m * s^2)$	No	1.470E+02	1.470E+02
anode_flow_rate	m^3/s	No	9.760E-04	9.760E-04
anode_density	$kg/(m^3)$	No	1.396E+03	1.399E+03
anode_x	m	Yes	0	7.620E-02
anode_y	m	Yes	0	2.540E-02
anode_voltage	m	Yes	0	0
in_viscosity	$kg/(m * s)$	Yes	1.000E-03	1.000E-03
in_pressure	$kg/(m * s^2)$	No	9.576E+02	9.580E+02
in_flow_rate	m^3/s	No	1.838E-09	1.838E-09
in_density	$kg/(m^3)$	Yes	9.999E+02	9.999E+02
in_x	m	Yes	0	7.620E-02
in_y	m	Yes	0	2.540E-02
out_viscosity	$kg/(m * s)$	No	4.500E+01	5.500E+01
out_pressure	$kg/(m * s^2)$	No	5.548E+02	5.556E+02
out_flow_rate	m^3/s	No	9.760E-04	9.760E-04
out_density	$kg/(m^3)$	No	1.396E+03	1.399E+03
out_x	m	Yes	0	7.620E-02
out_y	m	Yes	0	2.540E-02
ep_c_pressure	$kg/(m * s^2)$	No	9.578E+02	9.583E+02
ep_c_flow_rate	m^3/s	No	9.760E-04	9.760E-04
ep_c_viscosity	$kg/(m * s)$	Yes	1.500E-04	1.500E-04
ep_c_density	$kg/(m^3)$	No	1.396E+03	1.399E+03
ep_c_x	m	Yes	0	7.620E-02
ep_c_y	m	Yes	0	2.540E-02

Table A.2: Table of the data from the electrophoretic cross device along with the corresponding range of values output by Manifold (Continued on next page). The conditions were the same as referenced for Table 5.4.

Parameter	Units	User	Lower	Upper
Name		Specified	Limit	Limit
cathode_ep_c_length	m	No	1.000E-03	7.620E-02
cathode_ep_c_width	m	Yes	2.100E-04	2.100E-04
cathode_ep_c_height	m	Yes	2.000E-04	2.000E-04
cathode_ep_c_flow_rate	m^3/s	No	3.388E-08	3.401E-08
cathode_ep_c_viscosity	$kg/(m*s)$	No	4.062E+01	4.375E+01
cathode_ep_c_resistance	$kg/(m^4*s)$	No	7.550E+10	7.550E+10
ep_c_anode_length	m	No	4.000E-02	4.000E-02
ep_c_anode_width	m	Yes	2.100E-04	2.100E-04
ep_c_anode_height	m	Yes	2.000E-04	2.000E-04
ep_c_anode_flow_rate	m^3/s	No	9.760E-09	9.770E-09
ep_c_anode_viscosity	$kg/(m*s)$	Yes	1.500E-04	1.500E-04
ep_c_anode_resistance	$kg/(m^4*s)$	No	8.307E+11	8.307E+11
ep_c_anode_x_detector	m	No	6.270E-03	8.240E-03
in_ep_c_length	m	No	1.000E-03	2.540E-02
in_ep_c_width	m	Yes	2.100E-04	2.100E-04
in_ep_c_height	m	Yes	2.000E-04	2.000E-04
in_ep_c_flow_rate	m^3/s	No	1.838E-08	1.838E-08
in_ep_c_viscosity	$kg/(m*s)$	Yes	1.000E-03	1.000E-03
in_ep_c_resistance	$kg/(m^4*s)$	No	1.693E+11	1.693E+11
ep_c_out_length	m	No	1.000E-03	2.540E-02
ep_c_out_width	m	Yes	2.100E-04	2.100E-04
ep_c_out_height	m	Yes	2.000E-04	2.000E-04
ep_c_out_flow_rate	m^3/s	No	9.760E-09	9.760E-09
ep_c_out_viscosity	$kg/(m*s)$	Yes	1.500E-04	1.500E-04
ep_c_out_resistance	$kg/(m*s)$	No	9.245E+11	9.245E+11
e	$kg/(m^4*s)$	No	3.333E+01	3.333E+01
mu_0	$m^2/s*V$	No	9.998E-08	9.998E-08
v_0	m/s	No	3.332E-06	2.784E-05
t_peak_0	s	No	66.735	73.365
t_min_0	s	No	63.221	70.291
c_negligible	kg/m^3	No	4.474	4.474
c_floor	kg/m^3	No	2.237	2.237
sigma0	m	No	4.459E-05	4.459E-05
diff_0	unitless	No	1.000	1.000
diff_1	unitless	No	1.000	1.000
diff_2	unitless	No	1.000	1.000

Table A.3: Table of the data from the electrophoretic cross device along with the corresponding range of values output by Manifold. The conditions were the same as referenced for Table 5.4.

Appendix B

Replicating Results

The code for this project is publicly available at www.github.com/manifold-lang/pymanifold. Instructions for how to setup and run it are available in the README file in the root of that repo, and will be duplicated here. This project provides a Docker image that can be used to provide a consistent environment to all users. The version of dReal used is 4.18.05.2 and Python version 3.6.

B.1 Installation

This can be installed within Python 3 using *pip install –user pymanifold* However this will require building dReal4 from source and installing OMPython from GitHub along with OpenModelica if you need electrical simulations, alternatively we provide a Docker image which contains all of these libraries baked in. You can get the docker image by running:

```
docker pull jsreid13/pymanifold:latest
```

Run the single_channel_test.py script within this image using:

```
docker container run -it –rm -v $(pwd):/tmp -v $(pwd)/src:/tmp/src -w /tmp -e PYTHONPATH=/tmp jsreid13/pymanifold python3 tests/t_junction_test.py
```

Note: You need to run this command in your terminal while in the root of this repository

Any script within this repo can be run using this command, to run your own script you need to change the directory that you run this command from within your terminal to the directory containing that script and change tests/single_channel_test.py to the name of your script

B.2 Tests

The test used in Chapter 4 T-Junction Droplet Generator Results and Analysis was the tests/single_channel_test.py script. The test used in Chapter 5 Electrophoretic Cross Results and Analysis was the tests/ep_cross_test.py script. This tests folder is also used by the continuous integration tool CircleCI to test the Manifold code to ensure it is working. Users wanting to test devices of their own within Manifold should begin by looking at these scripts as working examples of devices designed within Manifold. The commit that was last tested was fa46937d4160dd864e131b697617282e63103760.

Appendix C

Stephen Chou Electrophoretic Cross Reports

UNIVERSITY OF WATERLOO
Faculty of Engineering
Department of Electrical and Computer Engineering

Systematic Design of Single-Phase Microfluidic Circuits for Electrophoresis

Prepared by

Stephen Chou
ID: 20382554
4B Computer Engineering
Monday, April 27, 2015

Table of Contents

1 Introduction

Electrophoresis is an experimental technique that can be used to detect specific substances of interest – or *analytes* – that may be contained within a sample by separating molecules based on their size and electric charge. In the context of microfluidic circuits, the procedure has been frequently employed to separate DNA and other biological molecules [1]. Indeed, it has been suggested that microchip electrophoresis could be potentially useful for a wide variety of practical biotechnological applications, since microchip electrophoretic devices could be designed to separate analytes rapidly as well as to operate on a massively parallel scale [2].

Unfortunately, the design of practical devices and procedures for performing microchip electrophoresis still tends to be done in a relatively ad-hoc manner that greatly relies on the designer's familiarity with the typical design process, since the result of microchip electrophoresis heavily depends on a large number of parameters that can be difficult to precisely control. The ad-hoc nature of the typical design process makes it extremely challenging for very complex circuits to be designed and constructed, thereby making it difficult to exploit the potential advantages of microchip electrophoresis. With that in mind, the ultimate objective of this project is to make it possible for microfluidic circuit designers to systematically design reliable microfluidic circuits by simply specifying certain attributes and intended behaviours of the circuit being designed, in a manner that is analogous to how a hardware description language like VHDL could be used to design digital hardware circuits. As a proof of concept, we extended the Manifold compiler framework to (partially) support the synthesis of electrophoretic devices in the context of single-phase microfluidic circuits, taking into consideration the problems associated with sample leakage and the effects of applying pullback voltages.

The remainder of this report will begin with an overview of microchip electrophoresis in general, followed by a brief overview of the Manifold compiler framework. This will be followed by a detailed description of the mathematical models and constraints that were implemented in the Manifold compiler backend in order to model the effects of sample leakage on electropherogram output in the context of microchip electrophoresis. The report will then conclude with a summary of potential next steps that can be undertaken for the project overall.

2 Microchip Electrophoresis

While there exist many variations of electrophoresis, the main idea behind any form of electrophoresis is to pass an electric field through a sample consisting of charged molecules, thereby causing molecules of different substances to move at varying velocities (dependent on the molecules' sizes and electric charges) and separate accordingly. One practical approach for performing electrophoresis in the context of single-phase microfluidics is to perform the procedure on microchips with channels that are arranged as shown in Figure 1. (For the remainder of this report, the configuration shown in Figure 1 will be referred to as an *electrophoretic cross*.) The entire electrophoresis procedure can be broken down – in sequence – into a loading phase, an injection step and a separation phase. During the loading phase, an electric potential difference is applied between the sample and waste reservoirs in order to transport sample material that had originally been deposited in the sample reservoir to the intersection of the electrophoretic cross. The injection step is performed once a sufficient amount of sample material had been transported to the cross intersection and involves the application of an electric potential difference between the anode and cathode reservoirs. During the separation phase, the potential difference between the anode and cathode reservoirs is maintained as a well-defined amount of sample material (i.e. the *sample plug*) is transported to the anode reservoir, in order to enable the separation of analytes within the separation channel.

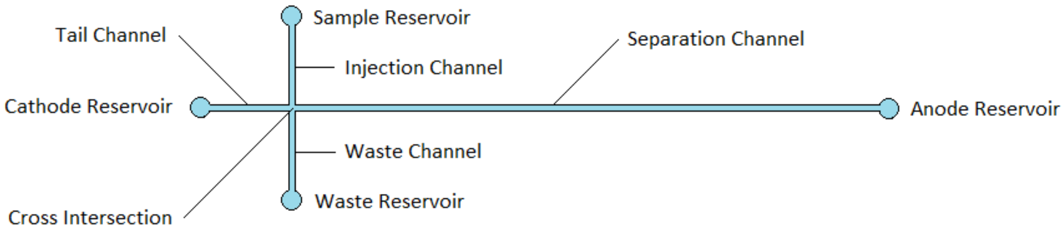

Figure 1. Schematic of an electrophoretic cross.

In order to determine whether a specific analyte is present in the sample, some form of detector can be positioned along the separation channel in order to measure the analyte's concentration over time at a specific location along the separation channel. If the analyte is naturally fluorescent or had been tagged with fluorescent tags prior to electrophoresis, then a fluorescence detector can be used to measure analyte concentration by measuring the level of fluorescence within the separation channel. The result of electrophoresis can be visualized in an *electropherogram*, which plots the concentration of analytes that can be detected at a specific position along the separation channel over a period of time.

3 Manifold Compiler

As alluded to earlier, the Manifold compiler enables microfluidic circuit designers to specify – using Manifold, which is a domain-specific language akin to VHDL – the high-level structure and behaviour of the microfluidic circuit that is being designed. Once a high-level schematic of the circuit has been specified in the Manifold language, the compiler backend could translate the schematic to mathematical equations that model and constrain the behaviour of the synthesized circuit, expressed as SMT formulas over the real numbers (SMT being essentially a generalization of Boolean SAT). The generated set of SMT formulas could then be evaluated by an SMT solver such as dReal [3] in order to identify a set of design parameters that could potentially enable the synthesized microfluidic circuit to satisfy specified requirements and constraints, after which the circuit could be more precisely simulated in Matlab in order to verify that the behaviour of the synthesized circuit conforms to specifications.

The compiler backend implements a set of what can be referred to as *strategies*, which are responsible for generating the SMT formulas that model and constrain the behaviour of the synthesized circuit. Each strategy is solely responsible for generating formulas that are either specific to a particular type of constraint or specific to a particular type of hardware component in the high-level schematic. For instance, one particular strategy is responsible for generating SMT formulas that constrain the physical placement of all hardware components to within the confines of a microchip, while another strategy is responsible for generating formulas that specifically model and constrain the behaviour of electrophoretic crosses.

The remainder of this report will concentrate on how the Manifold compiler backend had been extended to (partially) support the synthesis of single-phase microfluidic circuits that incorporate electrophoretic crosses. In particular, the report will describe the mathematical models that the compiler backend employs in order to model the effects of sample leakage on the electropherogram output, as well as describe how the compiler could compute the minimum amount of pullback voltage that has to be applied during the separation phase in order to sufficiently constrain the effects of sample leakage.

4 Mathematical Models

4.1 Sample Movement within an Electric Field

In general, a molecule that is contained in some form of bulk fluid moves within an electric field at a velocity that is governed by the relation

$$\vec{v} = \mu\vec{E} \tag{1}$$

where $\vec{v}$ denotes the velocity of the molecule, $\vec{E}$ denotes the electric field in which the molecule is placed and μ represents the *apparent mobility* of the molecule (which is a scalar quantity) [4]. The electric field can induce the molecule to move through two distinct mechanisms. Firstly, if the molecule itself has a non-neutral charge, then it would naturally move to a region of lower electric potential if it is positively charged or to a region of higher electric potential if it is negatively charged. Secondly, the presence of an electric field through the channel can induce the bulk fluid that is contained within the channel to move, thereby transporting molecules that are contained within the bulk fluid. Both electrophoretic flow (the motion that is independent of the bulk fluid) and electroosmotic flow (the motion of the bulk fluid) vary linearly with the electric field, meaning that the apparent mobility of a molecule can be decomposed as

$$\mu = \mu_{EP} + \mu_{EOF} \tag{2}$$

where μ_{EP} and μ_{EOF} denote electrophoretic mobility and electroosmotic mobility respectively [4].

In general, a molecule's electrophoretic mobility is dependent on a large number of parameters, including the charge, size and shape of the molecule as well as the viscosity, pH and composition of the bulk fluid in which the molecule is present [5]. For spherical polymers that have charge q and radius r and that have been placed in bulk fluid with viscosity η, [4] shows that electrophoretic mobility can be approximated as

$$\mu_{EP} = \frac{q}{6\pi\eta r}. \tag{3}$$

As the above equation suggests, a small spherical molecule that has a large electric charge would move within an electric field at a greater velocity than a large spherical molecule that has a small electric charge. Furthermore, within a given electric field, a molecule would move at a greater velocity if the bulk fluid has a lower viscosity, which should make intuitive sense.

Electroosmotic mobility is also dependent on a large number of parameters, some of which are properties of the bulk fluid while others are properties of the material from which the channel containing the bulk fluid is composed [4]. That being said,

$$\mu_{EOF} \approx 1.0 \times 10^{-8}\ \mathrm{m^2s^{-1}V^{-1}} \tag{4}$$

can be considered a reasonable rule-of-thumb for estimating the electroosmotic mobility [6]. It is important to note though that electroosmotic flow is not strictly speaking uniform across the entire cross-section of a channel; at distances that are relatively close to the channel wall (i.e. on the order of the Debye length), the velocity of the bulk fluid due to electroosmotic flow actually approaches zero. However, since the cross-sectional radius of a channel on microfluidic chips is usually several orders of magnitude greater than the Debye length, it is reasonable to treat electroosmotic flow as being practically uniform across the cross-section of a channel in the context of microfluidics [4].

4.2 Movement of Sample Plug

The movement of a sample within the separation channel can be visualized through the channel's concentration profile, which is a two-variable function that returns the concentration of a particular substance present at some specific position along the channel at some specific time. If one assumes that the movement of a sample within the separation channel is driven solely electrophoretic and electroosmotic flow, then the shape of the concentration profile within the channel would never deviate significantly from the shape of the concentration profile at the start of the injection phase, since all sample molecules would be moving at approximately uniform velocity. This means that if the initial concentration profile within the channel (which corresponds to the concentration profile $C(x,t)$ evaluated at $t=0$) is known and if one assumes that acceleration is instantaneous, then the concentration profile at any time t can be expressed in terms of the initial concentration profile as

$$C(x,t) = C(x - vt, 0) \tag{5}$$

where v denotes the velocity of the sample due electrophoretic and electroosmotic flow. Moreover, since fluorescence is directly proportional to sample concentration, the electropherogram output can be simply expressed in terms of $C(x_{detector}, t)$, where $x_{detector}$ is the distance of the fluorescence detector from where the sample was originally deposited (or the intersection of the channels in the context of an electrophoretic cross). This implies that if sample material does not diffuse within the separation channel, then the electropherogram output would actually retain the general shape of the initial concentration profile within the separation channel (but be horizontally scaled by the factor v).

In reality though, diffusion occurring within the separation channel would cause the length of the sample plug (along the axis of the separation channel) to increase at the same time the sample plug is moving down the separation channel. The change in the distribution of sample material over time can be modelled with Fick's second law of diffusion, for which analytical solutions exist for a wide variety of initial and boundary conditions when the diffusion coefficient is constant [7]. In particular, if the concentration profile of the sample plug at injection time can be approximated by a Gaussian function with the full width at half maximum (FWHM) equalling the radius of the channels (denoted $r_{channel}$), then [4] shows that the spread of the sample plug at any time t can be modelled as

$$\sigma(t) = \sigma_0 + \sqrt{2Dt} \tag{6}$$

where D is the diffusion coefficient for the analyte (assumed to be constant) and

$$\sigma_0 = \frac{r_{channel}}{2\sqrt{2\ln(2)}} \approx \frac{r_{channel}}{2.355}. \tag{7}$$

The concentration profile within the separation channel can then be expressed as

$$C(x,t) = \frac{C_0}{\sqrt{2\pi}\sigma(t)} \exp\left(\frac{-(x-vt)^2}{2(\sigma(t))^2}\right) \tag{8}$$

where C_0 is related to the concentration of sample material that was initially injected into the separation channel. From Equations (6 and (8, one can observe that the peak concentration of the sample plug decreases at the same time that the length of the sample plug is increasing, which is a straightforward consequence of the conservation of matter.

Determining the precise shape of the electropherogram output can be complicated when the general shape of the concentration profile varies over time as a consequence of diffusion. However, if the diffusion coefficient for the analyte is relatively small and if the sample plug is travelling at a relatively fast velocity, such that the constraint

$$t \ll \frac{\sigma_0^2}{2D} \tag{9}$$

holds true for the duration of the electrophoresis procedure, then $\sigma(t)$ can be simply approximated as σ_0 for any relevant value of t. In this scenario, the expression for the concentration profile within the separation channel reduces to approximately

$$C(x,t) \approx \frac{C_0}{\sqrt{2\pi}\sigma_0}\exp\left(\frac{-(x-vt)^2}{2\sigma_0^2}\right), \tag{10}$$

meaning that the shape of the concentration profile would be effectively constant and that, as the example in Figure 2 demonstrates, the shape of the electropherogram output could be approximated by a Gaussian function. In general though, if the effects of diffusion are relatively significant (i.e. if the diffusion coefficient is large), then the electropherogram output can have a shape that is drastically different from the concentration profile within the separation channel, as clearly evident in Figure 3.

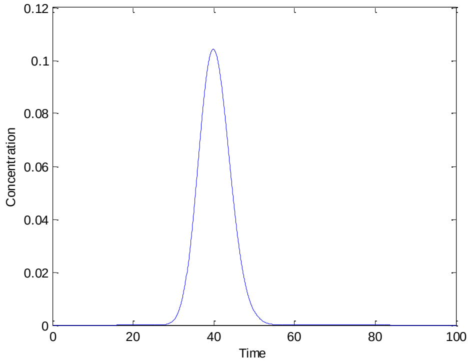

Figure 2. Electropherogram output for a hypothetical separation with $D = 0.1$, $x = 40$ and all other parameters in Gaussian concentration profile equalling 1. Observe that the shape of the electropherogram output curve is also approximately Gaussian.

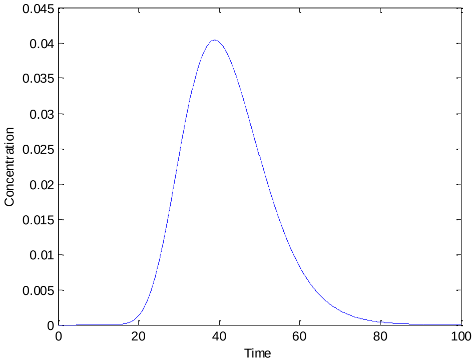

Figure 3. Electropherogram output for another hypothetical separation with $D = 1$, $x = 40$ and all other parameters in Gaussian concentration profile also equalling 1. Observe that the shape of the electropherogram output is no longer Gaussian.

The model that has been described so far should suffice in practice if the channels of the electrophoretic cross (in particular, the injection and waste channels) have circular cross-sections. If the channels have rectangular cross-sections though, then the concentration profile of the sample plug at the time of injection can be better modelled by a scaled rectangular function. In this scenario, the concentration profile within the separation channel can be more accurately expressed as

$$C(x,t) = \frac{C_0}{2}\left[\mathrm{erf}\left(\frac{\frac{w_{channel}}{2} - x + vt}{2\sqrt{Dt}}\right) + \mathrm{erf}\left(\frac{\frac{w_{channel}}{2} + x - vt}{2\sqrt{Dt}}\right)\right] \tag{11}$$

where $w_{channel}$ denotes the cross-sectional width of the injection and waste channels [7]. It should be noted though that, as can be observed in Figure 4, the concentration profile specified by Equation (11 begins to approximate a Gaussian function after separation has elapsed for a significant amount of time.

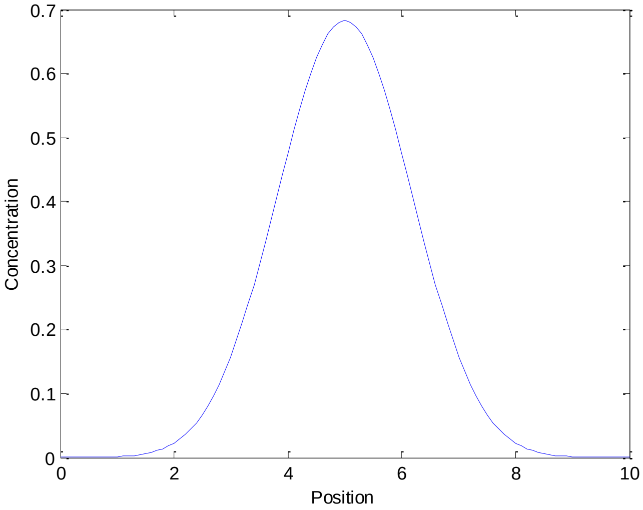

Figure 4. Concentration profile at $t = 5$ for a hypothetical electrophoretic separation in which the initial concentration profile is a scaled rectangular function. Observe that the shape of the concentration profile is approximately Gaussian.

4.3 Leakage of Sample Material

In an ideal world, only sample material that had been punched out during the injection step would be detectable by the fluorescence detector during electrophoresis, in which case the concentration profile within the separation channel is sufficient for completely describing the electropherogram output. However, as the sample plug moves away from the intersection of the electrophoretic cross during the separation phase, a concentration gradient between the cross intersection (which becomes increasingly devoid of sample material) and the injection and waste channels (which still retain sample material leftover from injection) would begin to emerge. If the sample material contained within the injection and waste channels are simply left untouched, then diffusion would naturally cause the leftover sample material to leak into the cross intersection. Since an electric field is maintained within the separation channel during the separation phase, the leaked sample material would be swept down the separation channel (in the same way that the sample plug is transported down the separation channel) and be eventually detected by the detector, thereby altering the electropherogram output.

Determining the precise concentration of sample material that has been leaked into the separation channel is challenging due to the fact that the concentration of sample material that is already present in the cross intersection (which affects the rate of leakage of sample material from the injection and waste channels) varies as the sample plug travels down the separation channel. However, for the purpose of synthesizing a design that satisfies specified constraints on a circuit's behaviour, it can be sufficient to establish a reasonable upper bound on the concentration of sample material that is leaked into the separation channel at any point in time. This upper bound can be obtained by considering the concentration profile within the cross intersection along the axis of the injection and waste channels, assuming that sample material that had been leaked into the cross intersection does not get swept down the separation channel. In particular, since sample material can only diffuse into the electrophoretic cross intersection if the sample concentration is lower within the cross intersection (as a consequence of Fick's first law of diffusion), and since sample material is swept down the separation channel almost as soon as the material leaks into the cross intersection, the concentration of sample material leaking into the separation channel at any point in time may be upper bounded by the sample concentration at the interface between the cross intersection and the end of the injection channel.

If one pessimistically assumes for simplicity that the cross intersection is initially devoid of sample material (i.e. there is no sample plug) and that the concentration of sample material within the injection channel is fixed with respect to time, then the concentration profile within the cross intersection – along the axis of the injection and waste channels – can be expressed as

$$C_{intersect}(x,t) = C_0 \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right) \tag{12}$$

for values of x that are significantly less than the diameter of the separation channel, where C_0 represents the initial concentration of analyte within the injection channel and x denotes the distance from the edge of the injection channel. (Note that this is simply the solution to the diffusion problem in a semi-infinite medium [7].) If a pullback voltage is applied at the sample and waste reservoirs (thereby generating electric fields between the cross intersection and the reservoirs), then leftover sample material within the injection and waste channels would be transported towards the reservoirs at some uniform velocity governed by Equation (5. The concentration profile given in Equation (12 could then be modified as

$$C_{intersect}(x,t) = C_0 \operatorname{erfc}\left(\frac{x+vt}{2\sqrt{Dt}}\right), \tag{13}$$

where v is the velocity (away from the cross intersection) of the sample material within the injection and waste channels due to electrophoretic and electroosmotic flows. With the modified concentration profile, the concentration of sample material that might be leaked into the separation channel at time t can then be upper bounded by the function $C_{intersect}(0,t)$.

4.3.1 Mass Transfer Model

Alternatively, the concentration of sample material that gets leaked into the separation channel at any arbitrary time can be estimated by considering how the quantity of sample material that remains within the injection and waste channels changes over time. In a scenario in which no pullback voltage is applied, the distance between the end of the injection channel and the boundary that marks the extent of sample material within the injection channel at time t can be reasonably approximated as $\sqrt{2Dt}$. If one assumes that the analyte concentration is initially C_0 throughout the entire injection channel, then the quantity of analyte that has leaked out of the injection channel – and leaked into the cross intersection – after time t has elapsed can be upper bounded by

$$m(t) = C_0 A\sqrt{2Dt} \tag{14}$$

where A denotes the cross-sectional area of the injection channel. The maximum quantity of analyte that could have possibly leaked out of the injection channel within an infinitesimally small period of time can then be expressed as

$$dm = C_0 A\sqrt{\frac{D}{2t}}dt. \tag{15}$$

Since sample material is leaked into the cross intersection from both the injection and waste channels, and due to the fact that sample material that has been leaked into the cross intersection would be almost immediately swept down the separation channel, the concentration of analyte that gets leaked into the separation channel at time t can be expressed as the ratio

$$C_{leak}(t) = \frac{2 \cdot dm}{Av \cdot dt} = \frac{2C_0}{v}\sqrt{\frac{D}{2t}} \tag{16}$$

where v denotes the velocity of electroosmotic and electrophoretic flow within the separation channel. (Note that $Av \cdot dt$ essentially represents the volume over which the leaked analyte is spread out.)

4.4 Constraining Sample Leakage Effects

From a high-level perspective, any allowable design for the electrophoretic cross must ensure that the electropherogram output is not significantly altered due to leakage of sample material from the injection and waste channels into the separation channel. More precisely, once the sample plug has moved away from the detector, the electropherogram output should not deviate significantly from the baseline level (i.e. the level of concentration at which one may deem the amount of analyte detected to be negligible).

One approach for defining the baseline level of the electropherogram output is to specify it as some fraction of the electropherogram output's peak value (specifically, the electropherogram output assuming there is no leakage of sample material from the injection and waste channels). In general, the peak value of the electropherogram output can be determined by treating the problem as a continuous function optimization problem. Since the electropherogram output can be expected to initially increase with time and then eventually decrease with time, the solution to the equation

$$\frac{\partial}{\partial t}C(x_{detector}, t) = 0 \tag{17}$$

would correspond to the time at which the electropherogram output reaches its peak value. If, for instance, the shape of the initial concentration profile is a Gaussian function, then the rate of change of the electropherogram output can be expressed as

$$\frac{\partial}{\partial t}C(x,t) = \frac{C_0}{\sqrt{2\pi}\big(\sigma(t)\big)^2}\left(\frac{v(x-vt)}{\sigma(t)} + \left(\frac{(x-vt)^2}{\big(\sigma(t)\big)^2} - 1\right)\sqrt{\frac{D}{2t}}\right)\exp\left(\frac{-(x-vt)^2}{2(\sigma(t))^2}\right) \tag{18}$$

where $\sigma(t) = \sigma_0 + \sqrt{2Dt}$ as before. In this scenario, observe is that at $t = x/v$,

$$\frac{\partial}{\partial t} C(x,t)\bigg|_{t=x/v} = \frac{-C_0}{2\left(\sigma_0 + \sqrt{\frac{2Dx}{v}}\right)^2} \sqrt{\frac{Dv}{\pi x}} \tag{19}$$

is always negative if D is non-zero, from which one could infer that the electropherogram output had already reached its peak value by the time the sample plug had been transported a distance of x down the separation channel. If D is sufficiently small though (i.e. if the effects of diffusion are not relatively significant, implying that the shape of the concentration profile never deviates significantly from the shape of the initial concentration profile), then the rate of change of the electropherogram output can still be reasonably approximated as zero at $t = x/v$. This implies that the time at which the electropherogram output reaches its peak value can be reasonably approximated as the time (denoted t_{peak}) at which the peak of the concentration profile within the separation channel has moved to the position of the detector, which simply corresponds to

$$t_{peak} = \frac{x_{detector}}{v} \tag{20}$$

where $x_{detector}$ denotes the distance of the detector from the electrophoretic cross intersection and v is the velocity of analyte within the separation channel. The value of the electropherogram output at t_{peak}, which should be reasonably close to the peak value of the electropherogram output (and could potentially be constrained as such), can then be computed as

$$C_{peak} = C(x_{detector}, t_{peak}), \tag{21}$$

after which the baseline concentration level can be defined as

$$C_{baseline} = kC_{peak} \tag{22}$$

where k is some constant, non-negative fraction that should be significantly less than one (e.g. 0.01).

Once the baseline concentration level has been established, the time at which the electropherogram output drops back down to the baseline level (denoted $t_{baseline}$) can be determined by simply finding the solution to the equation

$$C(x_{detector}, t) = C_{baseline} \tag{23}$$

subject to the constraint that the solution has to be strictly greater than t_{peak}. At $t_{baseline}$, it becomes more reasonable to assume (pessimistically) that the electropherogram output simply corresponds to the concentration of leaked analyte plus the concentration of analyte molecules that were from of the original sample plug. Furthermore, if one makes the assumption that leaked sample material does not diffuse within the separation channel as it is carried towards the anode reservoir, then the concentration of leaked analyte that could be detected at $t_{baseline}$ would clearly be upper bounded by

$$C_{leaked,max} = C_{intersect}(0, t_{baseline} - t_{peak}). \tag{24}$$

Consequently, if $C_{leaked,max}$ is constrained to be less than the baseline concentration level, then the effects of sample leakage on the electropherogram output can be effectively limited in theory.

5 Experimental Results

The mathematical models and constraints that have been described in Section 4 were implemented in the Manifold compiler backend as part of a strategy that generates SMT formulas specific to electrophoretic crosses. (It should be noted that for the purpose of quantifying sample material leakage into the separation channel, the model described in Section 4.3 was implemented as opposed to the model described in Section 4.3.1, since the latter has not been sufficiently developed.) For the most part, the equations in Section 4 could be quite trivially translated by the compiler backend to SMT formulas that are expressed in a format supported by dReal. Since dReal does not directly support the error function in SMT formulas though, the generated formulas instead use the rational approximation

$$\operatorname{erf}(x) \approx 1 - \frac{1}{(1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4)^4}, \tag{25}$$

where $a_1 = 0.278393$, $a_2 = 0.230389$, $a_3 = 0.000972$ and $a_4 = 0.078108$, in order to approximate the error function with a maximum error of 5×10^{-4} [8].

With the additions to the Manifold compiler backend implemented, we then attempted to synthesize a circuit that consisted of a single electrophoretic cross, which was designed to perform electrophoresis with DNA fragments (consisting of 1000 base pairs each) as the analyte. Empirical results have shown that the diffusion coefficient for DNA within certain types of bulk fluid can be reasonably approximated as

$$D \approx \frac{3 \times 10^{-10}}{N_{bp}^{0.57}} \ \mathrm{m^2/s} \tag{26}$$

where N_{bp} denotes the number of base pairs in the DNA fragments, from which an approximation of $5.85 \times 10^{-12}\ \mathrm{m^2/s}$ for the diffusion coefficient in the hypothetical test case was obtained [9]. Furthermore, empirical results have shown that the electrophoretic mobility of DNA fragments consisting of more than 1000 base pairs can be reasonably approximated as $-3.75 \times 10^{-8}\ \mathrm{m^2 s^{-1} V^{-1}}$ [9]. For the hypothetical input parameters listed in Table 1, a pullback voltage of approximately 4.02 V was computed by the Manifold compiler to be the minimum that had to be applied between the sample reservoir and the electrophoretic cross intersection in order to sufficiently constrain the effects of sample leakage on the electropherogram output, which does not appear to be an unreasonable value.

Table 1. Input parameters corresponding to hypothetical test case

Input Parameter	**Value**
Electrophoretic mobility	$-3.75 \times 10^{-8}\ \mathrm{m^2 s^{-1} V^{-1}}$
Electroosmotic mobility	$1.0 \times 10^{-8}\ \mathrm{m^2 s^{-1} V^{-1}}$
Diffusion coefficient	$5.85 \times 10^{-12}\ \mathrm{m^2/s}$
Separation channel length	$3.0 \times 10^{-2}\ \mathrm{m}$
Tail channel length	$4.5 \times 10^{-3}\ \mathrm{m}$
Injection channel length	$4.5 \times 10^{-3}\ \mathrm{m}$
Waste channel length	$4.5 \times 10^{-3}\ \mathrm{m}$
Detector distance from intersection	$2.5 \times 10^{-2}\ \mathrm{m}$
Channel radius	$5.0 \times 10^{-5}\ \mathrm{m}$
Cathode injection voltage	$-1.0 \times 10^{3}\ \mathrm{V}$
Anode injection voltage	$0\ \mathrm{V}$

We also examined how the magnitude of the pullback voltage computed by the Manifold compiler varies as the specified diffusion coefficient varies (assuming all other input parameters are fixed). As Figure 5 illustrates, the magnitude of the pullback voltage that is computed to be the minimum required for sufficiently constraining the effects of sample leakage increases as the diffusion coefficient of the analyte increases, which is not unexpected considering that the amount of sample leakage into the separation channel is heavily dependent on the rate of diffusion.

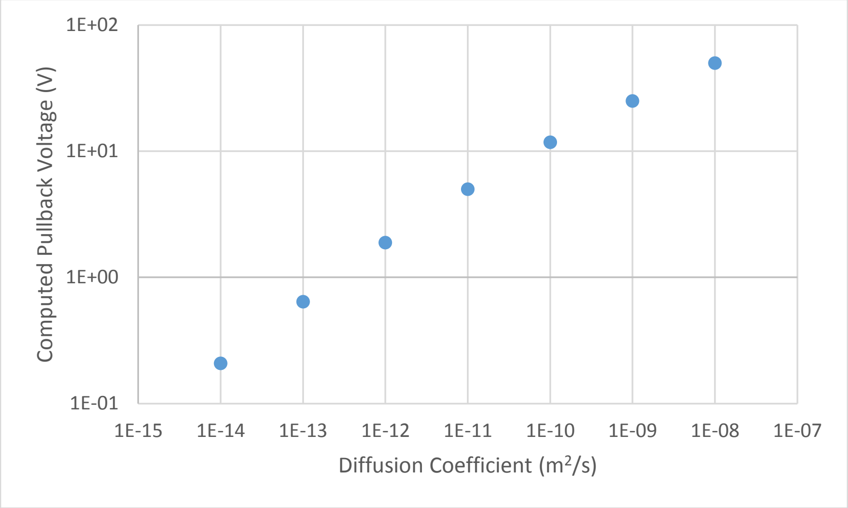

Figure 5. Computed pullback voltages as a function of the analyte’s diffusion coefficient.

6 Conclusion and Next Steps

The analyses and results that have been presented in this report suggest it is not infeasible to implement a synthesizer that would allow microchip electrophoresis devices to be designed in a more systematic fashion. That being said, much work remains to be completed before the Manifold compiler can be used to systematically synthesize reliable microchip electrophoretic devices. While the results presented in Section 5 do not appear unreasonable, without physically testing the computed parameters in a laboratory setting, it is impossible to conclude with certainty that the computed pullback voltages are in fact sufficient for constraining the effects of sample leakage on an actual electropherogram output. The mathematical model described in Section 4.3.1 could also be potentially extended to take into account the effects of applying a pullback voltage, which may enable the Manifold compiler backend to more accurately compute pullback voltages. Furthermore, while the models presented in Section 4 can provide pessimistic estimates of an electrophoretic cross’ behaviour, they do not necessarily describe the true behaviour of the device in an accurate manner; for the purpose of validating the synthesized design in a Matlab simulation, adjustments to some of the models – particularly those related to sample leakage – would be required. Finally, the compiler backend could be extended to model other physical processes that can potentially affect electropherogram output in the context of microchip electrophoresis, such as Joule heating and pressure-driven flow (which may occur within the separation channel after a significant amount of time has elapsed).

7 References

[1] V. M. Ugaz, "Electrophoresis in Microfluidic Systems," 2007. [Online]. Available: http://www.aesociety.org/areas/microchip_electrophoresis.php. [Accessed 23 April 2015].

[2] M. Vazquez, G. McKinley, L. Mitnik, S. Desmarais, P. Matsudaira and D. Ehrlich, "Electrophoretic Injection within Microdevices," *Analytical Chemistry,* vol. 74, no. 9, pp. 1952-1961, 2002.

[3] S. Gao, S. Kong and E. M. Clarke, "dReal: An SMT Solver for Nonlinear Theories over the Reals," in *Automated Deduction – CADE-24*, Berlin, Springer Berlin Heidelberg, 2013, pp. 208-214.

[4] C. J. Backhouse, *Engineering of Nanobiotechnological Systems,* Waterloo: Garland Publishing, 2015.

[5] World Health Organization, "WHO Pharmacopoeia Library," 2014. [Online]. Available: http://apps.who.int/phint/en/p/docf/. [Accessed 22 April 2015].

[6] H. A. Stone, A. D. Stroock and A. Ajdari, "ENGINEERING FLOWS IN SMALL DEVICES: Microfluidics Toward a Lab-on-a-Chip," *Annual Review of Fluid Mechanics,* vol. 36, pp. 381-411, 2004.

[7] J. Crank, The Mathematics of Diffusion, Oxford: Clarendon Press, 1975.

[8] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Washington: Dover Publications, 1972.

[9] B. J. Kirby, Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices, Cambridge: Cambridge University Press, 2010.

1. Background Information

Recall that if the initial concentration profile of some charged analyte (labelled i) within a separation channel can be described by a Gaussian curve, then the analyte's concentration profile at any point in time during electrophoresis can be described by the equation

$$C_i(x,t) = \frac{C_{0,i}}{\sqrt{2\pi}\sigma_i(t)} \exp\left(-\frac{1}{2}\left(\frac{(x - v_i t)}{\sigma_i(t)}\right)^2\right)$$

where $\sigma_i(t) = \sigma_0 + \sqrt{2D_i t}$, v_i denotes the velocity of the analyte along the separation channel, D_i denotes the analyte's diffusion coefficient (which is assumed to be constant), $C_{0,i}$ denotes the initial surface concentration of the analyte, t denotes the amount of time elapsed since electrophoresis began and x denotes the distance from the electrophoretic cross intersection. The rate of change of the concentration profile with respect to time can then also be expressed as

$$\begin{aligned}\frac{\partial}{\partial t} C_i(x,t) &= \frac{C_{0,i}}{\sqrt{2\pi}\left(\sigma_i(t)\right)^2}\left(\frac{v_i(x - v_i t)}{\sigma_i(t)} + \left(\left(\frac{(x - v_i t)}{\sigma_i(t)}\right)^2 - 1\right)\sqrt{\frac{D_i}{2t}}\right)\exp\left(-\frac{1}{2}\left(\frac{(x - v_i t)}{\sigma_i(t)}\right)^2\right) \\ &= \frac{C_i(x,t)}{\sigma_i(t)}\left(\frac{v_i(x - v_i t)}{\sigma_i(t)} + \left(\left(\frac{(x - v_i t)}{\sigma_i(t)}\right)^2 - 1\right)\sqrt{\frac{D_i}{2t}}\right)\end{aligned}$$

Now consider a sample that consists of n distinct charged analytes. As long as the concentration of each individual analyte in the sample is sufficiently low, one can reasonably assume that there is no interaction between the different analytes. In this scenario, the concentration profile of sample material in the separation channel can be approximated as

$$C(x,t) = \sum_{i=1}^{n} C_i(x,t),$$

while the concentration profile's rate of change with respect to time can be approximated as

$$\frac{\partial}{\partial t} C(x,t) = \frac{\partial}{\partial t}\sum_{i=1}^{n} C_i(x,t) = \sum_{i=1}^{n}\frac{\partial}{\partial t} C_i(x,t).$$

Assuming that some form of detector is positioned at a distance of $x_{detector}$ away from the intersection of the electrophoretic cross, we can define $F_i(t) = C_i(x_{detector}, t)$ to be the concentration of analyte i that can be detected at the position of the detector at time t. We can further define $F(t) = C(x_{detector}, t)$ to be the concentration of all sample material that can be detected at the position of the detector at time t – in other words, the electropherogram. Needless to say, the precise shape that $F(t)$ takes is dependent on a wide range of parameters, including attributes of the circuit being designed (such as the position of the detector and the length of the separation channel) as well as properties of the input sample (such as the diffusion coefficient and electrophoretic mobility of the analytes). For the remainder of this report though, we will make some reasonable assumptions about $F(t)$, namely that

1. $F(t)$ is first- and second-differentiable in the domain $(0, \infty)$,

2. $F(t)$ cannot contain more peaks than there are analytes in the input sample, where each peak is defined to start at some local minimum of $F(t)$ and end at the next local minimum, and
3. All local extrema of $F(t)$ are strict local extrema.

These assumptions allow us to assign labels to each local extremum of $F(t)$; in particular, we will let $t_{max,i}$ denote the i-th local maximum and $t_{min,i}$ denote the i-th local minimum. (For example, $t_{max,1}$ would denote the first local maximum to appear in the electropherogram.) Furthermore, we expect the local maxima and minima of the electropherogram to alternate; in other words, $t_{max,i} < t_{min,i} < t_{max,i+1}$ and $t_{min,i} < t_{max,i+1} < t_{min,i+1}$ for all i.

2. Precise Model

Intuitively, one can deem a sample to have been sufficiently separated if, in the electropherogram output, a peak can be discerned for each individual analyte contained in the sample. Mathematically, this implies that when a sample containing n distinct analytes has been sufficiently separated, the electropherogram must contain n (strict) local maxima and $n-1$ (strict) local minima, with each local maximum corresponding to a particular analyte (and vice versa). By treating the problem as a continuous function optimization problem, one can compute the values of the local maxima in an electropherogram by determining the times at which

$$\left.\frac{dF}{dt}\right|_{t=t_{max,i}} = 0 \text{ and } \left.\frac{d^2F}{dt^2}\right|_{t=t_{max,i}} < 0,$$

and similarly compute the values of the local minima by determining the times at which

$$\left.\frac{dF}{dt}\right|_{t=t_{min,i}} = 0 \text{ and } \left.\frac{d^2F}{dt^2}\right|_{t=t_{min,i}} > 0.$$

It should be apparent that in order for an individual peak to be discernable in practice, its maximum must be sufficiently greater than its two adjacent minima. Mathematically, this can be expressed as

$$\frac{F(t_{min,i})}{F(t_{max,i})} \leq c \tag{1}$$

and

$$\frac{F(t_{min,i})}{F(t_{max,i+1})} \leq c, \tag{2}$$

where c is a constant that must be greater than zero but less than one. (A smaller value of c would render the individual peaks more distinct.) At the same time, in order for analytes that have been separated to actually be observable in the real world, the concentrations of the separated analytes must be sufficiently greater than the minimum level that can be detected by the detector. Again, we can express this constraint mathematically as

$$F\left(t_{max,i}\right) \geq kC_{detectable}$$

where $C_{detectable}$ is the minimum concentration that can be sensed by the detector and k is a constant that must be greater than or equal to one. Furthermore, since practical detectors obviously cannot be expected to sample at arbitrarily high frequencies, one must ensure that a peak maximum is sufficiently far apart (in the time domain) from its adjacent minima so that the changes in sample concentration are more likely to be picked up by the detector. Mathematically, this can be expressed as

$$t_{max,i} + \delta < t_{min,i}$$

$$t_{min,i} + \delta < t_{max,i+1}$$

where δ must be no less – but should ideally be significantly greater, perhaps even by an order of magnitude – than the detector's minimum sampling period. (It should also be noted that preliminary experimental results have shown that selecting larger values for δ can drastically constrain the search space and thereby reduce the amount of time required for dReal to return an answer. Based on empirical observations, setting δ to be no less than one second appears reasonable.) Finally, to ensure each peak in the electropherogram roughly matches the concentration of a single analyte, one must require that

$$\frac{F_i\left(t_{max,i}\right)}{F\left(t_{max,i}\right)} \geq p \tag{3}$$

for all i, where p is a constant that must be less than one but should be relatively large (at least 0.5).

Excluding properties that are specific to analytes in the input sample (e.g. the diffusion coefficients), the mathematical model that has been described so far – as well as the other two models that this report will cover – can in principle be applied to solve for any design parameter that has not been explicitly specified in the incomplete circuit description. It should be pointed out though that the models described in the remainder of this report had been optimized to solve for the optimal placement of detectors along the separation channel (assuming all other parameters are known), since this would presumably be one of the more common use cases – if not the most common – in scenarios where a designer is attempting to construct circuits from electrophoretic crosses that had already been prefabricated.

3. Approximate Model

While it is technically possible to translate the precise model to SMT formulas that are solvable by dReal, it can be very expensive to compute all local maxima and minima in an electropherogram output when the sample contains more than a trivial number of analytes. Rather than attempting to precisely compute the local extrema then, it makes sense to instead find reasonable approximations for the local extrema.

To obtain reasonable estimates for the local maxima in an electropherogram, observe that there should actually be very little overlap between peaks corresponding to different analytes if a sample has been very well separated. In this scenario, one might reasonably approximate $F(t)$ as a piecewise function consisting of n pieces, with each piece corresponding to a particular $F_i(t)$. The i-th local maximum could then be reasonably approximated as $t_{peak,i}$, where $t_{peak,i}$ is defined as the time at which $F_i(t)$ reaches its maximum value and satisfies the relation

$$\left.\frac{dF_i}{dt}\right|_{t=t_{peak,i}} = 0.$$

Assuming that diffusion is not too significant though, one can further approximate $t_{peak,i}$ as

$$t_{peak,i} \approx \frac{x_{detector}}{v_i}. \tag{4}$$

The height of the i-th peak – i.e. $F(t_{max,i})$ – could then be approximated as $F(t_{peak,i})$. It is important to highlight that by approximating $t_{max,i}$ as $t_{peak,i}$, the approximate model drastically reduces the amount of time required for synthesis by eliminating the need for dReal to repeatedly evaluate $F'(t)$ and $F''(t)$.

Similarly, the local minima in an electropherogram can be reasonably estimated if we assume that a sample has been relatively well separated. In this scenario, it is reasonable to expect that at around time $t_{min,i}$, the only analytes that can be detected by the detector in any significant quantity would be the i-th analyte and the $(i+1)$-th analyte. It would therefore make sense to approximate $t_{min,i}$ as $\hat{t}_{min,i}$, where $\hat{t}_{min,i}$ is defined by the relation

$$\left.\frac{dF_i}{dt}\right|_{t=\hat{t}_{min,i}} + \left.\frac{dF_{i+1}}{dt}\right|_{t=\hat{t}_{min,i}} = 0. \tag{5}$$

When the heights of adjacent peaks are roughly equal (or at least in the same order of magnitude) though, it has been observed that the concentration of analyte i that can be detected at $t_{min,i}$ tends to also be roughly equal to the concentration of analyte $i+1$ that can be detected at the same time. In this scenario, rather than defining $\hat{t}_{min,i}$ by Equation (5), $\hat{t}_{min,i}$ can instead be simply defined by the relation

$$F_i(\hat{t}_{min,i}) = F_{i+1}(\hat{t}_{min,i}). \tag{6}$$

Both of the aforementioned approaches for approximating local minima in an electropherogram have their advantages and disadvantages. In particular, while the first alternative arguably has a sounder mathematical basis and is more likely to be more accurate when adjacent peaks have drastically different heights, the second alternative involves fewer mathematical operations and can therefore be expected to be solved by dReal more efficiently. Rather than solely relying on either one of the alternatives then, the Manifold compiler backend computes the heuristic

$$diff_i = \frac{C_{0,i}}{C_{0,i+1}}\sqrt{\frac{D_{i+1}\mu_i}{D_i\mu_{i+1}}}$$

for each local minimum, which can be viewed as a crude approximation of the ratio of the two adjacent peaks' heights. If $0.1 < diff_i < 10$, then the i-th peak and the $(i+1)$-th peak are assumed to be close enough in magnitude and the strategy relies on Equation (6) to estimate the corresponding i-th local minimum; otherwise, the strategy falls back on Equation (5) and emits the appropriate SMT2 formula. In either case, the magnitude of the i-th local minimum – i.e. $F(t_{min,i})$ – can be approximated as $F(\hat{t}_{min,i})$. And again, either of the two approximation approaches can drastically reduce synthesis time by significantly reducing – if not eliminating – the need for dReal to repeatedly evaluate $F'(t)$ and $F''(t)$.

Modelling electrophoretic separation with the approximate model rather than the precise model can enable the synthesis of microfluidic circuits that are designed to handle more than a trivial numbers of analytes. For the approximate model to actually be useful though, any solution that satisfies the approximate model should ideally also satisfy the precise model automatically; in other words, the approximations that have been described so far should be pessimistic. While it is difficult – if not impossible – to prove this for any arbitrary electropherogram, the remainder of this section will attempt to justify that the approximations are indeed pessimistic if we make some reasonable assumptions about the electropherograms that we can actually generate in a laboratory setting. To begin, we will provide justifications for some general properties of $F(t)$ that we can reasonably expect to hold. In particular,

> **Claim 3.1:** For all values of t_1 and t_2, $t_1 < t_2$, $F(t)$ must have at least one local maximum in the open interval (t_1, t_2) if there exists some $t_3 \in (t_1, t_2)$ that satisfies the constraints $F(t_1) < F(t_3)$ and $F(t_2) < F(t_3)$.
>
> **Proof:** By the mean value theorem, there exist some $a \in (t_1, t_3)$ and $b \in (t_3, t_2)$ for which $F'(a) > 0$ and $F'(b) < 0$. Now define a new function $f(t)$, which is equal to $F(t)$ in the interval $[a, b]$ and undefined elsewhere. We know that, by the extreme value theorem, $f(t)$ must have at least one global maximum. Since $f(t)$ is increasing at a and decreasing at b though, the global maximum clearly cannot be at either a or b. And since $f(t)$ is assumed to be differentiable at all points in the interval (a, b), the global maximum of $f(t)$ has to be at a stationary point, which implies that it must also be a local maximum. ■

Of course, a similar line of reasoning can be applied to justify that

> **Claim 3.2:** For all values of t_1 and t_2, $t_1 < t_2$, $F(t)$ must have at least one local minimum in the open interval (t_1, t_2) if there exists some $t_3 \in (t_1, t_2)$ that satisfies the constraints $F(t_1) > F(t_3)$ and $F(t_2) > F(t_3)$.

We can also show that

> **Claim 3.3:** For all i between 1 and $n - 1$ and all $t \in [t_{max,i}, t_{max,i+1}]$, $F(t) \geq F(t_{min,i})$.
>
> **Proof:** First assume that there exists some $a \in [t_{max,i}, t_{min,i})$ for which $F(a) < F(t_{min,i})$. Since $t_{min,i}$ is assumed to be a strict local minimum, there must also exist some $b \in (a, t_{min,i})$ for which $F(b) > F(t_{min,i})$. Then by Claim 3.1, $F(t)$ must have a local maximum in the interval $(a, t_{min,i})$, which contradicts the assumption that the local maxima at $t_{max,i}$ and $t_{max,i+1}$ are adjacent.
>
> By symmetry, there also cannot exist an $a \in (t_{min,i}, t_{max,i+1}]$ for which $F(a) < F(t_{min,i})$. ■

And again, a similar line of reasoning can be applied to show that

> **Claim 3.4:** For all i between 1 and n and $t \in [t_{min,i-1}, t_{min,i}]$, where $t_{min,0}$ and $t_{min,n}$ are defined to be 0 and some arbitrarily large value respectively in this context, $F(t) \leq F(t_{max,i})$.

We can now attempt to justify that if there exists a solution that satisfies the approximate model, then the corresponding design for the circuit should be capable of fully separating all analytes in the input sample. In particular, if we require any solution to the approximate model to satisfy the constraint $F(0) < F(t_{peak,1})$, then we can show that

> **Claim 3.5:** If the approximate model is satisfied, then $F(t)$ must have at least n local maxima.
>
> **Proof:** For all i between 2 and $n - 1$, we know from the model that $F(\hat{t}_{min,i-1}) < F(t_{peak,i})$ and $F(\hat{t}_{min,i}) < F(t_{peak,i})$. It then follows from Claim 3.1 that $F(t)$ must have a local maximum in the open interval $\widehat{(t}_{min,i-1}, \hat{t}_{min,i})$. Since we also know from the model that $F(\hat{t}_{min,1}) < F(t_{peak,1})$ and $F(0) < F(t_{peak,1})$, we can again apply Claim 3.1 to show that $F(t)$ must have at least one local maximum in the interval $[0, \hat{t}_{min,1}]$. Furthermore, since $F(t)$ approaches arbitrarily

close to 0 as t increases, we can infer from Claim 3.1 that there is at least one other local maximum in the interval $[\hat{t}_{min,n-1}, \infty)$. Altogether, $F(t)$ must have at least n local maxima. ■

This means if we accept that any electropherogram of interest can have at most n peaks, then the existence of a satisfying solution to the approximate model would imply that the corresponding electropherogram must have exactly n peaks, each of which must be "dominated" by a particular analyte as per Equation (3). In other words, the analytes in the input sample could be considered fully separated.

As pointed out previously though, in order for individual peaks to actually be discernable in practice, each peak's maximum must be sufficiently greater than its adjacent minima. To show that the approximate model does in fact guarantee this property, first observe that

Claim 3.6: For all i between 1 and $n-1$, there exists at most one value of j that satisfies the constraint $t_{max,i} < \hat{t}_{min,j} < t_{max,i+1}$.

Proof: Without loss of generality, assume there exist values of j_1 and j_2, $j_1 < j_2$, such that $t_{max,i} < \hat{t}_{min,j} < t_{max,i+1}$ is satisfied if $j \in \{j_1, j_2\}$. We know from the model that there exists some $t_{peak,k} \in (\hat{t}_{min,j_1}, \hat{t}_{min,j_2})$ for which $F(\hat{t}_{min,j_1}) < F(t_{peak,k})$ and $F(\hat{t}_{min,j_2}) < F(t_{peak,k})$. Then by Claim 3.1, we can deduce that $F(t)$ must have at least one local maximum in the interval $(\hat{t}_{min,j_1}, \hat{t}_{min,j_2})$. This would in turn imply that there exists a local maximum in the open interval $(t_{max,i}, t_{max,i+1})$, which contradicts the assumption that the local maxima at $t_{max,i}$ and $t_{max,i+1}$ are adjacent. ■

From this, we can deduce that

Claim 3.7: For all i between 1 and $n-1$, $t_{max,i} < \hat{t}_{min,i} < t_{max,i+1}$.

Proof: We know that $F'(t) > 0$ for all $t < t_{max,1}$, since $F_j(t) > 0$ for all j and $t < t_{max,1}$. Similarly, we know that $F'(t) < 0$ for all $t > t_{max,n}$, since $F_j(t) < 0$ for all j and $t > t_{max,n}$. However, we also know from the model that for every $\hat{t}_{min,i}$, there exist times before and after $\hat{t}_{min,i}$ at which the magnitudes of the electropherogram exceed $F(\hat{t}_{min,i})$. We can therefore conclude that there does not exist an i such that $\hat{t}_{min,i}$ is less than $t_{max,1}$ or greater than $t_{max,n}$.

Now first assume that there exists some i for which $\hat{t}_{min,i} < t_{max,i}$ holds true. We know from the model that there are $i-1$ distinct values of j for which $\hat{t}_{min,j} < \hat{t}_{min,i}$. We also know though that there can be at most $i-2$ distinct pairs $(t_{max,j}, t_{max,j+1})$ for which $t_{max,j+1} < \hat{t}_{min,i}$. Then by the pigeonhole principle, there must exist some j and k, $j \neq k$, such that both $\hat{t}_{min,j}$ and $\hat{t}_{min,k}$ fall in between the same pair of adjacent local maxima, which contradicts Claim 3.6.

By symmetry, there also cannot exist an i for which $\hat{t}_{min,i} > t_{max,i+1}$ holds true. ■

Following a similar line of reasoning as above, we can also justify that

Claim 3.8: For all i between 0 and $n-1$, there exists at most one value of j that satisfies the constraint $t_{min,i} < t_{peak,j} < t_{min,i+1}$, where $t_{min,0}$ and $t_{min,n}$ are defined to be 0 and some arbitrarily large value respectively in this context.

and make the following claim

Claim 3.9: For all i between 0 and $n-1$, $t_{min,i} < t_{peak,i} < t_{min,i+1}$.

Now taking into consideration both Claim 3.4 and Claim 3.9, it becomes clear that $F(t_{peak,i}) \leq F(t_{max,i})$ must hold true for all i between 1 and n. As well, taking into consideration both Claim 3.3 and Claim 3.7, it is clear that $F(\hat{t}_{min,i}) \geq F(t_{min,i})$ for all i between 1 and $n-1$. Knowing these two facts, it should be easily apparent that

$$\frac{F(\hat{t}_{min,i})}{F(t_{peak,i})} \leq c \Longrightarrow \frac{F(t_{min,i})}{F(t_{max,i})} \leq c$$

and

$$\frac{F(\hat{t}_{min,i})}{F(t_{peak,i+1})} \leq c \Longrightarrow \frac{F(t_{min,i})}{F(t_{max,i+1})} \leq c;$$

in other words, the approximations that the approximate model make are in fact pessimistic.

4. Constrained Model

Though the approximate model does scale much better (with respect to the number of analytes that the circuit is expected to handle) than the precise model, experimental results have shown that the approximate model still cannot be practically used by the Manifold compiler backend to synthesize circuits that deal with more than six or so analytes. This is not surprising, since computing approximate values for the magnitudes of the local extrema in the electropherogram output still requires the evaluation of n Gaussian curves for each individual extremum. Rather than attempting to actually estimate the values of the extrema then, it might make sense to instead compute reasonable bounds for the extrema and impose design constraints on those pessimistic bounds.

As with the approximate model, the times at which the electropherogram output reaches its local maxima and minima can be estimated with Equation (4) and Equations (5) or (6) respectively. Since $F(t)$ corresponds to the summation of all n individual peaks, $F(t_{peak,i})$ is obviously lower bounded by $F_i(t_{peak,i})$ for all i between 1 and n. In order to impose a reasonable upper bound on $F(\hat{t}_{min,i})$ though, first observe that since each individual peak is expected to monotonically increase prior to $t_{peak,i}$ and then monotonically decrease afterwards, we can actually define four additional points of interest – in addition to $t_{peak,i}$ – for each individual peak based on when it rises above or falls below certain thresholds. More concretely, we will define these points of interest – which we will label $t_{begin,i}$, $t_{focus,i}$, $t_{fade,i}$, and $t_{end,i}$ – by the mathematical relations

$$t_{begin,i} < t_{focus,i} < t_{peak,i} < t_{fade,i} < t_{end,i}$$

$$F_i(t_{begin,i}) = F_i(t_{end,i}) = \frac{1-q}{n-3} C_{negligible}$$

$$F_i(t_{focus,i}) = F_i(t_{fade,i}) = \frac{q}{2} C_{negligible}$$

where n again denotes the number of analytes in the input sample (assuming for simplicity that $n > 3$) and q is some arbitrary constant that must satisfy the condition

$$\frac{2}{n-1} < q < 1.$$

(Practically, q should be defined to be relatively large; based on results from preliminary experiments, $q = 0.9$ appears to be a reasonable choice in practice.)

Since each individual peak is expected to monotonically increase prior to $t_{peak,i}$ and monotonically decrease after $t_{peak,i}$, if we can guarantee that $t_{end,j} < \hat{t}_{min,i}$ holds for all $j < i$ and that $\hat{t}_{min,i} < t_{start,k}$ holds for all $i < k$ (both of which are in fact required of any solution that satisfies the constrained model), then we can actually establish

$$\begin{aligned} F(\hat{t}_{min,i}) &= \sum_{k=1}^{i-1} F_k(\hat{t}_{min,i}) + F_i(\hat{t}_{min,i}) + F_{i+1}(\hat{t}_{min,i}) + \sum_{k=i+2}^{n} F_k(\hat{t}_{min,i}) \\ &\leq \frac{(i-1)(1-q)}{n-3} C_{negligible} + F_i(\hat{t}_{min,i}) + F_{i+1}(\hat{t}_{min,i}) + \frac{(n-i-1)(1-q)}{n-3} C_{negligible} \\ &= F_i(\hat{t}_{min,i}) + F_{i+1}(\hat{t}_{min,i}) + \frac{(n-2)(1-q)}{n-3} C_{negligible} \end{aligned}$$

as a reasonable upper bound for $F(\hat{t}_{min,i})$. This means that in order to ensure the constraints expressed in Equations (1) and (2) are satisfied, it is sufficient to simply approximate the ratio $F(\hat{t}_{min,i})/F(t_{peak,j})$ as

$$\frac{F(\hat{t}_{min,i})}{F(t_{peak,j})} \approx \frac{F_i(\hat{t}_{min,i}) + F_{i+1}(\hat{t}_{min,i}) + \frac{(n-2)(1-q)}{n-3} C_{negligible}}{F_j(t_{peak,j})}.$$

for any i and j, rather than actually attempting to compute it precisely. This is very likely advantageous from the perspective of synthesis runtime, since the approximation essentially reduces the number of Gaussian functions that has to be evaluated for each peak from $2n$ to just 3.

If we further enforce that the constraints $t_{fade,i-1} < t_{peak,i} < t_{focus,i+1}$ and $t_{end,j} < t_{peak,i} < t_{begin,k}$ hold true for all $j < i - 1$ and $k > i + 1$ though, then we can additionally assert that

$$\begin{aligned} F(t_{peak,i}) &= \sum_{k=1}^{i-2} F_k(t_{peak,i}) + F_{i-1}(t_{peak,i}) + F_i(t_{peak,i}) + F_{i+1}(t_{peak,i}) + \sum_{k=i+2}^{n} F_k(t_{peak,i}) \\ &\leq \sum_{k=1}^{i-2} F_k(t_{peak,i}) + \frac{q}{2} C_{negligible} + F_i(t_{peak,i}) + \frac{q}{2} C_{negligible} + \sum_{k=i+2}^{n} F_k(t_{peak,i}) \\ &= \sum_{k=1}^{i-2} F_k(t_{peak,i}) + qC_{negligible} + F_i(t_{peak,i}) + \sum_{k=i+2}^{n} F_k(t_{peak,i}) \\ &\leq \frac{(i-2)(1-q)}{n-3} C_{negligible} + qC_{negligible} + F_i(t_{peak,i}) + \frac{(n-i-1)(1-q)}{n-3} C_{negligible} \\ &= (1-q)C_{negligible} + qC_{nelgigible} + F_i(t_{peak,i}) \\ &= F_i(t_{peak,i}) + C_{negligible}. \end{aligned}$$

Now observe that if we define $C_{negligible}$ in such a way that it is guaranteed to be less than $pF_i(t_{peak,i})$ (where p is some positive constant), then the relation

$$\frac{F_i(t_{peak,i})}{F(t_{peak,i})} \geq \frac{F_i(t_{peak,i})}{F_i(t_{peak,i}) + pF_i(t_{peak,i})} = \frac{1}{1+p}$$

would also be guaranteed to hold true. This is significant since as long as p is no greater than one, the ratio $1/(1+p)$ must be no less than 0.5; in other words, the i-th analyte could be considered the “dominant” analyte of the i-th peak in the electropherogram, which is what one would reasonably expect from a successful separation run.

Of course, the most straightforward method for ensuring that $C_{negligible}$ is no greater than $pF_i(t_{peak,i})$ for all i would be to simply define $C_{negligible}$ as

$$C_{negligible} = p \min_i F_i(t_{peak,i}).$$

The disadvantage with this approach though is that the value of $C_{negligible}$would have to be continuously recomputed by dReal during synthesis since it depends on what the value of $x_{detector}$ is. Rather than attempting to maximize $C_{neglgible}$ then, we can instead compute a reasonable lower bound by defining

$$C_{floor} = \frac{\min_{\mathrm{i}} C_{0,i}}{\sigma_0 + \sqrt{\frac{2 \max_i D_i \, x_{detector}}{v_n}}}$$

and setting $C_{negligible} = pC_{floor}$, which is guaranteed to be no greater than $p \min_i F_i(t_{peak,i})$.