
Joseph M. Pomo
University of New Mexico - Main Campus

Follow this and additional works at: https://digitalrepository.unm.edu/me_etds

Part of the Mechanical Engineering Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Mechanical Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.
Joseph M Pomo
Candidate
Mechanical Engineering
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Daniel Banuti, Chairperson

Dr. Peter Vorobieff

Dr. Patrick Wayne
EXPANDED-RANGE VENTURI FLOW METER:
Development and Testing of a 3D-Printed Flow Meter
and Solid-State Valve Concept

BY

Joseph M Pomo
Bachelor of Arts, Chemistry

THESIS
Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Mechanical Engineering

The University of New Mexico
Albuquerque, New Mexico

August 2022
Dedication

To Elizabeth and Owen
Acknowledgments

I owe great thanks to many individuals who have supported and walked alongside me as I worked through my Master’s degree and thesis research.

Thank you to Jon Allison at the AFRL for providing me this project and allowing me to work with his great team at the 'Thermal Lab'.

Thank you to Dr. Peter Vorobieff for his support as I began the my Master’s degree and for serving on my thesis committee.

Thank you to Dr. Patrick Wayne for both serving on my thesis committee and providing me feedback, equipment, and good-natured banter throughout my research at the AFRL.

Thank you to Dr. Daniel Banuti for agreeing to be my Master’s advisor and bearing with me as I struggled through research difficulties - and for encouraging his students to learn Python.

Thank you to my wife Lizzie and son Owen, for their continual encouragement and love.

This research was funded by AFRL Grant no. FA9453-19-2-0001
EXPANDED-RANGE VENTURI FLOW METER:
Development and Testing of a 3D-Printed Flow Meter
and Solid-State Valve Concept

by

Joseph M Pomo
B.A., Chemistry, University of New Mexico, 2015
M.S., Mechanical Engineering, University of New Mexico, 2022

ABSTRACT

In this thesis, I document the design, testing and performance of a novel Expanded-Range Venturi Flow Meter (ERVFM) concept. The ERVFM combines two, parallel-configured Venturi Flow Meters (VFM) with a Solid-State Selector Valve (SSSV). The goal was to create a reliable, high-turndown ratio flow meter which directs the fluid to either Venturi tube using the inherent pressure drop of the SSSV, and without moving parts. I constructed a pumped-fluid loop test-setup and quantified the ERVFM’s performance through pressure loss and mass flow rate measurements. In addition, I used python to develop models to assist in prototype design and data-processing. My results indicate that the SSSV does exhibit a switching phenomenon and further work is needed to make its performance more robust and definitively determine whether the system’s turndown ratio improves upon existing Venturi-based meters.
# Contents

## List of Figures

## List of Tables

1 **Introduction**

1.1 The ISS, OSAM-1, & the Need for Reliable Spacecraft Components ........................................ 1

1.2 Flow Meters ......................................................... 4

1.2.1 Survey of Flow Meters and Their Applications .......... 4

1.3 Flow Meter Applications ........................................ 8

1.4 Selecting a Flow Meter .......................................... 9

2 **Background**

2.1 Pipe Flow .......................................................... 18

2.1.1 Laminar & Turbulent Flow .................................. 19

2.1.2 Friction Factor & The Moody Chart ....................... 21

2.1.3 Major Losses .................................................... 22

2.1.4 Minor Losses .................................................... 28

2.1.5 Calculating Pressure Loss for Common Fluidic Features . . . . . 29

2.1.6 Pressure Loss Summary ....................................... 33

2.2 A Brief Summary of Passive Valve Designs ................. 35

2.2.1 Tesla Valve ..................................................... 35

2.2.2 Bubble-based Microvalve ..................................... 36

2.2.3 Jet-Diversion & Pressure-Driven Microvalves ............ 36
2.2.4 Quake Valves ........................................ 38
2.2.5 Coandă-Effect Microvalve ......................... 39
2.2.6 Geometry-Driven Oscillator ........................ 39
2.3 Solid-State Selector Valve .............................. 40
2.3.1 Proposed SSSV Mechanism .......................... 41

3 Methods ................................................. 45
3.1 Venturi Tube Pressure Loss Model ..................... 45
  3.1.1 Measured VFM $\Delta p$ and Flow Rate Calculations 46
  3.1.2 VFM Prototypes ................................... 46
  3.1.3 SSSV Pressure Loss Model ........................ 48
  3.1.4 Orifice Output ................................... 49
3.2 Test Setup & Data Collection .......................... 49
  3.2.1 Sensors ........................................... 51
3.3 Printed Fluidic Components ............................ 53
  3.3.1 Design and Printing Parameters ................... 54
  3.3.2 Print Post-Processing .............................. 55
  3.3.3 Integrating Printed Parts into the PFL ............ 57

4 Testing .................................................. 60
4.1 PFL Component Flow Rate Validation .................. 61
  4.1.1 KNF Diaphragm Pump ............................... 61
  4.1.2 Omega Turbine Flowmeter .......................... 61
  4.1.3 Flowrate Entering into the SSSV ................... 61
  4.1.4 Flowrate Exiting the SSSV (Entering the VFMs) .... 62
  4.1.5 Flowrate Entering the VFMs (DPT Measurements) .. 62
4.2 SSSV & VFM Prototype Evaluation ..................... 63
  4.2.1 SSSV Prototype Evaluation ........................ 63
  4.2.2 SSSV Prototypes .................................. 63
  4.2.3 VFM Prototype Evaluation ........................ 65
5 Results

5.1 Validation Testing

5.1.1 KNF Pump

5.1.2 Omega Turbine Flowmeter

5.1.3 Flow Rate Entering the SSSV

5.2 SSSV & VFM Performance

5.2.1 SSSV Prototype V0.54

5.2.2 SSSV Prototype V0.56

5.2.3 SSSV Prototype V0.57

5.2.4 Theoretical Turn-Down Ratio

5.2.5 SSSV V0.54 - Theoretical Turn Down Ratio

5.2.6 SSSV V0.57 - Theoretical Turn Down Ratio

5.3 Flow Visualization Using High-Speed Camera

6 Discussion

6.1 Validation Testing Results

6.2 SSSV and VFM Performance

6.3 Theoretical Turn-Down Ratio

6.4 Flow Visualization

7 Conclusion

Appendix

Python Models & Arduino Sketches for Sensors

A.1 Venturi Tube Pressure Loss Model

Python (Jupyter Notebook)

A.2 SSSV Pressure Loss Model (For SSSV V0.57)

Python (Jupyter Notebook)

A.3 DPT-0 Flow Rate Calculations (SSSV V0.57)

Python (Jupyter Notebook)
A.4  DPT-1 Flow Rate Calculations (SSSV V0.57)

    Python (Jupyter Notebook) ........................................... 121

A.5  Differential Pressure Transducer 0 (DPT-0) - Arduino Sketch ....... 126

A.6  Differential Pressure Transducer 1 (DPT-1) - Arduino Sketch ....... 127

A.7  Omega Turbine Flow Meter Pulse Counter - Arduino Sketch ......... 128

Bibliography ................................. 129
List of Figures

1.1 Astronauts repair a failed pump module on the ISS. . . . . . . . . . . . 2
1.2 Astronauts perform an EVA to replace a faulty flow control valve . . 3
1.3 Illustration of proposed OSAM-1 mission . . . . . . . . . . . . . . . . 3
1.4 Turbine flow meter . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.5 Turbine flow meter . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
1.6 Bradford Ultrasonic flow meter and Coriolis flow meter . . . . . . . . 6
1.7 Thermal flow meter . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
1.8 Types of differential-pressure flow meters . . . . . . . . . . . . . . . . 8
1.9 Drawing of a Venturi flow meter with pressure tap locations visible . 11
1.10 Free body diagram of a particle flowing along a streamline. . . . . . 12
1.11 Plot of relationship between flow rate and differential pressure for Ven-
turi flow meters. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
1.12 ERVFM concept drawing . . . . . . . . . . . . . . . . . . . . . . . . . 17
2.1 Difference between pipe flow (L) and open-channel flow (R) . . . . 18
2.2 Drawing of velocity profile for fully developed flow within a pipe. . . 19
2.3 Laminar and Turbulent Flow . . . . . . . . . . . . . . . . . . . . . . . 20
2.4 Comparison of smooth rough pipe walls and their effect on fluid flow 21
2.5 Moody chart demonstrating the relationship between the Reynolds
number $Re$, the relative pipe roughness $\frac{\epsilon}{D}$, and the friction factor, $f$. 22
2.6 Drawing of cylindrical flow element in laminar pipe flow. . . . . . . 23
2.7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.8 Drawing demonstrating the flow path through a globe valve. . . . . 29
2.9 Illustration of both flow separation and secondary flow within a bend. 31
2.10 Plot demonstrating the relationship between the bend radius-to-diameter ratio and the loss coefficient $K_b$. 31
2.11 Drawing showing the relationship between the orifice thickness and diameter ratio for a thick-edged orifice design. 32
2.12 Idealized pressure loss vs. flow rate plot demonstrating a hypothetical cross-over point. 34
2.13 Images from dye-streak experiment with Tesla valvular conduit. 36
2.14 Bubble-based microvalve. 37
2.15 Jet Diversion-type microvalve. 37
2.16 Differential pressure driven microvalve. 38
2.17 Anatomy of a Quake valve. 39
2.18 Coandă effect-driven microvalve. 40
2.19 Geometry-driven oscillating nozzle. 40
2.20 Drawing of solid-state selector valve concept. 41
2.21 Drawing of thick-edged orifice in SSSV with expected fluid interactions. 42
2.22 Drawing of serpentine path with expected flow direction within SSSV. 43
2.23 Illustration of the velocity differences within a bend which leads to secondary flow. 44
2.24 Idealized Pressure Loss vs. Flow Rate Plots for Darcy-Weisbach & Hagen-Poiseuille Frictional Loss Estimates. 44

3.1 Drawing of VFM designs tested in the ERVFM pumped-fluid loop. 47
3.2 Early VFM design indicating threaded-fittings and the undesirable expansions and contractions they create. 47
3.3 Images of early, printed Venturi prototypes and thread-cutting experiments. 48
3.4 Photo of pumped-fluid loop (PFL) test setup with labeled components. 50
3.5 Photos demonstrating the steps in the cup-and-stopwatch-method (CSM). 51
3.6 Omega Turbine flow meter installed in PFL. 53
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>Photo of PFL indicating the differential pressure transducer and Venturi flow meter connection points.</td>
</tr>
<tr>
<td>3.8</td>
<td>Screenshot of FormLabs Preform software.</td>
</tr>
<tr>
<td>3.9</td>
<td>Photo of supplies needed for creating custom silicone gaskets for the Venturi flow meters.</td>
</tr>
<tr>
<td>3.10</td>
<td>CAD and microscope images of VFM prototypes.</td>
</tr>
<tr>
<td>5.1</td>
<td>Validating the KNF pump flow rate.</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparing the flow rate output of the pump with the turbine flow meter</td>
</tr>
<tr>
<td>5.3</td>
<td>Volumetric flow rate entering the solid-state selector valve.</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparing the flow rate exiting the pump with the flow rate entering the solid-state selector valve prototypes.</td>
</tr>
<tr>
<td>5.5</td>
<td>Drawing of SSSV V0.54 with critical dimensions indicated.</td>
</tr>
<tr>
<td>5.6</td>
<td>Pressure loss estimation plots for SSSV V0.54.</td>
</tr>
<tr>
<td>5.7</td>
<td>DPT-measured flow rate measurements from SSSV V0.54 outputs.</td>
</tr>
<tr>
<td>5.8</td>
<td>CSM-measured flow rates from SSSV V0.54.</td>
</tr>
<tr>
<td>5.9</td>
<td>Drawing of SSSV V0.54 with critical dimensions indicated.</td>
</tr>
<tr>
<td>5.10</td>
<td>Pressure loss estimation plots for SSSV V0.56.</td>
</tr>
<tr>
<td>5.11</td>
<td>Cup and Stopwatch measured flow rates for SSSV V0.56.</td>
</tr>
<tr>
<td>5.14</td>
<td>VFM-measured plots for SSSV V0.57. This plot shows no cross-over point in contrast to the model estimations in 5.13.</td>
</tr>
<tr>
<td>5.12</td>
<td>Drawing of SSSV V0.57 with critical dimensions indicated.</td>
</tr>
<tr>
<td>5.13</td>
<td>Pressure loss estimation plots for SSSV V0.57.</td>
</tr>
<tr>
<td>5.15</td>
<td>Cup and Stopwatch-measured flow rates for SSSV V0.57.</td>
</tr>
<tr>
<td>5.17</td>
<td>Plot of DPT-measured flow rates and for SSSV V0.57. Like SSSV V0.54, the serpentine path has more fluid output throughout the test.</td>
</tr>
<tr>
<td>5.16</td>
<td>Plot of DPT-measured flow rates and for SSSV V0.54, and indicating the serpentine path has greater output throughout.</td>
</tr>
<tr>
<td>5.18</td>
<td>Translucency of SSSV V0.57 after wet-sanding coating with Krylon acrylic clear coat spray.</td>
</tr>
</tbody>
</table>
5.19 Prototype of Wayne injector coupler used to inject bubbles into the pumped-fluid loop. ........................................ 82
5.20 Still images of flow-visualization experiment. .................. 83

6.1 Comparison of metal and nylon ferrules in Swagelok fittings ...... 86
6.2 Examples of print artifacts in 3D-printed prototypes .................. 87
6.3 Detail of steel tube and printed-part interface sing RTV silicone. ... 88
6.4 Plot indicating the offset-correction implemented for the VFM data. 89
6.5 Comparison of CSM-measured and VFM-measured flow rates for SSSV V0.57. ....................................................... 90
6.6 Photo of Cup and Stopwatch-method measurement step. .......... 91
List of Tables

1.1 Qualitative review of common flow meters ........................................ 10
5.1 VFM-measured flow rates for SSSV V0.54 ........................................ 78
5.2 VFM-measured flow rates for SSSV V0.57 ........................................ 80
Chapter 1

Introduction

1.1 The ISS, OSAM-1, & the Need for Reliable Spacecraft Components

In July 2010, one of the International Space Station’s (ISS) ammonia pump modules went offline. The ammonia cooling loops, part of the Active Thermal Control System (ATCS), are responsible for heat-rejection through radiators on the station’s exterior. The ATCS ensures that the station’s interior remains habitable, and with only one functioning loop, the crew were forced to reduce their energy use by shutting down experiments for more than two weeks. After attempts to restart the pump failed, Astronauts Doug Wheelock and Tracy Caldwell-Dyson performed three EVAs over 25 hours to replace the pump module (Figure 1.1).

After returning the faulty pump module to earth on Space Shuttle Atlantis, an investigation determined that certain unexpected loading conditions caused the graphitic-carbon bearings to fail. The researchers concluded that although the centrifugal pump design had a rich space-flight heritage, new, more stringent tests are needed to qualify the pump for long-term use in a microgravity environment [35].

Only three years later, in 2013, the Flow Control Valve (FCV) within the new pump module failed to close, which again affected the efficiency of ammonia Loop A. The valve’s placement in the module made repairs impractical and astronauts Mike Hopkins and Rick Mastracchio replaced the pump module once more (figure 1.2) [30][15]. Space-qualification and testing procedures are rigorous, but eventual
component failure is always a possibility. That is why the ISS stored two pump module replacements on-board; critical systems require it. The ability to service aging or malfunctioned space vehicles, like satellites, has long been a goal of groups wishing to extend mission timelines and reduce costs. OSAM-1 (formerly "RestoreL") is a NASA mission focused on autonomous, telerobotic refueling, manufacturing and relocation. It’s first objective is an autonomous refueling procedure on the low-earth orbiting (LEO) satellite "LandSat 7", which far out-lived its five year mission and was finally decommissioned in 2021. LandSat 7, used hydrazine mono-propellant for on-orbit maneuvers and station keeping. Without fuel for attitude control, the satellite’s is left to slowly fall out of its intended orbit, until it eventually reenters earth’s atmosphere. (figure 1.3) [29] [28]
Figure 1.2: Expedition 38 astronauts Mike Hopkins and Rick Mastracchio perform an EVA to replace the pump module in ammonia Loop A. (L) A crew-member stands on the robotic Canadarm while holding the new pump module. The failed module is visible attached to another Canadarm. (R) The crew member positions in the new pump module. Photos: NASA [34]

Figure 1.3: Artist depiction of OSAM-1 (L) gripping onto a candidate satellite for servicing (R).[29]

OSAM-1 employs a novel "Propellant Transfer Technology" (PTT) which describes a suite of sub-assemblies necessary for the refueling process. One of these assemblies is a turbine flow meter which serves to monitor the fuel transfer 1.4. Such a meter must be well tested, reliable and accurate, especially as the mission is unmanned and repairs are out of the question. This provides an interesting case study
for choosing reliable components, and leads into the subject of this thesis: designing and testing an Expanded-Range Venturi Flow Meter (ERVFM).

Figure 1.4: Rendering of Hoffer turbine flowmeter under consideration for OSAM-1 (Restore-L) NASA mission[7]

1.2 Flow Meters

Flow meters are utilized in a diverse set of industries where fluid-flow measurement is important. Because of their wide adoption, many types of flow meters exist, and while they share the same purpose of measuring mass or volumetric flow rates, they can differ greatly in the measurement technique. The following sections gives several examples of flow meter types, but is not an exhaustive list. Also, among the other characteristics mentioned, I’ve included a turndown ratio for each type. Turndown describes the range over which a meter can accurately measure flow rates. For example, a meter which measures a minimum value of $0.1 \frac{l}{min}$ and a maximum value of $5 \frac{l}{min}$ has a 50:1 turndown. Such measurements are a factor of both the construction of the flow meter and the accuracy of its sensor.

1.2.1 Survey of Flow Meters and Their Applications

Common flowmeters can be grouped into several categories:

1. Mechanical
2. Ultrasonic & Coriolis
3. Thermal
Mechanical Flow Meters

Mechanical flow meters, sometimes called Positive-Displacement meters, usually involve the fluid doing work on some element within the meter. One example is turbine meters, which contain a turbine that spins with the flow at a speed proportional to the flow rate (figure 1.5). The turbine is commonly made of a magnetic alloy which is detected by a sensor (such as a hall-effect pickup) as it spins. Turbine meters are made in a variety of sizes, including pipe diameters greater than 10 inches (254mm), or as small as $\frac{1}{4}$ in ($\approx 6$mm) (such as the meter used later in this study). Turbine meters are best suited for clean fluids, free of debris which may accumulate and prevent the motion of the turbine. Filters can be added in the flow path to prevent this, however, motion components of any kind are susceptible to failure, especially at higher flow rates and with corrosive working fluids. Turbine meters usually have a turndown ratio of 1:10 to 1:20. Other mechanical meters are gear or diaphragm-based, which can have turndown ratios as high as 1:80 [40].

Ultrasonic & Coriolis Flow Meters

Ultrasonic flow meters use transducers inline or on the exterior of a pipe to measure flow rate. The transducers create ultrasonic pulses which pass through the working fluid and are received by the downstream transducer. Pulses can be sent in both
directions between the sensors and the “transit-time” of the pulses is proportional to
the flow rate. Some transducers have the benefit of being completely external to the
fluid. This is an important factor when working fluids are corrosive or flow rates are
high. Turndown ratios can be as high as 1:250 [24]. Coriolis flow meters typically
direct flow through a u-shaped channel and, as the flow rate increases, the fluid’s
inertia causes the channel to oscillate in a predictable manner. Sensors on either side
of the channel can measure its deflection and ascertain the mass flow rate directly
using the Coriolis principle. Like ultrasonic flow meters, Coriolis meters also may
have their sensors located external to the flow which confers the same benefits and
enable measurements even with debris-laden or heterogeneous fluids. These meters
may have a turndown ratio of at least 1:100 (figure 1.6).

Figure 1.6: Bradford Ultrasonic flow meter (L) and Coriolis flow meter, with its
characteristic, u-shape (R) [11][20]

Thermal Flow Meters

Inline thermal flow meters use probes inserted into (or external to) the flow. One probe
contains a heating element while the other contains a temperature sensor (figure 1.7).
The meter relies on measuring the heat convection through the working fluid (whether
liquid or gas) to measure the mass flow rate. The ’mass air flow’ meters in internal
combustion engines use this principle to adjust the air fuel mixture. Industrial thermal
flow meters claim turndown ratios of 1:500 or greater [21].
Differential-Pressure Flow Meters

Examples of differential-pressure flow meters include orifice and Venturi tubes. Orifice-plate flow meters create an obstruction in the flow path, usually in the form of a thin diaphragm. The diaphragm effectively reduces the cross-sectional area of the flow for a short time, creating a steep pressure difference upstream and downstream of the device. The static pressure on both sides is measured and the resulting difference is used to calculate the flow rate. In a similar way, Venturi tubes accept the flow and feed it into a region of smaller cross-section, called the throat. By contrast, Venturi tubes make this transition much more gradual by collimating the flow in a cone-shaped region. The throat section is relatively short and is followed by a diverging cone which allows the flow to expand to its original flow area. Pressure measurement locations, called ‘taps’ are located just before the flow converges and at the throat section. This pressure difference, along with the fluid density, and the known geometry of the Venturi tube, is used to calculate the flow rate entering the tube (see [9], chapters 13 & 14).
1.3 Flow Meter Applications

Flow meters are deployed in numerous applications. In the oil and gas industry, flow meters are used extensively in the drilling, transport, and sales-transaction process. ‘Custody Transfer’ describes the transfer of product from seller to customer, and on both sides of the transaction, accuracy is critical. Taking crude oil as an example, an error of only 0.03% in exported oil volume equates to millions of dollars lost *per day* [1]. This error becomes more important as energy prices increase. Accurate flow metering is essential for other applications, such as in spacecraft. Two common examples of this are Active Thermal Control Systems (ATCS), and fuel-metering for on-orbit propulsion systems. ATCSs are responsible for collection, transport and rejection of heat on spacecraft. As mentioned before, the International Space System (ISS), uses two large, pumped fluid loops (PFLs) to ensure on-board electronics, equipment, and crew environments remain at optimal operating temperatures. In this application, accurate flow meters are needed for feedback to the pump speed control to maintain the heat rejection balance [35]. In addition to PFLs, satellites may require flow meters to measure fuel dispensed to thrusters in their reaction control systems (RCS). Once in orbit, satellites use these systems for attitude control and station keeping (i.e., keeping the satellite oriented and positioned correctly on orbit).
RCSs may also be used for decommissioning operations in which a malfunctioning or end-of-life spacecraft is commanded to move into a graveyard orbit [16]. Flow meters can be a useful tool in these cases for tracking how much propellant or fuel has been used, which can inform a ground crew about end-of-life estimations.

1.4 Selecting a Flow Meter

Many factors should be considered when selecting a flow meter, including: the expected flow rates, compatibility with the working fluid, uniformity of the fluid’s density, size of the meter, permanent pressure loss associated with the meter, and the meter’s accuracy. As mentioned before, a turbine flow meter may work well for intermediate flow rates with a clean, non-corrosive working fluid, but it would be a poor choice for wastewater, where debris could interfere with its rotation.

Use in spacecraft adds several more factors to consider. In 1999, the European Space Agency (ESA) conducted a feasibility study for using commercial flow meters in spacecraft and found that two of the meters, both turbine-based, required complete redesigns to fulfill their testing requirement. Ultimately, they found the accuracy of each to be limited [8]. The ESA provides guidelines for pump selection in ATCs, and while they are not specifically related to flow meters, they are still applicable for most fluidic components (see reference [14], Chapter 12, pgs. 418-419):

1. High Efficiency
2. Low Mass
3. Low mass-to-output-power ratio
4. Hermetically-sealed structure
5. Minimum operational noise
6. Ability to withstand mission vibration and shock loads
7. Compatibility with onboard electrical system
8. Applicability to aerospace-environment usage
9. Ability to handle typical coolants as working fluids

10. High operational reliability

When considering flow meters, it is useful to add accuracy, repeatability and calibration to this list.

Reliability is one of the most important features of critical components, especially when the components are difficult to maintain or replace. Keeping in mind the guidelines from before, one way to improve reliability is to reduce complexity and part count, when possible. Turbine flow meters carry the advantage of having wide flow rate ranges, however the turbines themselves usually require bearings for the turbine rotor, and if the turbine rotation is sensed magnetically, the turbine itself is likely made of a ferrous alloy, perhaps raising fluid compatibility concerns. The Netherlands-based company, Bradford, offers a space-qualified ultrasonic flow meter. By comparison, this flow meter has no moving parts, and still offers high accuracy and a long fluid-compatibility list. This flow meter is a great option for higher reliability, however, it also has a large footprint (36”/ 900mm measurement tube) and mass (> 1kg) limiting its use in smaller spacecraft. Table 1.1 provides a qualitative scoring of the discussed criteria. Another option, and the one selected for this study, is a Venturi flow meter (VFM). The design comprises a set of trumpet-shaped cones which converge to a smaller ‘throat’ section and then diverge back to the original

<table>
<thead>
<tr>
<th>Fluid/Phase Compatibility</th>
<th>Turbine</th>
<th>Coriolis</th>
<th>Ultrasonic</th>
<th>Thermal</th>
<th>Orifice</th>
<th>Venturi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Moving Parts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Footprint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turndown Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Irrecoverable Pressure Loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Qualitative review of common flow meters. When comparing characteristics, green cells indicate a desirable score, red cells indicate an undesirable score, and the score for yellow cells fall somewhere in between.
pipe diameter. The fluid’s static pressure is measured before the converging section and at the throat to determine the differential pressure (figure 1.9). The fundamental principle of the VFM’s function is the Venturi effect which states that as the flow rate (a representation of the kinetic energy) of the flow increases, the pressure energy in the flow must decrease in order to satisfy the law of Conservation of Energy. For the VFM specifically, this means that as the flow converges into the throat section, the flow rate in the throat must necessarily increase, such that the pressure decreases, and the energy balance is maintained (see pg. 119 in [27]). This concept is actually an application of the Bernoulli equation, which considers the forces on a fluid particle flowing along a streamline.

**The Bernoulli Equation**

The Bernoulli equation necessarily makes several assumptions about the flow which directly effect the forces acting on a fluid particle:

1. The fluid is inviscid (that is, its viscosity is negligible)
2. The fluid is incompressible (density is constant)
3. The flow is steady (flow velocity is constant)
4. A particle in the flow follows a distinct streamline and does not cross into other streamlines
If we consider a particle with thickness $\delta y$, length $\delta s$, width $\delta n$, and mass $\delta m$ (figure 1.10), we can make some assumptions about the forces which act on it while flowing along a streamline, including pressure forces on all sides as well as the force of gravity. The particle has no shear forces acting on it because the fluid is assumed to be inviscid.

![Free body diagram of a particle flowing along a streamline with the force of gravity and various pressure forces acting on it. Note that the shear forces equal zero because the flow is assumed inviscid.](image)

Figure 1.10: Free body diagram of a particle flowing along a streamline with the force of gravity and various pressure forces acting on it. Note that the shear forces equal zero because the flow is assumed inviscid [27].

We can apply Newton’s second law ($F = ma$) to the information in the free body diagram (figure 1.10) and write the sum of the forces in the $s$ (streamline) direction as

$$
\sum \delta F_s = \delta m \ a_s = \delta m \ V \frac{\partial V}{\partial s} = \rho \ \delta V \ V \frac{\partial V}{\partial s}
$$

eqn. 1

with $V$ as the velocity, $\frac{\partial V}{\partial s}$ as the acceleration, $\rho$ as the density and $V$ as the particle volume ($\delta n \ \delta s \ \delta y$). The gravity force on the particle in the $s$ direction is

$$
\delta W_s = -\delta W \sin \theta = -\gamma \delta V \sin \theta
$$
where $\theta$ is the angle from the horizontal center line of the flow and $\gamma$ is the specific weight of the fluid particle ($\gamma = \rho g$, $g$ being the acceleration due to gravity. See pg. 96-97 in [27]). Note that if $\theta=0$ (the streamline is horizontal), then there is no acceleration due to gravity in that particular direction. Since gravity acts on the fluid, a pressure gradient exists throughout the flow and will vary with position $s$ and $n$. The pressure field in the $s$ direction can then be represented by

$$\delta p_s = \frac{\partial p}{\partial s} \frac{\delta s}{2}$$

If we let the net pressure force on the particle in the streamline direction be $\delta F_{ps}$ it is then defined as

$$\delta F_{ps} = (p - \delta p_s) \delta n \delta y - (p + \delta p_s) \delta n \delta y = -2 \delta p_s \delta n \delta y$$

$$= -\frac{\partial p}{\partial s} \delta s \delta n \delta y = -\frac{\partial p}{\partial s} \delta V$$

and the net force acting on the particle in the streamline direction is

$$\sum \delta F_s = \delta W_s + \delta F_{ps} = \left(-\gamma \sin \theta - \frac{\partial p}{\partial s}\right) \delta V$$

eqn. 2

We then combine equations 1 & 2 from above to get the following ‘equation of motion’ along the streamline (note that $\delta V$ has been factored out)

$$-\gamma \sin \theta - \frac{\partial p}{\partial s} = \rho V \frac{\partial V}{\partial s} = \rho a_s$$

eqn. 3

Equation 3 can then be manipulated and integrated provided we make the following substitutions: $\sin(\theta) = \frac{ds}{ds}$, $V \frac{dV}{ds} = \frac{1}{2} \frac{d(v^2)}{ds}$ and also recognize that the the value of $n$ along the streamline does not change (as the particle
rides it like a rail) and the change in pressure \( dp = \frac{dp}{ds} ds \) (only important along the streamline). This simplifies the pressure gradient function such that \( p(s, n) \) becomes a function of \( p(s) \) only, and \( \frac{dp}{ds} = \frac{dp}{ds} \). The result, now only valid along a streamline, is

\[
-\gamma \frac{dz}{ds} - \frac{dp}{ds} = \frac{1}{2} \rho \frac{d(V^2)}{ds}
\]

which further simplifies to

\[
dp + \frac{1}{2} \rho d(V^2) + \gamma dz = 0 \quad \text{(along a streamline)}
\]

This terms of this resulting equation can be integrated with the assumption that the fluid’s density is constant, yielding the Bernoulli equation:

\[
p + \frac{1}{2} \rho V^2 + \gamma z = \text{constant along a streamline}
\]

(1.1)

Nozzle-shaped devices, like the Venturi tube, obey the principle of Conservation of Mass. In the context of incompressible fluid flows, this law states that for a fixed volume or geometry, the rate of flow into the fixed volume must equal the rate of flow out of the volume (and no mass is created or destroyed in the process) (see chapter 3 in [27]). Assuming, the fluid density is constant (as is the case for the Bernoulli equation), the Continuity Equation of Incompressible Flow can represent this phenomenon as

\[
A_1 V_1 = A_2 V_2, \text{ or } Q_1 = Q_2
\]

With \( A_1 \& A_2 \) being the cross-sectional area of the flow at two distinct points and \( V_1 \& V_2 \), the fluid velocity. Returning to the Venturi effect mentioned before, for steady, incompressible flow, the flow rate entering the small cross-section of the throat must remain constant, and this means the fluid velocity must increase, to compensate. If we assume some flow is horizontal (\( z \) is constant), the Bernoulli equation becomes
Knowing that the velocity along a streamline is constant (according to the Bernoulli equation), we can rearrange this equation to solve for the velocity, and then substitute this into the continuity equation to find the measured flow rate entering the VFM:

\[ p_1 + \frac{1}{2}\rho v_1^2 = p_2 + \frac{1}{2}\rho v_2^2 \]

With \( p_1 \) as the static pressure measured at prior to the converging section, \( p_2 \) the static pressure measured at the throat, \( A_1 \) the cross-sectional area of the VFM inlet, \( A_2 \) the cross-sectional area of the throat, and the density, \( \rho \). It should be noted that although this calculation for the flow rate is suitable, it carries along with it all of the necessary assumptions of the Bernoulli equation from before. As a result, flow rates measured on a test setup will differ from calculated estimates because of real-world factors. This is why flow meter calibration and validation via some other method (separate, calibrated flow meter) is highly recommended.

Still, the VFM appears to be a good choice for applications requiring high-reliability as its design reduces part count, the internal geometry can be fabricated out of only one or two sub-components, and the same material can be maintained throughout its construction. When compared to orifice plate meters, the irrecoverable pressure loss is lower, and is often minimized in the design stage to mitigate its impact on performance. Finally, depending on the diameter of the inlet pipe, the actual measurement region between pressure taps on a VFM can be as small as a few inches, making its overall footprint smaller, compared to other meters. These benefits, combined with VFMs having an extensive history of research, made the VFM a good potential candidate for this study. However, compared to other meters, VFMs have lower turndown ratios; typically around 1:10 [22][18]. This reduced ‘range’ is due to the pressure loss mechanism of the Venturi, which, when plotted against the flow rate, has a horizontal parabola shape (figure 1.11). This behavior presents an issue for measuring both low
and high flow rates. For low flow rates ('low' being a relative term in this context), the Venturi tube may produce such a small differential pressure ($\Delta p$) at the pressure taps that the measurement is unreliable. By contrast, too high a flow rate may produce $\Delta ps$ which exceed the range of the attached sensor. Pairing the VFM with a wider-range sensor may work in some cases, but this runs the risk of reducing accuracy of measurements, especially if the sensed $\Delta ps$ are taken near the upper or lower limits of the sensor. Still, the flow rate's $\Delta p$ square-root dependency requires that potentially very large pressure differentials are needed for a given flow rate; which may not be possible given the constraints on other components in an application, such as the max speed of the pump.

Figure 1.11: Plot demonstrating that for a VFM, the flow rate $Q$ is proportional to the square-root of the pressure differential $\Delta p$. This creates a horizontal parabolic-shape and indicates that to ascertain higher flow rates, larger pressure differentials are necessary. Source: Pg. 119 in [27], and modified by JMP.

Another possibility is to use two VFMs in a parallel-arrangement: one VFM which can measure 'low' flow rates and the other, 'high' flow rates. Both VFMs have the same inlet diameter, but different throat diameters; one is relatively larger than the other (figure 1.12). Upstream of the VFMs, it would only be necessary to divert the flow to the appropriate path depending on the flow rate. If the VFMs are carefully designed, this should allow each to accurately measure $\Delta ps$ across an expanded range and increase the turndown ratio when compared to one VFM alone.

This concept is the basis for the Expanded-Range Venturi Flow Meter (ERVFM) design which seeks to enhance reliability by decreasing part count while overcoming
the inherent range limitation of VFMs. Provided that the parallel-configured VFMs perform as expected, it is still necessary to divert the flow to either one at the appropriate time. Adding a valve with a commanded flap would be trivial, however, in the spirit of minimizing part-count, Jon Allison at the Air Force Research Labs (AFRL, Kirtland AFB, NM) proposed a Solid-State Selector Valve (SSSV). The SSSV is a fluidic device which would exist upstream of the two (parallel-configured) VFMs and would feed either one via a dedicated output. Without a commanded-flap, the SSSV would divert the flow by carefully-designed internal geometry of each output. The valve design is critical to the ERVFM functioning properly. Because of this a careful investigation of pipe flow and pressure loss mechanisms was necessary, and is the subject of the next chapter.

Figure 1.12: ERVFM concept drawing showing, from left to right, an inlet pipe, an SSSV prototype design, and two ball valves each connected to a VFM
Chapter 2

Background

Preface

As mentioned in the Introduction, the SSSV is proposed to divert a flowing fluid to one output or the other simply by utilizing its designed-in geometry. This chapter explores fundamental pipe flow characteristics, the theory behind the SSSV mechanism, and also provides a brief summary of other passive valve designs.

2.1 Pipe Flow

Pipe flow specifically refers to flow within a fully-filled conduit. By comparison, open-channel flow, in which a gas (e.g. air) fills a portion of the pipe, also occurs within pipes, but is not the type of flow under consideration here. Whereas in open-channel flow, the flow rate is primarily driven by gravity (aqueducts), pipe flow is driven by a pressure gradient in the pipe [figure 2.1 Also see Chapter 8 in [27]].

Figure 2.1: Drawing demonstrating the difference between pipe flow (L) and open-channel flow (R). In the first case, the flow is driven primarily by a pressure gradient in the pipe. In the second case, the flow is driven primarily by gravity [27].

The type of pipe flow in this thesis can be further narrowed down to \textit{viscous}, leading to important interactions at the pipe wall, and \textit{incompressible}, in this case using water
as the working fluid. As the water enters a pipe, the region of the flow closest to the pipe wall slows down due to friction, with the fluid’s velocity reaching zero at the interface (called the no-slip condition). This leads to shear stresses propagating through the fluid and eventual creates a stable velocity profile. For laminar flow, the profile shape is characteristically parabolic with the region closest to the pipe wall having a velocity (effectively) equal to zero and the vertex having little to no shear stresses and leading to the highest fluid velocity at that point (figure 2.2).

Figure 2.2: Drawing of velocity profile for fully developed flow within a pipe. The fluid passes from a larger reservoir into an entrance and subsequently undergoes a shear stress gradient with the highest stress (and lowest velocity) at the pipe wall and the lowest stresses at the center of the flow (highest velocity). These physical effects within the flow create a parabola-shaped velocity profile.

2.1.1 Laminar & Turbulent Flow

The fluid’s flow behavior can be represented by the 'dimensionless' Reynolds number, $Re$, which is defined as the ratio between the inertial and viscous effects in the flow (equation 2.1).

$$ Re = \frac{\rho V D}{\mu} \quad (2.1) $$

$\rho$ is the fluid’s density, $V$ is the mean velocity for the flow (for laminar flow, this is the largest velocity divided by two), $D$ is a ‘characteristic linear dimension’ for the flow path (usually the pipe diameter for pipe flow) and $\mu$ is the fluid’s dynamic viscosity.

At Reynolds numbers below 2100, the flow is considered Laminar and steady with little or no perturbations in the flow and with viscous effects being dominant. For
Reynolds numbers above $\approx 4000$, the flow is considered turbulent with chaotic and seemingly spontaneous formation of vortices within the flow. Turbulent flow is dominated by inertial effects and significant shear stresses throughout the cross section of the flow. It is thought that the vortices degrade the uniform boundary layer formed in the laminar flow to the extent that the pipe roughness becomes an important consideration. Between a Reynolds number of 2100 and 4000 (not hard limits), transitional flow occurs and displays both steady, laminar-like flow and intermittent vortex formation which typically resolves quickly (see chapter 8 in [27]).

![Figure 2.3: Comparison of Laminar and Turbulent flow regimes.](image)

When considering a straight section of pipe, turbulent flow (higher $Re$) is highly dependent on the surface roughness of the pipe, $\epsilon$. In laminar flow, the degree of roughness of the pipe wall certainly contributes drag, but it is typically superseded by the thickness of the viscous boundary layer, leaving the rest of the flow mostly unaffected. With a much thinner boundary layer in turbulent flow, the pipe’s roughness is understood to induce eddies leading to the chaotic flow described before (figure 2.4). It is possible to create regions of turbulence with Reynolds lower than $Re = 4000$ by using a pipe with significant roughness.
Figure 2.4: Comparison of rough (L) and smooth (R) pipe walls and their effect on the flow. Rough regions are understood to ‘snag’ the flow and produce eddies. However, laminar flow may still be possible if the flow rate is ‘low enough’ and the viscous boundary layer is ‘thick enough’ to mask those rough regions. Source [27].

2.1.2 Friction Factor & The Moody Chart

The Moody chart (figure 2.5) demonstrates the relationship between $Re$, the relative pipe roughness $\frac{\epsilon}{D}$, which is the ratio of the absolute pipe roughness $\epsilon$ and the internal pipe diameter $D$, and an additional term called the friction factor $f$. The friction factor $f$ is a dimensionless quantity which is used to quantify the viscous (frictional) losses in a flow. In laminar flow, $f = \frac{64}{Re}$ and is independent of the pipe wall’s relative roughness (Plots on left-side of figure 2.5). For turbulent flow there are a number of correlations available. In this thesis I used the Churchill ‘all-regime’ equation (equation 2.4). On the right-hand side of figure 2.5, at high Reynolds numbers, the plots become nearly horizontal as the friction factor becomes more dependent on the relative roughness and less on $Re$ (See section 8.4.1 in [27]).

In laminar pipe flow, shear stresses within the flow lead to fluid particles losing kinetic energy and create pressure losses within the flow. This is also the case of turbulent flow where it is generally accepted that formation and movement of eddies through the flow create further shear stresses. In both cases, the pressure loss will increase as flow rate increases. However, because laminar and turbulent flow are distinct for the reasons discussed, their loss mechanisms are also distinct, and require
Figure 2.5: Moody chart demonstrating the relationship between the Reynolds number $Re$, the relative pipe roughness $\frac{\epsilon}{D}$, and the friction factor, $f$. For laminar flow, $f = \frac{64}{Re}$ and is independent of the relative roughness. In turbulent flow, $f$ can be calculated using the implicit Colebrook equation, or by numerous other correlations (such as the 'Churchill All-Regime Equation'). For larger $Re$’s the friction factor is almost entirely dependent on the relative roughness. Source: [27].

different treatment when calculating the pressure loss.

2.1.3 Major Losses

Pressure losses due to friction are called Major losses. For laminar flow, once the flow is fully developed and a stable velocity profile has formed, it is assumed that there is a force balance between the pressure differential (driving the flow through the pipe) and the viscous forces resisting this flow. We can then consider a cylindrical-shaped flow element in static equilibrium (acceleration = 0) with some length $l$ and some radius $r$. We can apply Newton’s second law and find the element is driven through the pipe by a pressure force (created by the pressure differential), and is resisted by pressure and shear forces acting in the opposite direction (figure 2.6).

Assuming that the flow in question is fully developed, laminar, and occurs within
Figure 2.6: (T) Drawing of a cylindrical flow element in static equilibrium (acceleration = 0). (B, free-body diagram) The element is driven to the right by the positive pressure differential and encounters resistance from the pressure difference at the head of the cylinder as well as shear forces.

A horizontal, circular pie, the relationship between pressure loss ($\Delta p$) and flow rate ($Q$) can be derived (also see pg. 390-393 in [27]):

The force balance is written as

$$\left(p_1\right)\pi r^2 - (p_1 - \Delta p)\pi r^2 - (\tau)2\pi r \ell = 0$$

and simplified to

$$\frac{\Delta p}{\ell} = \frac{2\tau}{r}$$

where $l$ is the length of the flow element, $r$ is the element radius, $\Delta p$ is the pressure loss and $\tau$ is the shear stress on the element. The shear stress itself is defined as linear function of $r$

$$\tau = \frac{2\tau_w r}{D}$$

where $\tau_w$ is the maximum shear stress at the pipe wall (no-slip condition).

The pressure loss is then related to $\tau_w$ by
The shear stress distribution across the pipe radius leads to the previously mentioned velocity profile (i.e. a velocity gradient). The shear stress, then is a function of the velocity gradient

\[ \tau = -\mu \frac{du}{dr} \]

with \( \mu \) being the fluid’s dynamic viscosity. The term is negative because the velocity is expected to decrease as the value of the radius increases from the center-line of the flow. Using a direct substitution of the shear stress from before, we find that the velocity gradient can be represented as

\[ \frac{du}{dr} = -\left( \frac{\Delta p}{2\mu \ell} \right) r \]

this is then integrated to yield the velocity profile equation \( u \)

\[ u = -\left( \frac{\Delta p}{4\mu \ell} \right) r^2 + C_1 \]

If we simplify the flow as having distinct velocities in certain areas, we can represent these regions by concentric rings, propagating from the center line with some radius \( r \) and thickness \( dr \). We obtain an expression for the volumetric flow rate by integrating the velocity equation across the radius with respect to the radius and obtain

\[ Q = \int_{r=0}^{r=R} u(r) 2\pi r \, dr = 2\pi V_c \int_{0}^{R} \left[ 1 - \left( \frac{r}{R} \right)^2 \right] r \, dr \]

\[ Q = \frac{\pi R^2 V_c}{2} \]
Here, $V_c$ represents the maximum fluid velocity (at the center-line of the flow) and $R$, the radius. The average velocity, $V$, is the flow rate divided by the cross-sectional area; $V = \frac{Q}{\pi R^2}$, and because the velocity profile is parabolic the average velocity is $\frac{V_c}{2}$. Through some manipulation and substitution, we have that

$$V = \frac{\pi R^2 V_c}{2 \pi R^2} = \frac{V_c}{2} = \frac{\Delta p D^2}{32 \mu \ell}$$

and

$$\Delta p = \frac{8 \mu L Q}{\pi \ell^4} \quad (2.2)$$

Equation 2.2 is commonly referred to as the Hagen-Poiseuille (HP) pressure loss equation. HP gives a direct measure of the pressure loss ($\Delta p$) due to the the viscous shear forces within the flow, and shows that the loss is directly proportional to both the length of the flow path, $L$ and the flow rate, $Q$. HP is only applicable for fully-developed laminar flow in horizontal, circular pipes, which means a different method for calculating pressure loss is required for turbulent flow and other types of fluidic components, such as bends and orifices.

The relationship between $\Delta p$ and $Q$ can also be derived via dimensional analysis (also see pg. 396-397 in [27]):

Here, $\Delta p$ is a function of the average velocity $V$, the length of the flow path $L$, the pipe diameter $D$, and the kinematic viscosity, $\mu$.

$$\Delta p = F(V, \ell, D, \mu)$$

This can then be described by 2 dimensionless groups as

$$\frac{D \Delta p}{\mu V} = \phi\left(\frac{\ell}{D}\right)$$
in which $\phi(L/D)$ is some function of $L$ and $D$. Using what we know from the Hagen-Poiseuille derivation, we can assume that the pressure loss is proportional to the length of flow

$$\frac{D}{\mu V} \frac{\Delta p}{C \ell} = \frac{C \ell}{D}$$

where $C$ is a constant. This can be further rewritten as

$$Q = AV = \frac{(\pi/4C) \Delta p D^4}{\mu \ell}$$

The value of the constant $C$ is 32 for a round pipe [27], and differs for conduits of other shapes. The above equation can be manipulated further through division by the dynamic pressure ($\frac{1}{2} \rho V^2$) and substitution of the equation for $\Delta p$ for horizontal, laminar pipe flow $\Delta p = 32 \mu L \frac{V}{D^2}$ to obtain

$$\frac{\Delta p}{\frac{1}{2} \rho V^2} = \frac{(32 \mu \ell V/D^2)}{\frac{1}{2} \rho V^2} = 64 \left( \frac{\mu}{\rho V D} \right) \left( \frac{\ell}{D} \right) = \frac{64}{Re} \left( \frac{\ell}{D} \right)$$

which is commonly stated in this form as the Darcy-Weisbach equation:

$$\Delta p = f_d \rho V^2$$

(2.3)

The Darcy friction factor $f_d$ for laminar flow is equal to $\frac{64}{Re}$. In comparison with Hagen-Poiseuille, the Darcy-Weisbach equation is directly proportional to the mean-flow velocity squared, $V^2$. As this is only applicable for laminar flow, dimensional analysis can also be used to derive the $\Delta p/velocity$ relationship for turbulent flow (also see pg. 410-411 in [27]):

We assume the pressure loss is some function of the mean-flow velocity $V$, the pipe diameter $D$, the flow length $L$, the absolute pipe wall roughness $\epsilon$, the kinematic viscosity $\mu$, and the fluid density, $\rho$. 

Chapter 2  
Joseph Pomo
\[ \Delta p = F(V, D, \ell, \epsilon, \mu, \rho) \]

Note that, as discussed before, turbulent flow and its associated friction factor are dependent on the pipe wall roughness \( \epsilon \) which explains its inclusion here. The above equation can be represented in dimensionless groups by

\[ \frac{\Delta p}{\frac{1}{2} \rho V^2} = \phi \left( \frac{\rho V D}{\mu}, \frac{\ell}{D}, \frac{\epsilon}{D} \right) \]

Like for laminar flow, it is reasonable to assume the pressure loss in turbulent flow will be proportional to the flow path length \( L \). Making this distinction simplifies the equation:

\[ \frac{\Delta p}{\frac{1}{2} \rho V^2} = \frac{\ell}{D} \phi \left( \text{Re}, \frac{\epsilon}{D} \right) \]

and further

\[ \Delta p = f \frac{\ell}{D} \frac{\rho V^2}{2} \]

Figure 2.7

which is identical to the Darcy-Weisbach equation (equation 2.3) except that the friction factor (specifically the Darcy friction factor \( f_d \)) is some function of the Reynolds number and the relative pipe roughness

\[ f = \phi \left( \text{Re}, \frac{\epsilon}{D} \right) \]

In this thesis, I chose to use the Churchill all-regime equation to calculate the Darcy friction factor because it is yields values comparable to the Colebrook equation, and
is suitable for calculations in the laminar, transitional and turbulent flow regimes (see section 8.5 in [9]).

\[ f_d = 8\left(\frac{8}{Re}\right)^{12} + \frac{1}{(A + B)^{\frac{1}{2}}} \]  
(2.4)

where,

\[ A = \left[2.457 \ln\left(\frac{1}{Re} + 0.27\epsilon\right)\right]^{16} \]  
(2.5)

and

\[ B = \left(\frac{37530}{Re}\right)^{16} \]  
(2.6)

\( Re \) in equation 2.6 is the dimensionless Reynolds number mentioned before, and the quantity \( \frac{\epsilon}{d} \) in equation 2.5 represents the relative wall roughness of the pipe (absolute roughness \( \epsilon \) divided by the pipe diameter). I did not measure the wall-roughness of the 3D printed prototypes in this study, however the authors of reference [3] used a very similar printer and resin, and found that the absolute roughness (\( \epsilon \)) was normally less than 10\( \mu \)m (\( \approx \)0.0004 inches).

2.1.4 Minor Losses

Whereas Major losses refer to frictional losses in straight sections of pipe, Minor losses occur in other types of fittings and fluidic devices, such as bends, expansions, contractions, tees, and valves [9][27]. Flow through these devices may be complex (figure 2.8) and it is often more expeditious to measure the pressure loss experimentally, and obtain a loss coefficient \( K \) for the specific device.

One method for calculating the loss coefficient is by \( K = \frac{\Delta p}{\frac{1}{2}\rho V^2} \), which defines \( K \) as the ratio of pressure loss \( \Delta p \) and the fluid’s dynamic pressure, \( \frac{1}{2}\rho V^2 \). When rearranged, this bears a striking resemblance to equation 2.3 (Darcy-Weisbach), except that the friction factor \( f \) is replaced by \( K \).

\[ \Delta p = \frac{1}{2}K\rho V^2 \]  
(2.7)
As the loss coefficient is highly dependent on geometry, there may exist specific approximations for certain valves or other flow components. It is also not uncommon for manufacturers to publish loss coefficients (or a similar metric) for their parts.

![Figure 2.8: Valves such as the globe valve on the left are commonplace. The flow path through a valve might be quite complex depending on its geometry (R), making direct pressure loss calculations laborious, inaccurate or impossible. As an alternative, the loss is typically determined experimentally and then represented by a loss coefficient \( K \).](image)

### 2.1.5 Calculating Pressure Loss for Common Fluidic Features

The pressure loss for a component is primarily governed by that component’s designed-geometry. As mentioned before, the form of loss may fall into two categories: Major and Minor losses.

#### Losses in Straight Pipe Flow

As mentioned above, the losses in the straight sections are due primarily to friction between the fluid and the pipe it flows through. There are two ways to calculate frictional losses in a pipe. The first, is the Darcy-Weisbach (DW) equation. DW incorporates a friction factor \( f_d \) and can be used to estimate pressure loss for both laminar and turbulent flows in circular pipes.

\[
\Delta p = f_d \frac{\rho V^2}{2} \tag{2.8}
\]
The second, Hagen-Poiseuille (HP), is an analytically-derived equation applicable only to fully-developed, laminar flow in horizontal, circular pipes.

\[ \Delta p = \frac{8\mu LQ}{\pi r^4} \]  

(2.9)

Besides their differing derivations, note that for DW, \( \Delta p \) is proportional to the square of the mean-flow velocity, whereas in HP, \( \Delta P \) is proportional to the volumetric flow rate and the pipe length. This distinction is important, as it means the pressure loss plot (vs. flow rate) will have a distinct shape for each. For DW, the shape is expected to be parabolic, and for HP, linear. Because we required an expression from HP to complete the DW derivation (for laminar flow), we should expect that the results of either the HP or DW will be similar for the laminar regime. However, because of the assumptions made during the derivation of HP (laminar, horizontal, steady flow) we should also expect HP to be unsuitable for losses in the turbulent regime. When studying frictional pressure losses, it is likely acceptable to use only DW for both laminar and turbulent flows, however, including the approximations from HP may also be enlightening particularly because of its 'linear' dependence on flow rate.

**Losses in Bends**

As a fluid travels around a bend, the inside of the bend will experience a higher velocity than the outside. This difference creates a pressure gradient and causes the higher-pressure fluid on the outside of the bend to flow inward. The fluid’s 'forward' momentum persists and results in a characteristic spiral-shape for the flow. This is called 'secondary flow' and generates inertial losses within the fluid. If the bend’s geometry is too severe (as is the case in a 180-degree turn), flow separation may also occur leading to vortex formation and causing further pressure loss (figure 2.9, also see see [9]).

These losses are difficult to quantify on their own, but a loss coefficient, \( K_b \) can
Figure 2.9: Illustration of the two common sources of inertial (energy) losses in bends. (L) flow separation as the fluid rounds a tight 90-degree turn as detaches from the pipe wall. (R) Secondary flow with its characteristic spiral-shape. (See Chapter 15 in [9])

be estimated using H. Ito’s bend-loss equation (see Chapter 15 in [9]):

\[ K_b = f \frac{r \alpha}{d} + (0.10 + 2.4f) \sin \left( \frac{\alpha}{2} \right) + \frac{6.6f \left( \sqrt{\sin \left( \frac{\alpha}{2} \right)} + \sin \left( \frac{\alpha}{2} \right) \right)}{\left( \frac{r}{d} \right)^4} \]  

(2.10)

Here, \( \alpha \) is the bend angle (in degrees), \( r \) is the bend radius and \( d \) is the pipe diameter. Figure 2.10 shows how \( K_b \) and \( \frac{r}{d} \) contribute to the pressure loss mechanism. As the bend radius-to-diameter ratio increases, the frictional losses dominate.

Figure 2.10: Plot demonstrating the relationship between the bend radius-to-diameter ratio and the loss coefficient \( K_b \). As the ratio increases, the losses are dominated more by surface friction (See Chapter 15 in [9]).

To calculate the total losses (including frictional losses) for a bend, we use a form of the Darcy-Weisbach equation:

\[ \Delta p = \frac{1}{2} \left( f_d \rho \frac{r_b}{d} \pi \frac{\theta}{180} V^2 + K_b \rho V^2 \right) \]

(2.11)
Where $K_b$ is the bend loss coefficient, $r_b$ is the bend radius, $V^2$ is the mean velocity of the fluid. Also note that the first term in equation 2.11 represents frictional losses for a pipe-length equivalent to the length of the bend’s center line. $f_d$, the Darcy friction factor is calculated by the Churchill equation mentioned before (equation 2.4).

**Losses in Orifices**

As mentioned before, an orifice creates a pressure drop first by inertial losses as the fluid hits the orifice edge, then through frictional losses within the contraction, and finally, losses due to flow separation as the fluid exits the orifice and quickly expands back to the original pipe diameter (figure 2.11). Regions of sudden expansion or contractions are common in pipe flow componentry, especially at entrances and exits (see chapters 9 & 12 in [9]). There are a variety of orifice designs, including those with flat, sharp, beveled, or chamfered edges. In addition, the orifice contraction has some diameter and some thickness associated with it. Regardless of the exact design, their pressure loss mechanism relies on the fluid sustaining inertial losses as it hits the edge of the orifice, followed by frictional losses in the contraction as the fluid’s velocity increases and interacts with the walls of the contraction, and possible flow separation as it exits the contraction.

![Figure 2.11: Drawing of thick-edged orifice with critical dimensions indicated. The orifice on the left has a lower thickness-to-diameter ratio than the one on the right. On the right, the streamlines suggest the flow reattaches to the walls inside the contraction (see Chapter 13 in [9]).](image)

Thick-edged orifice are a common, simple orifice design. When estimating the total pressure loss from the multiple sources mentioned above, a loss coefficient, $K_o$,
is used. $K_o$ can be calculated from the following equation, using the known geometry of the orifice (See Chapter 13 in [9]):

$$K_o = 0.0696(1 - \beta^5)\lambda^2 + (\lambda - 1)^2 + (1 - \beta^2)^2 + f_o(\frac{t}{d_o} - 1.4) \tag{2.12}$$

for $\left(\frac{t}{d_o}\right) > 1.4$, where $\beta$ is the ratio of the inlet and orifice diameters ($\frac{d_i}{d_o}$) and $\lambda$ is the jet contraction ratio given by:

$$\lambda = 1 + 0.622(1 - 0.215\beta^2 - 0.785\beta^5) \tag{2.13}$$

The jet contraction ratio gives a measure of how the cross-sectional area changes before and after entering a contraction. For thick-edged orifices, the flow may contract significantly as it passes over the sharp-edged entrance. A rounded entrance would result in less contraction (i.e. a ratio closer to unity). If the thickness-to-diameter ratio $\frac{t}{d_o}$ is greater than about 1.4, the flow may also reattached to the orifice throat and create greater pressure losses due to friction.

To calculate the total pressure loss, we can incorporate $K_o$ into the equation 2.7:

$$\Delta P = K_o\frac{\rho V^2}{2} \tag{2.14}$$

### 2.1.6 Pressure Loss Summary

There are several key points we can garner from the analysis above:

1. The pressure loss due to friction may be estimated using either the Hagen-Poiseuille or Darcy-Weisbach Equations

2. The Darcy-Weisbach equation is applicable to both laminar and turbulent flows when using the appropriate friction factor

3. Loss coefficients are useful for estimating pressure loss in components with complicated flow paths or multiple sources of loss.

4. The pressure loss is proportional to the flow rate $Q$ for the Hagen-Poiseuille
estimation, but is proportional to the square of the flow velocity, \((V^2)\) for Darcy-Weisbach friction losses, and minor loss estimations.

The last point is critical because it lays the foundation for ERVFM concept. Knowing that frictional losses (for laminar flow) should have a linear relationship with flow rate, and the losses for certain other fluidic designs have a quadratic relationship to the fluid velocity (and by extension the flow rate), as in the case of an orifice, we may be able to exploit each to create a device with two distinct loss mechanisms. Specifically for the ERVFM, we may be able to design a type of valve, with two outputs, which diverts the flow to one output or another based solely on the relative pressure loss in that output. For instance, one output could have an orifice placed in the flow path while the other could have a long, straight pipe leading to the output. If each output is carefully designed, it is possible that for lower flow rates, the orifice-containing output would be preferred, but for higher flow rates, the long, straight path would have the larger fraction of flow.

Figure 2.12: Idealized pressure loss vs. flow rate plot. Based on the pressure loss mechanism of each output, the plotted pressure loss may have either a parabolic (orifice output) or a linear plot (long, straight output). If this assessment holds, we should expect a distinct cross-over point below which the orifice will have a higher fractional output of flow, and above which the long, straight section will have a higher flow output.

Such a valve would have no moving parts and may fulfill the reliability goal of the
If the design concept performs the way I expect, the fractional flow exiting each output would never be zero. Put another way, the valve will always leak some amount of fluid from the outputs at all times. The hope is that the 'relative fraction' of flow exiting either output will change with the flow rate. A valve of this type might be appropriately called \textit{passive}, however, it is not the only valve concept of its kind. The following section explores other examples of passive valve designs.

\section*{2.2 A Brief Summary of Passive Valve Designs}

A passive valve is a flow control device which uses non-mechanical means to regulate flow rate, such as specially-designed geometry of the flow path, or by exploiting some physical effect\cite{41}. The fluid flow industry is brimming with gate valves, butterfly valves, ball valves, globe valves, check valves and solenoid actuated-valves. Passive valves appeared to be less common until I looked to microfluidics research. In that field, creating commandable flaps or gates is more difficult than on the macro-scale due to the fabrication techniques, so instead, other methods are explored. The following are a few notable examples of both macro and micro passive valves:

\subsection*{2.2.1 Tesla Valve}

Nikola Tesla's "Valvular Conduit" (Tesla Valve) is one of the most well-known examples of a 'passive' valve (figure 2.13). It functions like a flow diode by allowing low resistance to the flow in the 'forward' direction and relatively high-resistance in the 'backward' direction. This is accomplished through alternating rows of flow channels which, when driven in the backward-direction, forces the flow back on itself and creates significant losses \cite{25}.
This valve design is understood as being 'leaky' when flowed in the backward direction (flow is not completely impeded); which means it is most useful only in applications where this property is acceptable[6].

2.2.2 Bubble-based Microvalve

Khoshmanesh et al. describe a 'hydrodynamically actuated, bubble-based microfluidic system' for pumping and mixing fluids, as well as selectively diverting the flow. This works by introducing gas bubbles at specific points in the enclosed flow path, by means of an external pump, laser pulses, or electrolysis[23]. This strategy works because the fluids do not readily mix with the gas bubble at the tested flow rate (10 micro-liters per minute). But its efficacy at higher flow rates and with lower viscosity working fluids may be compromised.

2.2.3 Jet-Diversion & Pressure-Driven Microvalves

Jet-diversion valves pump a stream of fluid normal to, or at an oblique angle respective to the primary flow. This means that, in addition to the inlet and outlet, two other channels (or cavities) exist to supply the diverting jet (called the control flow), and to collect the diverted fluid (see figure 2.15).
Figure 2.14: Bubble-based valve from Khoshmanesh et al. A. Full open mode (no bubble obstruction). B. Partial bubble obstruction reduces red stream flow. C. Full bubble stops the red stream completely. D. Bubble removed, stream allowed to reintegrate. Source: [23]

Figure 2.15: Jet-diversion type valve which uses a jet-stream originating at X to force the supply flow (S) into the collection cavity (V), thereby bypassing the output (Y). Source:[39]

Even without the fluid jet, simply creating a pressure difference in adjacent cavities can limit the flow, however, this is admittedly not as effective.
2.2.4 Quake Valves

Quake valves are microvalves produced through a cast lithography process. They are traditionally made of elastomers (such as PDMS) in a three-layer structure which contains a flow channel and a control channel. The channels exist in separate layers so that their contents do not interact, and the valve is actuated by filling the control channel with another fluid or gas. The control channel expands (normally by pushing against a rigid surface) and causes the flow channel to collapse, which crimps the flow (see figure 2.17)[2]. While this technically violates the 'passive valve' paradigm of this section because it contains a moving part, it maintains the simplicity and apparent reliability I am seeking.
Figure 2.17: Diagram describing the mechanism of a 3D printed quake valve. A. shows the input and output ports for the control and flow channels. B. Shows the printed structure. C - F Top and side views of the channels. G. Valve open H. Valve closed. Source: [2]

2.2.5 Coandă-Effect Microvalve

The Coandă-effect describes the tendency for a flowing fluid to attach-to and follow a curved surface. If the curvature become too aggressive, the flow will eventually separate from the surface, but prior to that point, the effect can be exploited to bias a flow in a desired direction[38]. In a 'Y'-shaped, two-way valve, the flow may prefer one output due to the coandă-effect but can be forced into the other output using a diverting jet (as mentioned above), or by increasing the pressure loss of the preferred path (figure 2.18)

2.2.6 Geometry-Driven Oscillator

Microfluidic oscillators, like the one in figure 2.19 are symmetric devices which back-flow the working fluid (via adjacent channels) and direct it at the inlet. This forces the primary (axial) flow to one side of the device or the other, in a ping-pong fashion. The oscillation frequency is dependent on the inlet flow rate and the device’s geometry. This is not a valve, however, I include it in this list because it accomplishes a valve-like behavior without a commanded feature[26].
Figure 2.18: Coandă valve in its open state with flow attaching to the curved surface (L). Valve in closed-state in which the control flow diverts the inlet flow to the vent (R). [36]

Figure 2.19: A progression of simulated contours for an oscillator design. The backflow channels on either side feedback to the inlet and disrupt the flow, forcing it to exit on one side of the output, or the other. The geometry and flow rate determine the oscillating frequency. Source: [26]

### 2.3 Solid-State Selector Valve

Building off of the passive valve concepts from before, we can examine the Solid-State Selector Valve (SSSV), which, along with the parallel-configured VFMs, completes the ERVFM design.

The SSSV consists of three major parts: the input, which accepts the incoming flow and splits it in a Y-shaped junction, and two distinct outputs, which each direct the flow through a dedicated VFM. Based on the pressure loss mechanisms of mentioned
before, I designed one output to have a thick-edged orifice, and the other to have a serpentine comprising straight sections linked by 180-degree bends. For the orifice output, we should expect the pressure loss to increase as a second-order (quadratic) function of the flow rate. The other output would, ideally, be a long, straight path for which friction is the main source of pressure loss. This would give the output a linear pressure loss function. However, I learned from early prototypes that such an output would be prohibitively long, and I instead chose to break the straight path into sections connected by 180-degree bends to create a reciprocating serpentine shape. (figure 2.20).

![Figure 2.20: SSSV concept design with one input and two outputs. The orifice and a serpentine path are visible with a close-up of the orifice to show design detail.](image)

### 2.3.1 Proposed SSSV Mechanism

Like some of the examples of passive valves, the SSSV is proposed to divert, or 'switch', outputs as the relative pressure loss changes in each output. At low flow rates, because of the specific geometry and pressure loss mechanism, the orifice path will have relatively lower resistance compared to the serpentine path, but as flow rate increases, the orifice’s resistance will eclipse the serpentine’s. If we plot the pressure loss versus flow rate for each output, we would expect to see a distinct 'cross-over' point in which outputs swap flow rate dominance. An increase in flow rate will always
accompany an increase in pressure loss, but we are concerned with the relative pressure loss among the outputs.

The first orifice output (figure 2.21) utilizes a thick-edged orifice mentioned before. Its pressure loss depends primarily on the geometry of the orifice, including the ratio of the pipe inlet and orifice diameters, $\beta$, and the ratio of the orifice thickness (length) and diameter $\frac{t}{d_o}$. These are rolled up into a loss coefficient called $K_o$. For the ser-

![Figure 2.21: Example of thick-edged orifice with expected fluid interactions. As the fluid flows from left to right, it first interacts with the orifice edge, then passes through the constriction in the orifice, and finally exits the orifice and expands back to the original flow diameter.](image)

pentine output (figure 2.22), the straight sections will have pressure losses associated with viscous effects (friction) and can be calculated with either the Darcy-Weisbach or Hagen-Poiseuille (laminar only) equations. Like the orifice, the losses in the bends are also represented by a loss coefficient: $K_b$ which accounts for the angle of the bend $\alpha$, as well as the friction factor $f_d$. A drawing of the differing velocities within the bend (and subsequent pressure gradient) is shown in figure 2.23.

The ‘switching’ capability of the SSSV is critical to the ERVFM functioning as expected. We can look again at some idealized plots of the pressure loss and flow rate (figure ??) and notice that choosing to calculate the losses due to friction via Darcy-Weisbach versus Hagen-Poiseuille shows a different valve behavior. When estimating the pressure loss using Darcy-Weisbach, the second-order shape of the plot is expected
Figure 2.22: Drawing which shows the expected flow in the straight sections and bends within the serpentine path to overlap with the similarly shaped orifice loss plot. In this case, it is possible that the fluid exits each output equally until the plots diverge at some time. After diverging, the orifice’s pressure loss is expected to supersede the serpentine’s suggesting that more fluid will exit through the serpentine. In the case of Hagen-Poiseuille estimates, the linear shape of the serpentine’s pressure loss plot intercepts the orifice plot at some flow rate. Prior to this ‘cross-over point’, the serpentine output had higher pressure losses resulting in more fluid flow through the orifice output. After the cross-over point, this behavior flips as the orifice’s pressure loss quickly eclipses that of the serpentine.

Based on how I expect the valve to work, the Hagen-Poiseuille-based plot is the preferred behavior as it provides a distinct behavior above and below a distinct flow rate. These idealized plots work as an initial concept, however, the next chapter outlines the methods I used to both model the pressure loss in python, and run real-world experiments on valve prototypes.
Figure 2.23: The differing velocities within a flow lead to a pressure gradient throughout the fluid. The end result is a helical-shaped secondary flow and further pressure loss.

Figure 2.24: Idealized pressure loss vs. flow rate plots for Darcy-Weisbach (DW) & Hagen-Poiseuille (HP) frictional loss estimates. (L) The DW plot shows that the losses for both outputs is quadratic in shape suggesting that the plots will be colinear and eventually diverge. After this, the orifice pressure loss climbs faster than the serpentine output’s indicating more fluid will exit through the serpentine. (R) The HP frictional loss estimate is linear, resulting in a distinct, and theoretical, output switching point. Prior to the switch, more fluid will exit the orifice output (lower pressure loss compared to serpentine output). After the cross-over point, the behavior switches, and the orifice has the larger pressure loss (resulting in less fluid output). The plot on the right is the preferred scenario.
Chapter 3

Methods

Preface

The experimental methods in this study are organized into two categories:

1. Mathematical Models
2. Test Setup (Design and Build) & Data Collection

In the first, I used models to estimate the pressure loss in the VFMs and SSSV prototypes. In the second, I designed and constructed a pumped fluid loop (PFL) test setup and created a workflow for acquiring and analyzing data from the PFL.

3.1 Venturi Tube Pressure Loss Model

Focusing first on the VFMs, I estimated the range of pressure losses in each Venturi tube prototype. Knowing this ensured I would not exceed the upper limit of the DPTs. Pressure losses within VFMs are normally due to surface friction from the throat, and both the converging inlet, and diverging diffuser, cones. The diverging section may also have expansion-losses (due to flow separation), however, this is usually mitigated by designing a gradual expansion ([9], p. 158).

I estimated the pressure differential in the VFMs over a range of flow rates using the equation derived from the Continuity of Mass and the Bernoulli equation:

\[ Q = A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho[1 - (A_2/A_1)^2]}} \]
3.1.1 Measured VFM $\Delta p$ and Flow Rate Calculations

The differential pressure transducers (DPTs) which I used in the test setup produce distinct voltages as the flow rate in the VFM changes. I measured these voltages using an Arduino and output the data into an excel spreadsheet and, later a python program for data analysis. Each DPT has an associated calibration worksheet used to ascertain the differential pressure (this voltage corresponds to that pressure). I mapped the range of measured voltages to the range of corresponding differential pressures, and finally used equation 3.3 to calculate the inlet flow rate.

3.1.2 VFM Prototypes

The benefit of having a pressure loss model for the VFMs is I could use it to estimate the expected differential pressure ($\Delta p$), and could later use experimental data to validate the model’s accuracy.

When designing the VFMs, I realized that a balance must be achieved between too small and too large of a throat. The former would produce larger $\Delta p$s at lower flow rates, which is desirable if the DPTs’ resolution is low; but it may also produce $\Delta p$s which exceed the upper limit of DPT’s range. The latter has the opposite issue, potentially producing $\Delta p$s which are indiscernible at the lower limit.

One goal of the ERVFM is to maximize the range over which accurate flow rate estimates can be made. At the system-level, this means that one VFM ought to perform best at lower flow rates, and the other at higher flow rates. Ideally, there should be minimal overlap in between the sensor ranges.

I tested several prototypes throughout this study; all of which had a similar overall design, and differed only in the relative cross-sectional area of the throat (figure 3.1).

Some of my early designs were functional, but included unintended sources of pressure loss, such as expansions and contractions at the entrances and exits (figure 3.2).
Figure 3.1: Cross-section drawing of final VFM designs. B. has the largest throat diameter, A. is smaller than B, and C. is the smaller still. I estimated the pressure loss range for each using a python model before prototyping and testing.

Figure 3.2: Early VFM design showing potential sources of minor losses in the form of sudden expansions and contractions. The threaded fittings would not fully ’seat’ in the printed prototypes. This was later mitigated with a design overhaul.

I originally intended to use standard, 1/4" NPT swagelok fittings to make the connections to each tube and encountered several challenges:

1. The Formlabs Form3B printer could not print the 1/4" pipe-threads with sufficient precision. The threads looked nice, however the thread profile was not accurate enough to be usable (Left and middle images in figure 3.3).

2. Internal threads could be *chased* with a cutting tap, yielding usable results.
Chasing the external threads did not work, as the printed parts were too brittle, and easily cracked (Image furthest to right in figure 3.3).

3. Both external and internal threads had to interface with Swagelok-style fittings which created areas of expansion or contraction for the flow (figures 3.2 and 3.3 (bottom)).

Figure 3.3: Thread-cutting experiments on early Venturi prototypes. Internal 1/4 in NPT threads (pre-chasing) (L), External-Threads (pre-chasing) (M), External-threads after failed attempt to chase with thread-cutting die(R). Successful fitting install in internal threads (Note crooked fitting on left) (B). I ascertained from these and other trials that external threads were too brittle to be cut to the correct profile, while the internal threads led to potentially large minor losses mentioned before.

3.1.3 SSSV Pressure Loss Model

With a method established for estimating and measuring flow rates within the VFMs, I created a separate pressure loss model for the SSSV.

The mode of operation for the SSSV relies on the relative pressure differential of each output, over a range of flow rates. If designed with care, we should expect more working fluid to exit one output for ’low’ flow rates, and more fluid to exit the other
output for 'high' flow rates. The terms 'Low' and 'High' here are arbitrary, and the actual cross-over point depends highly on the valve’s geometry.

I created a python program into which I could input the SSSV’s geometry and working fluid properties, and output an estimated pressure loss for each output. Knowing this, I then plotted the loss for each output over a series of flow rates and could then easily estimate where the valve’s flow may switch.

3.1.4 Orifice Output

The orifice output utilizes a thick-edged orifice design to induce pressure loss. The important design parameters are the diameter of the orifice constriction, and the constriction’s length-to-diameter ratio. This type of orifice creates inertial losses as the working fluid hits the orifice leading-edge, as well as losses due to flow separation as the fluid exits the constriction into a sudden expansion. Also, if the constriction’s length is about 1.4 times the diameter, frictional losses should also be taken into account ([9], pg.147)

3.2 Test Setup & Data Collection

I designed the pumped-fluid loop (PFL) as a modular test setup into which I could easily install and remove printed prototypes. The majority of the flow-path uses grade 304L stainless steel tubing with an outer diameter of $\frac{1}{4}$-inches and inner diameter of about $\frac{3}{16}$-inches (actual measured = 0.18”/4.57mm). The steel tubing connects all flow components together via either Swagelok compression fittings (when interfacing with the steel tube), or direct press-fits into printed parts. I secured the components to a 36 by 36-inch optical table (figure 3.4).
Figure 3.4: Pumped fluid loop test setup with main components labeled. I frequently changed the SSSV and VFM prototypes, and also used three different pumps before settling on the KNF-FP-150

I used a five gallon bucket as a water reservoir. A length of roughly 4 feet of 1/4 inch I.D. tygon polymer tubing stretched from the bottom of the bucket to the inlet on the pump. The pump outlet connected via a short length of tygon tubing to a $\frac{1}{4}$-inch barbed fitting which itself connected to the steel tubing just preceding the Omega flow meter. These components mark the beginning of the PFL. After passing through the Omega flow meter, the pumped fluid encounters two 90-degree turns and then enters the SSSV. The fluid flow splits into two paths at a ‘Y’ within the SSSV and enters one of the VFMs. The outputs from each VFM return the water to the reservoir, each using a 4-foot length of tygon tubing.

I used a KNF FP150 diaphragm pump (model: KPDCB-4) to drive the fluid in the PFL. I previously used a Cole-Palmer peristaltic pump but discovered that the pulsations produced by the pump caused intermittent spikes in the VFM pressure readings. I attempted to mitigate this using a commercial, inline pulsation damper (Cole-Palmer), however, the issue remained. By comparison, the FP150 produces very low pulsations which have an imperceptible effect on the differential pressure data.
The motor on the KNF pump accepts 24 volt DC power and a 0 to 5 volt signal to set the speed. The pump data sheet states that the pump flow rate ranges from $0 \ \frac{l}{min}$ to $1.5 \ \frac{l}{min}$, with a maximum pressure head of 29 pounds-per-square-inch (PSI, Gauge pressure). To verify the former, I utilized the cup and stopwatch method (CSM) to measure the mass flow rate. First, I tared a scale with a small plastic cup. Then, I set the pump to a specific speed and measured the mass of water which entered the cup in a 5-second time span (figure 3.5). I repeated this for five replicates, over fourteen different pump speeds (including zero), and tabulated the results. Although slow, the method resulted in a standard deviation of only 1 to $1.5 \ \frac{ml}{s}$ across measurements.

![Figure 3.5: Photos demonstrating the 'cup and stopwatch method' to measure mass flow rate at different points in the pumped fluid loop.](image)

### 3.2.1 Sensors

I tested several different differential pressure transducers (DPTs) throughout this study and eventually settled on the setup outlined below. I used the "Venturi Tube Pressure loss Model" as a guide for choosing a pressure sensor with an appropriate range, and was fortunate to find suitable pressure sensors already available in the AFRL lab. I also used a turbine flow meter to capture baseline data for the flow rate as the working fluid exited the pump. For data acquisition, I used several Arduino Mega micro-controllers. The Arduino Mega uses a 10-bit analog-to-digital converter, and in
the case of the DPTs I used, the 0-5 volt input corresponds to a 0-1023 integer output at the Arduino. This gives an $\approx 4.9 \text{mV}$ measurement resolution; or, approximately $\pm 70 \text{ Pa}$. For the Omega turbine flow meter I also used an Arduino to measure the meter’s pulse generation, which I expected to increase with the flow rate. The meter’s specifications sheet states that a 1-liter volume passing through the meter corresponds to about 30,000 total pulses. The Arduino mega uses an Atmel ATmega 2560 microcontroller with a maximum sampling frequency of $\approx 15,000 \text{ samples/second}$ (See Section 26 in [4]). At the maximum flow rate of $1.5 \text{l/min}$, this corresponds to a $750 \text{ pulses/s}$ output from the turbine meter, which falls well below the maximum threshold for the Arduino.

The Omega turbine meter is the first sensor to follow the pump and serves as the "ground-truth" for the system’s flow rate measurements (figure 3.6, model: FTB-1311). The meter has a designed measurement range of 0.3 to 1.5 liters per minute (LPM) and purports to have a $\pm 1\%$ accuracy to the actual flow (see [31] and [33]). The meter contains a 'pickup' which senses a magnetic flux as the turbine spins. I supplied the meter with 24V DC and fed the signal wire into a digital input in the Arduino. Using the meter’s stated number of sensed pulses per unit volume, I created a script for the Arduino (adapted from [17]) which counts the number of pulses and outputs a flow rate ($\frac{l}{\text{min}}$) to the Arduino serial monitor. I consolidated these flow rate readings and plotted them against the pump signal voltage using the Python MatPlotLib library. Downstream from the turbine meter are two DPTs (Omega PX409-10WDWUV). The DPTs use a silicon diaphragm and strain gauge arrangement to generate voltages (0-5V) when a differential pressure is present. This transducer model claims to have a $\pm 0.8\%$ accuracy to the actual, gauge differential pressure. Omega defines this accuracy as the combination of the Best Straight Line (BSL), Hysteresis and Repeatability error (see [32]). The usable range for each DPT is 0-10 $\text{inH}_2\text{O}$ (0 - 2488.4 Pa). Each DPT designates a high and low-pressure input which I connected to the corresponding high and low pressure taps of each VFM (figure 3.7). For these DPTs, an increase in differential pressure leads to an increase in the sensor’s output voltage. I fed each DPT output into the analog input of an
additional Arduino and used a script to display voltages in the serial monitor. I then took these voltages and used a Python program to convert them into differential pressures and, knowing the differential pressure, I used equation 3.3 to calculate the flow rate at the Venturi TUBE input.

### 3.3 Printed Fluidic Components

I used a Formlabs Form3B liquid-resin stereolithography (SLA) 3D printer to produce both the SSSV and VFM prototypes. At the start of this study, I suspected that using traditional manufacturing should be avoided for the following reasons:

1. 'Subtractive' manufacturing via a milling machine or lathe would be more time-consuming because of the feature count of the parts, which would require multiple machining operations and specialized tooling.

2. Likewise, I did not know the exact number of prototypes I would need to test, but I anticipated that manufacturing each could take considerable time.

3. Creating the serpentine or orifice paths using subtractive manufacturing would
require the SSSV to be machined in multiple pieces. The pieces would then need to be precisely aligned and made water-tight during assembly.

Additive manufacturing remedied each of these issues. I was able to print all SSSV and VFM prototypes as monolithic structures with no multi-part assemblies or fasteners. Because of their size, I could 3D print each of the prototypes and have them installed in the system within one working day. Complex features like the internal channels required no extra effort on the manufacturing side. When iterating, a change to the design in CAD was normally all that was needed and the printing process remained the same.

3.3.1 Design and Printing Parameters

I designed all prototypes with Autodesk Fusion360 parametric design software. When it was necessary to iterate a design, I often rolled back the design ‘timeline’ to add new features or copied-and-pasted an existing prototype for more significant changes. This ensured I would not have to start from scratch for each new prototype design.

I exported each solid body from Fusion360 in the stereolithography (.STL) format.
with the mesh refinement set to 'high'. This setting controls the resolution of the exported file; if set too low, the tessellated, polygon structure of the mesh would be more prominent and could lead to problematic print artifacts in the end result. Setting the refinement higher leads to larger files on the disk, but will ensure that geometry, like circles, are as circular as possible.

I imported the .STL files directly into the Formlabs slicing software, Preform, for preparation (figure Version 3.14.0, Formlabs). For all prototypes and test prints I used Formlabs Clear V4 resin (Part Number: RS-F2-GPCL-04) with 50 \( \mu \)m layer thickness (fixed, not adaptive). Preform gave warnings if the part geometry contained captive features, such as the reciprocating serpentine path. These areas of the print will create suction against the resin vat and could lead to layer irregularities or print failure. I generally ignored this warning and chose to minimize these areas by reorienting the .STL file in the software. I used the automatic support generation feature and made only slight adjustments to the support areas, such as deleting any supports which generated inside the flow-paths as they were difficult to remove otherwise. I printed all parts on a support ‘raft’ which made removing the print from the build platform much easier. Following this workflow, most prints were successful, however I did notice several artifacts such as distinct lines, ribbing and deformed channels in some parts (discussed in the results chapter). These artifacts were inconsistent enough that I did not need to pursue fixes. I deliberately did not change the software and firmware version for the printer throughout this study.

### 3.3.2 Print Post-Processing

Once a print completed, I removed it from the build-plate by prying up the raft with a flat, scraper tool and immediately started the cleaning process by submerging the print in a solution of 99% isopropyl alcohol (IPA). The print comes out of the printer vat covered in resin and not entirely cured, which is referred to as the ‘green’ state. The resin cross-links when exposed to ultraviolet (UV) light, so a prompt cleaning is
Figure 3.8: Screenshot of Preform software with SSSV prototype version 0.57. The model is tilted at roughly 45-degrees relative to the horizontal plane and requires tree-branch-like support structures during the printing process.

necessary to ensure small features do not get filled and cured into the rest of the print by ambient light sources. I used a series of three IPA wash tubs, into which I dunked the printed parts multiple times to remove excess resin. For the SSSV and VFMs, I used a syringe and blunt-end needle to wash resin out of channels or small holes; this is an important step, as even a small amount of leftover resin could change the dimensions of critical features once fully cured. The exact wash procedure differed between parts, and I continued the process until the part surface was free of resin and was no longer tacky after drying. Following the washes, for the SSSV prototypes only, I placed the prints in the Form Cure UV chamber (Formlabs, Part #: FH-CU-01) for 15 minutes at room temperature.

The Material Properties Data Sheet for the Clear V4 resin makes a distinction between 'Green' parts (directly from the printer) and 'Post-Cured Parts', which are cured under specific conditions (60 minutes at 60°C with a 405nm LED at 1.25mW/cm² power). Post-cured parts generally have higher heat deflection temperatures, tensile-strength, tensile-modulus, and flexural-modulus [13]. I chose not to post-cure the VFM prototypes because curing made the parts noticeably brittle and more diffi-
cult to cut threads into the pressure tap holes. The post-cured parts also develop a yellow hue, although both the translucency, and overall function of the parts, were unaffected.

I explored several methods for improving translucency of the printed parts and settled on a combination of wet-sanding surfaces to a glass-smooth finish (progression of 600-2000 grit sandpaper), followed by a light coat of Krylon ‘Crystal Clear Acrylic’ spray paint. This treatment for the outer surfaces, combined with the distilled water flowing within the internal channels made flow visualization possible (see figure 5.18 in the results chapter). This seems to work because the acrylic coating fills the surface imperfections of the print and allows more light to pass through the structure. A similar effect can be achieved by putting several drops of water or mineral oil on a section of the print surface.

3.3.3 Integrating Printed Parts into the PFL

I originally planned to use SwageLok-style fittings to interface the printed parts with the PFL. My first attempt used SwageLok fittings with 1/4", female National Pipe Thread (NPT) threads. NPT threads are designed with a slight taper which creates a seal at the interface, without the need for an additional gasket. As an initial test, I designed a Venturi tube with 1/4" male NPT thread which I planned to screw into the SwageLok fitting, but found that the thread profile did not properly form (see figure 3.2 from before). I suspect this may be due to the print orientation (printed upright, perpendicular to the build-plate) as well as a resolution limit for the Form3B printer. Thinking the threads were likely close to a near-net-shape, I attempted to chase the printed external threads with a thread cutting die to achieve better results. This too proved difficult, as the printed threads easily shattered under the force of the die (I tested this with green and post-cured parts with similar results).

Following this experience, I changed my approach and attempted to cut internal threads into the parts. I designed a \( \frac{3}{16} \)-inch (11.1mm) diameter hole and used a \( \frac{1}{4} \)-inch NPT bottoming tap to cut the threads. This produced much better results compared
to my attempt with external threads, and I was able to thread the SwageLok fittings into the printed parts and run the pump without leaks. I placed the parts in a vise and tapped them by hand using a tap handle. I found that if I did not keep the tap completely straight, the threads would be slightly tilted. This led to air leaks and usually meant that a new print was needed. Also, without exception, there was always a gap between the Venturi tube flow path and the inlet for the fitting. This is primarily a limitation of the thread tapping-process as the cutter will usually not reach the very end of a blind hole. This gap would act as a sudden expansion followed by a sudden contraction and create large, irrecoverable losses (see chapters 9 and 12 in [9]).

In an effort to mitigate pressure losses, I redesigned the interface between the stainless-steel tube and the printed parts. For both the SSSV and VFM, I designed the inlets and outlets to have an outer-diameter slightly smaller than that of the stainless-steel tube. The prints themselves have some amount of 'shrinkage' due to the printing-process, so I created a series of parts with through-holes with 50µm differences in the diameter. I then found which iteration gave a reliable interference-fit and used that diameter for parts going forward. This method provided a sufficient way of securing the steel tube within the parts without any threaded feature. I also wanted to ensure the interface was both air and water-tight, so I designed channels into the printed parts which I back-filled with room-temperature vulcanizing (RTV) silicone (figure 3.9). In practice, this interface was not absolutely secure (in some cases, the tube could be easily removed by-hand), however, it was sufficient for the PFL test setup and ensured that the tube was fully inserted and butted against the flow path in the printed parts.

To further mitigate irrecoverable losses, I chamfered several areas within the printed parts and also chamfered both ends of the entrance and exit tubes (figures 3.10. Also see Chapter 9, pg. 91 in [9]) This creates a small, diamond-shaped expansion and contraction as the tube meets the print, but it represents a large improvement compared to the threaded fittings.
Figure 3.9: Preparing VFMs for the RTV silicone gasket process. I used the syringe to fill pre-designed cavities with RTV silicone at the interface between the steel tubes and the printed parts.

Figure 3.10: CAD model (T) and microscope images (B) showing chamfered features. At this scale (less than 2mm), flat chamfers become more rounded due to the 3D printer’s feature-size limitations. The red arrows indicates the print orientation.
Chapter 4

Testing

Preface

My testing procedure comprised two main areas:

- PFL Component Validation
- SSSV & VFM Prototype Evaluation

In the first place, I sought to check whether the PFL and its sensors functioned as intended. This includes checking the function of the pump, the turbine flowmeter, the DPTs, the arduinos and the data acquisition scripts. I dedicated a large portion of testing to validation because I was unsure how each component, being its own, stand-alone device, would work as an assembled system. Then, once I was sure the sensors were working, I evaluated the printed prototypes. When applicable, I used a simple technique called the cup-and-stopwatch method (CSM) to measure the mass flowrate of the fluid entering or exiting a specific portion of the PFL. In the CSM, I used a digital scale (Ohaus Explorer) to measure the mass of an 8-oz cup (10g HDPE) and then tared the scale. I set the pump speed signal to a distinct voltage, between 0 and 5 volts and then captured the water dispensed in the cup for 5 seconds. I then measured the mass of the cup with the water inside, logged the mass and repeated this process for a total of 5 replicates at the same voltage. I attempted to control for human error by measuring more than 5 replicates if the difference in mass among a replicate cohort was greater than 3g/5sec.
4.1 PFL Component Flow Rate Validation

4.1.1 KNF Diaphragm Pump

I controlled the KNF pump by inputting a 0 to 5 volt signal from a variable power supply. According to the data sheet, a voltage signal from 0 to 5 volts should correspond to pump speeds from 0 to 1.5 liters per minute (LPM, Cite/Link to appendix for KNF data sheet here). To test this, I disconnected the outlet of the pump from the PFL and selected 15 distinct voltages (from 0-5V) in 0.36V increments. Following the CSM outlined above, I logged the mass flowrate and confirmed the advertised flowrate range.

4.1.2 Omega Turbine Flowmeter

The turbine flowmeter served as the 'ground-truth' for the downstream DPTs so that I could verify, to some degree, the mass flowrates entering each of the VFMs. The data sheet for the turbine flowmeter specified the number of pulses per liter for the meter (1 liter = 33021.52), and I used this value to calculate the volumetric flowrate at the same, distinct pump signal voltages from before. I then captured the data from the arduino serial monitor in real-time.

4.1.3 Flowrate Entering into the SSSV

Using the same CSM, I measured the mass flowrate entering the SSSV. As mentioned before, I integrated each SSSV prototype into the loop by inserting a section of steel tube into the valve input and sealing it with RTV silicone. To measure the mass flowrate, I simply removed the valve and measured the fluid exiting the tube section. I then passed this data as a parameter to the SSSV Pressure Loss python script which aided the SSSV design.
4.1.4 Flowrate Exiting the SSSV (Entering the VFMs)

Again, using the CSM, I measured the output from the SSSV for each SSSV prototype. I first measured the fluid exiting both outputs simultaneously, which I could then corroborate with the measured flowrate entering the SSSV. Then, I measured the flowrate exiting each output, individually. In this way, the simultaneous measurements should match the sum of the two individual measurements. The measurements from the individual outputs change among prototypes, and comparing the outputs should indicate whether the valve is ‘switching’ as intended.

Knowing the flowrate entering the VFMs, I could then use the data output from each DPT and calculate the flowrate, for comparison.

4.1.5 Flowrate Entering the VFMs (DPT Measurements)

Following the same process as before, I set the pump speed and then gathered data from each of the DPTs. I then took the raw voltages from the DPTs and processed them in the python script mentioned before.

Regarding the DPTs, the data-sheets indicated a minimum and maximum sensor output value depending on the sensed flowrate. For instance, a flowrate equal to 0 should output a sensor value equal to -0.004V. In practice, elevation of the DPT itself caused a discrepancy in the minimum value. Simply lifting one side of the DPT relative to the other caused the minimum value to change, and I was not able to reach the specified value, even when carefully shimming each side of the DPT. Instead, I decided to treat this minimum value as an offset-correction which I applied to all other data in that set of measurements. This value changed each time I installed a new SSSV prototype because even a small shift in the DPT sensor’s position would manifest in a different baseline value. However once I began the measurements, the minimum value was repeatable, and I could run the pump from the minimum to the maximum flowrate and back to the minimum and achieve the same baseline value. This gave me confidence that a simple offset-correction would be a suitable solution.
4.2 SSSV & VFM Prototype Evaluation

4.2.1 SSSV Prototype Evaluation

The principle reason for evaluating the SSSV prototypes was to determine whether any output switching occurred. To aid in the valve designs, I used the ‘SSSV Pressure Loss Model’ python script into which I input a predetermined range of flowrates (section 4.2.4). The results for this script formed the basis for the designs and gave me some idea of the valve performance.

After installing a given valve prototype, I left the VFM outputs disconnected and used the CSM to measure the fluid exiting each output individually, as before. Then I added the VFM outputs back into the loop and collected data from the DPTs over the same flowrate range. I repeated this process for all SSSV prototypes.

4.2.2 SSSV Prototypes

I tested seven valve prototypes, each differing in some significant, but often subtle way from the previous iteration. The pressure loss in each output was a function of the designed-in geometry of the outputs. For instance, the pressure loss due to friction in the serpentine path could be increased or decreased by adjusting the length of the straight sections. The pressure loss in the orifice output can be adjusted by changing the diameter or length of the orifice.

In the early prototypes, testing showed that the serpentine path consistently had a higher fluid output. I attempted to reverse this behavior such that the orifice became the preferred path. My approach was to increase both the length and number of turns in the serpentine path while keeping the orifice measurements constant. Even this was not effective. After discussing this with Dr. Banuti, he suggested that I drill the orifice larger in small increments until the behavior switched. This plan worked well and demonstrated that the pressure loss mechanism of the orifice was very sensitive to the orifice diameter. By contrast, adjusting the serpentine geometry resulted in smaller relative changes in the pressure loss.
I evaluated each prototype according to the same procedure outlined in the methods chapter. As mentioned before, I first disconnected the SSSV from the VFMs and took measurements directly from the valve outputs.

I will introduce three note-worthy valve prototypes here and examine each more closely in the results chapter.

SSSV Prototypes:

1. V0.54 - A valve with the serpentine output comprising 4 turns with an orifice diameter equal to 2.55mm and length equal to 2.83mm. (figure 5.5)

2. V0.57 - The serpentine path is 1.31 times longer than V0.54 and with identical orifice design. (figure 5.12)

3. V0.56 - The serpentine path is 1.62 times longer than V0.54 and with identical orifice design. (figure 5.9)

Although distinct from one another, each of the valve prototypes share certain design details. For instance, all valves have an inlet flow diameter, equal to 4.416mm, which matches the steel tubing (measured with vernier calipers). At the valve entrance, the prototypes all split from the inlet into the two output sections at a 37.7° angle from the axis of the flow. This effectively doubles the cross-sectional flow area and thus reduces the flowrate by a factor of 2 (see equation 3.1). In the orifice output, the orifice entrance is preceded by a 53mm long path equivalent to 12 diameters (52.992mm/4.416mm = 12). The orifice diameter varied according to the prototype version, however the orifice length remained fixed at 2.83mm.

The serpentine path comprised a combination of straight sections, with 4.416mm diameter, and 180-degree bends. The number of bends was always an even number and the straight sections were always odd. This design scheme ensured that the outputs pointed in the same direction. The bends maintained the 4.416mm diameter and the bend radius was also 4.416mm (see figure 2.22).

When measuring fluid output, I oriented the valve such that the orifice output was on the left and the serpentine on the right.
4.2.3 VFM Prototype Evaluation

The purpose of evaluating the the VFMs was two-fold. First, I could take the raw differential pressure data acquired from the DPTs and compare to the values estimated according to the procedure in the methods chapter. Second, I could compare the known input flow rate with the flowrate calculated using equation 3.3.

With all the printed parts in the study, I anticipated that the empirical data would yield greater pressure losses than my estimates. In the first place, equation 3.3 is derived from the Bernoulli equation, which neglects viscous (frictional forces). Also, it is always the case that 3D printed parts are not exactly as you designed them with certain geometries becoming smaller, edges becoming rounded and surface finish being uneven or inconsistent, sometimes imperceptible without a microscope. For liquid resin printers, there are likely many factors to consider, such as the ambient temperature of the room, the age and homogeneity of the resin, the laser calibration, the post-process procedure and the efficacy of the cleaning fluids. For all intents and purposes, the parts will come out very close to what we want, but it is reasonable to assume that the losses may be greater in a real-world, microfluidic application.

In the results chapter, I have indicated which valve prototype the data and plots correspond to.

I created three VFM designs which I labeled VFM-A, VFM-B, and VFM-C. After settling on a general shape for the Venturi tube, the only difference among these designs is the cross-sectional area of the throat. The designed throat diameter for VFM-A is 2.75mm, VFM-B is 3.75mm and VFM-C is 2.476mm. I began this study with only VFM-A and B, a small Venturi and a 'larger' Venturi, respectively. However, in testing at the lower flowrate range, VFM-B did not create pressure differentials large enough to be read reliably by the DPT. I attempted to remedy this, by creating VFM-C, which had a throat about $\frac{1}{3}$ smaller than VFM-B.
4.3 ERVFM Performance & Turn-down Ratio

The results from the testing outlined above should provide all necessary data to determine the ERVFM’s performance. The primary questions to answer are, 1. Does the SSSV provide any ‘switching’ capability, and 2. Does this allow the parallel VFMs to create a larger turn-down ratio than a standard VFM.
Chapter 5

Results

Preface

This chapter presents results from experiments and testing outlined in the methods. As mentioned before, the investigation of the ERVFM concept seeks to answer these questions: Does the SSSV work, and does this improve the overall turn-down ratio compared to a standard Venturi flow meter?

The results are structured in the following way:

1. Validation Testing
   i. KNF Pump
   ii. Omega Turbine Flowmeter
   iii. Flow Rate Entering the SSSV

2. SSSV & VFM Performance
   i. SSSV Results by Prototype
   ii. VFM & DPT Performance
      i. Theoretical Turn-down Ratio
      iii. Flow Visualization Using a High-Speed Camera
5.1 Validation Testing

As mentioned in the testing chapter, part of the testing focused on determining whether the components in the PFL were working as expected and what error, if any, I should consider when using the data elsewhere.

5.1.1 KNF Pump

The data-sheet for the KNF FP-150 pump gives a range of flowrate values when the pump is supplied with between 0 and 5 volts. I tested this using the cup and stopwatch method (CSM) as outlined in section 3.2. The results indicate that the claimed volumetric flowrate and the measured flowrate are fairly consistent (5.1).

![Figure 5.1: Comparison of volumetric flow rate vs. control voltage (0-5V) from the pump’s datasheet (L) and as-measured via the cup and stopwatch method (R)](image)

Point-for-point there are some noticeable differences, however, for the purpose of this study, the minimum and maximum values in the measured data are suitably similar to the data-sheet (maximum measured flowrate = 1.502 L/min).

5.1.2 Omega Turbine Flowmeter

The omega turbine flowmeter was the first flow element to follow the pump. I captured the flowrate data from the meter using the procedures outlined in sections 3.2.1 and 4.2.2. In figure 5.2, the data indicates a similar flow rate to those shown in figure 5.1,
however, an overlay of the plots reveals that the turbine meter read marginally higher values than the CSM pump measurements.

![Graph](image_url)

**Figure 5.2:** Volumetric flow rate output from the turbine flow meter (3-replicates) (L). Overlay of the turbine output and as-measured pump values using the cup and stopwatch method (R). The turbine meter measured slightly higher flow rates compared to the pump.

With a control voltage greater than approximately 1V, this offset is distinct and represents, at most, a difference of $60.4 \frac{mL}{min}$. I speculate the reason for this discrepancy is a possible error in the way the flow rate is calculated, and perhaps the number of pulses/volume are inaccurate.

### 5.1.3 Flow Rate Entering the SSSV

To complete the validation testing, I used the CSM (as outlined in section 4.2.3) to measure the volumetric flow rate entering the SSSV prototypes. I later used this data in the SSSV Pressure Loss Model (section 3.1.5).

Figure 5.3 shows the behavior of the data is similar to the previous validation plots. Once the working fluid enters the valve, it has passed through the turbine flow meter, two 90° bends and three sections of tubing. Figure 5.4 demonstrates that the pressure head provided by the pump (29PSI) is more than capable of maintaining a steady flow rate in spite of the minor losses created by these flow elements.

### 5.2 SSSV & VFM Performance

In sections 4.3.2 and 4.3.3, I explained the testing procedure for both the SSSV and VFMs. This current section includes results from both, and is organized by the SSSV
Figure 5.3: CSM-measured volumetric flow rate entering the solid-state selector valve. The shape is consistent with other flow rate validation plots and indicates the flow rate remains consistent up to this point in the pumped-fluid loop.

prototype version number. I tested numerous prototypes, however, the valve switching behavior was somewhat elusive due to the potent influence of the orifice diameter. In the past, I tested valves with too restrictive an orifice diameter, in which the serpentine path was always preferred, and other valves with too generous an orifice diameter, in which the opposite was true. These data were useful for my own learning, however, I will only include data from the most pertinent prototypes in this thesis. For clarity, I have formatted the following sections such that collected flow rate data for the SSSV prototype and the VFM prototypes are held within the same section.

As mentioned in the background section, when calculating major losses due to friction we can use either the 'empirical' Darcy-Weisbach (DW) equation or the analytically-derived Hagen-Poiseuille (HP); the former being applicable for both laminar and turbulent flows, and the latter only for fully-developed laminar flow in circular pipes. For each of the SSSV prototypes, I’ve included pressure loss estimations using both DW and HP calculations for comparison.
5.2.1 SSSV Prototype V0.54

This valve has an orifice diameter equal to 2.55mm and an orifice length equal to 2.83mm. It has 4 turns, and each of which is connected by a 52.992mm straight-section (designed length). I chose these dimensions based on the SSSV pressure loss model which showed a possible 'cross-over' or 'switching' point for the two outputs at approx. $0.22 \text{ l/min}$ using the DW equation and $0.32 \text{ l/min}$ via HP (Figure 5.6).

These estimates mirror the pressure loss predictions for each path; namely that the serpentine path should have a more linear shape and the orifice should be parabolic. The difference in the predicted switching point in figure 5.6 is caused by an upward shift in the serpentine plot, while the orifice plot remains the same. Curiously, even when calculated using DW, in which the $\Delta P$ is proportional to $V^2$, the plot remains fairly linear and has a distinct bend roughly in the middle of the flow rate range. The bend is less severe, but still remains when calculated using HP.

Moving on from the estimates, the flow rate data measured by the VFMs for each output show a much different result (figure 5.7):

There is now no distinct cross-over point in the plots. Please note that while figure
Figure 5.5: Drawing of SSSV V0.54 highlighting the orifice diameter and thickness and serpentine straight-section length. This overall design is similar to early prototypes but includes a larger orifice diameter. Units are mm.

5.6 plots volumetric flow rate vs. pressure loss, figure 5.7 plots the control voltage vs. the measured, volumetric flow rate. To reconcile this, I recognized that the pressure loss and volumetric flow rate are related insofar as the output with a higher pressure loss should have less volumetric output by comparison. Another way to think of this is the output with more "hydraulic resistance" will have less output. Although figure 5.7 has no switching point, it still shows that the orifice has a higher hydraulic resistance at higher flow rates, which is expected.

Finally, for SSSV V0.54, figure 5.8 shows the CSM-measured output (with the VFMs removed from the PFL).
Figure 5.6: Pressure loss estimates for SSSV V0.54: Darcy-Weisbach frictional losses (L), Hagen-Poiseuille frictional losses (R). Both plots indicated a cross-over point, and the Hagen-Poiseuille estimates push this point further up in flow rate range.

Figure 5.7: DPT-measured flow rates exiting the outputs of SSSV V0.54 and entering the VFMs. The plots show no distinct switching point in contrast to figure 5.6.

Figure 5.8: Cup and Stopwatch-measured flow rate measurements from both the serpentine and orifice outputs for SSSV V0.54. As distinct switching point is visible. Note these measurements were taken without VFMs attached.
Here, a distinct switching point is visible and demonstrates that one fundamental goal of this study, creating a functioning, solid-state valve, is possible. The end-behavior of each plot is consistent with figures 5.6 and 5.7, and the switching point of approximately \( \frac{0.33}{\text{min}} \) is consistent with the HP pressure loss estimates.

### 5.2.2 SSSV Prototype V0.56

After testing the performance of SSSV V0.54, I designed V0.56 and attempted to shift the observed cross-over point upward from \( 0.33 \frac{L}{\text{min}} \). In my previous testing, I learned that adjusting the length of the straight-sections in the serpentine path was the best way to fine-tune the pressure loss. Adjusting the diameter of the orifice was effective, but much too sensitive and difficult to control accurately. Adding bends to the serpentine path also created higher losses but in larger increments than were desirable (See Appendix XXX: SSSV V0.55). For SSSV V0.56, I chose to increase the lengths of each straight section until the pressure loss model showed a cross-over point higher in the flow rate range. I increased the length of all the serpentine straight-sections equally, as this minimized adjustments to the python model code. V0.56 had a total ’straight-section’ length 1.62 times longer than V0.54 (figure 5.9). Increasing the

![SSSV V0.56 Diagram](image)

**Figure 5.9:** Drawing of SSSV V0.56. The cumulative length of straight-sections in the serpentine path are 1.62 times longer than V0.54. Units are mm
length of the straight sections had no effect on the print process, however, more care was needed to ‘flush’ uncured resin from the longer channels during post-processing.

The pressure loss model plots for V0.56 again show a distinct difference between the DW and HP calculations (figure 5.10). According to the DW estimations, the end-

Figure 5.10: Pressure loss estimates for SSSV V0.56: Darcy-Weisbach frictional losses (L), Hagen-Poiseuille frictional losses (R). The Darcy-Weisbach plot indicates two distinct cross-over points for the outputs, while the Hagen-Poiseuille plot shows a cross-over point near upper flow rate range.

behavior for the plots agrees with those of V0.54, however it predicts two cross-over points; one at $0.33 \frac{1}{\text{min}}$ and another at $\approx 0.5 \frac{1}{\text{min}}$. The HP estimation places the cross-over point much higher in the flowrate range (perhaps $0.71-0.73 \frac{1}{\text{min}}$). Unfortunately, due to a technical error, I could not include the DPT-gathered data for V0.56, and I was unable to retest this prototype. The CSM-measured plot more closely agrees with the HP estimation above. In this case, the cross-over point occurred far up in the flow rate range, and it is unclear if it actually ‘switched’ outputs.

5.2.3 SSSV Prototype V0.57

Up to this point I found that V0.54 had a cross-over point too low in the flow rate range and V0.56 crossed over too high in the flow rate range. I attempted to split the difference and design a valve which switched somewhere in between. I utilized the same approach as before and adjusted the length of the straight-sections along the serpentine path. SSSV V0.57 was the result and includes straight-sections within the
Figure 5.11: Flow rate measurements from both the serpentine and orifice outputs for SSSV V0.56. The switching point exists high in the flow rate range and it is unclear if the switch actually happens and exactly where it happens. The plot is consistent with the shape of the HP-calculated pressure loss estimates in figure 5.10

serpentine path which are approximately 1.31 times longer than V0.54 (figure 5.11).

As before, the pressure-loss model plots show distinct differences. Like V0.56, the DW plot shows two cross-over points, $0.35 \frac{l}{min}$ and $0.55 \frac{l}{min}$. The HP plot predicts a cross-over point at approximately $0.62 \frac{l}{min}$ (figure 5.13) The VFM-collected data (figure 5.14) shows a similar behavior to figure 5.7. Again, although the pressure loss model predicts it, we see no cross-over point in the plot. Finally, the CSM-measured flow rate (figure 5.15) shows a cross-over point at roughly $0.43$ or $0.44 \frac{l}{min}$.

Figure 5.14: VFM-measured plots for SSSV V0.57. This plot shows no cross-over point in contrast to the model estimations in 5.13.
Figure 5.12: Design for SSSV V0.57. The length of straight-sections in the serpentine path are 1.313 times longer than V0.54, Units are mm

This falls well below the predictions in the pressure loss model, however, it does demonstrate that adjusting the length of the serpentine path is an effective and repeatable way to change the cross-over point.

5.2.4 Theoretical Turn-Down Ratio

The turn-down ratio of a flow meter refers to the ratio of the highest and lowest values perceived by the meter[24]. In the ERVFM, the DPT senses a differential pressure across the pressure taps on each VFM. The raw values from the sensor are actually voltages, and I used the process described in the methods section to map the values to differential pressures (in Pascals). Due to the parallel flow configuration of the Venturi tubes, we should expect the theoretical turn down ratio to be higher than a VFM used alone.

5.2.5 SSSV V0.54 - Theoretical Turn Down Ratio

In figure 5.7 I plotted the differential pressure sensed by each DPT (for VFM-A & VFM-C) relative to the control voltage. Figure 5.16 demonstrates the physical
Figure 5.13: Pressure loss estimates for SSSV V0.57: Darcy-Weisbach frictional losses (L), Hagen-Poiseuille frictional losses (R). The slightly shorter straight-sections in this prototype move the estimated cross-over point down in the flow rate range. The Darcy-Weisbach estimates again show two distinct cross-over points.

Table 5.1: **VFM-Measured Flow Rates - SSSV V0.54**

<table>
<thead>
<tr>
<th>VFM-A (Serpentine)</th>
<th>VFM-C (Orifice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0710</td>
<td>0.0569</td>
</tr>
<tr>
<td>0.101</td>
<td>0.075</td>
</tr>
<tr>
<td>0.175</td>
<td>0.115</td>
</tr>
<tr>
<td>0.223</td>
<td>0.171</td>
</tr>
<tr>
<td>0.283</td>
<td>0.208</td>
</tr>
<tr>
<td>0.341</td>
<td>0.254</td>
</tr>
<tr>
<td>0.406</td>
<td>0.305</td>
</tr>
<tr>
<td>0.463</td>
<td>0.357</td>
</tr>
<tr>
<td>0.521</td>
<td>0.407</td>
</tr>
<tr>
<td>0.581</td>
<td>0.449</td>
</tr>
<tr>
<td>0.641</td>
<td>0.497</td>
</tr>
<tr>
<td>0.695</td>
<td>0.550</td>
</tr>
<tr>
<td>0.741</td>
<td>0.600</td>
</tr>
<tr>
<td><strong>0.777</strong></td>
<td>0.628</td>
</tr>
</tbody>
</table>

Table 7.1 indicates the high and low values. In this case the theoretical turn down ratio is \(\frac{0.777}{0.0569}\), and equal to approximately a 1:13.
Figure 5.15: Cup and stopwatch-measured flow rates for SSSV V0.57. The switching point at \( \approx 0.44 \frac{l}{min} \) is slightly lower than the HP-calculated pressure loss estimates in figure 5.10, but confirms that adjusting the length of the straight-sections in the serpentine path is an effective method for manipulating the cross-over point.

5.2.6 SSSV V0.57 - Theoretical Turn Down Ratio

For prototype V0.57, the result is similar.

Figure 5.17: SSSV V0.57 shows close similarities to the data for SSSV V0.54. The lowest flow rate reading appears in the orifice output and the highest in the serpentine output.
Figure 5.16: Flow rate output from the DPTs attached to VFM-A and VFM-C. There is a clear split in the plots at lower and upper ends of the flow rate range, indicating no cross-over point and showing that the serpentine path maintains higher flow rates throughout the test.

Table 5.2: **VFM-Measured Flow Rates - SSSV V0.57**

<table>
<thead>
<tr>
<th>VFM-A (Serpentine)</th>
<th>VFM-C (Orifice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0694</td>
<td><strong>0.0361</strong></td>
</tr>
<tr>
<td>0.117</td>
<td>0.0542</td>
</tr>
<tr>
<td>0.183</td>
<td>0.0927</td>
</tr>
<tr>
<td>0.216</td>
<td>0.134</td>
</tr>
<tr>
<td>0.265</td>
<td>0.173</td>
</tr>
<tr>
<td>0.323</td>
<td>0.223</td>
</tr>
<tr>
<td>0.381</td>
<td>0.278</td>
</tr>
<tr>
<td>0.434</td>
<td>0.332</td>
</tr>
<tr>
<td>0.498</td>
<td>0.384</td>
</tr>
<tr>
<td>0.557</td>
<td>0.439</td>
</tr>
<tr>
<td>0.607</td>
<td>0.494</td>
</tr>
<tr>
<td>0.670</td>
<td>0.533</td>
</tr>
<tr>
<td>0.718</td>
<td>0.575</td>
</tr>
<tr>
<td><strong>0.765</strong></td>
<td>0.608</td>
</tr>
</tbody>
</table>
When considering the data itself (Table 7.2), the theoretical turn down ratio is $\frac{0.765}{0.0361}$, or approximately 1:21.

5.3 Flow Visualization Using High-Speed Camera

The CSM-measured data for each of the prototypes above indicated that a cross-over point somewhere in the flow rate range. To further test this result, I conducted a flow visualization experiment in which I set the pump to a predetermined flow rate and then bled small bubbles ahead of the valve. My hope was that more bubbles would pass into the output with lower hydraulic resistance. I then raised the flow rate and repeated the observations.

Following the post-processing steps in the methods section, I improved the translucency of SSSV prototype V0.57. Upstream from the valve, I installed a printed 'Wayne Injector Coupler' (WIC) with a cylindrical through-hole and a thick, silicone diaphragm on the top face. I punctured the diaphragm with a 22-gauge, blunt-tip needle and fed bubbles into the flow path with an attached 10 milliliter syringe.

Figure 5.18: SSSV V0.57 installed in the PFL after sanding and coating with Krylon Acrylic Clear Coat
The WIC and syringe setup worked well for injecting small bubbles into the loop, however, even at low flow rates, the bubbles traveled too quickly to determine whether they preferred one output or the other.

To remedy this, Dr. Wayne let me borrow a high-speed camera (Phantom). I mounted the camera to a standard tripod above the SSSV and backlit the valve with a LED light-sheet. I then set the pump to output at approximately $0.2 \frac{l}{min}$ and captured a video at 2000 frames per second and then repeated this at $1.0 \frac{l}{min}$. The resulting videos (still images here) provide more evidence that the output switching occurred with this prototype. By observing the videos, it is clear the orifice output is preferred at the lower flow rate, and the serpentine, at the higher flow rate (figure 5.20). I counted the number of bubbles passing into each output at each flow rate to confirm this. For the $0.2 \frac{l}{min}$ flow rate, a total of 3 and 8 bubbles passed into the serpentine and orifice outputs over a 10-second period, respectively. For the $1.0 \frac{l}{min}$ flow rate, I counted 45 and 36. I recognize that this test is, at best, only a qualitative means for evaluating the valve function. There are likely other, uncontrolled factors that may influence a bubble to pass into one output or another.
Figure 5.20: Still images from the high-speed bubble flow visualization experiment. As expected, the $0.2\frac{L}{min}$ flow rate showed more bubbles passing into the orifice output (L), and the $1.0\frac{L}{min}$ showed more bubble passing into the serpentine output (R).
Chapter 6

Discussion

Preface

This chapter seeks to interpret the results from the previous chapter. I do not present new results here, however, I do provide details regarding the testing procedures, data collection, and other items, as needed, for context and further explanation.

6.1 Validation Testing Results

As 'low-tech' as the cup-and-stopwatch method (CSM) sounds, it actually provided great data which helped build my confidence as I tested components in the PFL. For instance, the pump flowrate measurements were very close to the company’s published data. In nearly all cases, the CSM measurements usually did not vary more than 0.5 - 1 \( \frac{g}{sec} \); equivalent to 30 - 60 \( \frac{mL}{min} \), and the upper end of the variance typically happened at the higher flow rates. My method involved holding the cup beneath the water for a set amount of time (5 seconds). So I expected that on some measurements, I would be more slow to remove the cup, and in others I would be faster. Regardless, it seemed to be consistent and reliable enough way to determine the actual mass flow rate at the different points in the loop. Also, I did not include data for this, however, I tested the pump in multiple experiments over multiple days and found that the motor speed was consistent.

The turbine meter measured flow rates slightly higher than pump measurements. I would have expected the opposite, so I am curious if my Arduino sketch influenced this
in some way. I adapted this code from another Arduino project [17] which includes a provision to prevent the hall effect sensor from triggering twice as the turbine makes a full rotation. That snippet of code seems to work, otherwise the end result would have deviated much more. However, without knowing more about how the Omega meter senses the rotation of the turbine, it is difficult to speculate further than that. I should note that the meter was at least 10 years old, and I do not know its history. The meters come tested from the manufacturer with a set \( \frac{\text{pulses}}{\text{unit-volume}} \) calibration, and it is possible this number has changed slightly in the intervening years. Measuring the mass flow rate via the CSM as the fluid entered the SSSV showed that the flowrate had changed very little, even after passing through two 90-degree SwageLok bends and the turbine meter. I think this demonstrates that while the minor losses may be significant in sections of the loop, the pressure head provided by the pump was sufficiently high to negate those effects. I also tested a peristaltic pump and a model FP-70 KNF pump. I retired the perisaltic pump because it created large pulsations in the flow. The FP-70 worked well until it began intermittently slowing down and speeding up during tests. I owe Dr. Wayne many thanks for letting me borrow the FP-150 so that I could repeat and finish my testing. Regarding the PFL, I am generally satisfied with the structure of the loop. The initial construction was laborious, however once the basic loop was complete, its modular components made it very easy to reconfigure and swap parts for testing. For the tubing itself, I had only a few leak issues (both air coming into the flow and water leaking out), but they were easily fixed by tightening the fittings. For the SwageLok fittings specifically, the polymer (nylon) ferrules were very handy as I could tighten them as much as I liked and did not have to worry about deforming the steel tubes (I am unsure about the pressure-rating and fluid compatibility for other applications, however). Printing the SSSV and VFM prototypes had advantages mentioned in the methods chapter. Even when adding more features, for instance, more bends to the SSSV, I was limited more by the available print volume than I was the complexity of the design. I encountered several artifacts such as ribbing, and malformed bends (figure 6.2). Although, these
issues were not frequent, I attempted to avoid them by printing only on specific areas of the build plate which I was certain produced good results. The rippling might be caused by misalignments as the build platform lifts and re-positions in between layers (sometimes called the 'peel'). This process allows resin to flow back under the part for the next layer. It may also be that printing some parts vertically with minimal supports did not provide enough stability throughout the print process. The malformed bend might be caused by the print sagging, or otherwise deforming due to shrinkage, or the peel process pulling on a thin, uncured section of the print. This became less of an issue (by accident) as I started printing the SSSV prototypes at a 45-degree angle. Installing and removing the prototypes from the PFL became a routine process. The tubes could usually be inserted by-hand with little force. Sealing the tubes with an in-place, silicone gasket was necessary (figure 6.3); without this step, air would easily leak into the flow path. When I needed to remove a part and reuse the tubes, I used a vise to grip the tube end while I pulled the printed part off. Often, silicone residue remained on the metal tube, which I removed with a knife. Sometimes
the gasket would remain inside the printed part, and I used a sharp pick-type tool to fish them out. This experience showed me that, after curing, the RTV silicone released fairly easily from the print. I later used this method with a 3D printed mold to cast a high-temp part for another project. With that said, I cannot guarantee the longevity of this method. I noticed over time that the tube gaskets developed air pockets as the gasket pulled away from the print slightly (no photos of this). This did not cause any issues, and while it worked well for my purpose, I question the durability and pressure tolerance of this method for other applications.

6.2 SSSV and VFM Performance

The previous results for the SSSV prototypes demonstrated that the SSSV concept is valid. The 'switching' capability of the included prototypes was both repeatable and manipulable. However, it is clear that the switching phenomenon, at least with my test setup, is not robust. The most compelling piece of evidence for this is the DPT-acquired data from the VFM. Among all the presented prototypes, this data suggests no switching point at all.

I see two possible explanations for this:

1. The DPTs produced unreliable data (less likely)
Figure 6.3: Photo of the tube sealing process with RTV silicone (T) and a detail-drawing of the designed-in gasket channels (B).

2. The fractional flow between outputs is easily influenced (more likely)

In the first case, as mentioned before, I was unable to achieve a perfect ('zero') baseline measurement for either DPT. In theory, an equal gauge pressure present at both inputs on the DPT would result in a differential pressure equal to zero. If the DPT was elevated on either side, from the horizontal, this would still register as a voltage. In practice, no amount of leveling was effective in reaching the 'zero-voltage' specified on the data sheet. Instead, I elected to capture the baseline measurement in each set of data (voltage with pump off) and subtract it from all other measurements as a simple offset correction. Figure 6.4 shows the difference between the original data and offset-corrected data. One issue with this is that it assumes a linear behavior for the baseline value across the entire measurement range. This may be a poor assumption to make, and perhaps a thorough calibration of each DPT with a trusted flow meter would have cleared this up.
Figure 6.4: Flow rates calculated from the DPT-acquired data (for SSSV V0.57) with and without the offset-correction applied. The corrected flow rates were lower by as much as $0.3 \frac{l}{min}$ for the orifice output, and $0.1 \frac{l}{min}$ for the serpentine output.

On the other hand, even the offset-corrected data itself is consistent with the results of the validation tests. For instance, the plots are all fairly linear, which mirrors the shape of the original pump and turbine meter flow rate tests. These plots also have the same end behavior as the CSM-measured data for all prototypes; the serpentine path always had the greater fraction of flow at the upper-end of the flow rate range. Also, the VFM-calculated flow rates fall in line fairly well with all CSM-measured data. This suggests that there are not significant issues with acquiring and processing the raw voltage data. While the suitability of the offset-correction may be in question, the above reasons convince me that the discrepancy presented in the VFM plots may lie with explanation number 2.

As a reminder, using the plots for prototype V0.57 as examples (figure 5.15), the CSM data were measured with the Venturi tubes removed from the loop. This means the loop effectively ended at the valve outputs (where the water flowed into a bucket (figure 6.6)). By contrast, the VFM data was measured with the Venturi tubes installed in their intended place, immediately following the valve.

When considering the end behavior of the plots (figure 6.5), they both show the serpentine with a greater fraction of the flow. Without a distinct cross-over point, the VFM plot shows the serpentine path with a higher fraction of flow across the entire
Figure 6.5: Side-by-side comparison of the CSM and VFM-acquired data for SSSV V0.57. The main experimental difference between the two is the CSM data was measured with the VFMs removed from the loop. This likely reduced the upstream effects of the VFM’s and allowed the SSSV to maintain its cross-over point.

range. V0.54 shows the same result. This leads me to believe that installing the VFM on the orifice output increases the hydraulic resistance enough to force most of the flow through the serpentine path.

For testing, I attached VFM-C to the orifice output and VFM-A to the serpentine output. I knew, based on my Venturi tube pressure loss model, that the expected differential pressure would not exceed the DPT sensor range, however I had not considered that the differential pressure produced at the taps could be (too low) for the DPT to reliably discern. This was the case with an early prototype Venturi tube called VFM-B. For clarity, VFM-C has the smallest throat diameter, VFM-A has the next smallest, and VFM-B had the largest. I originally attached VFM-B to the orifice output and found the differential pressure too small to be reliable, and I replaced it with VFM-C as a remedy. This made the differential pressures larger, but with its very small diameter (2.476mm designed), I suspect it creates too much resistance, especially at low flow rates.

6.3 Theoretical Turn-Down Ratio

When designing a VFM, it is important to consider the flow rate range over which we want to measure, and then design the Venturi to suit this range. This is most directly accomplished by adjusting the diameter of the throat section relative to the inlet flow.
As I mentioned with my VFM-B prototype, it is possible for pressure differentials to be so small that slower flow rates cannot be reliably sensed by the meter. I opted to replace VFM-B with the much smaller VFM-C to induce larger pressure differentials and hopefully create more trustworthy data for low flow rates.

The 1:13 turn-down ratio for V0.54 is reasonably close to a standard VFM. The 1:21 ratio for V0.57 should raise some eyebrows. Looking back at the data for V0.57, the lowest flow rate sensed by VFM-C corresponds to a differential pressure of $\approx 132 \text{ Pa}$. This is quite close to the 70 Pa calculated resolution of the arduino, which indicates it may be unreliable. The next data point above that resolution threshold is $\approx 206 \text{ Pa}$. 

---

Figure 6.6: Photo of the fluid exiting the SSSV outputs during CSM flow rate measurements. During the tests, I captured the working fluid in bucket and periodically reintroduced it to the reservoir.
Recalculating the turn-down ratio with the corresponding flow rate gives a more reasonable, 1:14.

To improve this turn-down ratio, one option is to use a DPT and data acquisition solution capable of detecting even lower flow rates. Another option is to use a pump with a wider flow rate range. The KNF FP-150 has an upper limit of $1.5 \frac{l}{min}$. Increasing this would make the ratio larger, so long as the DPT $\Delta P$ limit is not exceeded, and all detected flow rates are trustworthy.

The success of the ERVFM lies both with the switching capability of the SSSV, and the increased range of the VFMs. VFM-C caused an issue with the overall function of the assembly, and it is clear that further investigation is needed to make the switching more resilient to downstream elements.

6.4 Flow Visualization

I pushed to include a flow visualization experiment because I wanted some additional verification of the switching phenomenon. I eventually settled on the bubble injection after trying several other methods. I hoped that a simple dye streak injection would work, however, even at the lowest flow rate, the dye immediately diffused into the surrounding fluid. I also tried mineral oil and glycerol (with and without dye), but there was not enough contrast to see the injected fluid. I also inserted a piece of thread in the flow at the junction between the orifice and serpentine entrances, in the hope that the thread would bias to one side or the other as I adjusted the flow rate. This too did not work.

Injecting bubbles proved to be simple enough for a visual, and the bubble count provides some extra evidence that the switching is occurring. However, the visual should likely not be taken as conclusive evidence; I was unable to control both the size and speed with which the bubbles exited the syringe and I am unsure how the bubbles passing through the orifice might influence the upstream bubbles.

The bubble visual also demonstrated that the junction between each output cause flow disturbances. At the low flow rate, a bubble gets stuck every now and then at the tip
of the junction, but eventually finds its way to either output. At higher flowrates, the junction shears any bubbles it interacts with and then causes quite severe vortices to form on either side. Minor losses at the junction is something I had not considered at all when I first designed the valve, and is probably not something I would have noticed without the high-speed camera. It may not have a large effect on the pressure loss of the outputs, but is a large considerations for the (overall) pressure loss of the valve. I recognize now that a CFD model may have likely tipped me off to this much earlier.
Chapter 7

Conclusion

In this thesis I presented a design for an Expanded-Range Venturi Flow Meter (ERVFM). The flow meter comprises a novel, solid-state selector valve (SSSV) and two Venturi flow meters arranged in a parallel configuration. This work covers the theory, methods, testing and results for the ERVFM concept.

The ERVFM concept relies on the relative hydraulic resistance in each of its outputs to preferentially divert the working fluid to either flow meter. Based on this definition, I demonstrated that the concept is valid, however, I have several final thoughts about its performance, the testing procedure, and future work.

The SSSV appears to function correctly and the ‘switching’ phenomenon is tunable across prototypes. However, the CSM-plots in the results chapter show that the switching behavior happens very gradually. This means that, although the fluid output undoubtedly changes dominance, the flow rates measured at each output are very close. This results in a switching point that is easily disturbed, even by adding minimal resistance downstream from the valve. Perhaps there are different SSSV designs that could make the switching behavior more robust and create more reliable flow rate readings at the differential pressure transducers (DPTs). As it is, I cannot currently conceive of a design that would allow this, but I am confident that it can be done, especially when leveraging the flexibility of 3D printing for both rapid-prototyping and creating complex designs. It is probable that an improved valve design would look very different from my prototypes.

The python-based pressure-loss models provided useful guidance but did not ac-
accurately predict the valve’s function. After completing the flow-visualization experiments and seeing the induced turbulence within the valve, I realized that there are likely many sources of unaccounted pressure loss in the prototypes. Future work would benefit from using simulation and experiments to characterize the pressure loss, not only in the outputs, but throughout the valve.

I was generally satisfied with the Arduino data-acquisition setup, however, especially at lower flow rates, a more precise solution is likely needed. The Venturi flow meter (VFM) data showed me that the Arduino’s resolution may be limiting factor.

With the benefit of hindsight, I would focus my approach more on understanding how the minute changes in geometry affected the system’s performance. By the end, my multiple prototypes allowed me to grow a bit of ’experimental-intuition’ to the point where I could confidently design a part and be very sure it would work. However, the flow visualization was an eye-opener, and I am certain I could have realized this a lot sooner with a computational fluid dynamics simulation.

Going forward, I believe the ERVFM and SSSV concepts have promise, and in the years to come, I am hopeful that the results and knowledge gained will help inform my future learning.
Appendix A

Python Pressure Loss Models & Arduino Sketches for Sensors

A.1 Venturi Tube Pressure Loss Model

Python (Jupyter Notebook)

```
# This notebook calculates the pressure difference between the ports on a,
# venturi tube, when given:
# 1. Flowrate (m^3/s)
# 2. Cross-sectional areas for the venturi’s geometry
# 3. Fluid density (kg/m^3)
# Output from Previous function should be Pascals (Pa)

# Import needed libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# create an array of flowrate you want to loop through
fr_vfn_a = np.linspace(0, 700, 100) # Linearly spaced array of values. Last parameter determines number of points. Increase for a smoothed plot
vol_fr_vfn_a = [0, 0.074352, 0.100344, 0.144872, 0.187368, 0.2406, 0.28464, 0.337272, 0.380064, 0.423864, 0.474168, 0.526176, 0.564984, 0.601488, 0.643684] # These measurements were determined empirically from direct bucket and stopwatch

vol_fr_vfn_a =[]
for i in fr_vfn_a:
    volfr_vfn_a = i*1000
    vol_fr_vfn_a.append(volfr_vfn_a)
```

Venturi_Tube_Plots
```
fr_vfm_c = (0.0.12516, 0.176664, 0.226968, 0.272976, 0.323808, 0.377808, 0.43464, 0.49776, 0.556104, 0.609192, 0.654672, 0.711792, 0.75102, 0.789648)  # These are flowrates determined empirically from direct bucket and stopwatch, measurements (L/min)
vol_fr_vfm_c = []
for i in fr_vfm_c:
    volFr_vfm_c = i*1000
    volFr_vfm_c.append(volFr_vfm_c)

[36]: # Converts a ml/min flowrate to m^3/s - Intuitive units, but needed in this form for pressure drop in Pascals 
     # def flowrate (vol_fr):
     #     Q = []
     #     for n in vol_fr:
     #         q = n*(1.0e-6)/60.0
     #         Q.append(q)
     #     return Q

[37]: # Calculates the pressure difference given the physical parameters def PDiff (A1, A2, Q, rho):
     dp = []
     for i in Q:
         DP = (((i^2) rho*(1-(A2/A1)^2))/(2*A2^2))
     dp.append(dp)
     return dp

[38]: # Calculate the flowrate (m^3/s) for VFM-A specifically

Q_vfm_a = flowrate(vol_fr_vfm_a)
# print(Q_vfm_a)

[39]: # Calculations for VFM-A

A1 = 1.66e-5  # Area for flow inlet cross-section (m^2) Radius = 2.208mm = 0.002208mm radius.
A2 = 5.52e-6  # Area for tube throat cross-section (m^2) Radius = 1.375mm = 0.001375mm radius.

rho = 998.21  # Density of water at 20C, kg/m^3

dp_vfm_a = PDiff(A1, A2, Q_vfm_a, rho)
# print(dp_vfm_a)
```
\[40\]: Calculate the flowrate (m\(^3\)/s) for VFM-C specifically

\[41\]: Calculations for VFM-C (smallest throat venturi)

\[42\]:

\[43\]: # Calculations for VFM-C (smallest throat venturi)

\[44\]: # Calculate the flowrate (m^3/s) for VFM-C specifically

\[45\]: # Convert a m\(^3\)/s flowrate to m\(^3\)/min
return Q

# Calculates the pressure difference given the physical parameters
def Pdiff (A1, A2, Q, rho):
    dp = []
    for i in Q:
        DP = ((1+2)*rho*(1-(A2/A1)**2)/(2*A2**2))
        dp.append(DP)
    return dp

Q = flowrate(vol_fr)
# print(Q)

# Calculations for 'small' venturi
A1 = 1.534107566e-5 # area for large cross-section (m^2) ! THIS IS FIXED DUE TO, --DIAMETER OF INCOMING FLOW!!
# A2 = 4.908738521e-6 # OD = 2.5mm
A2 = 5.939573611e-6 # OD = 2.75mm
# A2 = 7.055394939e-6 # area for small cross-section (m^2) # OD = 3.00mm
# A2 = 8.295768101e-6 # OD = 3.25
# A2 = 9.6511275e-6 # OD = 3.50mm
# A2 = 1.104466167e-5 # OD = 3.75mm
# A2 = 1.256637061e-5 # OD = 4.00mm

rho = 1000.0 # density for water (kg/m^3)

dp_small = Pdiff(A1, A2, Q, rho)
# print(dp_small)

# Calculations for 'large' venturi
A1 = 1.534107566e-5 # area for large cross-section (m^2) ! THIS IS FIXED DUE TO, --DIAMETER OF INCOMING FLOW!!
A2 = 1.104466167e-5 # OD = 3.75mm

rho = 1000.0 # density for water (kg/m^3)

dp_large = Pdiff(A1, A2, Q, rho)
# print(dp_large)

# print(len(wol_fr))
# print(dp_small)
plt.plot(wcl_fr, dp_small, color="orange")
plt.title("HDPV Venturi (Larger Estimated Differential Pressure Range)")
plt.xlabel("Flowrate (ml/min)")
plt.ylabel("Differential Pressure (Pa)")
plt.show()

plt.plot(wcl_fr, dp_large, color="blue")
plt.title("HDPV Venturi (Smaller Estimated Differential Pressure Range)")
plt.xlabel("Flowrate (ml/min)")
plt.ylabel("Differential Pressure (Pa)")
plt.show()

plt.plot(wcl_fr, dp_small, color="orange")
plt.plot(wcl_fr, dp_large, color="blue")
"LDPV" Venturi
(Smaller Estimated Differential Pressure Range)

Flowrate (mL/min)

Differential Pressure (Pa)

[52]: [matplotlib.lines.Line2D at 0x3114a6480]
# Using measurements from the PXI19-001DWSV Differential Pressure Sensor, calculate the flowrate

dp_small = [0, 627.3868456, 1171.061196, 2762.484719, 6245.080555]

# Calculates the pressure difference given the physical parameters
def flowrate (A1, A2, dp, rho):
    fr = []
    for i in dp:
        FR = np.sqrt(((i*(2*A2**2))/(rho*(1-(A2/A1)**2))))
        fr.append(FR)
    return fr

A1 = 1.534107564e-5 # area for large cross-section (m^2) !!THIS IS FIXED DUE TO DIAMETER OF INCOMING FLOW!!
A2 = 5.59957361e-6 # OD = 2.75mm

small_venturi_fr = flowrate(A1, A2, dp_small, rho)
# print((small_venturi_fr))

# Converts a m^3/s flowrate to mL/min
def flowrate_conv (vol_fr):
    Q = []
    for n in vol_fr:
        q = n*60000000.0
        Q.append(q)
    return Q

Q_small = flowrate_conv(small_venturi_fr)
# print(Q_small)

dp_large = [0, 132.9670555, 233.1845502, 267.8926031, 677.3232544]

A1 = 1.534107564e-5 # area for large cross-section (m^2) !!THIS IS FIXED DUE TO DIAMETER OF INCOMING FLOW!!
A2 = 1.104466167e-5 # OD = 3.75mm

large_venturi_fr = flowrate(A1, A2, dp_large, rho)
# print((large_venturi_fr))

Q_large = flowrate_conv(large_venturi_fr)

print(Q_large)

[0.0, 492.3895471053229, 652.0586671584942, 685.8679253726516, 1111.309001117286]

# Calculate venturi flowmeter loss coefficient
# def h_loss (d, d_t, alpha_I, alpha_D, f_I, f_D, f_T):

## No longer needed, but keeping just in case

# # Calculations for low-dp venturi

# A1 = 1.46e-5 # Area for large cross-section (m^2) Radius = 2.208mm - 0.05mm
# (Estimated over-cure) = 2.168mm radius.
# A2 = 1.05e-5 # Area for small cross-section (m^2) Radius = 1.875mm - 0.05mm
# (Estimated over-cure) = 1.825mm radius.
# rho = 998.21 # Density of water at 20C, kg/m^3
# dp_large = Pdiff(A1, A2, Q, rho)
# print(dp_large)
A.2 SSSV Pressure Loss Model (For SSSV V0.57)  

Python (Jupyter Notebook)

```
# This document calculates the pressure loss, in Pascals, for the two outputs in the Solid State Selector Valve (SSSV) V0.57

# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import itertools

# Specific properties for SSSV Version V0.56
num_ss_serp = 5 # Number of straight-sections in the serpentine path
num_bend_serp = 4 # Number of bends in the serpentine path
num_ss_orif = 1 # Number of straight-sections preceding the orifice

# Physical properties for water at 20C
rho = 998.21 # Density of water at 20C, kg/m^3
nu = 1.0023E-6 # Kinematic viscosity of water at 20C, m^2/s
nu = 0.0010005 # Dynamic viscosity of water at 20C, N*s/m^2

# Characteristics of the water flowing into the SSSV
Q_mass_input = (1, 3.17, 4.70, 6.33, 7.72, 9.63, 11.01, 13.33, 14.91, 16.71, 17.
2, 19.76, 21.33, 23.48, 24.29) # Mass flowrate at the SSSV input, g/s
These flowrates were measured by bucket & stopwatch on 03/01-02/22 by JNP

# Q_mass_input = np.linspace(1, 100, 50)
```
# Q_mass = (1, 3.17, 4.70, 6.33, 7.72, 9.63, 11.01, 13.33, 14.91, 16.71, 17.82, ...# Mass flowrate at the SSSV input, g/s. These flowrates were measured by bucket # stopwatch on 03/01-02/22 by JMN

Q_mass = []

for i in Q_mass_input:
    Q = i/2
    Q_mass.append(Q)

[46]: # Volumetric Flowrate Added so that I can calculate the pressure drop according to H-P

Q_vol_ls = (.1, 0.19, 0.28, 0.38, 0.46, 0.56, 0.66, 0.80, 0.89, 1.00, 1.07, 1.19, 1.26, 1.41, 1.49) # L/min

Q_vol_ls= (.00100, 0.00317, 0.00470, 0.00633, 0.00772, 0.00963, 0.01101, 0.01333, 0.01491, 0.01671, 0.01782, 0.01976, 0.02133, 0.02348, 0.02429) # L/sec

Q_vol = []

for i in Q_vol_ls:
    QV = i*0.001 # Converts Q_vol_ls to have units: m^3/s
    Q_vol.append(QV)

[47]: # Physical characteristics of SSSV & SSSV entrance

epsilon = 1.71E-6 # Average surface roughness height (‘ra’), meters - Value borrowed from Arnold, et al. 2019. # I'm a bit suspicious of this surface roughness value. Might be good to find another paper or two to confirm this value.

D = 0.00416 # Hydraulic diameter of SSSV entrance path, meters

[48]: # Physical length (flow length) of different parts of the SSSV

# Note that the serpentine path is made of a series of straight and bend sections

L_ss = 0.071192 # Average length of straight section, meters
L_bend = 0.013873 # Length of centerline of one bend (SSSV version 0.52), meters
r_bend = 0.00416 # radius of bend

theta = 180 # Bend angle in degrees
alpha = np.pi/2 # Bend angle in radians

# Physical characteristics of the orifice section
L_orif = 0.00283 # thickness of orifice (length of orifice), meters
# L_orif = 0.0035 # thickness of orifice (length of orifice), meters

D_orif = 0.00255 # orifice diameter, meters
# D_orif = 0.00278 # orifice diameter for SSSV V0.52 AFTER drilling to 7/64", L
# units are meters
# D_orif = 0.0021 # Orifice diameter for SSSV V0.52

l_d_ratio = L_orif/D_orif # Ratio of the orifice length (thickness) and the
# orifice diameter

beta = D_orif/D # Ratio of orifice diameter to incoming flow diameter

# Mass flowrate to bulk velocity conversion, m/s

def QtoVbar(Q_mass, D, rho):
    rho_gran = rho*1000 # converts kg/m^3 to g/m^3
    A = np.pi*(D/2)**2 # calculates the cross sectional area of flow, m^2
    Vbar = Q_mass/(rho_gran*A) # Calculates the bulk velocity, m/s
    return Vbar

Vbar = [] # Units: m/s

for i in Q_mass:
    V = QtoVbar(i, D, rho)
    Vbar.append(V)
    # print(Vbar)

# Added 20220316 Hagen-Poiseuille equation, only applicable for laminar flow in circular pipes.
# In H-P, the pressure drop, dp is directly proportional to the volumetric flowrate.
# This calculation is, allegedly, equivalent to the Darcy-Weisbach equation for calculating pressure drop.
def dp_hb(nu, L, Q, R):
    dphp = (8*nu*L*Q)/(np.pi*(R**4))
    return dphp

hp_ss = []
for i in Q_vol:
    hp_test = dp_hb(nu, L_ss, i, D/2)
    hp_ss.append(hp_test)

# Function calculates the reynolds number, 'N_re', given several parameters.
def re_number (rho, V, D, nu):
    N_re = (rho*V*D)/nu
    # if N_re < 2000:
    #    print('laminar flow')
    # elif 4000 > N_re > 2000:
    #    print('transitional flow')
    # elif N_re > 4000:
    #    print('turbulent flow')
    return N_re

for j in Vbar:
    N_re = re_number (rho, j, D, nu)
    N_re.append(N_re)
    # print(N_re)

# Function uses the Churchill correlation to calculate the darcy friction factor (dimensionless). Note that the constants 'A' and 'B' are required for this calculation.

def darcyff (N_re, D, epsilon):
    A = (2.45+np.log((1/(0.016*N_re)**0.9)+(0.27*epsilon)/D))**16
    B = (37550/N_re)**16

    if N_re < 2000:
        f_d = 64/N_re # For laminar flow, use the 64/N_re relation
    else:
        f_d = 8*(0.016*N_re)**12*(1/(A*B)**1.5))**(1/12)
    return f_d

# Function call, calculate the Darcy friction factor (f_d = darcy friction factor)
f_d = []
for r in N_re:
    fd = darcyff(r, D, epsilon)
    f_d.append(fd)
# print(f_d)

[68]: # Darcy-Weisbach Equation - Calculates the frictional loss in Pascals in a pipe with length 'L'
def delta_p (f_d, rho, Vbar, D, L):
    dp = f_d*(L/D)*((rho*Vbar**2)/2)
    return dp

[69]: X = delta_p (0.4184179, 1000, 0.03685, 0.00416, 0.052992)
# print(X)

[60]: # Calculate the frictional loss in 1 straight section of pipe.
   
   dp_ss = []
   
   for i,j in zip(f_d,Vbar):
       del_p = delta_p (i, rho, j, D, L_es)
       dp_ss.append(del_p)
   
   # print(dp_ss)

[61]: # Function calculates the loss coefficient for a pipe bend
def k_bend (f_d, D, r_bend, alpha):
    k_b = f_d*alpha*(r_bend/D)**(0.1+2.6*f_d)*np.sin(alpha/2)+(np.sqrt((np.sin(alpha/2))+(np.sin(alpha/2)))**(4+alpha/np.pi))
    return k_b

[62]: # Function call, calculate the loss coefficients in the bend geometry for SSSV
   
   k_b = []
   
   for i in f_d:
       kb = k_bend(i, D, r_bend, alpha)
       k_b.append(kb)
   
   # print(k_b)

   # Check these K_b values against those in the revised calcs document

[63]: # Modification of Darcy-Weisbach Equation - Calculates the pressure loss in a bend (Accounts for losses due to friction, secondary flow, and flow separation)

def delta_p_bend (f_d, rho, Vbar, D, r_bend, theta, k_b):
dp\_bend = (0.5*f\_d*rho*(r\_bend/D)*np.pi*(\theta/180)**2)*((\rho_{bend}*Vbar)**2) + (0.5*\rho_{bend}*Vbar**2)
return dp\_bend

[64]: # Function call, pressure loss in one bend
dp\_bend = []
for i, j, k in zip(f\_d, Vbar, k\_b):
dp\_bend = delta\_p\_bend(i, rho, j, D, r\_bend, theta, k)
dp\_bend.append(dp\_bend)

# print(dp\_bend)

[65]: # The following function calculates the loss coefficient for an orifice !!!Note:
---to self: check darcy friction factor use here, might require a recalc for
---the orifice

def k\_orif (beta, f\_d, D\_orif):
alpha = i+0.025*(1-0.215*(beta**2)-0.765*(beta**5))
C\_th = (1-0.50*(L\_orif/(1.4*D\_orif))-0.50*(L\_orif/(1.4*D\_orif))**3)**4.5
if L\_d\_ratio <= 1.4:
k\_o = 0.
else:
k\_o = 0.016*(1-\beta**5)*\alpha**2*C\_th*((\alpha-beta**2)**2)*(1-C\_th)*((\alpha-1)**2)*((1-beta**2)**2)
---f\_d/(L\_d\_ratio-1.4)
return k\_o

[66]: # k\_orif

ek\_o = []
for i in f\_d:
k\_orif = k\_orif(beta, i, D\_orif)
k\_o.append(k\_orif)

# print(len(k\_o))

[67]: # This function calculates the fluid velocity in the orifice, based on the
---continuity equation (V1A1 = V2A2)

def Vbar\_orif (D, D\_orif, Vbar):
Vbar\_o = ((np.pi*(D/2)**2)/(np.pi*(D\_orif/2)**2))*Vbar
return Vbar\_o

[68]: # Vbar\_o - the fluid velocity will increase when it enters the orifice, this
---will calculate that new velocity (Vbar\_o)
Vbar_o = []
for i in Vbar:
    Vborif = Vbar_orif(D, D_orif, i)
    Vbar_o.append(Vborif)
    print(Vbar_o)

[69]: # Function calculates the pressure loss in the orifice

def delta_p_orif(k_o, rho, Vbar_o):
    dp_orif = 0.5*k_o*rho*(Vbar_o**2)
    return dp_orif

[70]: # Calculate pressure drop in orifice (does not take into account the pressure
    # loss due to friction in orifice)
    dp_orif = []
    for i in range(len(Q_nss)):
        for j in range(len(Q_nss)):
            if i == j:
                dp_orif = delta_p_orif(k_o[i], rho, Vbar_o[j])
                dp_orif.append(dp_orif)
                break

    # print(dp_orif)

[71]: # Calculate the total pressure loss in the serpentine path (just a sum of
    # previously calculated values times the number of straight sections and bends)
    tot_dp_ser = []
    for j in range(len(Q_nss)):
        tot_dp_ser.append(num_ser*dp_ser[j] + num_bend*dp_bend[j])
        tot_dp_ser.append(num_ser*hp_ser[j] + num_bend*dp_bend[j])
    print(tot_dp_ser)

[72]: # This function calculates the total pressure loss in the orifice path (it's a
    # sum of the dp_orif values with 1 corresponding dp_orif value)
    # The way this total is calculated depends on how the SSSV is designed. i.e.,
    # the orifice in SSSV V0.52 is preceded by one straight section.
    tot_dp_orif = []
    for i in range(len(Q_nss)):
        tot_dp_orif.append(dp_orif[i] + dp_ser[i])
        print(tot_dp_orif)
[73]:
N_re_orif = []
for j in Vbar_o:
    N = re_number (rho, j, D, nu)
    N_re_orif.append(N)
# print(N_re_orif)

[74]:
# Function call, calculate the Darcy friction factor (f_d = darcy friction_factor) inside the orifice
f_d_orif = []
for r in N_re_orif:
    fd = darcyff(r, D, epsilon)
    f_d_orif.append(fd)
# print(f_d_orif)

[75]:
### Actually, this is likely not needed. The calculation for the loss_coefficient K_orf takes into account whether frictional losses are significant, based on the geometry of the orifice.

### Need to return to this. JMP - 20211121

# Calculate the new frictional losses in the length of the orifice using the new friction factor f_d_orif
dp_l_orif = []
for i, j in zip(f_d_orif, Vbar_o):
    del_p = delta_p (i, rho, j, D_orif, L_orif)
    dp_l_orif.append(del_p)
# print(dp_l_orif)

[76]:
# print(tot_dp_serp)
# print(tot_dp_orif)

[77]:
# Convert mass flowrate (g/s) to volumetric flowrate (mL/min)
# 1 g of water = 1mL of water
Q_vol = []
for i in Q_mass:
    Q = i*60
    Q_vol.append(Q)
# plt.plot(Q_mass, tot_dp_serp, label = 'dp Serpentine', linestyle = 'dotted', color = 'blue')
# plt.plot(Q_mass, tot_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
plt.plot(Q_mass, tot_dp_serp, label = 'dp Serpentine', linestyle = 'dotted', color = 'blue')
plt.plot(Q_mass, tot_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
# plt.plot(Q_mass, tot_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
plt.plot(Q_mass, tot_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
# plt.plot(Q_mass, tot_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
plt.plot(Q_mass, tot_dp_serp, label = 'dp Serpentine', linestyle = 'dotted', color = 'blue')
pl.xlabel("Mass Flowrate (g/s)")
pl.ylabel("Volumetric Flowrate (mL/min)"")
pl.title("Pressure Loss in SSSV V0.57")
pl.legend()
pl.show()

Note: The high pressure loss in the orifice path is consistent with what was observed in experiments.
plt.plot(Q, vol, tct_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
plt.xlim([0, 500])
plt.ylim([0, 1000])
# plt.xlabel("Mass Flowrate (g/s)")
plt.xlabel("Volumetric Flowrate (mL/min)")
plt.ylabel("Pressure Loss (Pa)")
plt.title("Pressure Loss\nSSV V0.57")
plt.legend()
plt.show()

# Note: The high pressure loss in the orifice path is consistent with what was observed in experiments.

[81]: plt.plot(0, re, tct_dp_serp, label = 'dp Serpentine', linestyle = 'dotted', color = 'blue')
plt.plot(0, re, tct_dp_orif, label = 'dp Orifice', linestyle = 'dotted', color = 'red')
plt.axline((0, 0), color = 'black', linestyle = 'dotted', label = 'Transition')
plt.show()
plt.xline(n-1000 , color = 'black', linestyle = 'dotted')
plt.xlabel("Reynolds Number")
plt.ylabel("Pressure Loss (Pa)")
plt.title("Pressure Loss w.r.t. Reynolds Number\nSSSV V0.57")
plt.legend()
plt.show()
A.3 DPT-0 Flow Rate Calculations (SSSV V0.57)

Python (Jupyter Notebook)

DPT-0_VFM-C_Flowrate_Calculator_SSSV_V0.57

[1]: # This notebooks takes the voltage output by DPT-0 on the pumped fluid loop,
    # (PFL) test bench, and outputs a mass flow rate through the 'C'-venturi flow
    # meter (VFM).
    # The steps are:
    # 1. Map the acquired voltages to the corresponding differential pressure (in Pa)
    #    Pascals) via the Calibration worksheet for DPT-1
    # 2. Use the geometry of the VFM, the known differential pressure, and the
    #    Bernoulli equation to solve for the mass or volumetric flow rate in the VFM.
    # On the PFL test bench, the signal wire (white) from DPT-0 is fed into an
    # analog input on an Arduino Mega 2560.
    # DPT-0 is powered with 24V DC via the red and black wires.
    # The arduino is running a sketch which takes the raw input data (values range from
    # 0 to 1023) and converting it to a voltage
    # signal ranging from -0.004 to 4.994V (per the calibration worksheet provided by
    # the manufacturer with the DPT)
    # The voltages, which change as the flow rate through the VFM changes, are
    # simply copied from the arduino serial monitor
    # and pasted into a text file for this Jupyter notebook to pick up.

[2]: # Information about "DPT-0" (Orifice output to lower pressure drop venturi)
    # Model Number: PX419-001DWUSV
    # Serial Number: 403380
    # Capacity: 1.00 PSID (or 6894.76 Pascals - Differential)
    # Calibration Date: 2/14/2012

[3]: # Import libraries
    # numerical math
    import numpy as np
    # data manipulation
    import pandas as pd
```python
# plots
import matplotlib.pyplot as plt

import itertools

# Step 1. Map the acquired voltages to the corresponding differential pressure.
# The calibration worksheet gives a range of differential pressures which correspond to a range of voltages.
# The calibration worksheet gives three values only, so in this script, I will interpolate the calibration data over more than 25-50 points, so that the ending flowrate plots are easier to understand. NOTE: this assumes the voltage given from the sensor acts linearly with flow rate. That might not be the case, so
# inter-value interpolation may be unreliable.

# Setting up the pressures given by calibration worksheet (work sheet actually gives pressure in PSI, here, I use pascals)
max_P_d = 6894.757
min_P_d = 0

# Setting up the voltage output
max_volt = 4.994
min_volt = -0.004

# SSSV 0.57
# Voltage when flow is partially through NDPV and LDV. - SSSV 0.57
actual_volt = [0.2938, 0.3591, 0.4484, 0.5673, 0.6753, 0.8324, 1.1076, 1.3816, 1.6566, 1.9316, 2.0057, 2.3709, 2.7720, 3.0710, 3.4126, 3.6916]
offset_volt = []  # This array (list) fills with voltages measured from DPT-Q, which is readign from VFM-C
for i in actual_volt:
    volt_diff = i - actual_volt[0]
    offset_volt.append(volt_diff)

# This function remaps an array of values from one range to another, given only the max and min values of the two ranges.
def remap(x, in_min, in_max, out_min, out_max):
```
return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min

[6]:
dp_vfmc = [] # Empty array to capture pressure values from loop below
for i in offset_volt:
    press = renap(i, -0.004, 4.994, 0, 6894.757)
dp_vfmc.append(press)

[7]:
#print(dp_vfmc)

[8]:
P_d_vfmc = [] # Empty array to capture pressure values from loop below
for i in actual_volt:
    press = renap(i, -0.004, 4.994, 0, 6894.757)
P_d_vfmc.append(press)

[9]:
# print(P_d_vfmc)

[10]:
# Step 2 take the calculated differential pressures (in pascals) and calculate the mass flow rate (Note: DPT-1 is connected to the high dp venturi)

rho = 998.21 # Density of water at 20C, kg/m^3

# Cross-sectional areas of inlet and throat of the high-dp venturi
A1 = 1.46e-5 # Area for inlet cross-section (m^2) Radius = 2.208mm - 0.05mm
...(Estimated over-cure) = 2.158mm radius.
A2 = 4.43e-6 # Area for throat cross section (m^2) Radius = 1.289mm - 0.05mm
...(Estimated over-cure) = 1.186mm radius.

beta_vfmc = A2/A1 # ratio of areas (throat over inlet) for high-dp venturi

[11]:
# function which calculates the velocity in the throat U_2

def vel_2(dp, rho, beta):
    u_2 = sp.sqrt(((2*dp)/(rho*(1-((beta)**2)))))
U_2.append(u_2)

[12]:
# Call vel_2 function to calculate the velocities in the venturi throat, "U_2"
...(Units are meter/second)
U_2 = []

for i in P_d_vfmc:
    v2 = vel_2(1, rho, beta_vfmc)
Chapter A

Joseph Pomo

```python
# Use the continuity equation to solve for the velocity in the inlet, U_1
U_1 = []
for i in U_2:
    v1 = i*beta_vfsc
    U_1.append(v1)
# print(U_1)

# Use rho, the throat cross-sectional area, and U_1 to solve for mass flow rate
Q_vfsc = []
for i in U_1:
    q = i*Ai*rho*1000
    Q_vfsc.append(q)

# Convert mass flow rate (g/s) to volumetric flowrate (ml/min)
# 1 g of water = 1mL of water
Q_vfsc_vol = []
for i in Q_vfsc:
    Q = i*60
    Q_vfsc_vol.append(Q)

# Enter these values into the appropriate sheet in the 'FP150_recharacterization' excel file
Q_vfsc_vol
```

[243.80374012430644, 279.89209364429053, 297.95692174754634, 336.4901412822465, 377.32746203624704, 417.26435226535095, 467.1273464040485, 521.5447963850475, 575.8742435595888, 626.0374111894946, 682.6769184991066, 738.0789248386243]
776.8085433109129, 818.8091946817901, 881.5836809105676]

[18]: Q_vfsc_vol_lpm = []
    for i in Q_vfsc_vol:
        Q_lpm = i/1000
        Q_vfsc_vol_lpm.append(Q_lpm)

[19]: print(Q_vfsc_vol_lpm)
    [0.248380374012430646, 0.2798820936642905, 0.29795921747545634,
     0.336495014126224665, 0.37732746203824703, 0.4172643522555597,
     0.4671273666604856, 0.5213476793850675, 0.5755742430598688, 0.6280374111884969,
     0.6826756154910656, 0.7380789243350423, 0.7758058433109129, 0.81856919446817902,
     0.8516836809105677]

[20]: %store -r Q_vfsc_vol
%store -r P_d_vfsc

[21]: %Q_vfmc_vol

[22]: %Q_vfma_vol

[23]: # plt.scatter(Q_vfmc_vol,P_d_vfmc, label = 'VFMC')
# plt.scatter(Q_vfma_vol,P_d_vfsc, label = 'VFMA')
    plt.scatter(P_d_vfmc, Q_vfsc_vol, label = 'VFMC')
    plt.scatter(P_d_vfsc, Q_vfsc_vol, label = 'VFMA')
    plt.legend()
    #plt.title("Differential Pressure vs. Volumetric Flowrate for LOPV")
    plt.xlabel('Volumetric Flow Rate (mL/min)')
    plt.ylabel('Differential Pressure (Pa)')
$y = \text{np.linspace}(0, 1.25, 500)$
A.4 DPT-1 Flow Rate Calculations (SSSV V0.57)

Python (Jupyter Notebook)

DPT-1_VFM-A_Flowrate_Calculator_SSSV_V0.57

[1]: # This notebooks takes the voltage output by DPT-1 on the pumped fluid loop
     →(PFL) test bench, and outputs a mass flowrate through the high
     →pressure-differential (smallest) venturi flow meter (VFM)
     # which is connected to the serpentine output
     # The steps are:
     # 1. Map the acquired voltages to the corresponding differential pressure (in)
        →Pascals) via the Calibration worksheet for DPT-1
     # 2. Use the geometry of the VFM, the known differential pressure, and the
        →Bernoulli equation to solve for the mass or volumetric flowrate in the VFM.

     # On the PFL test bench, the signal wire (white) from DPT-1 is fed into an
     →analog input on an Arduino Mega 2560.
     # DPT-1 is powered with 24v DC via the red and black wires.

     # The arduino is running a sketch which takes the raw input data (values range
     →from 0 to 1023) and converting it to a voltage
     # signal ranging from -0.002 to 4.997% (per the calibration worksheet provided,
     →by the manufacturer with the DPT)
     # The voltages, which change as the flowrate through the VFM changes, are,
        →simply copied from the arduino serial monitor
     # and pasted into a text file for this jupyter notebook to pick up.

[2]: # Information about "DPT-1"
     # Model Number: FX419-001DWUSV
     # Serial Number: 403446
     # Capacity: 1.00 PSID (or 6894.76 Pascals - Differential)
     # Calibration Date: 2/14/2012

[3]: # Import libraries
     # numerical math
     import numpy as np
     # data manipulation
```python
import pandas as pd

# plots
import matplotlib.pyplot as plt
import itertools

# Step 1, Map the acquired voltages to the corresponding differential pressure.
# The calibration worksheet gives a range of differential pressures which correspond to a range of voltages.
# the calibration worksheet gives three values only, so in this script, I will interpolate the calibration data over more than
# that, say 25-50 points, so that the ending flowrate plots are easier to understand. NOTE: this assumes the voltage given from the
# sensor acts linearly with flow rate. That might not be the case, so inter-value interpolation may be unrealistic.

# Setting up the pressures given by calibration worksheet (work sheet actually gives pressure in PSI, here, I use pascals)
max_P_d = 6894.757
min_P_d = 0

# Setting up the voltage output
max_volt = 4.997
min_volt = -0.002

# actual_volt = np.linspace(-0.002, 4.997, 50) # This list will later be replaced with real data from the DFT.

# SSVY V0.57
# Voltage when flow is partially through VFM-A.
actual_volt = [0.01948,0.07179,0.12604,0.22426,0.28533,0.38599,0.5277,0.69044,0.86504,1.08628,1.31888,1.53386,1.82852,2.07139,2.32012]

offset_volt = [] # This array (list) fills with voltages measured from DPT-Q, which is readign from VFM-C
```
for i in actual_volt:
    volt_diff = i - actual_volt[i]
    offset_volt.append(volt_diff)

# This function remaps an array of values from one range to another, given only the max and min values of the two ranges.
def remap(x, in_min, in_max, out_min, out_max):
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min

dp_vfma = [] # Empty array to capture pressure values from loop below
for i in offset_volt:
    press = remap(i, -0.002, 4.997, 0, 6894.757)
    dp_vfma.append(press)

#print(dp_vfma)

P_d_vfma = [] # Empty array to capture pressure values from loop below
for i in actual_volt:
    press = remap(i, -0.002, 4.997, 0, 6894.757)
    P_d_vfma.append(press)

#print(P_d_vfma)

# Step 2 take the calculated differential pressures (in pascals) and calculate, the mass flow rate (Note: DP-1 is connected to the high dp venturi)
rho = 998.21 # Density of water at 20C, kg/m^3

# Cross-sectional areas of inlet and throat of the high-dp venturi
A1 = 1.46e-5 # Area for inlet cross-section (m^2) Radius = 2.200mm - 0.05mm
    # (Estimated over-cure) = 2.150mm radius.
A2 = 5.52e-6 # Area for throat cross section (m^2) Radius = 1.375mm - 0.05mm
    # (Estimated over-cure) = 1.325mm radius

beta_vfma = A2/A1 # ratio of areas (throat over inlet) for high-dp venturi

# function which calculates the velocity in the throat U_2

def vel_2(dp, rho, beta):
    u_2 = np.sqrt(((2*dp)/(rho*(1-(beta)**4))))
    U_2.append(u_2)

# Call vel_2 function to calculate the velocities in the venturi throat, "U_2"
    # Units are meter/second
U_2 = []

for i in P_d_vfma:
    v2 = vel_2(1,998, beta_vfma)
    #print(U_2)

[13]: # Use the continuity equation to solve for the velocity in the inlet, U_1

U_1 = []

for i in U_2:
    v1 = i*beta_vfma
    U_1.append(v1)
    #print(U_1)

[14]: # Use rho, the throat cross-sectional area, and U_1 to solve for flowrate(g/s)

Q_vfma = []

for i in U_1:
    q = i*rho*1000
    Q_vfma.append(q)

[15]: #print(len(Q_hdpv))

[16]: # Convert mass flowrate (g/s) to volumetric flowrate (ml/min)

Q_vfma_vol = []

for i in Q_vfma:
    Q = i*80
    Q_vfma_vol.append(Q)

[17]: Q_vfma_vol_lpm = []

for i in Q_vfma_vol:
    Q_lpm = i/1000
    Q_vfma_vol_lpm.append(Q_lpm)

[18]: #print(Q_vfma_vol_lpm)

[19]: # Share the following values among the notebooks, so that they can be used elsewhere.

 STORE Q_vfma_vol
%store P_d_vfna

 Stored 'Q_vfna_vol' (list)
 Stored 'P_d_vfna' (list)
A.5 Differential Pressure Transducer 0 (DPT-0) - Arduino Sketch

```cpp
// Sensor read code for Omega_PX419-001DWU5V Differential pressure
// transducer
// Reads signals from DPT-0

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  float x = analogRead(A2);
  float in_min = 0.00;
  float in_max = 1023.00;
  float out_min = -0.004;
  float out_max = 4.994;
  float k;

  k = maptest(x, in_min, in_max, out_min, out_max);
  Serial.println(k, 4);
}

float maptest(float x, float in_min, float in_max, float out_min, float out_max){
  float result;
  result = (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
  return result;
}
```
A.6 Differential Pressure Transducer 1 (DPT-1) - Arduino Sketch

```c
// Sensor read code for Omega_PX419-001DWSV5 Differential pressure transducer
// Reads signals from DPT-1

void setup() {
   // put your setup code here, to run once:

   Serial.begin(9600);
}

void loop() {
   // put your main code here, to run repeatedly:
   float x = analogRead(A0);
   float in_min = 0.00;
   float in_max = 1023.00;
   float out_min = -0.002;
   float out_max = 4.997;
   float k;

   k = maptest(x, in_min, in_max, out_min, out_max);
   Serial.println(k, 4);
}

float maptest(float x, float in_min, float in_max, float out_min, float out_max) {
   float result;
   result = (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
   return result;
}
```
/ A.7 Omega Turbine Flow Meter Pulse Counter - Arduino Sketch

```cpp
// This code gives the volumetric flowrate for the Omega FTB-1311
// turbine flowmeter
// I adapted this code from one of Joshua Hrisko’s projects:
// https://makersportal.com/blog/2018/10/3/arduino-tachometer-using-a-hall-effect-sensor-to-
// measure-rotations-from-a-fan
// digital pin 2 is the hall pin
int hall_pin = 2;
// set number of hall trips for RPM reading (higher improves
// accuracy)
float hall_thresh = 100.0;

void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
    // make the hall pin an input:
    pinMode(hall_pin, INPUT_PULLUP);
}

void loop() {
    // preallocate values for tach
    float hall_count = 1.0;
    float start = micros();
    bool on_state = false;
    // counting number of times the hall sensor is tripped
    // but without double counting during the same trip
    while(true){
        if (digitalRead(hall_pin)==0){
            if (on_state==false){
                on_state = true;
                hall_count+=1.0;
            }
        } else{
            on_state = false;
        }
    }
}
```
Bibliography


[34] Space Ref. *International Space Station: December 2013.* 2013. URL: http://%20spaceref.com/international-space-station/2013/12/.


