

# Analog Fluidic Networks

Towards computational non-electronic systems

Christopher Julien Stocker Salas

# Analog Fluidic Networks

**Towards computational non-electronic systems**

by

**Christopher Julien Stocker Salas**

Student Name	Student Number
Christopher Stocker	566575

Professor: Jamie Paik  
Host Professor: Katia Bertoldi  
Teaching Assistant: Anne Meeusen  
Project Duration: June 6 - November 24, 2023  
Faculty: School of Engineering, EPFL

**EPFL**

I would like to express my profound gratitude to Prof. Paik for providing me with the invaluable opportunity to conduct my Master's thesis research at Harvard's Materials and Structures by Design Laboratory.

I extend my sincere thanks to Prof. Bertoldi and Doc. Anne Meeussen, as well as Doc. Adel Djellouli for their unwavering guidance, generous time, and constant efforts in helping me complete this academic journey. They shared exceptional patience in imparting the necessary techniques and concepts crucial to this project, and they were always readily available to address my questions and concerns.

Furthermore, to all who played a part in this journey, your contributions have been crucial to my personal growth, both academically and personally. I am forever grateful for all your support.

A handwritten signature in black ink, appearing to read "J. Stocker". The signature is fluid and cursive, with a horizontal line underneath it and a small mark resembling a checkmark or a stylized 'e' at the end.

— Christopher J. Stocker

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Literature Review</b>	<b>3</b>
A	Historical Context . . . . .	3
B	Alternative Computing Systems . . . . .	4
B.1	Advancements in Fluidic Networks . . . . .	5
B.2	Digital Logic for Soft Devices . . . . .	5
<b>3</b>	<b>Theoretical Framework</b>	<b>6</b>
A	Principles of Analog Fluidic Networks . . . . .	6
B	Calculating Response of Fluidic Network . . . . .	7
C	Describing our nonlinearities in the model . . . . .	8
<b>4</b>	<b>Methodology</b>	<b>9</b>
A	Hardware Overview . . . . .	9
B	Automation Overview . . . . .	9
C	Characterization and Testing . . . . .	11
D	Pressure Sensors Setup . . . . .	11
E	Characterizing Network Elements . . . . .	13
F	Testing Networks . . . . .	13
<b>5</b>	<b>Experimental Setup and Results</b>	<b>14</b>
A	Experimental setup of linear network . . . . .	14
B	Experimental setup of nonlinear network . . . . .	17
<b>6</b>	<b>Results and Discussion</b>	<b>19</b>
<b>7</b>	<b>Challenges and Opportunities</b>	<b>20</b>
<b>8</b>	<b>Conclusion</b>	<b>21</b>
<b>9</b>	<b>Personal Contributions</b>	<b>22</b>
<b>10</b>	<b>Supplementary Information</b>	<b>23</b>
A	Linear and Nonlinear 3D Plots . . . . .	23
B	Plots: Pressure-Time series of tube network . . . . .	24
C	Plots: Pressure-Time series of check valve network . . . . .	26
D	Contributions . . . . .	28
E	Code Availability . . . . .	28
<b>References</b>		<b>29</b>
<b>Appendix</b>		<b>30</b>
A	Datasheet: Saleae Logic Pro 8 . . . . .	31
B	Datasheet: NXP Pressure Sensor MPVX7002DP . . . . .	38
C	Datasheet: NXP Pressure Sensor MPVX7007 . . . . .	39
D	Datasheet: NXP Pressure Sensor MPMPXV7025 . . . . .	40
E	Custom Printed Circuit Board . . . . .	41



**Harvard John A. Paulson  
School of Engineering  
and Applied Sciences**

# Analog Fluidic Networks Towards Computational Non-Electronic Systems

**CHRISTOPHER J. STOCKER<sup>1,2,\*</sup>, SUPERVISED: ANNE MEEUSSEN<sup>1</sup>, KATIA BERTOLDI<sup>1</sup>, AND JAMIE PAIK<sup>2</sup>**

<sup>1</sup>Harvard University John A. Paulson School of Engineering and Applied Sciences 29 Oxford Street, Cambridge, MA 02138, USA

<sup>2</sup>Swiss Federal Institute of Technology in Lausanne, CH-1015, Lausanne, Switzerland

\* christopher.stockersalas@epfl.ch, cstocker@seas.harvard.edu

---

Soft robotics is a domain that seeks to construct robots composed primarily of soft materials. Although progress has been substantial in developing non-rigid actuators and sensors, the evolution of soft control systems remains less explored. This research introduces an analog fluidic network as a novel approach to alternative computation methods. Such analog networks could be utilized to offload computation from the brain to body of our robot. This enables a computational task to be entirely mechanically operated. The fluidic network can be design to map different linear and non linear functions. The output of the network is entirely determined by the physical dynamics of its constituent elements. To achieve this a comprehensive characterization of the fluidic network's elements were needed to calibrate and validate our analytical model. This involved sweeping the flow and pressure response of our individual elements (tubes, check valves) and our different assembled networks. The results suggest that analog fluidic computers could be used to perform entirely mechanical calculations. This could eventually transform the control of a complex body to be reduced to a simple learning task, where mechanical computer contribute to the overall orchestration of the intelligent behaviour.

---

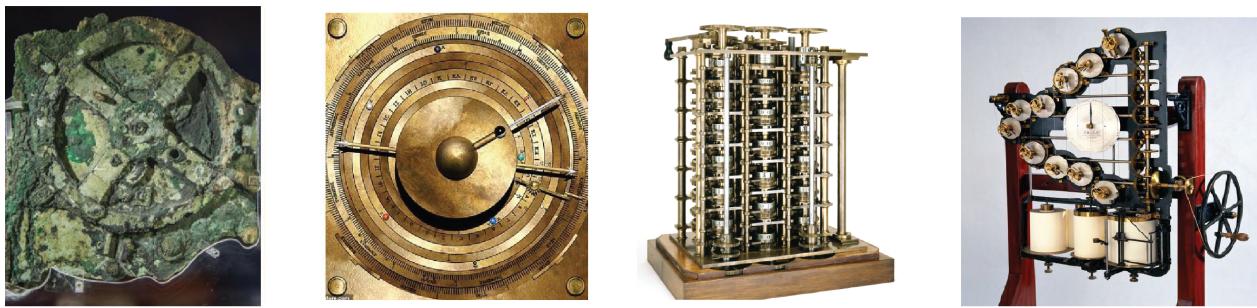
## 1. INTRODUCTION

Throughout history, mechanical mechanisms have been pivotal in the realm of information processing, with notable examples including the ancient Antikythera[1] mechanism and Charles Babbage's analytical machines[2]. In present times, the transistor computer has become the dominant machine, primarily due to its superior miniaturization and integration capabilities[3]. Nonetheless, the emergence of unconventional computing approaches, interlacing concepts from materials science, information processing, and robotics, have opened avenues for novel mechanical computing systems [4, 5]. These systems hold the potential to augment and extend traditional electronic computing by directly interacting with their environment and simplifying the information process along they way.[6].

My thesis explores the utilization of analog fluidic networks as an unconventional computing system, leveraging fluidic mechanisms and their inherent nonlinearities for computation in a modeled network. We propose a framework wherein tubes and check valves form a network for processing information, where the mathematical calculations are intrinsic measures of the output pressure signals[7]. Our focus is on abstracting analog logic within these fluidic systems, examining how they diverge from conventional electronic computing examples [8]. We discuss the unique challenges and opportunities presented by these systems, including their design, integration, and potential applications in soft robotics.

## 2. LITERATURE REVIEW

### A. Historical Context



**Fig. 1.** Original and reconstructed Antikythera mechanism and the Charles Babbage general purpose computing machine.

The first mechanical computer is dated to have originated around 100 BC in Greece. The Antikythera mechanism was an intricate device designed to predict astronomical positions and eclipses for calendar purposes. Its discovery and analysis, demonstrated a sophistication in mechanical computing far ahead of its time [1]. This device, with its complex system of gears, laid the groundwork for future mechanical computers.

Centuries later in the 1830s, Charles Babbage made a monumental step forward with his design of the Analytical Engine[2]. Charles Babbage conceived this computer as a general-purpose computing machine, although he never saw its completion, this Analytical Engine could perform simple arithmetic tasks, store memory, and perform conditional if statements. Similar to its Antikythera predecessor, it operated by harnessing gears and mechanical processes to perform calculations. The structure of the analytical engine was essentially the same that came to dominate computer design in the electronic era. Mechanical machines were starting to peak interests to perform serious laborious mathematical tasks.

By the 19th century a surge of mechanical computers had risen, particularly to address complex scientific problems. The prominent Lord Kelvin was fascinated by the tides at sea and wanted to devise a mechanism to predict low and high tides. He knew the theoretical equations but he was missing a device capable of performing the strenuous calculations. His fascination with Joseph Fourier's work on decomposing functions into sine waves, drove him to design machines capable of automating the laborious process of tidal analysis[9]. Lord Kelvin used devices like the scotch yoke for generating sinusoidal motion and his mechanical analogs for addition, marked a significant leap in practical mechanical computing. These machines did not just simplify existing calculations but enabled the execution of computationally intensive tasks that were previously impractical to perform manually.

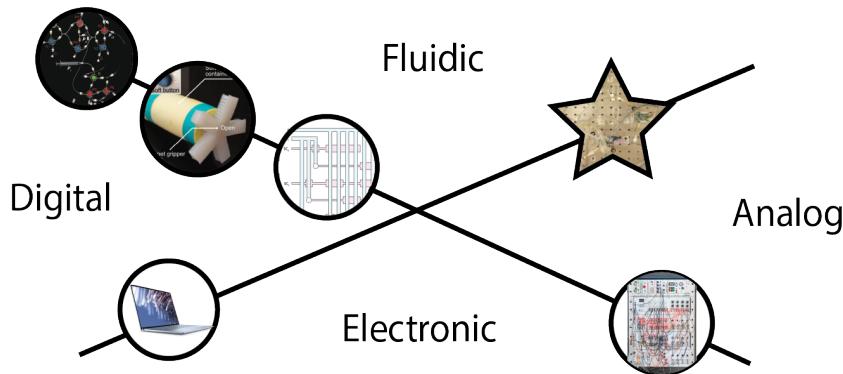
Lord Kelvin even went on to make a mechanical integrator, using a ball and disk system to perform the integral of a function. This invention exemplified the transition from manual to mechanized computation for more sophisticated mathematical calculations, and laid the groundwork for more advanced mechanical computers like the differential analyzer, which evolved into a landmark analog computer. Kelvin's machines, especially the harmonic analyzers, highlighted the ingenuity of mechanical computation. These computational analog machines came to play a pivotal role forecasting the tide and choosing drop points for the Allied invasion on D-Day during World War II [10, 11]. Up to the 1970s analog computer remained unrivaled in the analysis of dynamics systems and were consistently used by every large company and administration concerned with dynamical problems (NASA, Lockheed, CEA, BAC) etc.[12]

Analog computers were surpassed by the rise of electrical digital computers. This shift was largely driven by the development of the transistor and advancements in large-scale integration, which significantly enhanced the computational capabilities of digital systems[3, 13].

As we approach the physical and economic boundaries defined by Moore's Law, a paradigm shift in computing technology appears inevitable. The economic and physical challenges in creating smaller yet equally powerful computers will have obvious consequences on our chip-dependant technologies especially those reliant on miniaturization (i.e. wearables, mobile phones). This transition likely signals a move away from general-purpose hardware and towards specialized architectural designs, along with a diversification in the types of computing systems employed [3, 14].

In the realm of soft robotics, there is a renewed interest in the application of mechanical computers. This resurgence aims to mitigate or entirely remove the necessity for a rigid control unit within soft robotic systems. The integration of mechanical computers introduces a trade, where increased mechanical complexity could reduce computational demands. This is particularly pertinent in the development of soft control systems, as part of the broader goal to engineer entirely soft robotic systems. Our research delves into the use of analog fluidic networks within soft robotic control. By leveraging the inherent capabilities of historical analog computers, we aim to advance the functionality and integration of these systems [15, 16].

## B. Alternative Computing Systems



**Fig. 2.** Cross section between digital and analog computers and the electronic and fluidic interfaces.

Our research looks at orthogonal paradigms of computational power as represent by the star symbol in Fig2. Two types of computers exists: digital and analog. Digital computers function within a discrete domain, employing symbolic representations for computation. For instance, the arithmetic operation "3+4" is executed using binary representation, where "3" (binary: 110) and "4" (binary: 001) are added to yield "7" (binary: 111) with only three bits. In contrast, analog computers leverage physical properties for computation. This approach is illustrated through the the similar example of addition where two one pressure input of 3mbar and a second of 4mbar yield 7mbar of total pressure. The 7mbar value is an actual physical quantity and not symbolic as its digital counterpart, and not discrete but continuous.

In traditional rigid body robotics, control systems are predominantly electronic and digital, executing complex control loops for operation. However, in the domain of soft robotics, different challenges and opportunities arise. Soft actuators, for instance, can attain configurations difficult or impossible to achieve with rigid linkages due to their inherent flexibility and compliance. Yet, these soft actuators often rely on external, rigid electronic control units for pressure modulation, which can introduce limitations like increased stiffness, slowed reaction times, and obstacles to miniaturization. In order to fully think of soft robotic as a compliant system one must think of alternative forms of computing for soft materials.[17].

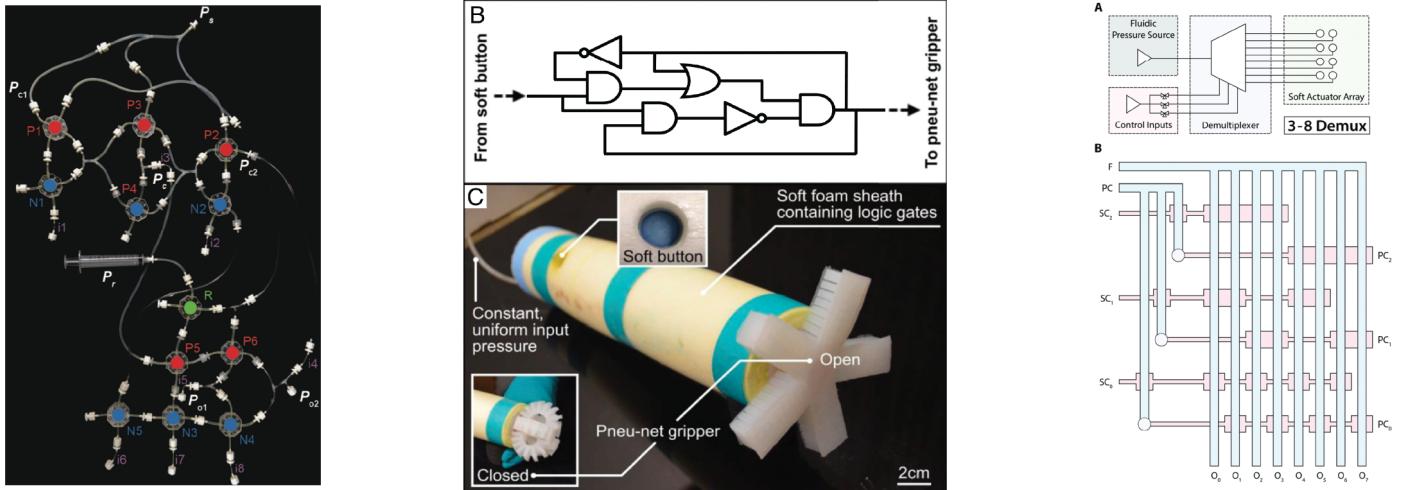
A critical aspect of soft robotics is the integration of actuation and control mechanisms. Typically, multiple actuators are necessary for complex tasks such as navigation, each requiring an external mechanism for pressure control. This dependency not only limits the autonomy of the robot but also imposes physical constraints due to the tethering of tubes, impacting the robot's dynamic capabilities [18–20]. Recently, a variety of techniques have been used to integrate control hardware into soft fluidic robots. These methods are varied and draw from disciplines including microfluidics, digital logic, and material science.

The focus of our research is on developing fluidic circuits for soft robot actuation, emulating biological systems with control and actuation components made entirely of soft materials. This approach, inspired by digital fluidic systems, incorporates elements like fluidic circuits and microfluidic multiplexing[21], offering an adaptable solution that complements traditional computing methods. Notable advancements in this area include the development of an octopus-shaped soft robot with an onboard fluidic self-oscillator for leg actuation [22] and the creation of a bistable valve for generating undulating motion in a crawling robot [23, 24]. Similarly fluidic gates have been demonstrated to perform modulation tasks on soft earthworm robot [25] Fig3.

Soft robots provide significant advantages over their rigid counterparts. These compliant, dexterous devices can navigate delicate environments with ease without damage to themselves or their surroundings. With many degrees of freedom, a single soft robotic actuator can achieve configurations that would be very challenging to obtain when using a rigid linkage. Because of these qualities, soft robots are well suited for human interaction. While there are many types of soft robot actuation, the most common type is fluidic actuation, where a pressurized fluid is used to inflate the device, causing bending or some other deformation.

Obviously the focus of our project is directed towards fluidic circuits for soft robot actuation. In this way the control and actuation are entirely made out of soft materials similar to biological systems. Different to biological systems, previous work in this space has predominant in the fluidic digital space, where biological system operate in the analog space. These novel systems, which include fluidic circuits and microfluidic multiplexing, offer a flexible and adaptable approach. We are looking to explore the capabilities of its analog counterpart to incorporate a different form of mechanical intelligence.

### B.1. Advancements in Fluidic Networks



**Fig. 3.** (Left) circuit of pressure controlled oscillator. (Middle) soft logic gates gripper. (Right) fluidic demultiplexer design.

Recent developments in fluidic networks have been directed towards the integration of soft robotics into everyday applications. Song et al. discuss the challenges in adjusting fluidic circuit's digital logic operations due to the absence of methods for creating analog circuit-like behavior in digital fluidic circuits. They propose a CMOS-inspired fluidic circuit, using fluid valves analogous to the CMOS architecture in electronic circuits, which allows for the creation of fluidic analog circuits [26]. This CMOS-inspired approach facilitates active control over circuit characteristics such as pressure gain amplitude and time response.

Wood et al. (2020) highlight the limited work done in developing soft control systems for soft robots, despite significant advances in soft actuators and sensors. They propose a soft microfluidic demultiplexer as a control system for soft robots, enhancing functionality by simplifying control and increasing efficiency. This demultiplexer reduces the reliance on external valves and components, accommodating high pressures and flow rates essential for soft robotics[27].

### B.2. Digital Logic for Soft Devices

Preston et al. (2019) also describe completely soft pneumatic digital logic gates appropriate for use with current soft actuators. These gates include NOT, AND, and OR gates, built using a single bistable valve and capable of operating with binary high/low input and output pressures. This development paves the way for autonomous control of soft devices without hard components or electronic interfaces. These soft digital logic gates are scalable, power-efficient at steady states, and reconfigurable, supporting functionalities like data storage, decision making, digital-to-analog conversion, and tactile interaction between humans and soft devices[28].

These soft logic gates can be combined to create complex digital logic circuits. For instance, a NOT gate inversely translates its binary input signal, while OR and AND gates are reconfigurable, depending on the input pressure connections. This approach allows the implementation of any logical function, essentially replicating the functionality of electronic computers, albeit with size and speed limitations. Notable implementations include a set-reset latch for data storage, a shift register for reducing input complexity, a leading-edge detector for precise control of data recording, and a digital-to-analog converter for translating digital signals into analog outputs.

### 3. THEORETICAL FRAMEWORK

Here we present an analysis of the physical model. The flow dynamics within the network can be accurately described by the Darcy-Weisbach equation. This equation, denoted as Eq.3, establishes a fundamental relationship between the flow rate ( $q$ ) and the head loss ( $\Delta p$ ) in a closed fluid system. The subsequent section outlines the theoretical scaffolding behind the Darcy-Weisbach equation as well as the mathematical tools used to solve this system in a nonlinear space.

#### A. Principles of Analog Fluidic Networks

In order to understand the inner workings of fluidic circuits it is important to understand the basic underlying physical phenomena. In fluid dynamics, *head* is a crucial concept that relates the energy in an incompressible fluid to the height of an equivalent static column of that fluid, which is essential for understanding fluid flow in analog fluidic networks. There are four types of head: velocity head ( $h_v$ ), elevation head ( $h_e$ ), pressure head ( $h_p$ ), and resistance head ( $h_r$ ). The total head equation, derived from the Bernoulli principle for incompressible fluids, is simplified as

$$h = \psi + z \quad (1)$$

where  $h$  is the hydraulic head in meters,  $\psi$  is the pressure head, and  $z$  is the elevation. In our experiments the elevation head plays a significant role if the inlet and outlet pressure sensors are at different heights. For this reason we made sure to make all our measurements at same heights to eliminate any elevation head bias.

The hydraulic gradient between two or more points in a fluidic network, known as the **Darcy Slope**, determines the **Darcy Flux**, a concept central to our analysis of fluidic networks. This gradient can be expressed through vector calculus as

$$\nabla h = \left( \frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}, \frac{\partial h}{\partial z} \right) = \frac{\partial h}{\partial x} \mathbf{i} + \frac{\partial h}{\partial y} \mathbf{j} + \frac{\partial h}{\partial z} \mathbf{k} \quad (2)$$

where flow occurs in the direction of the negative gradient.

Darcy's law describes the relationship between the Darcy flux  $q$  (defined as  $q = \frac{Q}{A}$ , with  $Q$  in  $m^3/s$  and  $A$  in  $m^2$ ) and the medium's permeability  $k$ , dynamic viscosity  $\mu$ , and pressure drop  $\Delta p$  over a distance  $L$ .

$$q = -\frac{k}{\mu L} \Delta p \quad (3)$$

Darcy's law applies only to laminar flows. Consequently, we selected a fluid mixture of glycerol and water with a specific viscosity conducive to maintaining laminar flow within our predetermined pressure values. In our system, we employ a mixture consisting of 85% glycerol and 15% water by weight. This composition provides a dynamic viscosity for temperatures between 18-23 °C, registering between [0.18568- 0.11385]  $\frac{kg}{m^3}$ .

The permeability  $k$  is linked to the geometry of the tube as the inverse of hydrodynamic pressure. The flow through a cylindrical pipe can be described as part of the Hagen-Poiseuille equation where the hydrodynamic resistance is  $R_c$ , and the dynamic viscosity is  $\mu$  and the tube diameter is  $d$ .

$$R_c = \frac{128\mu L}{\pi d^4} \quad (4)$$

Our network's fluidic flow is characterized as pipe flow, distinct from open channel flow due to the absence of a free surface. This results in the exertion of hydraulic pressure on the conduit rather than atmospheric pressure. The energy within pipe flow is defined as **head** and is governed by Bernoulli's Equation. Frictional losses in pipe flow are described by the Darcy-Weisbach formula.

The Darcy-Weisbach formula is instrumental in our analysis of fluidic networks, describing the average velocity of an incompressible fluid through a pipe and accounting for friction-induced pressure loss. Given cylindrical tubing with uniform diameter  $D_c$ , and assuming full flow, the pressure loss  $\Delta p$  due to viscous effects is proportional to the length  $L$ . The formula integrates these elements, including the **Darcy friction factor**  $f_D$ .

$$\frac{\Delta p}{L} = f_D \cdot \frac{\rho}{2} \cdot \frac{\langle v \rangle^2}{D_c} \quad (5)$$

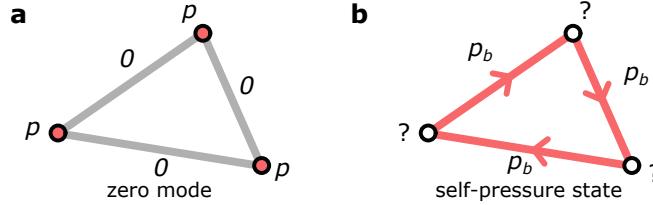
where  $\frac{\Delta p}{L}$  is the pressure loss per unit length, and  $\langle v \rangle$  is the mean flow velocity.

Under laminar conditions, the Darcy friction factor is inversely proportional to the Reynolds number, with  $f_D = \frac{64}{Re}$ . Thus, the Darcy-Weisbach equation can be reformulated as

$$\frac{\Delta p}{L} = \frac{128}{\pi} \cdot \frac{\mu Q}{D_c^4} \quad (6)$$

where  $\mu$  is the dynamic viscosity, and  $Q$  is the volumetric flow rate, with  $Q = \frac{\pi}{4} D_c^2 \langle v \rangle$ . This formulation aligns with the Hagen-Poiseuille equation derived from the Navier-Stokes equations.

## B. Calculating Response of Fluidic Network



**Fig. 4. Special network states.** **a**, Zero mode: equal pressures  $p$  at all (dots) nodes yield no flow through any bonds (lines). **b**, Forbidden state of bond flow: an Escherian state of looping bond pressures ( $p_b$ , arrows) that cannot be realized by any node pressure pattern.

In order to calculate the response of our fluidic network we model it similar to its electrical counterpart. A fluidic network includes  $N$  nodes at which pressure  $\vec{p}$  or net fluid flow  $\vec{q}$  can be imposed. The nodes are connected by  $B$  tubes, through which fluid flows at rates  $\vec{q}_b$  as a consequence of pressure drops  $\vec{p}_b$  between connected nodes, related via conductance functions  $k_b()$  so that  $q_b = k_b(p_b)$ . The network response to imposed node pressures is governed by the following matrix equations:

$$\begin{aligned}\vec{p}_b &= R\vec{p} \\ \vec{q}_b &= K(\vec{p}_b) \\ \vec{q} &= R^T \vec{q}_b\end{aligned}\tag{7}$$

where  $R$  is the network graph's oriented incidence or rigidity matrix and  $K()$  the conductance matrix with functions  $k_b()$  on its diagonal. Specific work is required to calculate the network's response to the imposed node fluxes. Since  $R$  may not be of full rank and  $K()$  may not be analytically invertible, inversion of Eq. 7 requires pseudoinversion and function approximation:

$$\begin{aligned}\vec{q}_b &= W_r D_r^{-1} U_r^T \vec{q} + W_{SF} \vec{x} \\ \vec{p}_b &= K^{-1}(\vec{q}_b) \\ \vec{p} &= U_r D_r^{-1} W_r^T \vec{p}_b + U_{ZM} \vec{y}\end{aligned}\tag{8}$$

Here, matrices  $U$ ,  $W$ , and  $D$  are defined via the singular value decomposition of  $R^T = U D W^T$  [29].  $U$  and  $W$  are orthonormal matrices with the respective left and right singular vectors of  $R^T$  as their columns, while  $D$  contains the singular values of  $R^T$  on its diagonal. For convenience, we partition  $U = [U_r \ U_{ZM}]$  into the vectors spanning the column space and kernel of  $R^T$ ;  $W = [W_r \ W_{SF}]$  into the vectors spanning the row space and cokernel; and  $D = \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix}$  so that  $D_r$  is a square diagonal matrix of ordered (from big to small) nonzero singular values.

Note that the cokernel of  $R^T$  contains *zero modes* (ZM, node pressures that produce no edge pressure differentials, Fig. 4a), and its kernel consists of *self-pressure states* (SF, Escherian edge pressure patterns that map to zero node flux and cannot be realized by node pressurization, Fig. 4b); the rank  $r$  therefore obeys  $N = r + N_{ZM}$  and  $B = r + N_{SF}$ . The unknown  $\vec{y}$  may be chosen freely;  $\vec{x}$  is set via a consistency equation  $W_{SF}^T \vec{p}_b = \vec{0}$ , ensuring that no self-pressure states are present in the network response. This consistency equation may not be analytically solvable if any tube conductances  $k_b()$  are nonlinear, but its solution can be numerically approximated by minimizing  $W_{SF}^T \vec{p}_b$  with a standard root-finding algorithm.

We approximate the response of a network with nonlinear elements to a partially specified node pressure field as follows. A standard vector root-finding method is used to obtain a pressure field  $\vec{p}^u$  at the network's free nodes that ensures no net local node flow under the imposed pressure field  $\vec{p}^i$ . The resulting network state, found by using  $\vec{p} = \vec{p}^u + \vec{p}^i$  as input for Eq. 7, is compatible with both the imposed node pressure and the conditions of fluidic equilibrium.

In the specific case of purely linear conductances, solutions may be found directly using the following three-step approach. First, we calculate node fluxes  $\vec{q}$  resulting from an initial pressure field  $\vec{p} = \sum_i \vec{p}_i$  via Eq. 7, where  $\vec{p}_i = (0, \dots, p_i, \dots, 0)$  are desired individual node pressures. Second, we determine how the network relaxes to mechanical equilibrium so that node flux vanishes except at the specified nodes: an appropriate compensating flux is obtained via  $\vec{q}_p = \vec{q}_n - N_p N_p^T \vec{q}_n$ , where  $N_p$  is the matrix with forced nodes  $n_i = (0, \dots, 1, \dots, 0)$  as its columns. The corresponding relaxation pressure  $\vec{p}_p$  is then calculated from the compensating flux  $\vec{q}_p$  and the reduced compatibility matrix  $R_p = R - R N_p N_p^T$  via Eq. 8. Lastly, the final pressure state of the network is given by  $\vec{p}_{\text{full}} = \vec{p} + \vec{p}_p$ , and the matching bond pressures, fluxes and node fluxes can be obtained from this pressure state via Eq. 7.

### C. Describing our nonlinearities in the model

In the exploration of function generation, linear functions are often considered to lack complexity due to their inherent simplicity. The tube network, in this context, serves primarily as a rudimentary computational device capable of executing basic linear operations described by the Darcy-Weisbach Eq.3. To enable the representation of nonlinear functions, the integration of nonlinear components is essential.

Mathematically speaking a continuous numerical function can be represented as an expansion series of either a Fourier or Taylor analysis. These expansion offer the decomposition of any continuous function as a sum polynomials in the case of Taylor Series, or *cosine* and *sinus* function in the case of Fourier Series. In both instances a large breath of possibilities can be grasped with a handful of building function.

The incorporation of nonlinearity in the network is vital for more sophisticated calculations. Check valves were chosen as the nonlinear element for their easy numerical modeling. These function similar to diodes in the electrical domain, they facilitate unidirectional fluid flow, which is initiated only after surpassing a specific threshold cracking pressure  $p_c$ , expressed as:

$$q = \begin{cases} 0 & \text{if } \Delta p < p_c \\ k(\Delta p - p_c) & \text{if } \Delta p \geq p_c \end{cases} \quad (9)$$

Systems that rely solely on diodes are limited to diode logic, which does not encompass full functional completeness. This means they cannot express all possible truth tables by combination of themselves.

## 4. METHODOLOGY

This methodology overview serves to explicate the technical foundation of our study and provides a basis for replicating the experimental setup. All code files can be found in the project's repository and the data sheets to all the experimental components are in the supplementary information.

### A. Hardware Overview

This section delineates the hardware components utilized in the execution of our research, underscoring their pivotal roles in data acquisition, processing, and controlling experimental conditions. The complete set-up of testing and measuring equipment is listed here below.



**Fig. 5.** Dashboard of all hardware utilized for data acquisition and experimental testing.

- Harvard Apparatus: Pump 33 Dual Drive System
- Fluigent: LineUp™ Push-Pull
- Fluigent: LineUp™ LINK
- Sensirion: SLS-1500
- Fluigent: Flow Unit M+
- Saleae: Logic Pro 8
- NXP Pressure Sensor 2kPa: MPXV7002DP
- NXP Pressure Sensor 7kPa: MPXV7007DP
- NXP Pressure Sensor 25kPa: MPXV7025DP

### B. Automation Overview

This section provides a comprehensive overview of the automation systems employed in our research, highlighting their significance in augmenting both the precision and efficiency of our experimental processes. Automation plays a crucial role in our methodology, enabling consistent and repeatable operations, reducing the potential for human error, and facilitating complex data collection and analysis tasks that would be impractical manually.

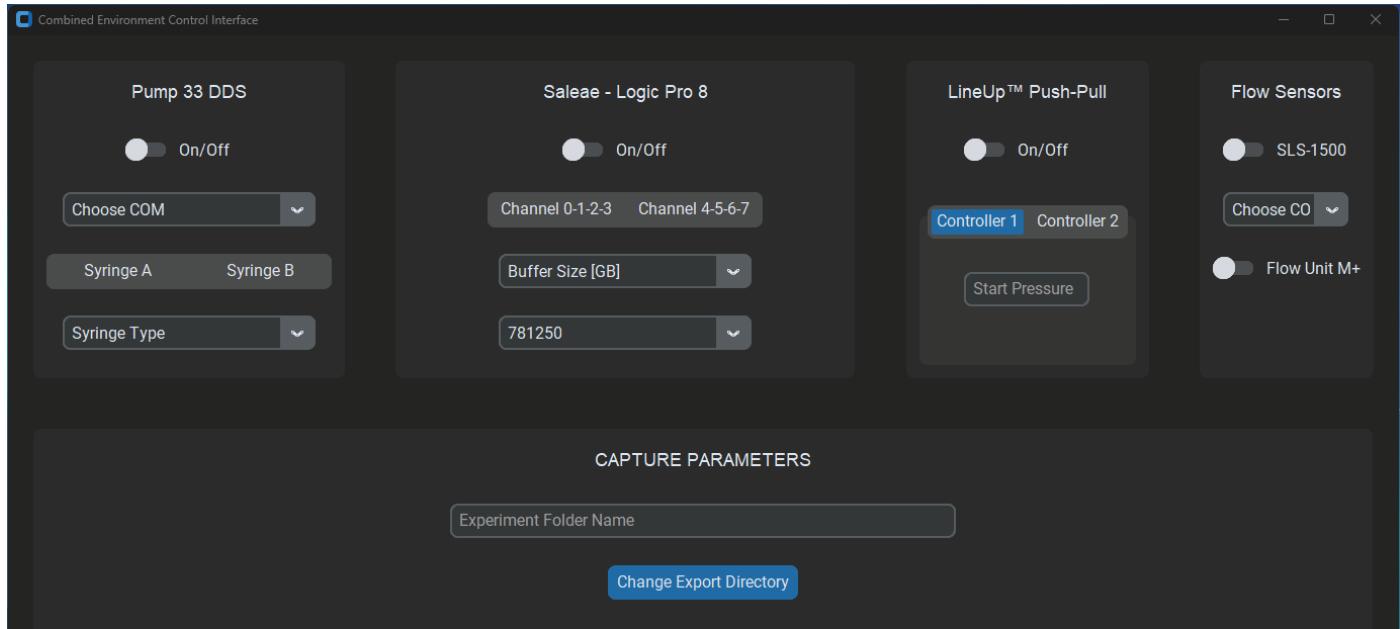
Key elements of our automation strategy include:

1. Build the library codes developed to manipulate all the aforementioned testing equipment.
2. The multiprocessing of these library codes to run in parallel.
3. The data collection automation and post-processing.

Each element contributed to ensure seamless operation, while allowing for sophisticated control and monitoring of experimental variables, thus ensuring the integrity and reliability of our data. The automation files for each hardware equipment can be found in the "Modules" folder in the project's code repository. In this folder we have the following modules:

1. Fluigent: Flow Unit M+ [FLG\_M\_Plus.py]
2. Saleae (both for the version 2.0 and 1.0) [Saleae.py]
3. Fluigent: Flow Unit M+ [FLG\_M\_Plus.py]
4. Fluigent LineUp™ Push-Pull [Push\_Pull\_Pressure.py]
5. Sensirion: SLS-1500 [sls\_1500.py]
6. Harvard Syringe Pump: 33DDS [syringe\_control.py]
7. Saleae: Logic Pro 8 [Saleae.py]
8. Multiprocessing one pressure controller [mp\_one\_controller.py]
9. Multiprocessing two pressure controllers [mp\_two\_controller.py]

The combination of all these developed driver for data acquisition and testing lead to the creation of CECI (Combined Environment Control Interface) [6](#).



**Fig. 6.** CECI (Combined Environment Control Interface) window page for parallel measurements and commands.

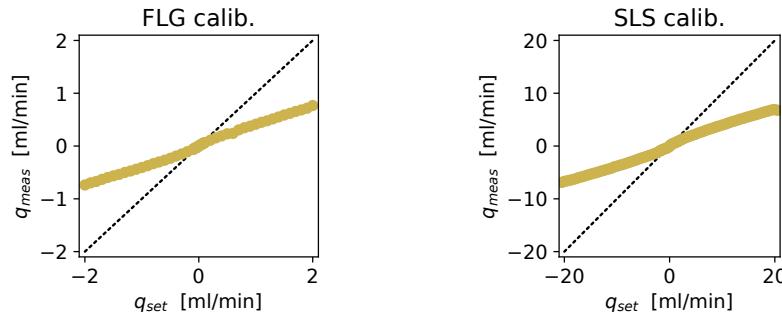
CECI is a graphical user interface for controlling and multiprocessing of the aforementioned lab equipment. Users can configure equipment settings, initiate experiments, and monitor progress through the GUI. The script manages the GUI interface, file handling, and hardware control. Each piece of hardware presents its own frame within the GUI containing different switches, buttons, and entry fields for specific manipulation of that equipment. The interface also provides a calibration option to test and capture the correct functioning of the selected hardware equipment prior to starting the experimentation. During testing a timer shows the progress of the experiment on display. Once done the selected folder location is used to automatically save the experiment relevant files and sub-folders.

CECI simultaneous controls all the measuring and operating devices. It enables the multiprocessing of different scripts to run in parallel thus assuring an accurate and optimized testing environment.

### C. Characterization and Testing

This section outlines the comprehensive characterization and testing methodologies employed in our study, which are crucial for validating the functional integrity and performance metrics of our fluidic network. Characterization involves a detailed examination of the inherent properties and behaviors of our network elements (tubes, and check valves) under various flow rates.

#### *Flow Flow Calibration for Glycerol-Water mixture*

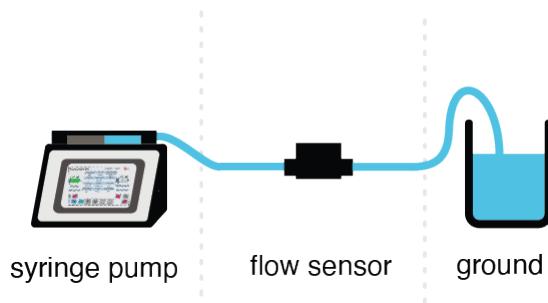


**Fig. 7.** (Left) Calibration of Flow Unit M+. (Right) Calibration of Sensirion SLS-1500.

The Darcy-Weisbach equation (3) highlights our need to independently measure the flow  $q$  and the pressure  $p$  to characterize its permeability  $k$ . The calibration of the flow sensor, particularly for our viscous fluid mixture, involved a more nuanced approach. Both flow sensors are calibrated to measure water, isopropanol, or other readily available chemical liquids. However to have an accurate reading of flow sensors we needed to conduct our own calibration.

To measure flow  $q$  we first used the syringe pump [Harvard Apparatus Pump 33 Dual Drive System] to deliver a steady stream of known flows  $q$ . The syringe pump [Harvard Apparatus Pump 33 Dual Drive System] was utilized for delivering a steady stream of fluid at known flow rates  $q$ . Calibration of flow sensors was performed against the set flow rates of the syringe pump. Two flow sensors, the SLS-1500 [Sensirion] for larger flow ranges and the Flow Unit M [Fluigent] for smaller flow ranges, were calibrated for our specific fluid viscosity. Each sensor have a window of operation of  $[0 - \pm 40]$  mL/min and  $[0 - \pm 2]$  mL/min respectively.

There is no sweep function on the syringe pump. I created a custom script to cycle the syringe pump to push flow at increasing flow rates every 10 seconds. Careful attention was given to the type of syringe used for the specific flow rates, as large flow rates needed large syringes and vice versa. The different sweeps with different syringes were then aggregated to produce the calibration curve show in Fig.8.



**Fig. 8.** Calibration of flow sensors using syringe pump.

### D. Pressure Sensors Setup

In our research, we utilized three different pressure sensors, each tailored for specific ranges of pressure measurement. It is important to note we integrated the sensors to a custom pcb (see appendix) to reduce noise. Each sensor is governed by its own unique transformation function that converts the sensor output voltage to pressure. The transformation functions adhere to the general formula:

$$V_{signal} = V_{source} \times (a \times P(mbar) \times 10 + 0.5) \quad (10)$$

where  $V_{signal}$  is the output voltage from the sensor,  $V_{source}$  is the input voltage to the sensor,  $P(kPa)$  is the pressure in millibar , and  $a$  is a sensor-specific constant.

**1. For the 0.2 kPa to 2 kPa Range:** The transformation function is:

$$P_{\text{mbar}} = \left( \frac{\frac{V_{\text{signal}}}{V_{\text{source}}} - 0.5}{0.2} \right) \times 10 \quad (11)$$

**2. For the 0.057 kPa to 7 kPa Range:** Assuming a different set of constants, the formula might be:

$$P_{\text{mbar}} = \left( \frac{\frac{V_{\text{signal}}}{V_{\text{source}}} - 0.5}{0.057} \right) \times 10 \quad (12)$$

**3. For the 0.018 kPa to 25 kPa Range:** A more appropriate function for this range would have different constants:

$$P_{\text{mbar}} = \left( \frac{\frac{V_{\text{signal}}}{V_{\text{source}}} - 0.5}{0.018} \right) \times 10 \quad (13)$$

Each sensor operates optimally within its specific pressure range. The transformation functions are crucial for converting the raw voltage output into meaningful pressure readings. The choice of these sensors and their corresponding functions ensures that a wide range of pressures can be accurately measured.

## E. Characterizing Network Elements

### PVC Tubes Characterization

The first step of characterizing our connecting pvc tubes fluidic is to determine the permitivity. Following Darcy-Weisbach Eq.3 we need to run sweeps of different flow rates and simultaneously measure the pressure drop across our tube element. Each element will have a specific permitivity relative to the tube's length and diameter.

### Check Valves Characterization

To characterize our connecting check valves we needed to repeat the characterization procedure of pvc tubes but with a check valve added to the end of our tubes. Similarly, we ran sweeps of different flow rates and measured in parallel the pressure drop across our tube plus check valve. To characterize check valves we need to determine their cracking pressure and the permitivity beyond this point.

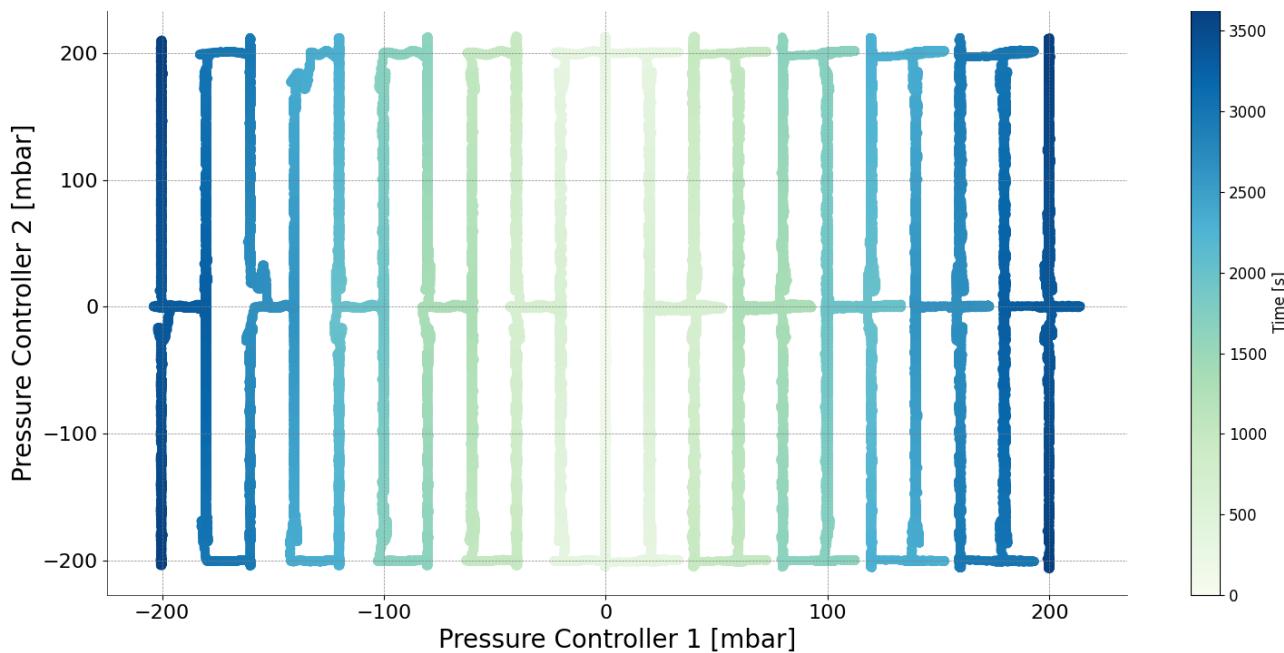
## F. Testing Networks

### Network Preparation

This protocol outlines the steps for measuring flow and pressure. The procedure is divided into three main stages: eliminating bubbles, calibration, and testing. The first stage involves adjustments to the pressure and vacuum valves and a check for any bubbles. The calibration stage involves opening the logic controller, the reservoir valves, and the differential valve, followed by running a pressure sensors calibration test. The final stage is the test, where one must start from a system at rest before launching the CECI automation tool. Concurrently the pressure difference between the output nodes are measured. Before any new test is conducted the pressures sensors are re-calibrated and visual inspection of any leaks or air bubbles are examined. Attention is given to eliminate any height difference between pressure sensor to avoid elevation head Eq.1 as this can skew the data.

### Linear and Nonlinear Network characterization

The firs step is assembling the nodes, tube and check valves to the network's schematic. The reservoirs must be filled to ensure no air bubbles enter the system one automation is launched. Then the two-pressure input space is explored in a zigzag manner as show in Fig 9. Each search starts at 0 pressures for both input pressure and the expands to explore on quadrant in a zig-zag manner. Each quadrant search takes about 1 hours to scan as it requires some time for the PID controller of the system to stabilize.

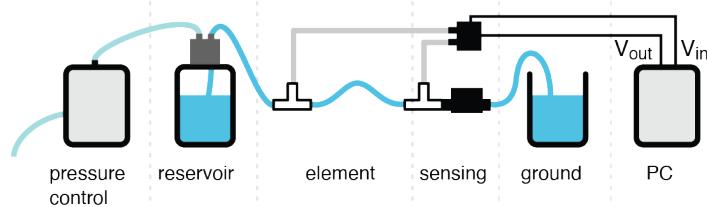


**Fig. 9.** Zig-Zag search of the two-pressure inputs.

## 5. EXPERIMENTAL SETUP AND RESULTS

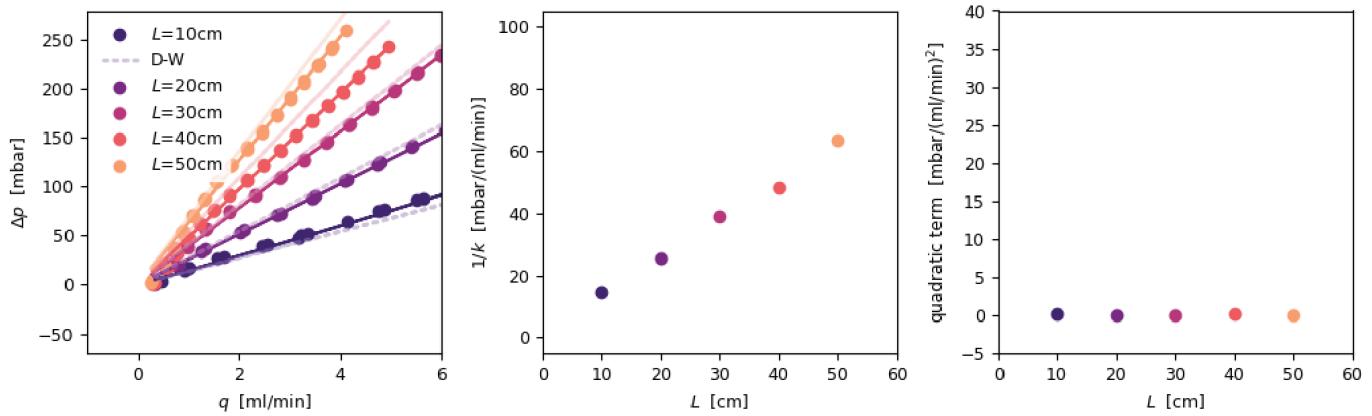
### A. Experimental setup of linear network

Once the flow rates were calibrated for our specific fluid we then proceeded to test the conductance of a simple pvc-tube element. This time however we needed to control the pressure delivered to the tube and simultaneously measure the pressure difference across the tubes while recording the flow. In Fig.10 we see how all these pieces are connected. The pressure controller pushes air to our reservoir that in turn push liquid through our testing element between two nodes. The pressure sensor are connected to these two nodes to measure the change in pressure  $\Delta p$  across our element and relays that data to our computer. Lastly the flow passes through our flow sensor and into ground.



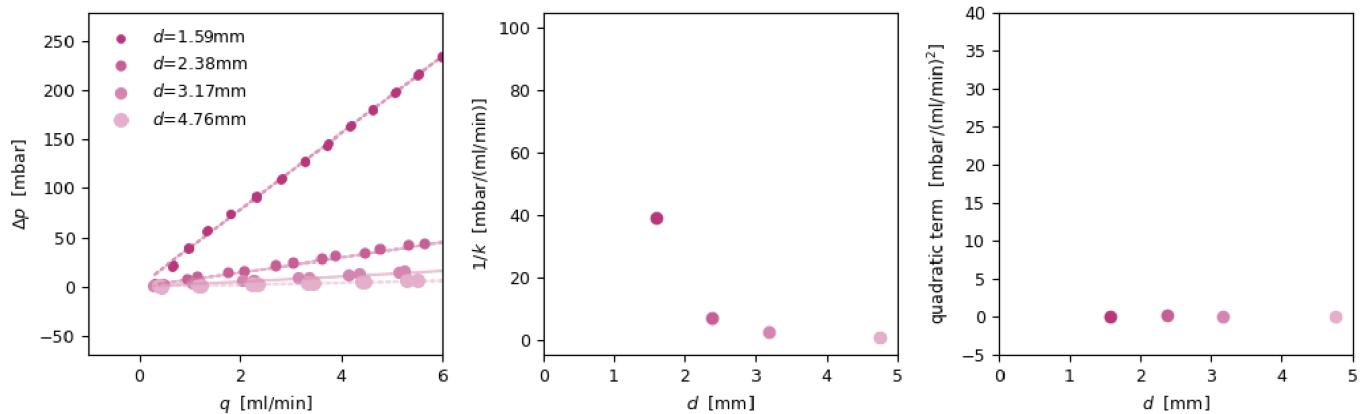
**Fig. 10.** Conductance characterization from different tube length and diameters using flow meter and pressure sensor.

This step was repeated to sweep different flow rates of pvc-tubes of different lengths and diameters. We tested five different lengths of pvc-tubes [10cm, 20cm, 30cm, 40cm, 50 cm] as shown left in Fig.11. The results showed to be consistent with the Darcy-Weisbach Eq.3 depicted here as the dotted line. Fig.11 middle shows how the resistance grows linearly with the increasing tube length. The data also shows there are no effects pressure per square of volumetric flow rate as shown in Fig.11.



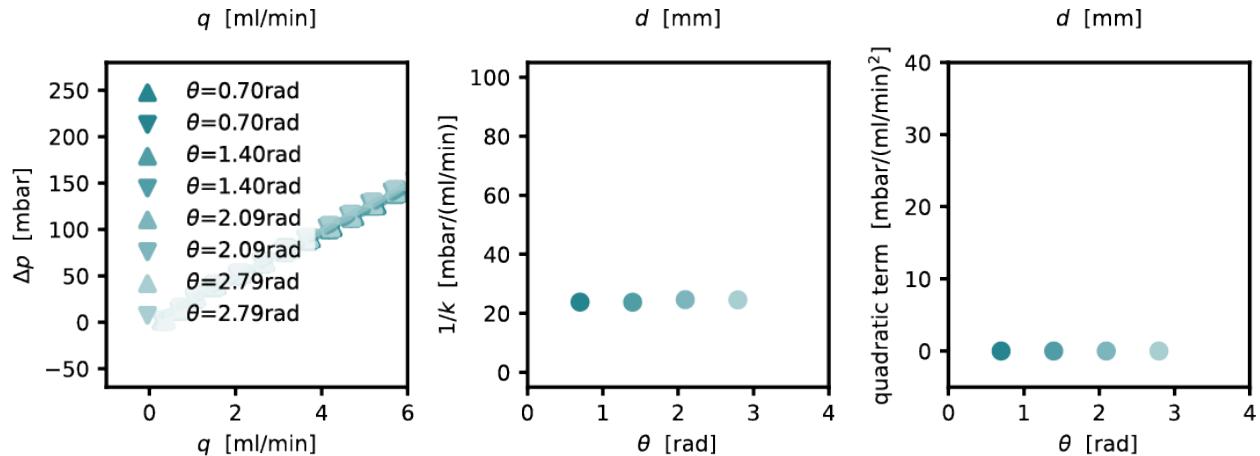
**Fig. 11.** (Left) Characterization of different tube lengths. (Middle) Conductance of each tube. (Right) null correcting factor.

This same experimental procedure had to be repeated to characterize the conductances of tubes of different diameter. We tested four different tube diameters [ 1.59mm, 2.38mm, 3.17mm, 4.76mm] as depicted left of Fig.12. The results again confirm that the change in pressure follows the Darcy-Weisbach Eq.3. Fig.12 (middle) shows the resistance varies inversely by the fourth power of the tube's diameter. This is consistent with the Hagen-Poiseuille Eq.4 for hydrodynamic resistance. Lastly Fig12 (left) shows there is no change in the pressure per square of volumetric flow rate.



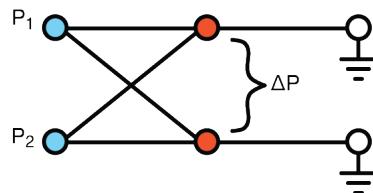
**Fig. 12.** (Left) Conductance characterization of tube diameter. (Middle) Hydrodynamic resistance of circular tube. (Right) null correcting factor.

We 3D-printed two custom nodes for our fluidic system. The first node has 10 luer lock connections, and the second node has 4 luer locks connections. We needed to dismiss any bias our node design would have on the fluidic dynamics of our system. Fig13 (left) shows the attachment to the nodes position does not play a role in changing the Darcy-Weisbach flow of our network. Fig13 (middle ) shows the conductance remains constant. Fig13 (right) shows there are no correcting factors to the Darcy-Weisbach equation.



**Fig. 13.** (Left) Different node attachment. (Middle) Hydrodynamic resistance different attachment points. (Right) null correcting factor.

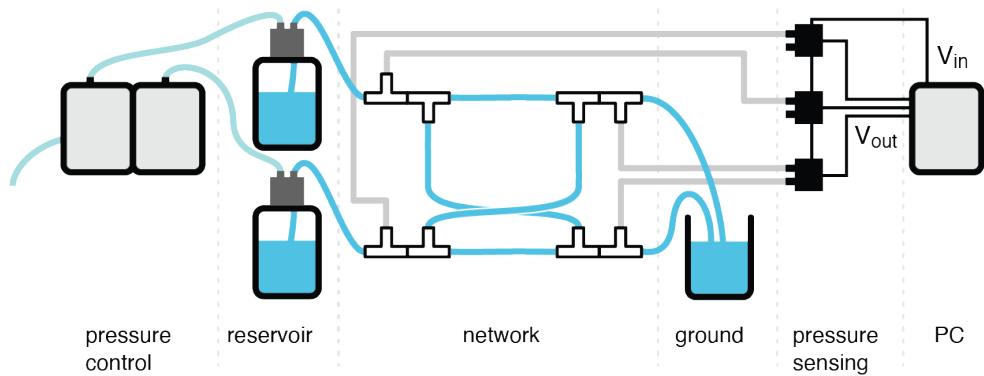
Once we had collected the conductances of all our pvc-tubes, we could assemble our first simple fluidic network. This network consists of two pressure inputs  $P_1$  and  $P_2$ . These are connected in cross like shape as depicted in Fig.14. We then measure the pressure different at the output nodes of this cross shaped network and empty the fluid to ground.



**Fig. 14.** Simple cross network representation.

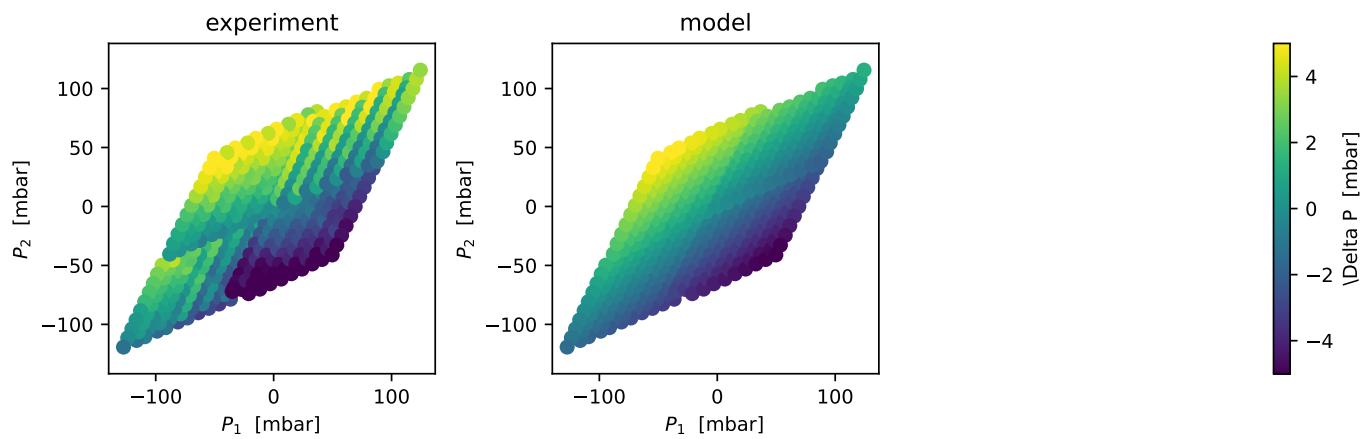
Experimentally this consisted of incorporating a second pressure controller to add a second input to our network. The two pressure controllers independently drive the pressure at our input nodes  $P_1$  and  $P_2$ . Each pressure controller can sweeps a range of  $[\pm 200\text{mbar}]$  giving us a four quadrant workspace to operate with. We sweep this entire workspace with our pressure controllers. The pressure is measured at each input node  $P_1$  and  $P_2$  by our pressure sensors. Our sweep were split in four tests each exploring a different quadrant of pressures. This was due to the constraints on the experimental setup given by the amount of fluid our

reservoir could store and the amount of cache ram our computers could allocate. In each test the fluid passes through the network and the change in pressure is measured at the output nodes before reaching ground. This experimental setup can be visualized in Fig.15.



**Fig. 15.** Experimental setup of crossed tube network.

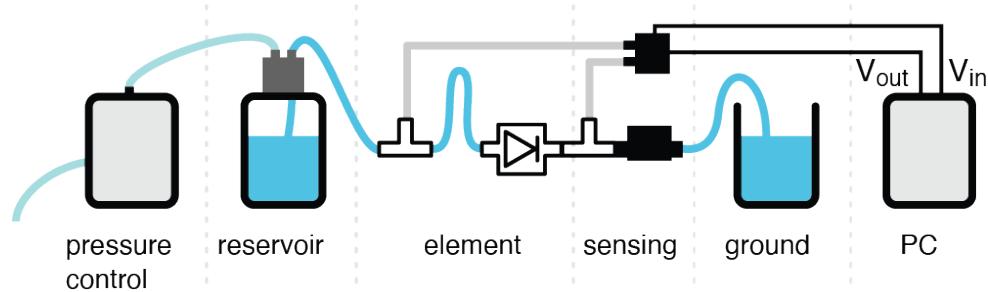
The results of this first tube network were tested with our numerical model by feeding the experimental input data at each node and calculating the expected network response. Then this was compared to the actual experimental values. Fig.16(left) shows the experimental results and Fig.16(right) show the modeled results.



**Fig. 16.** Plots of our pressure difference at our output nodes in our experimental(left) and numerical model (right).

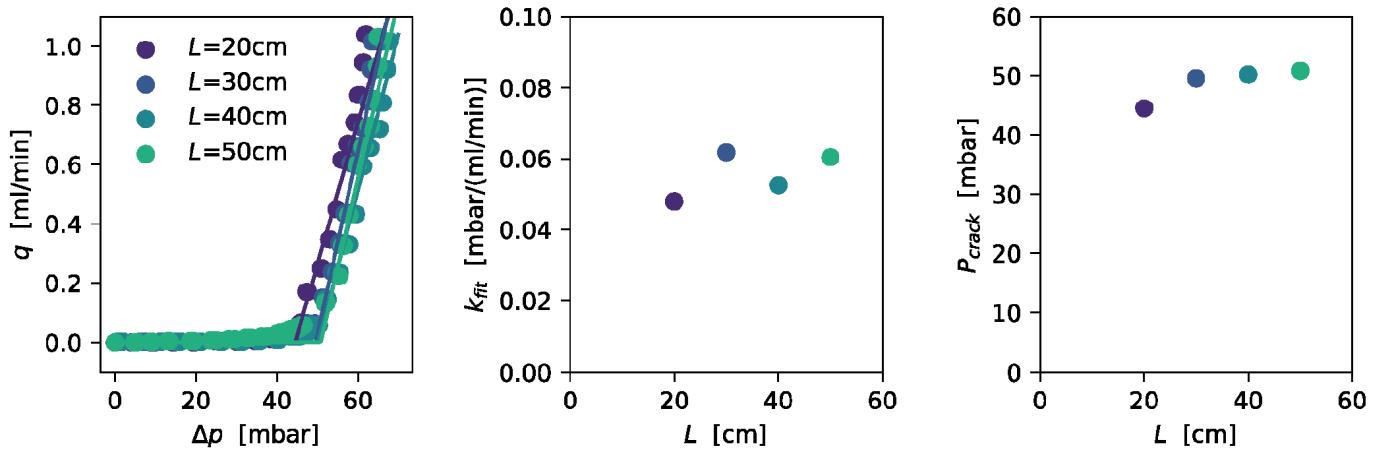
## B. Experimental setup of nonlinear network

Next we wanted to add nonlinearities to our system. For this reason we first needed to characterize our nonlinear element the check valve. To accomplish this we repeated the experimental setup we had devised for the pvc tubes but incorporated a check valve at one end. We swept the system under different pressures and recorded the measured pressure drop and flow rate as with the tube element. We repeated the test for four different tube lengths we would later use in the nonlinear network. Fig17 shows the experimental setup to characterizing the check valves.



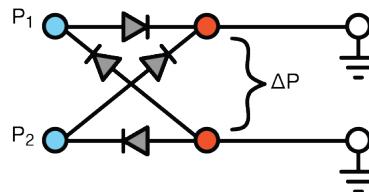
**Fig. 17.** Conductance and cracking pressure characterization from different tube length with check valves using flow meter and pressure sensor

The results showed in Fig.18 (left) shows the conductance of our check valve after a cracking pressure. Fig.18 (middle) shows how the conductance of the check valve and tube element varies with changing tube lengths. Fig.18 (right) shows this cracking pressure varies negligibly based on the length of the tube at the end of the check valve.



**Fig. 18.** (Left) Conductance and cracking pressure characterization for check valves with different tube lengths (Middle) Conductance results (Right) Cracking pressure results.

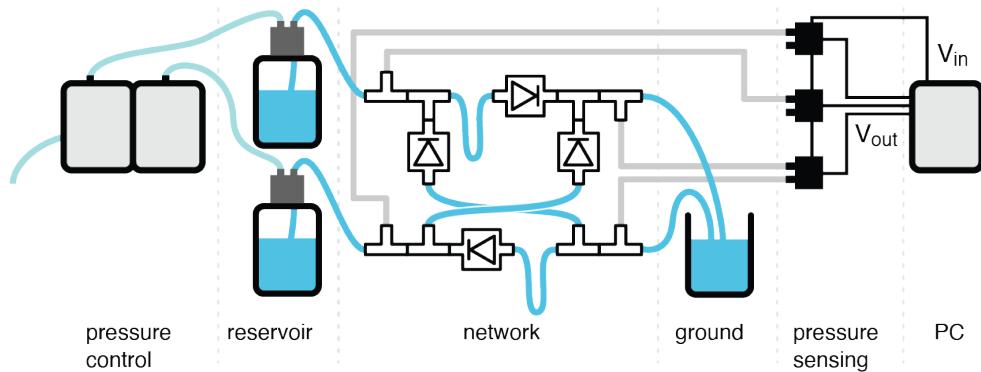
Once we had collected the conductances with the cracking pressures of all tube lengths, we could assemble our first nonlinear fluidic network. Similar as before this networks consists of two pressure inputs  $P_1$  and  $P_2$ . These are connected in cross like shape as depicted in Fig.19. This time however we add "diodes" in the form of check valves to our network. We then sweeps the pressure inputs and measure the pressure difference at the output nodes.



**Fig. 19.** Check valve cross network representation.

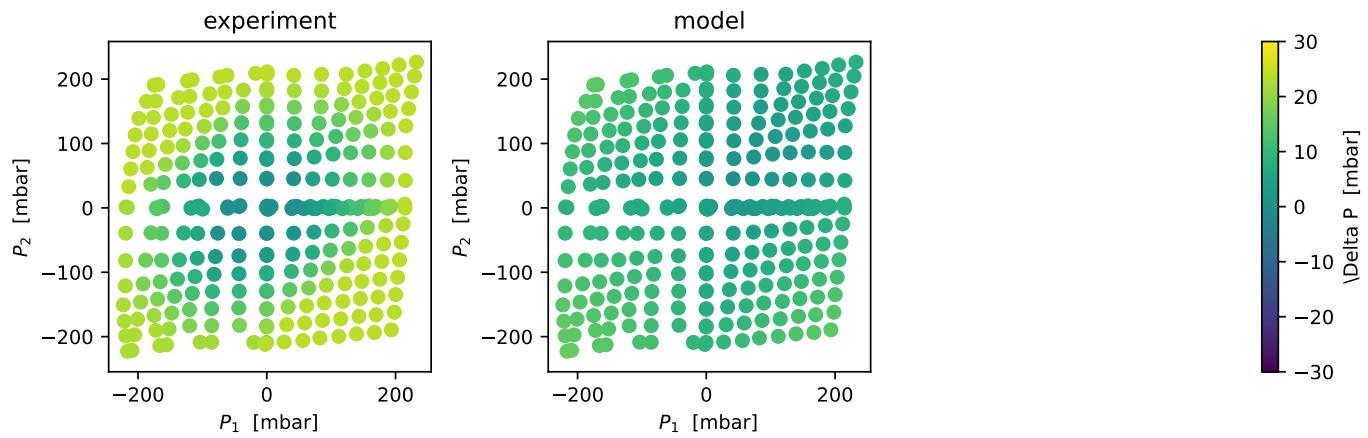
Experimentally this consisted of physically incorporating the check valves in the correct position and orientation as shown in Fig.20. Similarly as before we explore the four input quadrant. In each test the fluid only passes through the check valve in one direction and

does so if the change in pressure across it is superior to its cracking pressure. The experimental diagram of this network is present below.



**Fig. 20.** Experimental setup of check valve network.

The results of this nonlinear network were tested with our numerical model by feeding the experimental input data at each node and calculating the expected network response. Then this was compared to the actual experimental values. Fig.21(left) shows the experimental results and Fig.21(right) shows the modeled results



**Fig. 21.** Plots of our pressure difference at our output nodes in our experimental(left) and numerical model (right).

## 6. RESULTS AND DISCUSSION

### Results Overview

Our experimental outcomes demonstrate the successful inception of both linear and nonlinear function generators through the utilization of analog fluidic networks. This achievement underscores the viability of employing analog methodologies within fluidic circuitry for complex functionalities.

### Observations and Discrepancies

The underlying principle of our approach has been validated by the observed trends in our data. However, we noted discrepancies in magnitude, suggesting the necessity for a more comprehensive characterization. Particularly, this involves a detailed examination of tubes integrated with diodes, alongside an exploration of various diode types. Such discrepancies highlight the intricacies in fluid dynamics within our system, which merit further investigation.

### Future Research Directions

A pivotal future step involves aligning our theoretical models more closely with experimental results. This synchronization is essential for refining the accuracy of our models and enhancing the predictability of the system's behavior in real-world scenarios.

Another promising direction is the integration of our analog fluidic models with optimization algorithms. This integration could open new avenues in complex tasks such as classification problems or control mechanisms in soft robotics. By combining the physical properties of fluidic networks with advanced computational strategies, we anticipate the development of more sophisticated and intelligent soft robotic systems.

## 7. CHALLENGES AND OPPORTUNITIES

### Experimental Challenges

The testing procedures for the fluidic network were meticulously designed to ensure the integrity and accuracy of the experimental results. Prior to each test, a thorough inspection was conducted to ensure there were no leaks in the system. This was critical to maintain consistent pressure results and fluid flow. Additionally, the network was checked for air bubbles, which were removed to prevent any interference with fluid dynamics. Pressure equalization was carried out to stabilize the system before initiating any tests.

The reliability of the Random Access Memory (RAM) components was verified, ensuring their proper functioning in the fluidic setup. Another key step was the preparation of the flow liquid, which involved using a weight mixture of 85% glycerol to 15% water. This was to ensure a laminar flow consistent with our set pressure values. The network was carefully filled with this mixture, taking special care to avoid any air entrapment. Pressure sensors were crucial for the data collection of the pressure drops at specific locations of our network. In order to reduce the white noise picked up from our NXP ??were checked and dried to prevent any moisture-related malfunctions.

Finally, a critical step in the procedure was the calibration of these sensors. Calibration was performed according to the manufacturer's specifications to guarantee accurate measurement of fluidic parameters such as pressure, flow rate, and viscosity. This comprehensive approach to the testing procedures was designed to ensure the reliability and validity of the experimental data collected from the fluidic network.

### Technical Challenges

The development and implementation of fluidic networks in analog computing systems present several technical challenges. One significant hurdle is the inherent limitations imposed by the physical elements used in these networks. As analog computers, they are bound by the physical properties of the components and the physical laws governing them, which can introduce issues such as noise and signal amplification problems. This is particularly challenging in fluidic networks where the fluid dynamics are subject to variations due to temperature changes, viscosity fluctuations, and environmental disturbances.

Moreover, the precision of measurements in fluidic systems can be affected by these factors, leading to inaccuracies in computation. Ensuring consistent and reliable performance under varying conditions requires careful design and robust control mechanisms. These challenges necessitate ongoing research and development to enhance the stability and accuracy of fluidic networks in analog computing applications.

### Future Applications

The potential applications of fluidic networks in the realm of soft computing are vast and promising. We envision that the complexity inherent in controlling fully soft computers can be effectively managed through soft analog computing systems. These systems could leverage environmental inputs to process and deliver signals, thereby integrating more seamlessly with their surroundings.

Such fluidic networks could transform areas where flexibility and adaptability are paramount, such as in wearable technology, biomedical devices, and robotics. For instance, in the field of soft robotics, fluidic networks could provide a more natural and intuitive way of controlling robotic movements and responses. Additionally, in biomedical applications, these networks could lead to advanced prosthetics and diagnostic tools that are more responsive to physiological changes.

The exploration of fluidic networks in computing not only holds the potential for technological innovation but also opens up new avenues for interdisciplinary research. The integration of principles from materials science, fluid dynamics, and information processing could lead to the development of computing systems that are more in tune with the physical and biological world, offering a new paradigm in computing technology.

## 8. CONCLUSION

The advancements in unconventional computing, particularly in fluidic networks and soft digital and analog logic, highlight a growing interest in mechanical intelligence and morphological computing. In the context of our research we demonstrated the possibility of analog fluidic computer to perform linear and nonlinear computations. The linear functions were characterized by the linear elements in our system. To add non linearity to our system we added check valves to provide nonlinear elements to our system that were still simple enough modeled numerically.

In the future we see unconventional computing methods as an alternative to traditional digital computers that transfers complexity, enhances adaptability, and opens new possibilities for autonomous control in soft robotics. These systems embody a novel approach, integrating flexibility and responsiveness into computational processes, making them suitable for soft robotic tasks.

## 9. PERSONAL CONTRIBUTIONS

My personal contributions to this project have been in the fabrication, assembly, coding, and characterization, of the fluidic elements.

### Fabrication

- Node redesign for more robust luer lock connection.
- Printing, tapping, and curing luer lock nodes.

### Assembly

- Final screwing, curing and assembly of network nodes.
- Joint tube actuators.
- Assembly together of network

### Coding

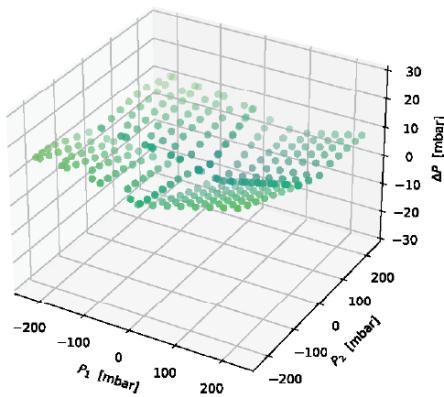
- Multiprocessing of Sensirion flow sensor.
- Multiprocessing of Fluigent micro flow sensor
- Multiprocessing of Fluigent single and double pressure controller.
- Multiprocessing of Harvard Apparatus syringe pump.
- Multiprocessing of Saleae logic controller.
- Data conversion from voltage to pressure values.
- GUI dashboard of components control environment.

### Characterization of fluidic elements

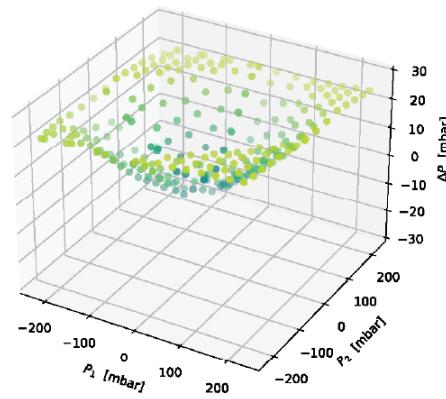
- Tubes of different lengths.
- Tubes of different diameters.
- Tubes of different lengths with different check valves.

## 10. SUPPLEMENTARY INFORMATION

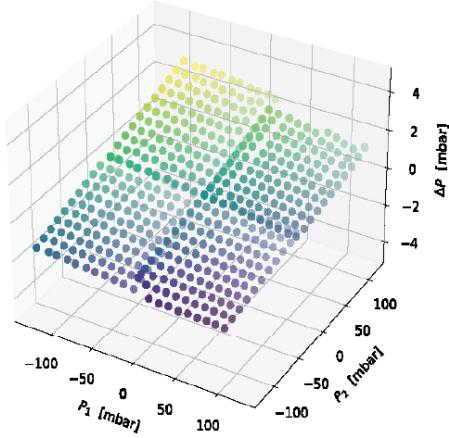
### A. Linear and Nonlinear 3D Plots



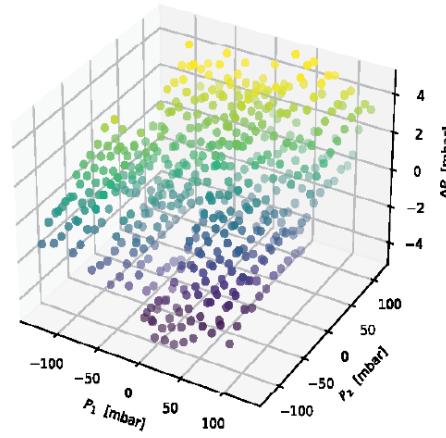
**Fig. 22.** Model check valve network result



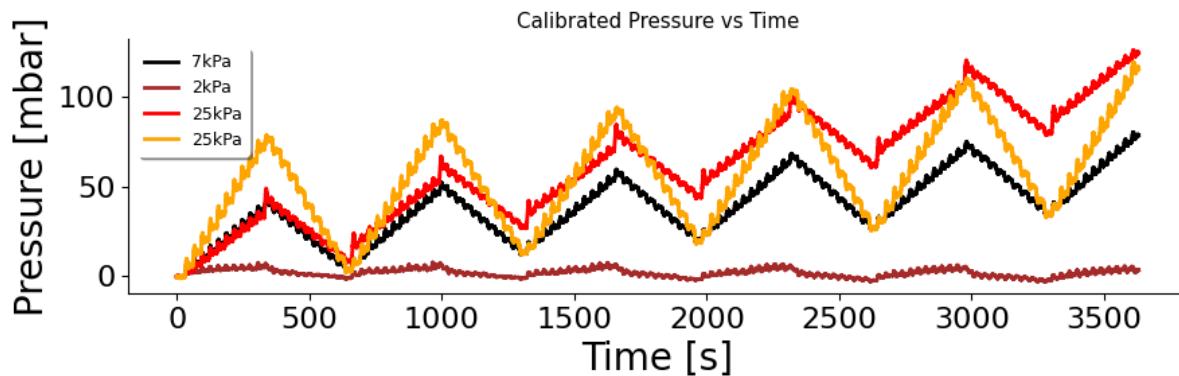
**Fig. 23.** Experiment check valve network result



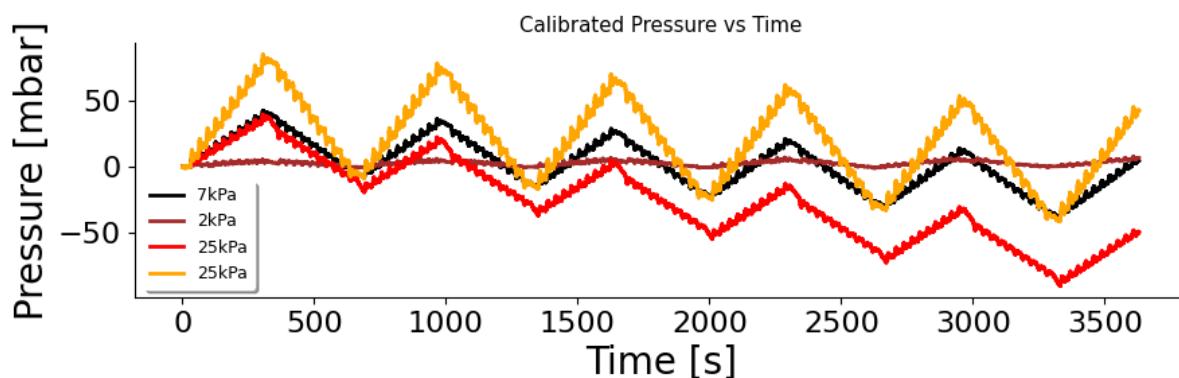
**Fig. 24.** Model tube network result



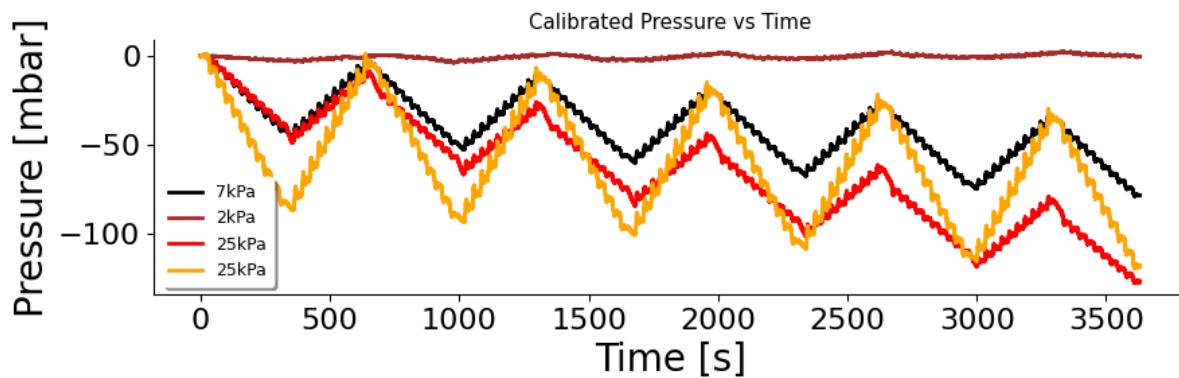
**Fig. 25.** Experiment tube network result

**B. Plots: Pressure-Time series of tube network**

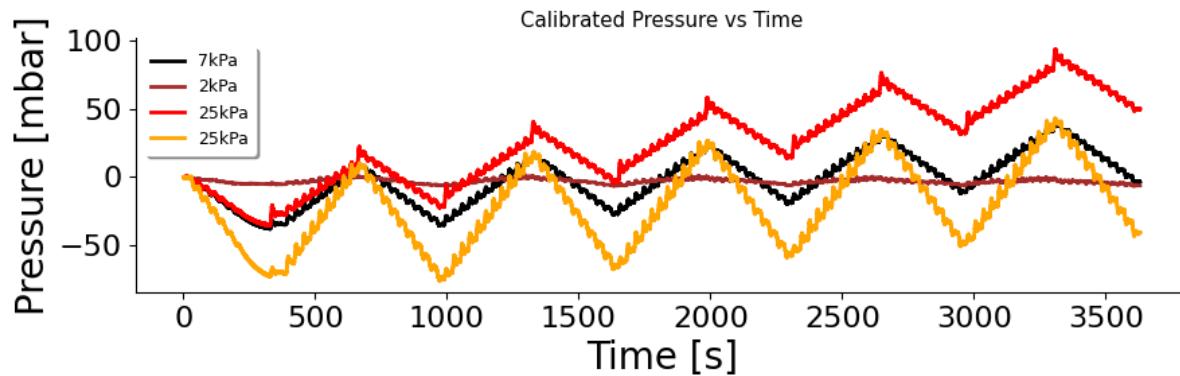
**Fig. 26.** Pressure vs time of tube network search scan in **first quadrant**.



**Fig. 27.** Pressure vs time of tube network search scan in **second quadrant**.

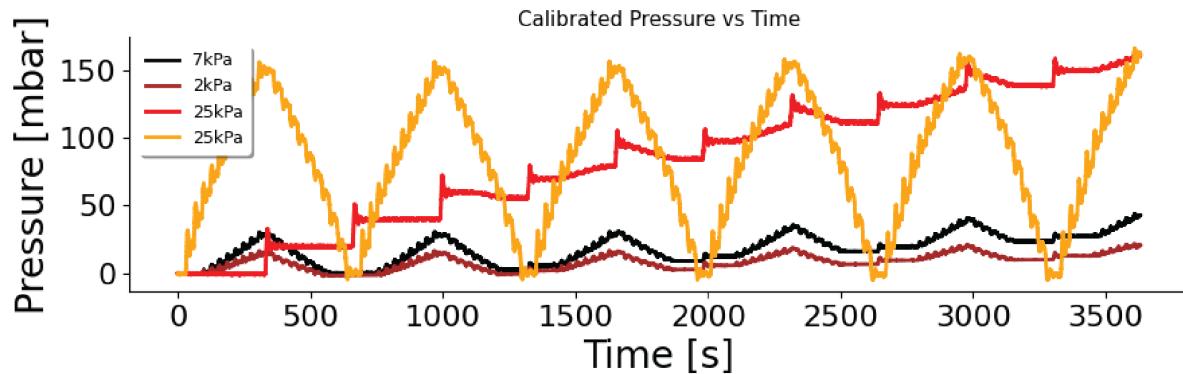


**Fig. 28.** Pressure vs time of tube network search scan in **third quadrant**.

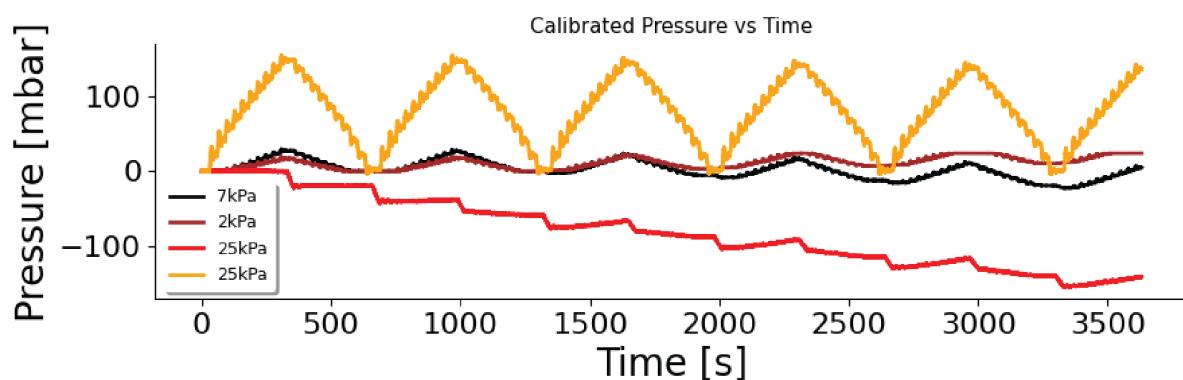


**Fig. 29.** Pressure vs time of tube network search scan in **fourth quadrant**.

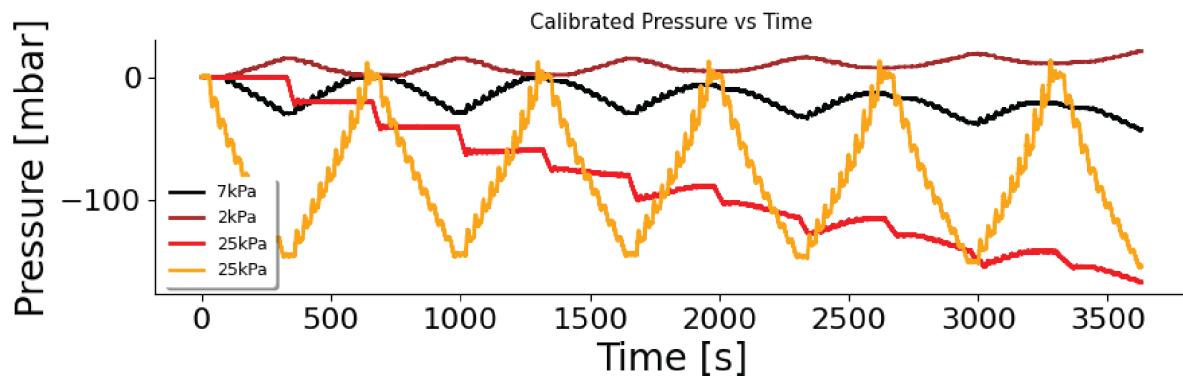
### C. Plots: Pressure-Time series of check valve network



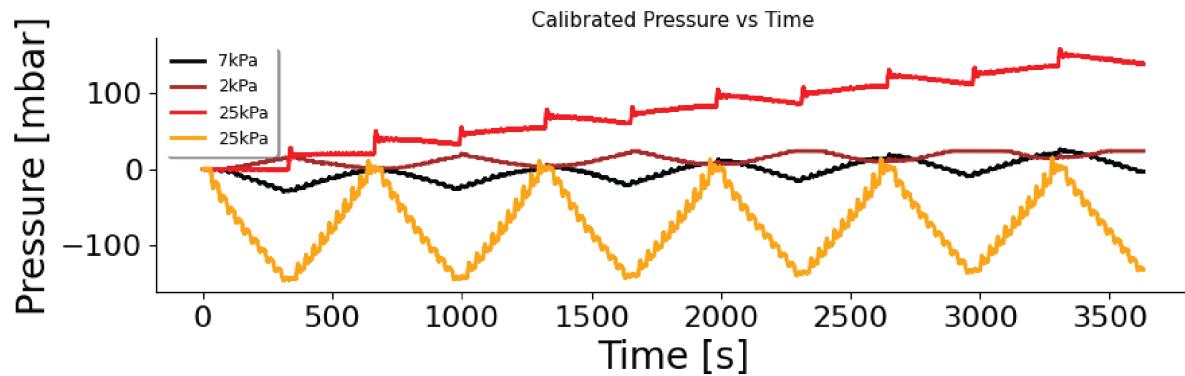
**Fig. 30.** Pressure vs time of check valve network search scan in first quadrant.



**Fig. 31.** Pressure vs time of check valve network search scan in second quadrant.



**Fig. 32.** Pressure vs time of check valve network search scan in third quadrant.



**Fig. 33.** Pressure vs time of check valve network search scan in **fourth quadrant**.

#### D. Contributions

Prof. Katia Bertoldi led the project's overall coordination. She also played a pivotal role in interpreting the results and outlining the experimentation steps .

Doc Anne Meeussen conceptualized the study, and developed the theoretical framework toward the numerical model. She also was key to onboarding, training, and troubleshooting.

Doc Adel Djelloul contributed to the experimental design and data collection process.

Mr. Jim MacAuthor, a research technician, provided electronic support for our pressure sensor printed circuit board.  
PhD. Jackson Wilt, a researcher in the Lewis Lab, printed the soft actuators.

#### E. Code Availability

The various scripts developed and utilized in this project are available in the various projects' GitHub repositories. Two different repositories were developed, the main *Fluidic Brain* was used to develop all the integration of the various sensor elements used to create the testing platform. The *Combined Environment Control Interface* was developed as open-source tool for the lab to use and profit from the extensive libraries and classes developed for this project to be used in future experiments of their own. This latter one was packaged in a **Docker-Image** to be consistent across all windows machines and the necessary python version and dependencies. Both of these repositories can be found at the following link. <sup>1</sup> <sup>2</sup>

---

<sup>1</sup>[https://github.com/ChristopherJulien/Fluidic\\_Brain](https://github.com/ChristopherJulien/Fluidic_Brain)

<sup>2</sup><https://github.com/ChristopherJulien/Combined-Environment-Control-Interface>

## REFERENCES

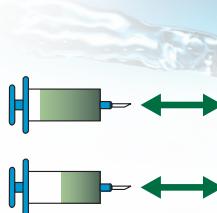
1. T. Freeth, Y. Bitsakis, X. Moussas, *et al.*, "Decoding the ancient Greek astronomical calculator known as the Antikythera Mechanism," *Nature* **444**, 587–591 (2006).
2. A. Bromley, "Charles Babbage's Analytical Engine, 1838," *IEEE Annals History Computing* **20**, 29–45 (1998).
3. J. Shalf, "The future of computing beyond Moore's Law," *Philosophical Transactions Royal Society A: Mathematical, Physical Engineering Sciences* **378**, 20190061 (2020).
4. V. Bush, "The differential analyzer. A new machine for solving differential equations," *Journal Franklin Institute* **212**, 447–488 (1931).
5. K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature* **575**, 607–617 (2019).
6. Y. Jiang, L. M. Korpas, and J. R. Raney, "Bifurcation-based embodied logic and autonomous actuation," *Nature Communications* **10**, 128 (2019).
7. T.-H. Lee, S. Bhunia, and M. Mehregany, "Electromechanical Computing at 500 °C with Silicon Carbide," *Science* **329**, 1316–1318 (2010).
8. B. Tremi, A. Gillman, P. Buskohl, and R. Vaia, "Origami mechanologic," *Proceedings National Academy Sciences* **115**, 6916–6921 (2018).
9. D. Cartwright, *Tides: A Scientific History* (Cambridge University Press, 2000).
10. B. Parker, "The tide predictions for D-Day," *Physics Today* **64**, 35–40 (2011).
11. A. Clymer, "The mechanical analog computers of Hannibal Ford and William Newell," *IEEE Annals History Computing* **15**, 19–34 (1993).
12. "Analog computing returns," (2016).
13. A. Grochowski, D. Bhattacharya, T. Viswanathan, and K. Laker, "Integrated circuit testing for quality assurance in manufacturing: history, current status, and future trends," *IEEE Transactions on Circuits Systems II: Analog Digital Signal Processing* **44**, 610–633 (1997).
14. "The death of CPU scaling: From one core to many – and why we're still stuck," (2012).
15. C. Majidi, "Soft Robotics: A Perspective—Current Trends and Prospects for the Future," *Soft Robotics* **1**, 5–11 (2014).
16. D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics Biomechanics* **5**, 99–117 (2008).
17. K. McDonald and T. Ranzani, "Hardware Methods for Onboard Control of Fluidically Actuated Soft Robots," *Frontiers Robotics AI* **8**, 720702 (2021).
18. S. Song, S. Joshi, and J. Paik, "CMOS-Inspired Complementary Fluidic Circuits for Soft Robots," *Advanced Science* **8**, 2100924 (2021).
19. K. Stanley and R. Miikkulainen, "Efficient evolution of neural network topologies," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2 (IEEE, Honolulu, HI, USA, 2002), pp. 1757–1762.
20. R. L. Truby, M. Wehner, A. K. Grosskopf, *et al.*, "Soft Somatosensitive Actuators via Embedded 3D Printing," *Advanced Materials* **30**, 1706383 (2018).
21. N. W. Bartlett, K. P. Becker, and R. J. Wood, "A fluidic demultiplexer for controlling large arrays of soft actuators," *Soft Matter* **16**, 5871–5877 (2020).
22. M. Wehner, R. L. Truby, D. J. Fitzgerald, *et al.*, "An integrated design and fabrication strategy for entirely soft, autonomous robots," *Nature* **536**, 451–455 (2016).
23. D. J. Preston, P. Rothemund, H. J. Jiang, *et al.*, "Digital logic for soft devices," *Proceedings National Academy Sciences* **116**, 7750–7759 (2019).
24. S. T. Mahon, A. Buchoux, M. E. Sayed, *et al.*, "Soft Robots for Extreme Environments: Removing Electronic Control," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, (IEEE, Seoul, Korea (South), 2019), pp. 782–787.
25. S. Song, S. Joshi, and J. Paik, "CMOS-Inspired Complementary Fluidic Circuits for Soft Robots," *Advanced Science* **8**, 2100924 (2021).
26. S. Song, S. Joshi, and J. Paik, "CMOS-Inspired Complementary Fluidic Circuits for Soft Robots," *Advanced Science* **8**, 2100924 (2021).
27. N. W. Bartlett, K. P. Becker, and R. J. Wood, "A fluidic demultiplexer for controlling large arrays of soft actuators," *Soft Matter* **16**, 5871–5877 (2020).
28. D. J. Preston, P. Rothemund, H. J. Jiang, *et al.*, "Digital logic for soft devices," *Proceedings National Academy Sciences* **116**, 7750–7759 (2019).
29. A. Ben-Israel and T. N. E. Greville, *Generalized inverses: theory and applications*, no. 15 in CMS books in mathematics (Springer, New York, 2003), 2nd ed.

## Operating Conditions

Three operating conditions are available to accommodate a wide range of setups and experimental protocols.

### Independent Condition

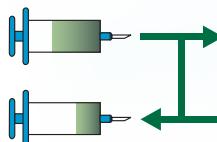
Independent Condition allows the Pump 33 DDS to operate as two separate syringe pumps named P1 & P2. Each syringe will operate independently with different syringe types, size, force, target (volume or time, mode dependent).



	Mode	Syringe	Rate	Target Volume/Time
P1	Infuse, Withdraw, Infuse/Withdraw, Withdraw/Infuse	Any size/type 0.5 µl - 60 ml	Any within syringe capability	Any (Mode Dependent)
P2	Infuse, Withdraw, Infuse/Withdraw, Withdraw/Infuse	Any size/type 0.5 µl - 60 ml	Any within syringe capability	Same as P1

### Reciprocating Condition

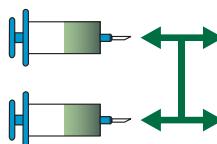
In reciprocating condition, both syringe channels move in opposite directions at the same rate using the same syringe size and type. When combined with a valve box, the reciprocating condition can provide the continuous fluidic delivery of a peristaltic pump with the accurate, pulseless, low flow rates provided by a syringe pump.



	Mode	Syringe	Rate	Target Volume/Time
P1	Infuse/Withdraw, Withdraw/Infuse	Any size/type 0.5 µl - 60 ml	Any within syringe capability	Any
P2	Opposite of P1	Same as P1	Same as P1	Same as P1

### Twin Condition

Twin Condition allows both syringes to operate in the same mode using the exact same syringe type, syringe size, force, target (volume or time) and flow rate settings. The pump also allows the user to combine both flows for higher speed and volume infusion applications.



	Mode	Syringe	Rate	Target Volume/Time
P1	Infuse, Withdraw, Infuse/Withdraw, Withdraw/Infuse	Any size/type 0.5 µl - 60 ml	Any within syringe capability	Any (Mode Dependent)
P2	Same as P1	Same as P1	Same as P1	Same as P1

## Specifications

Type	Microprocessor dual independent infuse/withdraw continuous syringe pump	
Accuracy	±0.25%	
Reproducability	±0.05%	
<b>Syringe:</b>		
Type	Glass, plastic and stainless steel	
Size Minimum	0.5 µl (0.103 mm minimum inner diameter)	
Size Maximum	60 ml (32.573 mm maximum inner diameter)*	
<b>Flow Rate:</b>		
Minimum	1.02 pl/min (0.5 µl syringe, 0.103 mm inner diameter)	
Maximum	106 ml/min (60 ml syringe, 32.573 mm diameter )	
Display	7" color display with touch screen	
<b>Connectors:</b>		
USB	Type B	
RS-232	9-pin D-sub connector	
RS-485	IEEE-1394, 6 pos for pump-pump communication	
TTL Input/Output	Two 15-pin D-sub connectors, one for each pump mechanism	
Footswitch	Two phonojack inputs, one for each pump mechanism	
<b>Average Linear Force</b>		
	70 lbs (31.75 kg) at 100% force setting up to a flow rate of 90 ml/min using up to a 60 ml syringe with a 32.573 mm inner diameter	
	50 lbs (22.6 kg) at 100% force setting for flow rates 90 ml/min to 106 ml/min using the same size syringe	
Power Supply	Input 100-240 VAC, 50-60 Hz, Output 30 V 1.66 A 50 W	

Weight	21 lbs (9.09 kg)
Dimensions (L x D x H)	11 x 15 x 8" (28 x 39 x 21 cm)
Classification	Class I
Pollution	Degree 1
Installation	Category II
Regulatory Certifications	CE, ETL (UL & CSA), CB Scheme, EU RoHS, WEEE

\*NOTE: Some larger syringes may be compatible with the Pump 33 DDS. Please contact Technical Support for more information.

Order #	Product
70-3333	Pump 33 DDS Dual Independent Syringe Pump
70-2215	Footswitch (with phone plug)





www.saleae.com

# Saleae Logic Pro 8 USB Logic Analyzer

## FEATURES

- Powerful, Easy-to-use Software
- Deep Sample Buffers
- Highly Portable, USB Attached
- 24 Included Protocol Analyzers
- Automation API
- Custom Protocol Decoder Plugin API
- Edge and Pulse Width Triggering
- Protocol Result Filter and Search
- Measurements, Bookmarks and Timing Markers
- Four Data Export Formats: CSV, Binary, VCD and MATLAB
- Cross Platform Windows, Linux, and OSX

## APPLICATIONS

- Firmware Debugging
- FPGA Debugging
- Functional Verification
- Performance Profiling
- Reverse Engineering
- Protocol Decoding
- Data Logging

## KEY SPECIFICATIONS

- Eight Digital Channels
- 500 MSPS Digital Sampling (max)
- 100 MHz Max Digital Bandwidth
- Eight Analog Channels
- 50 MSPS Analog Sampling (max)
- 5 MHz Analog Bandwidth
- Recording Length Limited by Available RAM and Density of Recorded Data
- RGB LED, Customizable 24 bit Color
- USB 3.0 Super Speed

## DESCRIPTION

The Saleae Logic Pro 8 USB Logic Analyzer is an 8 channel logic analyzer with each input dual purposed for analog data recording. The device connects to a PC over USB and uses the Saleae Logic Software to record and view digital and analog signals.

A logic analyzer is a debugging tool used to record and view digital signals. It operates by sampling a digital input connected to a device under test (DUT) at a high sample rate. These samples are recorded to a sample buffer, and at the end of the capture, the buffer is displayed in the software for review.

Logic analyzers are great for debugging embedded applications. In the most common case, a developer working on firmware for a microcontroller will write code to communicate with another component, possibly using protocols like serial, I2C, or SPI. To verify the functionality or to diagnose errors in the firmware, a logic analyzer is connected to the digital IO used for communication and records the activity during testing. The recording is then shown on the display so the user can view the actual behavior of the firmware, and compare that with the expected behavior to narrow down and identify the source of the issue – or verify that the operation is correct.

## INCLUDED COMPONENTS

Saleae Logic Pro 8 USB Logic Analyzer, 2x 4 Channel Wire Harnesses, 16 Micro-Gripper Hooks, Saleae Carrying Case, USB 3.0 cable, and a Getting Started Guide

## 1 Sensing Performance

Parameter	SLS-1500	Unit
Full scale flow rate	40	ml/min
Sensor output limit <sup>a</sup>	65	ml/min
Accuracy <sup>b</sup> (whichever error is larger)	5 0.25	% of measured value % of full scale
Repeatability <sup>b</sup> (whichever error is larger)	0.5 0.025	% of measured value % of full scale
Temperature coefficient (additional error per °C; whichever is larger)	0.25 0.00625	% measured value / °C % full scale / °C
Mounting orientation sensitivity <sup>c</sup>	<0.1	% of full scale

Table 1: Performance of SLS-1500 (all data for medium H<sub>2</sub>O, 23°C, 1 bar<sub>abs</sub> unless otherwise noted)

### 1.1 Specification Charts

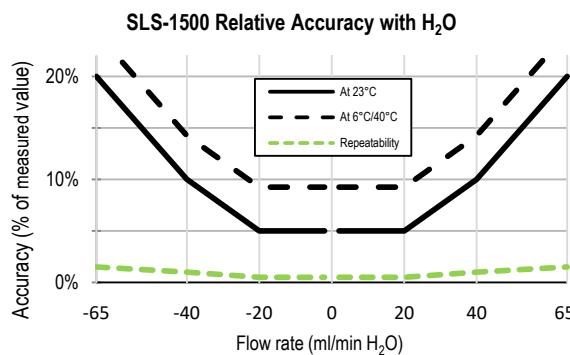


Figure 1: Sensor accuracy and repeatability (% of measured value) across the sensor's flow range

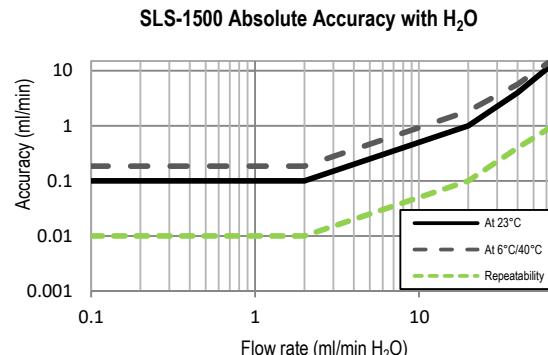


Figure 2: Sensor accuracy and repeatability (ml/min) across the sensor's flow range

<sup>a</sup> Flow rate at which the sensor output saturates, see section 1.2 for performance specification between full scale and saturation point.

<sup>b</sup> Accuracy respectively repeatability below ±20 ml/min. See the charts in section 1.2 for the accuracy respectively repeatability specifications between ±20 ml/min and full scale.

<sup>c</sup> Maximum additional offset when flow channel is vertical.

## INTRODUCTION

# SPECIFICATIONS

SENSOR PERFORMANCE				
Sensor model	XS	S	M+	L+
Calibrated media	Water	Water, IPA FC 40 Mineral oil HFT 7500*	Water, IPA FC 40 Mineral oil HFT 7500*	Water, IPA FC 40 Mineral oil HFT 7500*
Range	0±1.5µL/min	0±7µL/min 0±70µL/min	0±2mL/min	0±40mL/min
Accuracy (m.v.= measured value) also applies to negative values	10% m.v. above 75 nL/min  7.5 nL/min below 75 nL/min	5% m.v. above 0.42 µL/min  21 nL/min below 0.42 µL/min  20% m.v. above 4.2 µL/min  210 nL/min below 4.2 µL/min	5% m.v. above 10 µL/min  0.5 µL/min below 10 µL/min  10% m.v. above 50 µL/min  5 µL/min below 50 µL/min	5% m.v. above 1 mL/min  50 µL/min below 1 mL/min  10% m.v. above 2 mL/min  200 µL/min below 2 mL/min
Lowest detectable flow increment	3.7 nL/min	10 nL/min	/	/
MECHANICAL SPECIFICATIONS				
Sensor diameter	25 µm	150 µm	430 µm	1.4 mm
Max pressure	200 bar	200 bar	100 bar	15 bar
Wetted materials	PEEK & Quartz Glass	PEEK & Quartz Glass	PPS, stainless steel 316L	PPS, stainless steel 316L

\* By default calibration water, but can change using OxyGEN software or FEZ on local mode.

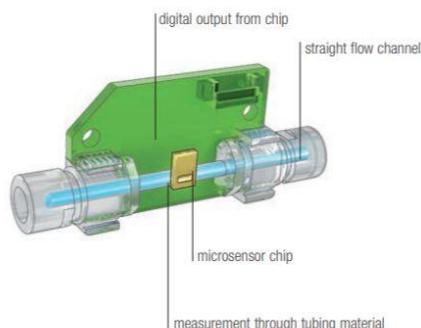
Please note that the maximum pressure depends on the FLOW UNIT model. Ensure that the pressure applied to a FLOW UNIT does not go beyond this value at all times.

The FLOW UNIT suits your own fluid controller. If you use a pressure regulator you may have to enter a maximum pressure below this value. If you use other flow controller, be aware that pressure may go higher than 100 bar very easily and may cause damage to your FLOW UNIT.

## 5 Understanding your Measurements

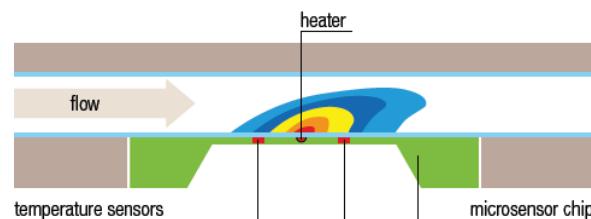
### 5.1 Liquid Flow Meter Operating Principle

Sensirion's highly advanced CMOSens® technology combines all associated electronics from signal conditioning circuitry to calibration memory and MEMS micro heater structure on a single integrated CMOS chip. The extremely compact chip creates a small thermal mass for very fast responses and consumes very little power. The chip is bonded to the outside of the flow channel and due to its media isolated sensing principle, there is no direct contact between the digital microchip and the fluid. Measurements are taken through the flow channel wall.



Schematic layout of a liquid flow meter

Your Sensirion liquid flow meter operates on a microthermal sensing principle. A negligible amount of heat is introduced to the fluid and monitored to correlate temperature differentials to actual flow. Temperature sensors before and after the heater detect temperature changes as flow transfers heat from one side to another. The illustration below highlights the temperature profile during flow:



Fluid thermodynamic properties (i.e., heat capacity, thermal conductivity) will directly affect sensor response. The standard sensor calibrations for H<sub>2</sub>O and IPA cover most commonly used fluids.

It is important to maintain laminar flow within the capillary during measurements. The flow meter maximum flow ranges have been specified having this in mind. Use inlet tubing with the same inner diameter or larger than the sensor's capillary inner diameter. More viscous fluids reduce turbulence although one must use caution to avoid exceeding the recommended pressure limits of the flow meter (see the datasheet for specifications of recommended maximum operating pressure and burst pressure).



## 2 COMMAND OVERVIEW

### 2.1 SENSOR CABLE COMMANDS

This commands are available for all sensor products.

ID	Bytes send	Bytes receive	Name	Comment	Pw level	Storage
0xD0	1	String	Get Device Information	Get name, article code and serial number of RS485 Sensor Cable	0	-
0xD1	1	7	Get Version	Get Firmware/Hardware/SHDLC version	0	-
0xD3	0	0	Device Reset	Execute a reset on RS485 Sensor Cable	0	-
0x90	0 / 1	1 / 0	Device Address	8 Bit Address of RS485 Sensor Cable	0	E
0x91	0 / 4	4 / 0	Baudrate	Baudrate of RS485 Interface	0	E
0x92	0	0	Factory Reset	Set back all settings to default values	0	E
0x93	0	4	System up Time	Get the time since device is powered up or reset	0	R
0x20	0 / 1	1 / 0	Termination	Enable or disable the Termination resistor	0	E
0x21	1 / 21	21 / 0	User Data	Save 20 bytes of Userdata in EEPROM	0	E
0x22	0	2	Device Selftest*	Execute an selftest with device	0	-
0x23	0 / 1	1 / 0	Sensor Voltage	Defines the sensor supply voltage	0	E
0x24	0 / 1	1 / 0	Sensor Type*	Defines the sensor type	0	E
0x25	0 / 1	1	Sensor Address*	I <sup>2</sup> C address for sensor access	0	E
0x26	0	2	Measure Sensor Voltage	Measure the sensor supply voltage of RS485 Sensor Cable	0	-
0x27	0/2	2/0	Reply Delay	Set a reply delay for RS485	0	E
0x28	0/2	2/0	I2C Delay	Set I2C communication delay	0	E
0x29	1	0...128	Scan I2C Address	Scan all I2C addresses for Ack	0	-
0x2A	5...205	0...200	I2CTransceive	Generic I2C Transceive	0	-



## 2.2 SF04 FLOW SENSORS

This commands are available for flow sensor products based on the SF04 chip used for flow meters and differential pressure sensors. (Sensor type = 0)

### 2.2.1 MEASUREMENT COMMANDS

ID	Bytes send	Bytes receive	Name	Comment	Pw level	Storage
0x30	0	1	Sensor Status	Get the status of sensor and continuous measurement.	0	-
0x31	0	0	Start Single Measurement*	Start single measurement	0	-
0x32	0	0 / 2	Get Single Measurement	Read out measurement from sensor if finished	0	-
0x33	0 / 1 / 2 / 3	0 / 2	Start Continuous Measurement*	Start continuous measurement with optional interval and resolution	0	-
0x34	0	0	Stop Continuous Measurement	Stop continuous measurement	0	-
0x35	0 / 1	0 / 2	Get Last Measurement	Read out last measurement while continuous measurement	0	-
0x36	0 / 1	0...254	Get Measurement Buffer	Read out all measurements from buffer	0	-
0x37	0 / 1	1 / 0	Totalizator Status	Enable or disable the totalizator,	0	-
0x38	0 / 1	8 / 4	Totalizator Value	Get the value of the totalizator or duration	0	R
0x39	0	0	Reset Totalizator	Set the totalizator value to zero	0	-
0x3B	15/6	0	Start Auto Detection Measurement	Start auto detection measurement	0	-
0x3C	0/2/38	38/0	Advanced Measurement Configuration	Set advanced measurement configuration	0	-

### 2.2.2 SENSOR SETTINGS

ID	Bytes send	Bytes receive	Name	Comment	Pw level	Storage
0x40	0 / 1	1 / 0	Measurement Type*	Measurement type (Flow/Temp/Vdd)	0	R
0x41	0 / 1	1 / 0	Resolution*	Resolution of flow, temperature, and Vdd measurement	0	SR
0x42	0 / 1	1 / 0	Heater Mode*	Heater mode for the flow sensor	0	SR
0x43	0 / 1	1 / 0	Calib Field*	Calibration field of the flow sensor	0	SR
0x44	0 / 1	1 / 0	Factory Settings*	Factory settings of the flow sensor	0	SR
0x45	0 / 1	1 / 0	Linearization*	Linearization of measurement	0	SR



### 2.2.3 SENSOR INFORMATION

ID	Bytes send	Bytes receive	Name	Comment	Pw level	Storage
0x50	0	21	Sensor Part Name*	Part name of the sensor	0	SE
0x51	0	13	Sensor Item Number*	Item number of the sensor	0	SE
0x52	0	2	Flow Unit*	Flow unit of sensor	0	SE
0x53	0	2	Scale Factor*	Scale factor of active measurement type and calibration field	0	SE
0x54	0	4	Sensor Serial Number*	Sensor serial number	0	SE
0x55	0	1	Measurement Data Type*	Get the data type of the measurements (signed or unsigned)	0	SE

### 2.2.4 ADVANCED SENSOR COMMANDS

ID	Bytes send	Bytes receive	Name	Comment	Pw level	Storage
0x65	0	0	Sensor Reset*	Execute a reset on the sensor	0	-
0x66	0/1...n	101/0	Autostart	Define the command sequence to be executed after powerup	0	E

\* Sensor must be idle for execution of this command

E: Eeprom RS485 Sensor Cable (if a value is set, the continuous measurement is interrupted while value is written to Eeprom)

R: RAM RS485 Sensor Cable

SR: Sensor Register

SE: Sensor Eeprom

For additional output filtering, refer to Application Note AN1646<sup>[1]</sup>.

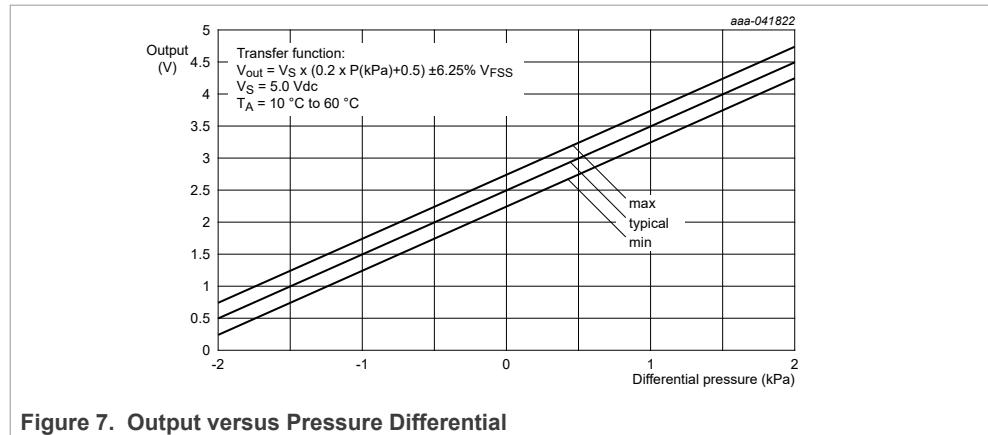


Figure 7. Output versus Pressure Differential

## 9.2 Pressure (P1)/Vacuum (P2) Side Identification Table

NXP designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing a gel die coat which protects the die from harsh media.

The Pressure (P1) side may be identified by using [Table 6](#).

Table 6. Pressure side identification

Part Number	Case Type	Pressure (P1) Side Identifier
MPXV7002GC6U/GC6T1	482A-01	Side with Port Attached
MPXV7002GP	1369-01	Side with Port Attached
MPXV7002DP	1351-01	Side with Part Marking

## 9.3 Minimum Recommended Footprint for Surface Mounted Applications

Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package. With the correct footprint, the packages self-align when subjected to a solder reflow process. NXP recommends designing boards with a solder mask layer to avoid bridging and shorting between solder pads.

NXP Semiconductors

MPXV7007

Integrated silicon pressure sensor, on-chip signal conditioned, temperature compensated and calibrated

For additional output filtering, refer to Application Note AN1646<sup>[1]</sup>.

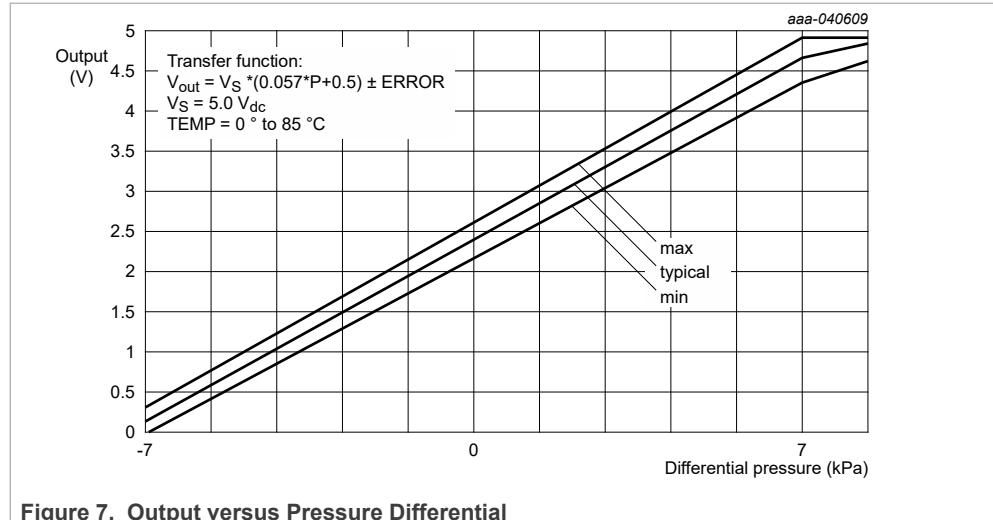


Figure 7. Output versus Pressure Differential

### Transfer function

Nominal transfer value:

$$V_{out} = V_S \times (0.057 \times P + 0.5) \pm (\text{Pressure Error} \times \text{Temp. Factor} \times 0.057 \times V_S)$$

$$V_S = 5.0 \text{ V} \pm 0.25 \text{ Vdc}$$

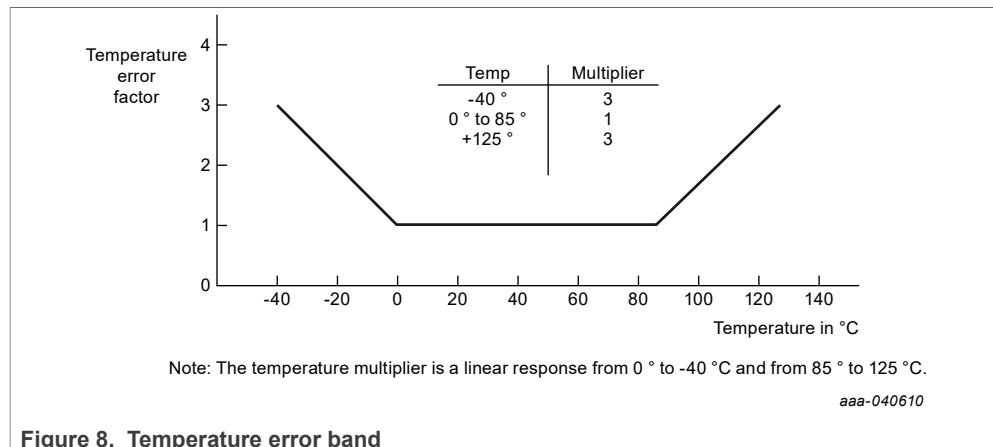


Figure 8. Temperature error band



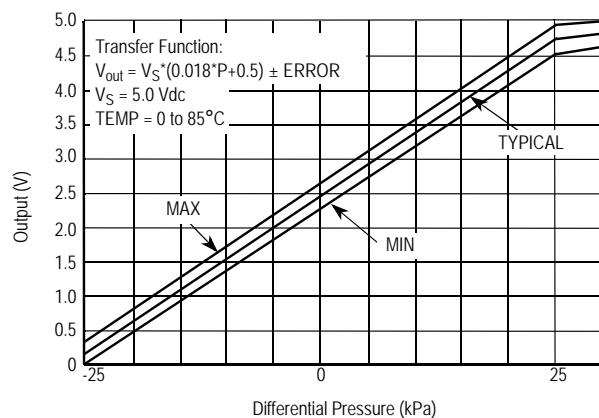
## On-chip Temperature Compensation and Calibration

The MPXV7025 series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor performance and long-term reliability. Contact the factory for information regarding media compatibility in your application.

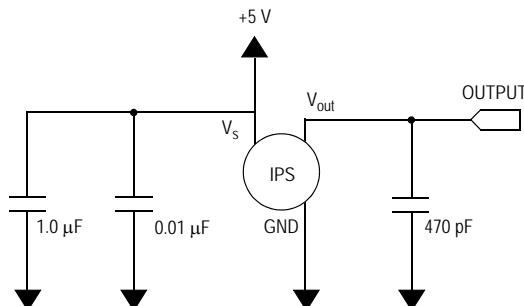
**Figure 2** shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output

curves are shown for operation over a temperature range of 0° to 85°C using the decoupling circuit shown in **Figure 3**. The output will saturate outside of the specified pressure range.

**Figure 3** shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.



**Figure 2. Output versus Pressure Differential**



**Figure 3. Recommended Power Supply Decoupling and Output Filtering**

(For additional output filtering, please refer to Application Note AN1646.)

