The Development of a Low-Cost Spirometer

Asa Garner, acey4000@gmail.com
Pilgrim School, Class of 2021
USC Viterbi Department of Aerospace and Mechanical Engineering, SHINE 2020

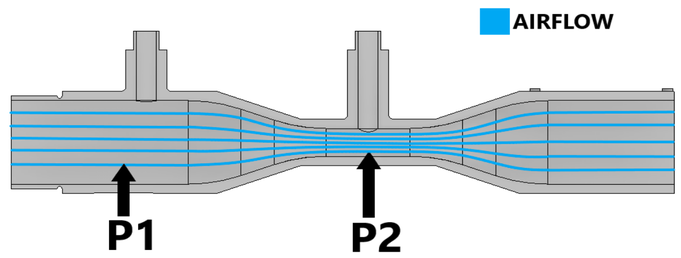
Introduction

Ultimately, the goal of a spirometer is to measure a patient's lung capacity. Since it gives readings over time, not only can it give the total velocity, but it can also measure the sustained airflow volume. This is relevant to the issues of today- possible COVID patients and recovering COVID patients alike must be diagnosed and treated properly, and measuring lung function is an important part of effectively taking care of patients throughout the course of the illness. In times of high demand, access to spirometers that are cheap, accessible, and retain precision is extremely important. Therefore, my research involved the development of a low-cost alternative to some of the more expensive options available today. This included designing the spirometer using CAD, 3D printing it, modifying it to improve flow quality, and integrating software to produce a useful output.

Objective & Impact of Professor's Research

The research conducted in Professor Luhar's Fluid Interactions Lab has very wide-reaching applications. In the past, this has included aeronautical design, analysis of various fluid related industrial applications, and more. However, current circumstances have led to a focus on computationally mapping infectious disease transmission and the development of airflow related devices to help prevent, cope, and mitigate the effects of viral infectious diseases such as COVID-19. The gap between these distinct scientific topics is bridged by fluid mechanics.

How Does the Spirometer Work?



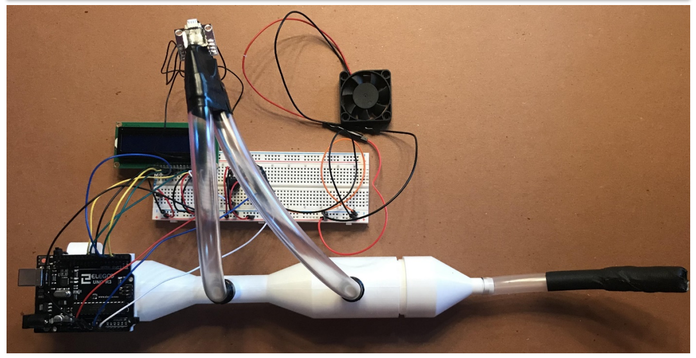
$$v_2^2 = \frac{P_1 - P_2}{\frac{1}{2}\rho\left(1 - \left(\frac{A_2^2}{A_1^2}\right)\right)}$$

ρ = Rho, air density
v = Airflow kinetic energy
A = Area (difference)
P = Pressure

```
inputVolt = analogRead(analogInPin); // Voltage read in (0 to 1024)
pressure_pa = read_pressure_pa(inputVolt, offset_pressure);
volFlow = (sqrt((abs(1000*pressure_pa)*2/rho)/(1-rarea)))*area_2; // Volumetric flow of air
massFlow = (volFlow*rho); // Massflow of air
//volume = volFlow*dt + volume; // Total volume (essentially integrated over time)
delay(100);
```

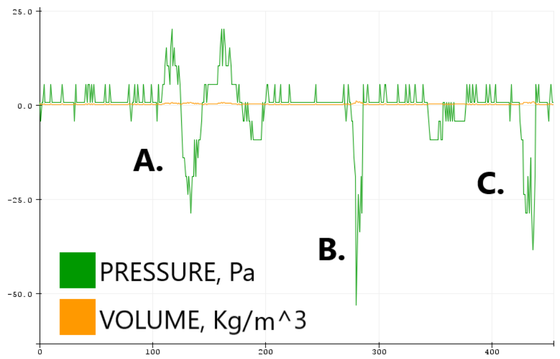
The cross section above describes the basics of a venturi tube: as air flows through the tube (from left to right), it enters through ***P1*** and is constricted through ***P2***. A measurement of pressure at the two points using a differential pressure sensor (located up the pipes seen in the cross section) is sent to an Arduino Uno board that is attached to the spirometer body. Based on the reading and constants for the geometry- specifically the area of the two sections- airflow velocity and volume are calculated on the Arduino using Bernoulli's Equation, also included above. These outputs can be displayed through an LCD connected to the Arduino via a breadboard. Below the equation is corresponding code that generates numerical values for the data output that is then sent to the data plotter and the LCD.

The Spirometer

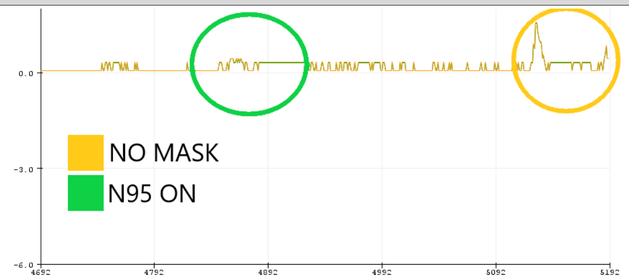


Pictured at right is the finalized 3D printed spirometer body with all the electronics connected. The mouthpiece is on the bottom right, and this can be taken off and swapped with the computer fan on the top right for airflow benchmarking purposes.

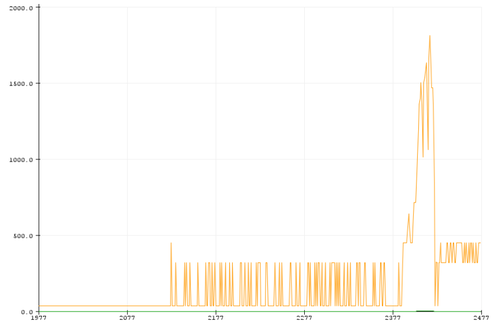
Spirometer Data



This graph depicts spirometer data acquired from the differential pressure sensor hooked up to the venturi tube. The values have been calibrated by converting to corresponding airflow velocity values and then checking these readings with the use of a digital anemometer. Depicted at ***Point A***. are pressure readings of the differential created by me breathing in and out. ***Point B***. is my sharp breath into the tube. ***Point C***. is my more gradual breath. Being able to visualize this patient data allows it to be put alongside a benchmark, and lung function can be measured through comparison.



This graph of airflow measured from the spirometer shows the importance of wearing a mask amidst COVID, the use of which significantly decreases the output of airflow volume.



Here are some exhalation results I achieved by blowing through the spirometer (ml/s).

Skills Learned

SHINE has taught me a massive amount about the nuances of fluid mechanics, and the wide range of applications throughout engineering. Through presentations, demonstrations, and applying the theories, I've learned a lot about the fundamental laws of fluid interactions that govern most CFD software. Beyond the theoretical knowledge, I was able to learn more about applying these skills in a scientific environment. I was also able to apply my own skills in new and exciting ways. 3D printing is something I already knew and did regularly, but the design and production of the spirometer hardware opened my eyes to the more medically oriented advantages of 3D printing. The place where the most of my knowledge had to be tested and applied was through the programming of the microcontroller- in order to calculate airflow velocity and volume I had to not only get familiar with the corresponding CFD equations but learn how to integrate them into a program. Structuring this all and producing a viable output has given me a very strong introduction to microcontroller use that will be useful in pretty much all aspects of engineering.

Future Ways to Improve the Low-Cost Spirometer

Although much has been done to optimize the pressure sensor readings through software, accuracy is a constant challenge with cheaper differential pressure sensors due to the moving parts they introduce. A possible solution to this would be completely changing the sensors to ultrasonic transducers, which are cheaper in part due to the lack of moving parts. Since sound has a fixed speed in air, changing the speed of air measured by the sensors would be reflected in the data.

Acknowledgements

I want to thank Professor Luhar, my mentors Andrew Chevarin and Shilpa Vijay, my lab mate Maddie Yee, my family for support, and finally I want to thank Dr. Mills, Cathalina, and the rest of the SHINE team for setting this whole program up, and then finding a way to safely run it amidst a global pandemic.