Lab 6 – Spirometer System

GOAL

Demonstrate a spirometer system incorporating a (1) Lilly-type flow tube (2) piezoresistive differential pressure sensor (3) instrumentation amplifier and low-pass filter (4) analog-to-digital converter (5) MATLAB processing and display.

OBJECTIVES

1) Build and test the electronics to measure the differential pressure from the Lilly-type flow tube.

2) Develop the Arduino and MATLAB interface to display flow-time and volume-time curves obtained with a Lilly-type spirometer.

GENERAL GUIDELINES

1) Each student must build his/her own circuit.
2) Students are allowed (even encouraged) to help each other. Of course, Buma will be around to provide assistance as well.

REQUIRED PARTS AND MATERIALS

- Lab kit
- Scope and probe kit
- Computer with Arduino IDE and MATLAB
- Spirometer flow head + MPX2010DP sensor + wiring
- Bacterial filter and mouthpiece
- 10V benchtop power supply and banana cables
- AD620 instrumentation amplifier (one)
- TL081 op amp (one) (you should already have one in your lab kit)
- 30 ohm resistor (orange/black/black) (one)
- 1 kohm resistor (brown/black/red) (one) (you should already have one)
- 3.3 kohm resistor (orange/orange/red) (one)
- 100 kohm (brown/black/yellow (one) (you should already have one)
- 0.1 uF capacitor (tan colored) (one) (you should already have one)

PART 0: INTRODUCTION

The spirometer measurement system will use a Lilly-type flow head, the MPX2010DP sensor, an instrumentation amplifier and low-pass filter, the Arduino for data acquisition, and MATLAB for data processing. Here's how it works:

1) The flow head:
    - Air flow causes a slight pressure difference ΔP between the two sides of the mesh screen.
2) The sensor convers ΔP into a differential voltage ΔV.
3) The sensor output ΔV is then:
    - Amplified by the AD620 instrumentation amplifier.
    - Smoothed (low-pass filter) by the TL081 op amp.
4) The low-pass filter output is recorded by the Arduino and sent to a computer.
5) The computer processes the data (using MATLAB).
    - An upper plot shows exhaled flow vs time.
    - The lower plot shows exhaled volume vs time.

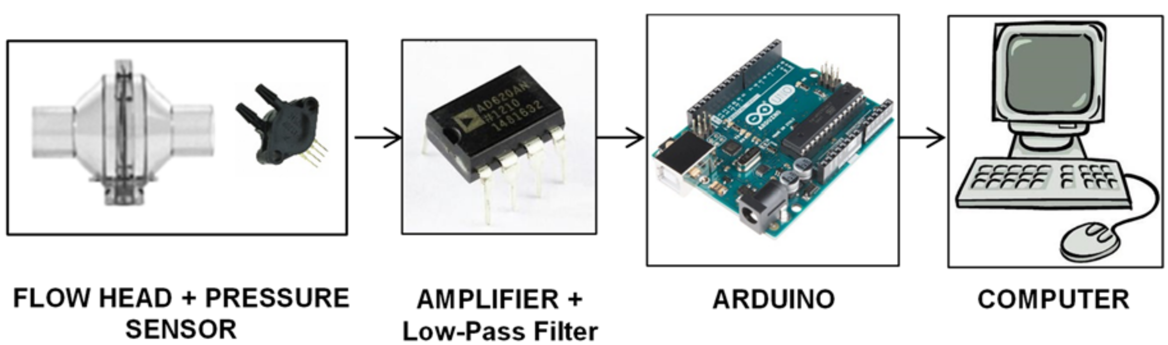


Fig. 1: Simplified block diagram of the spirometer measurement system.

PART 1: INSTRUMENTATION AMPLIFIER AND LOW-PASS FILTER

The purpose of this section is to make a "home-built" spirometer. The MPX2010DP differential pressure sensor has already been attached to the flow head (Buma is so kind ….). A four-wire cable is the interface between the sensor and your breadboard. The setup may not look very pretty, but it does give reasonable results after proper calibration!

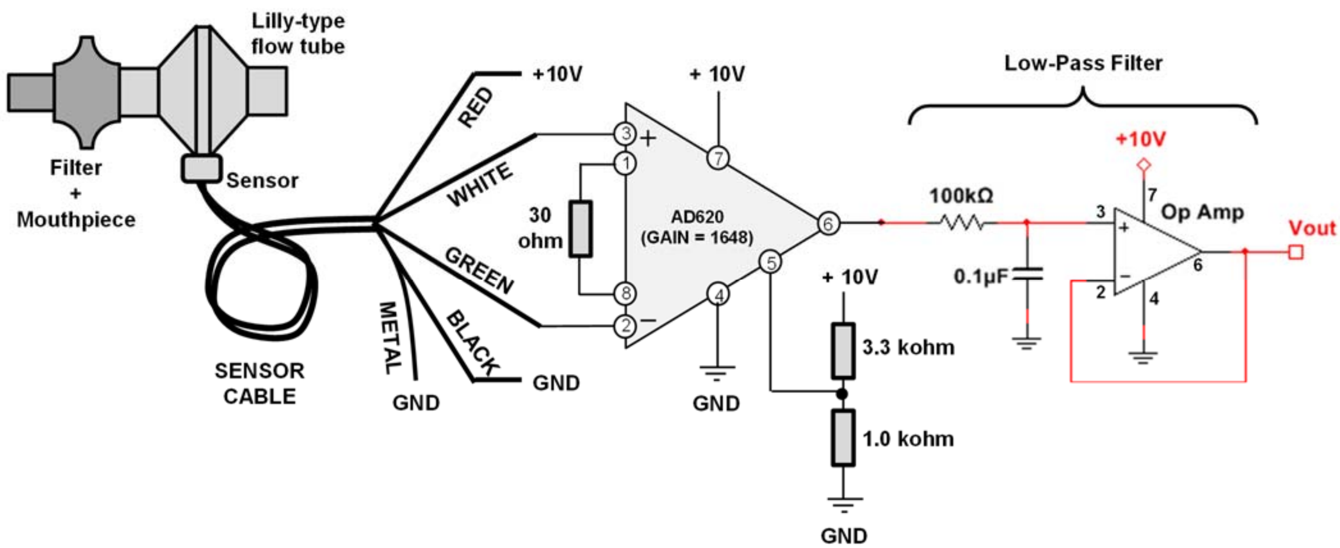


Fig. 2: Electronics for flow head pressure sensor, instrumentation amplifier, and low-pass filter.

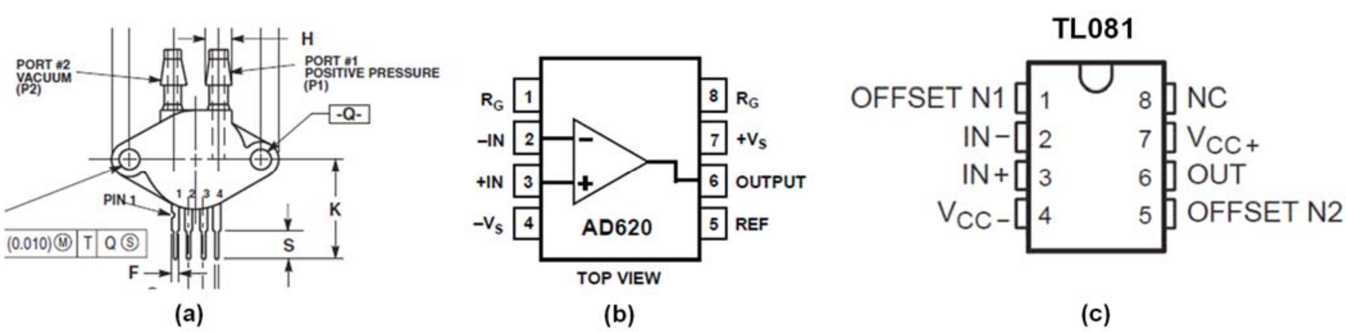
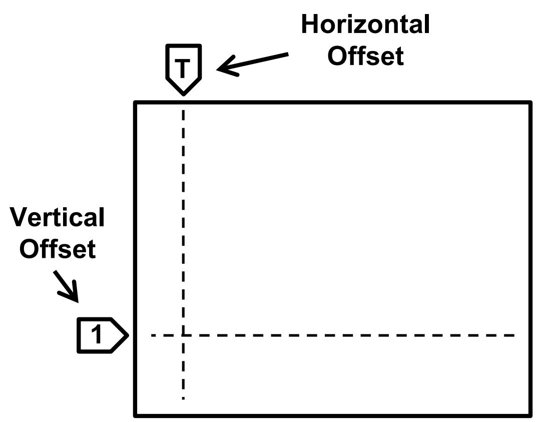


Fig. 3: (a) External appearance of the MPX2010DP pressure sensor (b) Pin diagram of the AD620 instrumentation amplifier (c) Pin diagram of the TL081 op amp.

- <u>**Step 1a**: Build the circuit in Fig. 2.</u>
  - Use NEAT and COLOR-CODED wiring!
    - Red = +10V Black = GND Yellow = other
    - Feel free to consult Buma's board as an example.
  - The pin diagram for the AD620 and TL081 are shown in Fig. 3.
  - Due to limited supplies, you may have to share the flow head + cable assembly.

- Please be GENTLE with the flow head and cable assembly.
- The wiring is given by:
  - BLACK = GND
  - WHITE = Pin 3 (+ signal input) of AD620
  - RED = +10V
  - GREEN = Pin 2 (- signal input) of AD620
  - METAL = GND

- <u>**Step 1b**: Measure the output of the AD620.</u>
  - Attach a scope probe to the output (Pin 6) of the AD620.
  - Turn on the scope. If necessary, press the "Force" button to enable local (e.g. Front Panel) control.
  - Press the "Default Setup" button to reset the scope.
    - Make sure both the scope and probe are set to "1X".
    - Vertical settings:
      - Scale = 0.5V/div
      - Offset: scroll down the waveform so it is near the bottom of the screen.
    - Horizontal settings:
      - Scale = 500 ms/div
      - Offset: scroll the waveform so that the "T" marker is near the left edge of the screen.



  - Push the "TRIG MENU" button, go to "SOURCE" and select "AC Line".
  - Press the "Run/Stop" button so that it appears RED.

- <u>**Step 1c**: Perform a forced expiratory maneuver (FEM) with the spirometer</u>.
  - Comments about the flow head:
    - For hygienic reasons, it is a good idea to use a separate mouthpiece for each person who exhales into the spirometer.
    - You will need to attach a nose clip for the FEM.
    - <u>**Try to keep the spirometer head as still as possible during the FEM (e.g. don't tilt it up and down or wave it around).**</u>
  - Acquisition procedure:
    - Press the "Single" button on the scope and quickly take a deeeeep breath.

- Bring the mouthpiece to your lips and exhale as quickly as humanly possible!
- Try not to inhale again until the scope trace is done (takes 5 seconds).
- The scope trace should look like a pulse with some "fuzz" (see Fig. 4a).
- **Save this scope trace for your lab report (e.g. use "swave" in MATLAB)**.

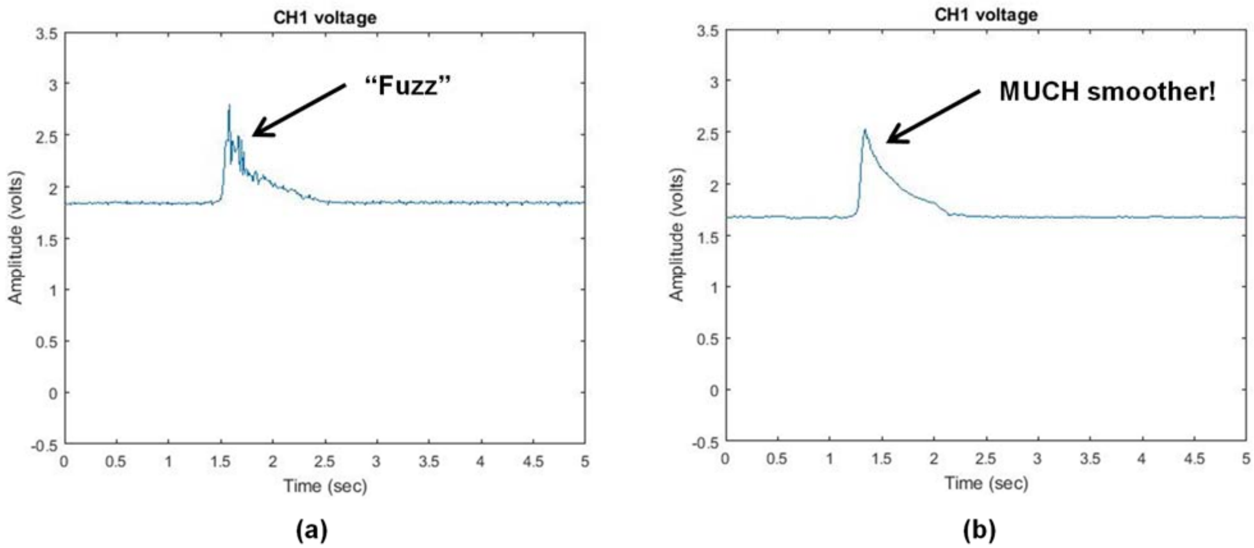


Fig. 4: (a) Scope trace of AD620 output should show a pulse with some "fuzz". (b) The output of the TL081 op amp should be MUCH smoother due to the low-pass filter.

- **Step 1d**: Measure the output of the low-pass filter.
  - Move the scope probe to the output of the TL081 op amp.
  - Repeat the FEM procedure.
    - The scope trace should look much cleaner (see Fig. 4b)!
    - **Save this scope trace for your lab report (e.g. use "swave" in MATLAB)**.
  - Once you have saved both scope traces, go ahead to Part 2.

(End of Part 1)

PART 2: ARDUINO

For this lab, we want to record 5 seconds of data from analog input pin (A0). We'll choose a sampling interval of dt = 10 ms and N = 500 samples.

- **Step 2a**: Starting with the Lab 5 (blood pressure) Arduino code, modify it to do the following:
  - Time between readings is T = 10 milliseconds.
  - Number of samples per trace is N = 500.
- **Step 2b**: Upload your code and observe the Arduino output using the **Serial Monitor** on the computer.
  - Make sure the bottom right of the window is set to "57600 baud".
  - You should notice the letter "**a**" on the first line.
  - Type in the letter "**y**" in the command line and press the **Send** button.
    - You should see a rapid burst of voltage values when you send 'y'.
    - The Arduino's yellow Tx LED should be blinking rapidly during the data burst.
  - Remember to save your program!

(End of Part 2)

PART 3: MATLAB (2 programs)

This week, we will once again write TWO MATLAB programs! ☺ The first program acquires data from the Arduino. The second program analyzes the measured voltage to display FLOW and VOLUME curves.

- **Step 3a**: Starting with your MATLAB code from Lab 5, make the following changes:
  - The number of samples is **N** = 500.
  - The sampling interval is **dt** = 0.010 seconds.
  - Save this program as something like "Lab6_acquiredata.m".

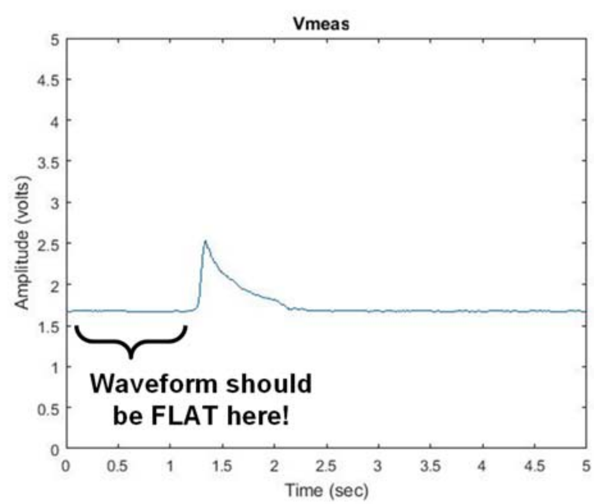


Fig. 5: The MATLAB plot of Vmeas should look something like this.

- **Step 3b**: Perform a trial data acquisition:
  - Start your MATLAB code … WAIT ≈ 2 SECONDS … take a deeeeep breath and blow FORCEFULLY into the spirometer!
    - Hopefully you will see a figure appear that looks something like Fig. 5.
    - You want the waveform to be FLAT during the first second (or slightly longer).

- **Step 3c**: Create a new MATLAB program (New >> Script) to display **Flow** and **Volume** curves.
  - Call this program something like "Lab6_DataProcessing.m".
  - First, convert **Vmeas** to **Flow**.
    - Define flow resistivity **R** (kPa·s/L), sensor sensitivity **S** (V/kPa) and amplifier gain **Ad**.
    - Define the reference voltage **Vref** as 1.5 and **Calibration** as 1.0 for now (tweak this later).
    - Convert **Vmeas** to **Flow** using your formula from PreLab6.
  - Next, integrate **Flow** to obtain **Volume**.
    - For the scope of this course, it is sufficient to use a "cumulative sum" to integrate flow:
      - This is the **cumsum** command in MATLAB.
      - What do you have to do to approximate an integral using the **cumsum** command?
        - Remember that **cumsum** simply adds together values (see MATLAB help).
        - Think about how to approximate an integral as a discrete sum.
        - Hint: The time increment **dt** plays an important role.
  - Create a new figure window showing both **Flow** and **Volume** curves (e.g. one above the other).
    - Remember to use the **subplot(2,1,1)** and **subplot(2,1,2)** commands to do this.
  - Run your code to see if you get any errors.
    - Your code works if you get a figure showing two plots (flow vs time and volume vs time).
    - The volume plot will probably look weird, but we'll fix that with the zeroing step.
- **Step 3d**: Zero the system (find correct **Vref**).
  - Re-run your data processing MATLAB code as many times as needed to make the **Volume** plot be ZERO and FLAT before the FEM occurs (see Fig. 6).
    - Your **Vref** is too small if the volume is steadily increasing with time (positive slope).
    - Your **Vref** is too large if the volume is steadily decreasing with time (negative slope).
    - The proper **Vref** produces a completely HORIZONTAL waveform during the first second!

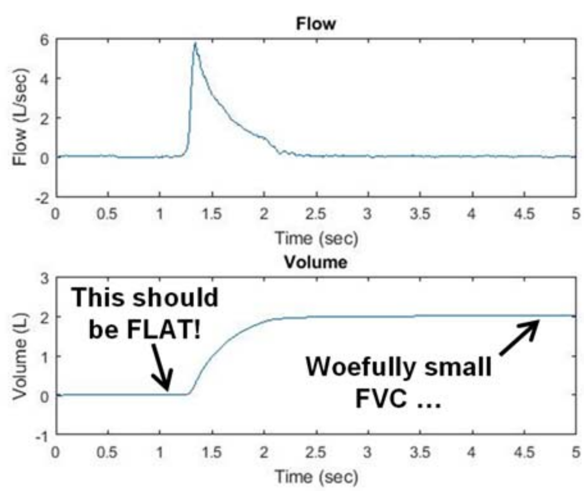


Fig. 6: The proper value of Vref will cause the volume curve to be ZERO and FLAT before the FEM occurs.

    - You will probably find that your forced vital capacity (FVC) is disappointingly low (it should be 4 to 6 liters). Do not worry! We'll fix this soon with the calibration step.

- **Step 3e**: Calibrate the system
  - This involves finding the proper value of **Calibration**.
    - This is similar to the calibration step in Lab 1, where a known 200 g weight is supposed to produce a displayed weight close to 200 g.
    - For this lab, we will inject a known volume of air into the spirometer using a **3 liter syringe** (Ellis Hospital was nice enough to give us an old calibration syringe ☺).
    - Carefully attach the syringe tip to the flow head (remove the mouthpiece and filter).
      - You can use some tape to seal the connection (e.g. prevent air leaks).
      - Let the syringe rest horizontally on the bench top.
    - Perform a calibration experiment by doing the following:
      - Pull the syringe knob all the way back.
      - Start your MATLAB data acquisition code.
      - WAIT ABOUT 2 SECONDS.
      - Now inject all the air into the spirometer (do this in about one second).
    - Next, run your MATLAB data processing code.
      - Think about how to interpret your "flow" and "volume" plots in order to adjust the **Calibration** value in your code.
        - Hint: What known quantity are you injecting into the spirometer?
      - Re-run your second MATLAB program as many times as needed until you get the correct result.
        - Depending on the system, your **Calibration** value will most likely be somewhere between 1 and 3.
        - **Save your syringe flow and volume plots for your lab report.**

- **Step 3f**: Make a new figure showing a plot of Flow vs. Volume
  - Create a new figure window (**figure** command).
  - Think about how to use the **plot** command to plot flow vs volume.
  - Your plot should look something like Fig. 7.
    - NOTE: Although there is no explicit time axis, you start at (V, F) = (0, 0) and move to the right as time elapses.

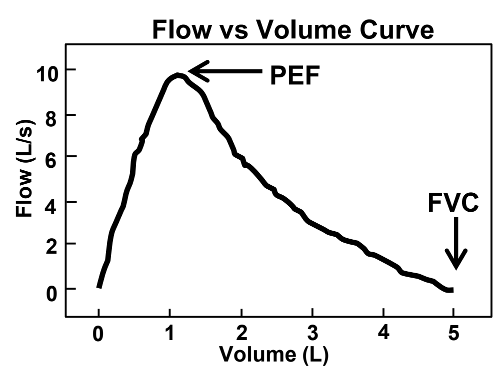


Fig. 7: Example of Flow vs Volume curve. The peak of the curve is PEF, while the right-most point is the FVC.

(End of Part 3)

PART 4: SPIROMETER MEASUREMENT

- **Step 4a**: Now that your system is zeroed and calibrated, it is time to measure yourself!
  - Replace the syringe with the mouthpiece + filter, then perform an FEM experiment.
  - When you run your second MATLAB code (data processing), you may need to slightly change the value of **Vref** to ensure a flat waveform during the first second.
    - **Vref** tends to drift a little bit with time, so a slight modification is perfectly normal.
  - Based on your plots, what is your PEF (peak expiratory flow) and FVC (forced vital capacity)?

- **Step 4b**: Your lab report needs the following:
  (1) Scope traces (like Fig. 4).
  (2) Plots of **Flow** vs time and **Volume** vs time from the SYRINGE (Step 3e).
  (3) Values of **Vref** and **Calibration**.
  (4) Spirometer plots of YOU: **Flow** vs time, **Volume** vs time (like Fig. 8), and **Flow vs Volume** (like Fig. 7).
  (5) Your PEF and FVC.
    - **Show your plots to Buma**.
    - Buma will tabulate everybody's peak expiratory flow (PEF) and forced vital capacity (FVC) and declare the respective winners!

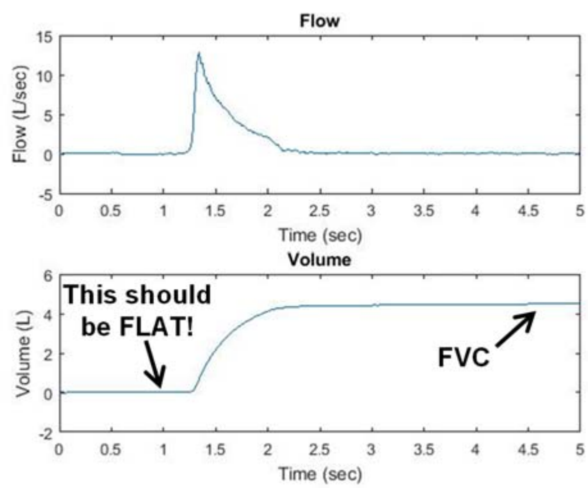


Fig. 8: When you breathe into the spirometer, your flow and volume curves should look something like this. Save the Calibration and most recent Vref values for your report.

(End of Lab 6)