## Flow control setup for a single nozzle:

Experimental setup was arranged according to the Figure 1, shown below. Our first step is to control the flow rate through the nozzle according to our demand. Flow meter is attached before the nozzle solenoid to get the flow meter reading and adjust the duty cycle of the nozzle solenoid according to the error produced in our system. The control flow diagram of our system is shown in Figure2.

As currently we don’t have the pressure transducer, so we are actuating bypass solenoid by taking inverse pwm value of the nozzle solenoid, e.g; if the duty cycle of nozzle solenoid is 40% then the bypass solenoid will actuate with 60% duty cycle.

In this step, we are not considering the constant pressure.

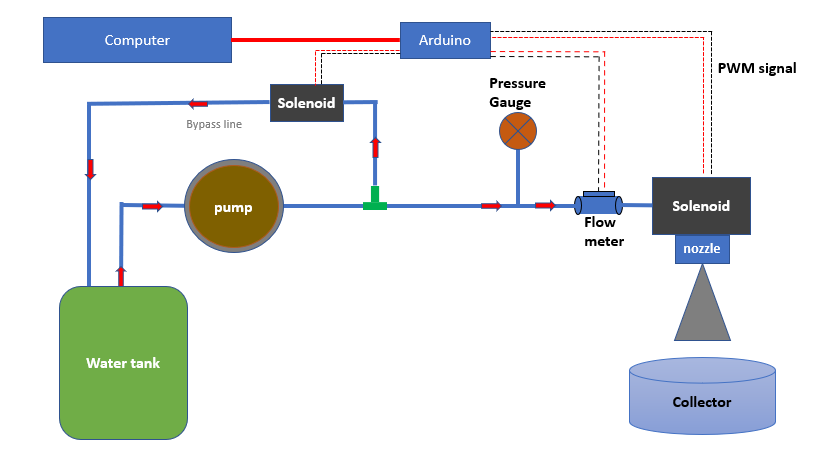


Figure :Experimental setup for one nozzle

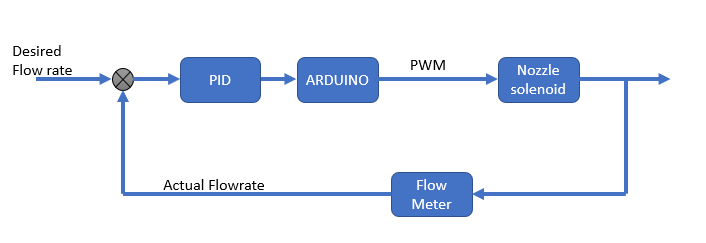


Figure :Flow control diagram

Arduino micro-controller is used to compute the PID control algorithm and then it computes the duty cycle according to the PID control signal.

**First experiment:**

we plotted experimental data in excel and generated the graph as shown in Figure 3, in this experiment Kp value is set at 50 ,ki and kd=0. The orange graph shows the desired flow rate, potentiometer is attached to change the desired flow rate. The blue graph shows the error between the desired and actual flow rate.  
As we can see in the graph that error gets zero when the desired flow rate is below 1000 mL/min because the nozzle can allow maximum flow rate upto 1200mL/min.  
There are a lot of fluctuations in the system, PID gains need to be tuned further.

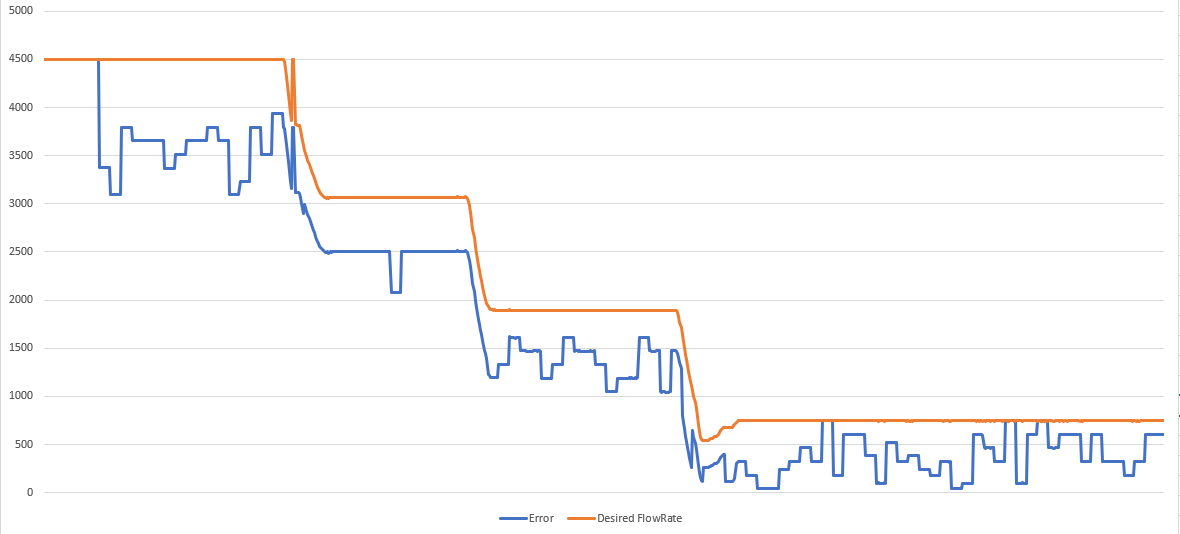
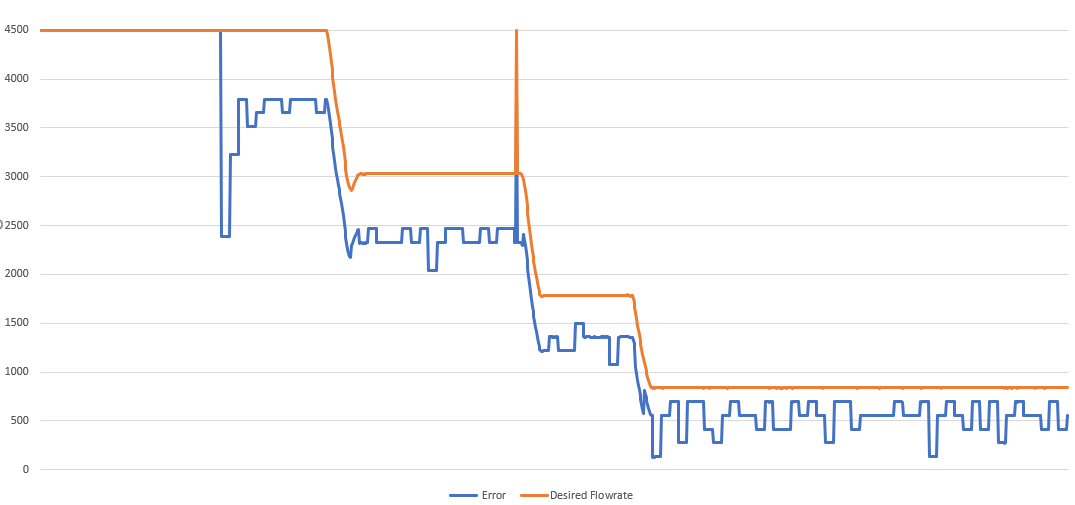


Figure : Flow control results

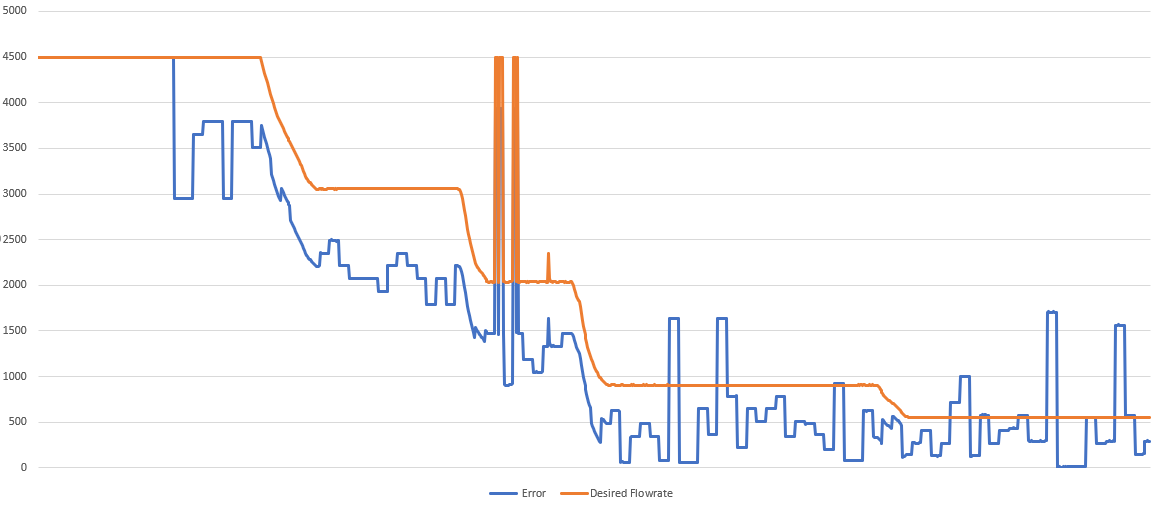
**Second experiment:**

In this experiment kp=80, kd and ki=0 is used. The results are better than the previous one. It has less oscillations as compared to the previous one.



**Third experiment:**

In this experiment kp=80, kd=10 and ki=0 is used. We can see the result are oscillating more as compared to the previous ones.

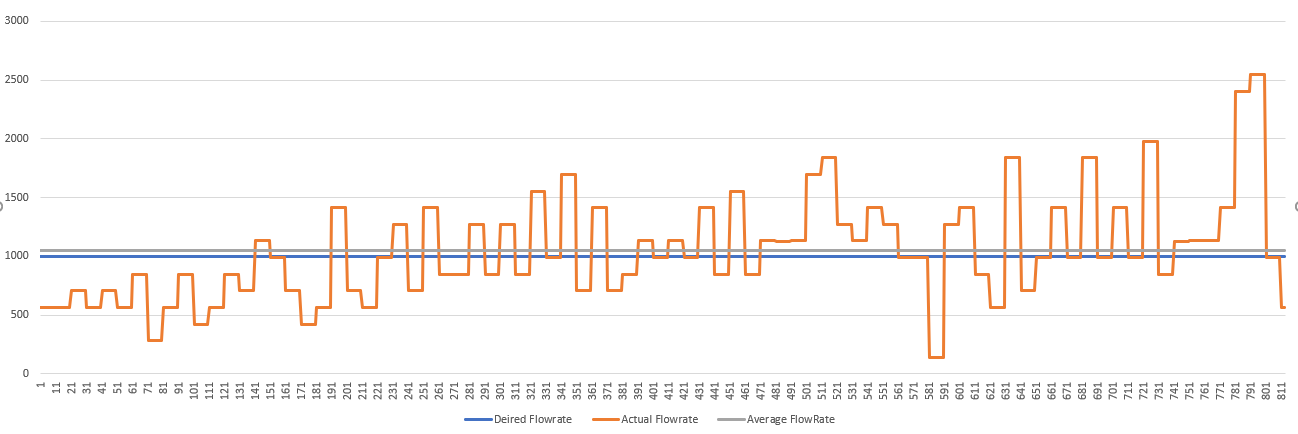


Previously we had generated graphs between the error and the desired flow rate and thus for better understanding we now generate the graphs between the actual and desired flow rate.

**Fourth experiment:**

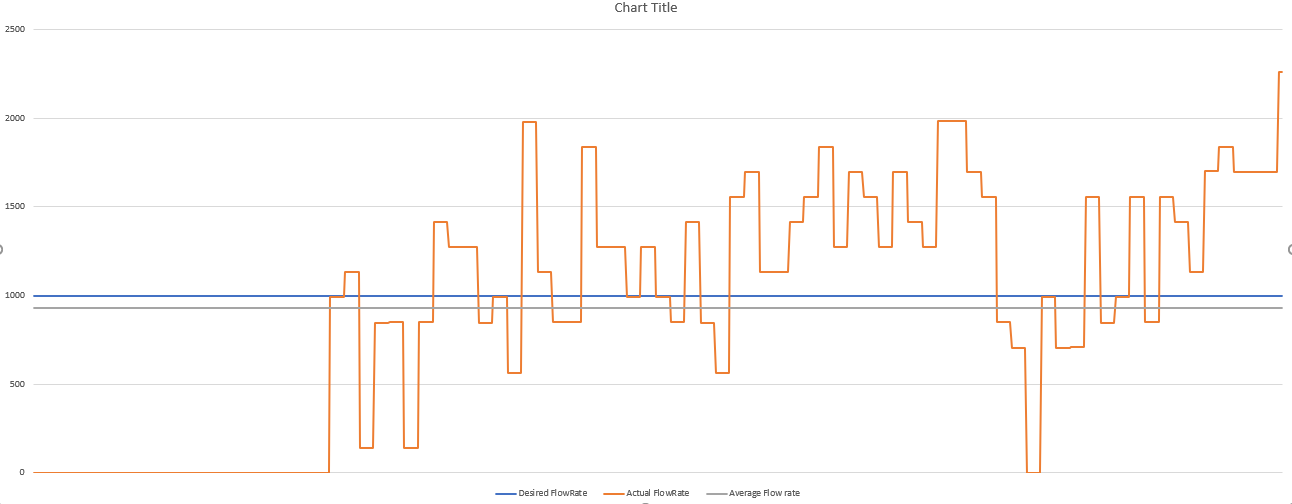
In this experiment kp=80,kd=0 and ki=0 is used. We generated graph between the desired and actual flow rate and an average flow rate through all the sample taken.

Our desired Flow Rate is 1000mL/min shown in blue line, the orange line is showing the actual flow rate taken from the flow meter readings and the gray line shows the average flow rate (1054mL/min) through this experiment. So, the average value is the closer to the desired flow rate value.



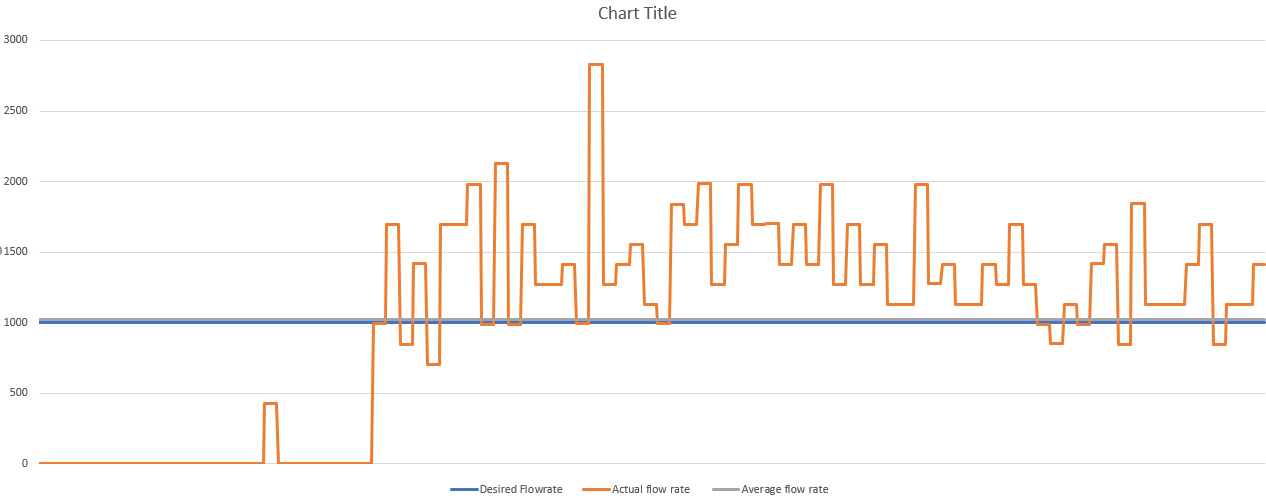
**Fifth experiment:**

In this experiment kp=80,kd=10 and ki=0 is used. Here again our desired Flow Rate is 1000mL/min shown in blue line, the orange line is showing the actual flow rate taken from the flow meter readings and the gray line shows the average flow rate (929.4 mL/min) through this experiment. So, the average value is the closer to the desired flow rate value.



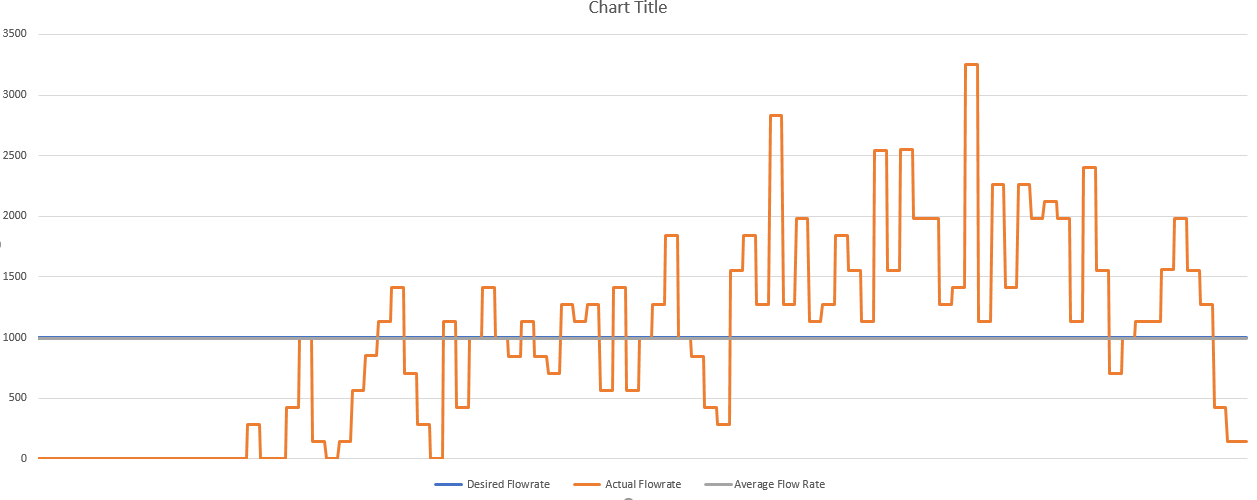
**Sixth experiment:**

In the previous experiment, the graph showed that there is lot of overshoot and undershoots. Thus, we tried to reduce the kp value in this experiment. In this experiment kp=50,kd10 and ki=0. in these results, the there is still overshoots but there is minimum undershoots. The average Flow rate achieve is 1024 mL/min.



**Seventh experiment:**

In this experiment kp=0.3, kd=6 and ki=0 is used. The average Flow rate is 990mL/min and the desired flow rate is 1000mL/min. although there is the oscillations in the system.



**Conclusion:**

As you can see from the above experimental results, there is an improvement in the desired and the average flow rate in the system by constantly changing PID gains. But there is a lot of fluctuation in the system. There are many factors which causes fluctuations;

1. Not included pulsation damper in the system, which is necessary for the diaphragm pumps.
2. Not incorporated constant pressure concept.
3. As we have used on/off solenoid for the bypass line also, which is the major factor for creating fluctuations.
4. Real time PID signal creates PWM signal is also a factor of fluctuations.

**Recommendation:**

As for as my knowledge is concerned, for generating pwm signals (for desired flow rate) through PID controller is not a good solution.

We should implement fuzzy logic or just simple categories for the flow rate VS Duty cycle. By experimentation we can find these relations, e.g for 10% duty cycle, the flow rate achieved is 300mL/min and 20% duty Cycle the flow rate achieved is 500mL/min.

So, use this data to adjust PWM to the nozzle according to the flow rate required.

Table : Rough value for DutyCyle Vs Flow rate

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Duty cycle % | Average Flow rates (Experiment1,2,3) | Average Flow Rate (mL/min) |
| 1 | 10 |  | 100 |
| 2 | 20 |  | 200 |
| 3 | 30 |  | 300 |
| 4 | 40 |  | 400 |
| 5 | 50 |  | 500 |
| 6 | 60 |  | 600 |
| 7 | 70 |  | 700 |
| 8 | 80 |  | 800 |
| 9 | 90 |  | 900 |
| 10 | 100 |  | 1000 |

When this data is achieved, we can easily use this data to adjust duty cycle according to the required flowrate.

if we need 500 mL/min as output of the nozzle, so the DutyCyle should be 50 % according to the above data.