Self-Powered Flow Rate Sensor

Vikram Ramanathan

1 Project Proposal

Currently, most sensors are powered by primary batteries (or non-rechargeable batteries). As a result, regular battery replacements would be required for long-term installations (with minimal maintenance). However, battery replacement can be expensive and not feasible for remote monitoring applications. In addition, sensors using primary batteries must also reduce the frequency and intensity of its power consumption to prolong its lifespan. This would mean lower sampling rates and reduced transmission ranges for wireless communication. To achieve long-term operation without compromising on the quality and quantity of information received, energy harvesting with secondary batteries (rechargeable batteries) is a viable and more effective alternative.

The purpose of this project is to design a self-powered flow velocity sensor. The sensor would be different from state-of-the-art implementations of flow rate sensing in that the mechanical design would not only be smaller but also optimized for a larger range of flow velocity measurements. The sensor will also be able to rectify input voltage signals, charge a battery, power an Arduino and have the ability to be calibrated easily. While this sensor is designed for civil applications (plumbing and piping systems), the mechanical design could be very easily altered to accommodate for flow sensing in water bodies (rivers, ocean currents, etc).

2 Background/Theory

The primary purpose of this sensor is to provide the user with the average flow velocity of the fluid passing through its inlet. In order to calculate the average flow velocity, volumetric flow rates from particular system must be measured. To simplify the model, the flow is assumed to be steady. Therefore, the volumetric flow rate can be found by Eq. 1 below.

$$Q = \frac{dV}{dt} \approx \frac{V_f - V_i}{\Delta t} \tag{1}$$

The measured volumetric flow rates are then associated with voltage readings from the sensor circuit. Therefore, a linear model relating the volumetric flow rate and the voltage readings can be established.

Once calibrated, the voltage readings can be used to find the volumetric flow rate and Eq. 2 below can be used to approximate average flow velocity.

$$V = \frac{Q}{A_{inlet}} \tag{2}$$

3 Mechanical Theory

Given that the primary objective of this project was to increase a flow sensor's measurable range of flow velocities, developing an entirely new mechanical design was essential. The need for a change in mechanical design is explained in the section below.

3.1 State-of-the-Art

3.1.1 Paddlewheel Self-Powered Flow Sensor

Relevant Technical Specifications:

- Flow Rate range is 0.3 to 6 m/s
- Linearity: ±1% of max range
- Repeatability: ±0.5% of max range

This sensor is designed for use in fluids that need not be entirely rid of large impurities (that could get stuck in the sensor). This is due to the fact that only a small paddlewheel (with 4 paddles) needs to be turned and a turbine is not used. However, this sensor entails an expensive and complex installation procedure as a hole or a separate fitting needs to be made/installed in the existing pipe. The mechanical design for this sensor is detailed in Figure 1 below.

3.1.2 Yosoo DC Water Turbine Generator (purchased from Amazon)

A analysis of this solution was written as a small report and can be found here: Yosoo DC Water Turbine Generator Performance Evaluation.

3.2 Proposed Design

In order to develop a mechanical design that would overcome the shortcomings of the state-of-the-art implementations discussed in the previous section, a cross-flow turbine implementation is proposed. Unlike most water turbines, which have axial or radial flows, a cross flow turbine is designed such that water passes through the turbine transversely. A cross flow turbine is a type of impulse turbine. Similar to a water wheel, the water strikes the turbine blades. However, for a cross-flow turbine, the water passes through the turbine after striking the first blade and then strikes another blade while exiting the turbine. Passing through the runner twice provides additional efficiency.

The cross flow turbine was designed according to the schematic shown in Fig.1.

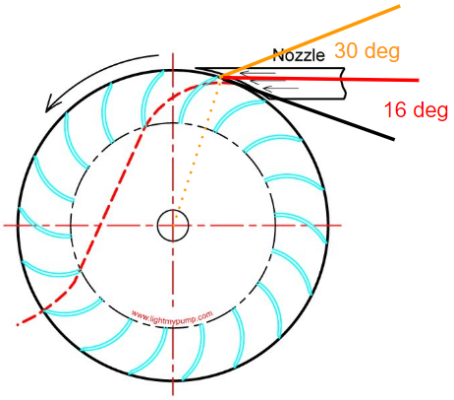

Figure 1: Side View of Turbine Design

Careful attention must be paid to the angle the blades make with lines tangent to their starting point (on the outer diameter of the turbine wheel) and the line indicating the direction of inlet flow as well as the angle made by the line parallel to the direction of the blade and the line indicating the direction of inlet flow. Optimal choices for these angles are detailed in Figure 2 as -16°and -30°respectively. The final turbine design can be viewed in the project folder. A representative view of the design can be seen in Fig. 3 for the reader's reference.

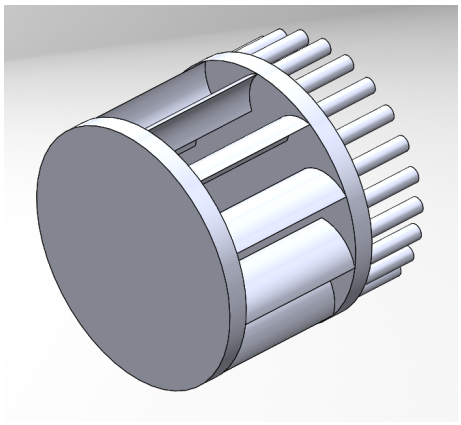

Figure 2: Isometric View of Final Turbine Design [Solidworks]

Given that much of the cross-flow turbine design's efficiency is dependent on the casing design, an entirely new design is proposed that takes inspiration from Miroslav Cink's cross-flow radial turbine [Sin+14] (seen in Fig. **??**. This system is very promising as it does away with the use of the "hydraulic flap" to direct the flow onto the blades (primarily seen in Ossberger cross-flow turbines). However, while the regulation system that involves a control device to guide the flow based on the flow rate was not implemented, the general design was emulated for a significantly smaller device.

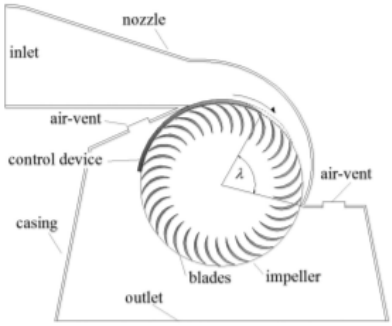

Figure 3: Miroslav Cink's Cross-Flow Radial Turbine

The casing (shown in Fig. 4)also had to fit onto a shower spout as well as fit the turbine assembly (which consisted of the turbine from the Yosoo DC Generator as well as the cross-flow turbine attachment in Fig. 3).

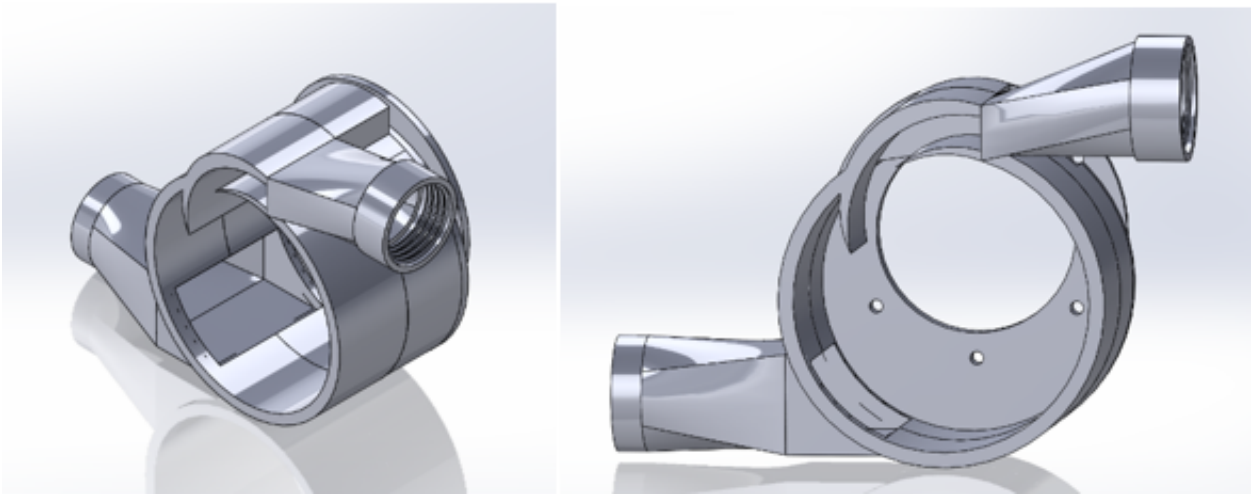

Figure 4: Cross-Flow Turbine Casing Design

Both the casing and the turbine were printed using CraftBot 3D Printers in the UT Austin MakerStudio. Some design flaws that were corrected include the threading (printing resolution and the lack of printing support for the non-threaded region following the threaded region resulted in distorted threads), input flow angle (changing the shape of the curved control surfaces at the inlet of the device) as well as the locations of the screw holes for the turbine assembly.

4 Circuit Design

Since a DC brushless motor with a three phase voltage output was used, a 3 phase voltage rectifier was required to ensure a DC power supply. In addition to the 3 phase voltage rectifier, a battery charging/protection module with a boosted 5V Output was used to both charge a 3.7V Lithium Ion Polymer battery as well as supply power to an Arduino Uno. Figure 5 depicts the circuit schematic.

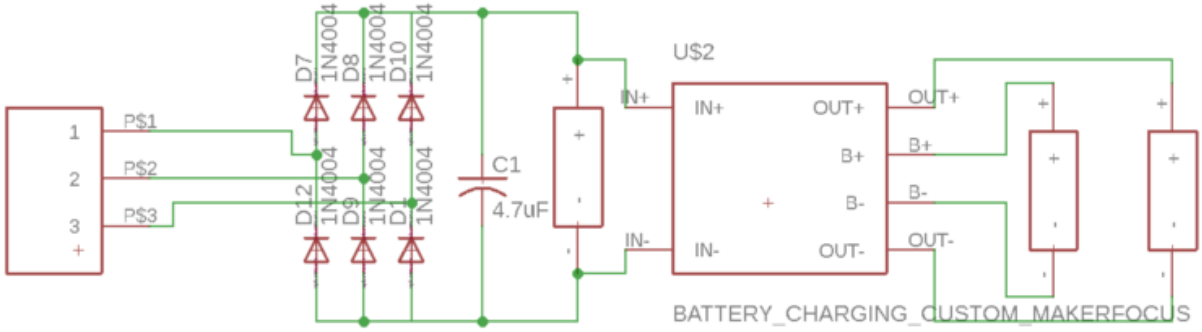

Figure 5: Circuit Schematic

4.1 PCB Fabrication

To manufacture the circuit detailed in Section 3 above, a PCB was fabricated. The PCB was designed using Eagle CAD. In order to make the process of soldering easier, a large trace (24 width) and larger pad diameters were used. The organized board layout of the single-sided PCB is shown in Fig. 6.

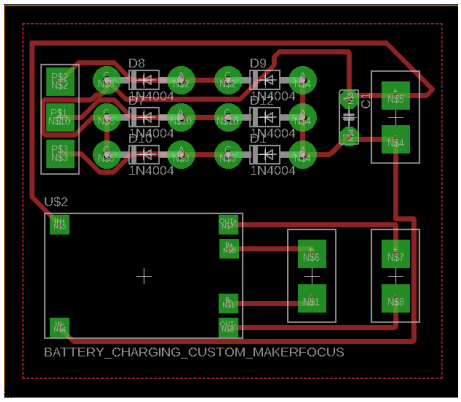

Figure 6: PCB Board Layout

Once the board was printed using the OtherMill Pro in the MakerStudio, headers were soldered onto the pads for the battery charging/protection module. Finally, the module, the 6 diodes and the capacitor were soldered onto the board.

5 Experimental Procedure

Since the only source of somewhat steady flow I could use for this experiment was my shower spout, the experimental procedure was built around this limitation.

5.1 Sensor Calibration

In order to calibrate the sensor, the volumetric flow rate needs to be calculated for various shower spout flow outputs.

1. Mark 3 or more positions of increasing volumetric flow rate on the shower handle
2. Turn on the flow at the first position
3. Place a 1L measuring cylinder in the flow
4. Use a stopwatch to record the amount of time it takes for the cylinder to fill to 0.5L
5. Repeat for the other positions

5.2 Average Flow Velocity Measurement

1. Determine the area of the shower spout outlet
2. In order to ascertain repeatability and uncertainty in area calculation, repeat step 1 five times.
3. Place the shower handle at the first position
4. Measure 20 samples of analog voltage output from the circuit
5. Repeat for all other shower handle positions

6 Results

In this section, the results obtained from testing the sensor and the calculated uncertainties/errors are presented.

6.1 Sensor Calibration

In Fig. 7, the linear regression model relating the analog voltage readings to the volumetric flow rate is depicted.

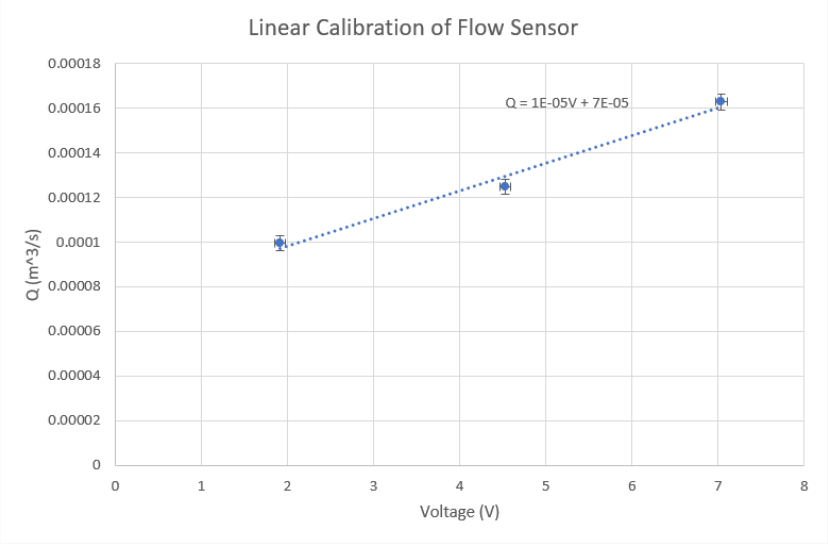

Figure 7: Linear Calibration of Flow Sensor with Uncertainty Whiskers

It is apparent from Fig. 7 that the linear model is fairly accurate. While the volumetric flow rate for position 2 lies further away from the model, we can attribute this error to a lack of data (since volumetric flow rate was noitceable steady only at a few shower handle positions). More positions to test would provide us with a better understanding of the accuracy of this linear model.

6.2 Flow Velocity vs Voltage

The flow velocity is calculated using Eq. 2. The uncertainty in both area and volumetric flow rate measurements are considered in the flow velocity uncertainty analysis (using the method of Sequential Perturbation). The result is a linear relationship between flow velocity and voltage as seen in Fig. 8.

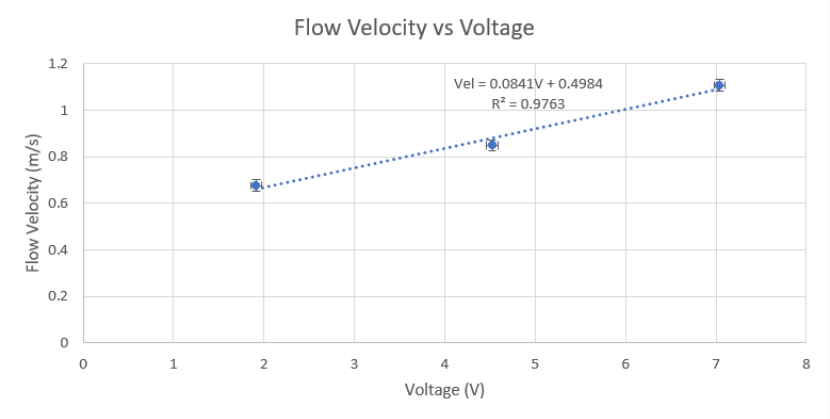

Figure 8: Flow Velocity vs Generator Output Voltage

In Fig. 8 above, we observe that the linear fit is adequately accurate even though the model results in a relatively large error for the second position's flow velocity. Once again, this error could be attributed to the unavailability of volumetric flow rate/flow velocity data points.

6.3 Relevant Technical Specifications

- Flow Rate Range (range of reasonable model accuracy): 0.68 to 1.11 m/s
- Linearity: (BFSL) 3.61%
- Repeatability: (uncertainty of flow velocity measurement) $\pm 0.24 m/s \approx 5.6\%$ of max range

6.4 Effect of Temperature [Hypothesis Testing]

It is essential that this sensor perform consistently regardless of the temperature of the fluid. To understand the influence of the temperature of the fluid on the sensor, the following analysis was conducted.

6.4.1 Hypothesis Testing [T-test]

$$H_o: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

Welch-Satterthwaite Equation

$$\nu = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{(\frac{S_1}{\sqrt{n_1}})^4}{n_1 - 1} + \frac{(\frac{S_2}{\sqrt{n_2}})^4}{n_2 - 1}}$$

Test statistic,

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

The results are summarized below.

$$t_0 = 0.6846$$
$$\nu = 37.9998$$
$$t_{0.025,37.9998} = 2.0252$$
$$t_{0.025,37.9998} > t_0$$
$$\therefore Accept\ H_0$$

We can conclude with 95% certainty that,

$$\mu_1 = \mu_2$$

Therefore, according to this analysis, the temperature of the fluid flowing through the sensor would not affect the sensor's performance (i.e. sensor output is similar for both hot and cool liquid flow).

7 Conclusion

This project aimed at developing a self-powered modular flow sensor. The results for this project indicate that I was able to calibrate the sensor fairly easily as well as have the sensor output average flow velocities within the range of 0.68 to 1.11m/s with a linearity error of 3.61% and a repeatability error of 5.6% of max range. However, there are a few design flaws that must be addressed. The case made to hold the Arduino and the PCB (this would prevent solder joints from breaking due to an unexpected fall) was attached to the turbine. Since the casing was too large and heavy, the turbine would shift in position. This slight shift in position caused the turbine to touch the flow control structures and friction would cause the turbine to stop. In order to take measurements for this report, the sensor turbine had to routinely be checked and fixed into placed. For future improvements to this design, I would suggest designing a lighter and smaller DAQ device (instead of using the Arduino). In addition, the mechanical design of the case had to be tuned by hand (filing) after 3D printing because of irregularities in the quality of the turbine print. These irregularities are apparent on the inner surface of the print as there are visible undulations. Removing these undulations by using the "Ultra" quality setting on the 3D printer, while time consuming, is recommended for added efficiency of the turbine.

The biggest source of error in our measured data is the lack of flow rate positions to test. As a result, we cannot test the upper bounds of this sensor's range of measurement. A necessary development for the next set of experiments would be to use a flow output with a range from 0 to 6m/s as well as a rotameter to take more accurate flow rate measurements. In addition, another future test that would be potentially helpful in improving the efficiency of the sensor would be to test different blade angles and separations in the cross-flow turbine. By running a hypothesis test we could determine whether the efficiency of the turbine is affected by blade angle.

Overall, this project was largely successful in that the sensor (once calibrated) outputted voltage values proportional to the average flow velocity. However, improvements to the mechanical design and optimizing the electrical hardware to make up a smaller footprint are necessary for future versions of this sensor.

References

[Sin+14] Marco Sinagra et al. "Cross-Flow turbine design for variable operating conditions". In: *Procedia Engineering* 70 (2014), pp. 1539–1548.