ULTRASONIC ANEMOMETER DESIGN AND TESTING IN WIND TUNNEL

Introduction

Anemometer is any device that is used for measuring wind speed and direction. Commonly installed mechanical anemometers consist of cup shaped vanes that allow them to rotate at a speed proportional to the wind speed. Due to moving parts, such type of anemometers is more susceptible to wear and tear and thus loss in accuracy. Another type of wind speed measuring device is "Ultrasonic Anemometer" which uses the time of flight (TOF) concept to measure the wind speed and direction. Compared with traditional cup type mechanical anemometers, ultrasonic anemometers have high accuracy and no abrasion. Ultrasonic anemometers can be one, two and three dimensional and they employ sound waves with frequency greater than 20kHz to work efficiently.

The working concept behind an ultrasonic anemometer is very simple and is based on the concept of relative speed i.e. Speed of sound travelling in wind direction will travel faster thus less TOF and vice versa.

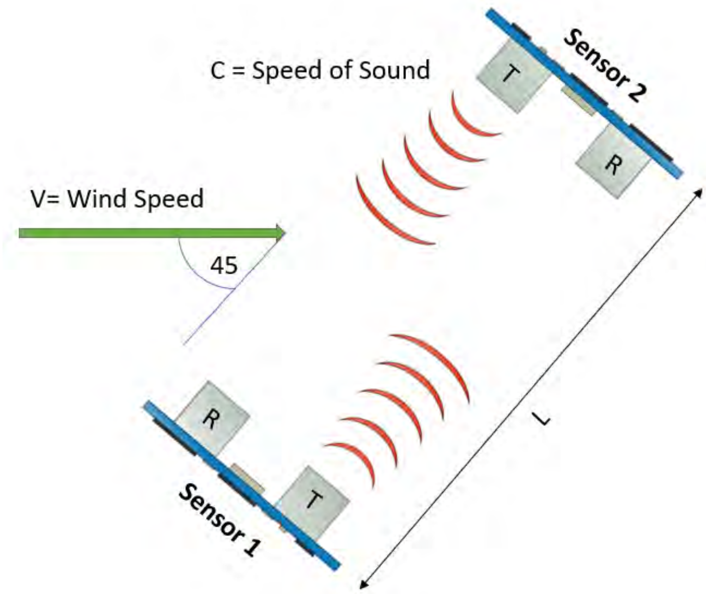
In the following experiment, we designed and tested the one-dimensional ultrasonic anemometer in the Wind tunnel at SEMTE ASU. The details of the experiment are described below.

Components

- Two HC-SR04 Ultrasonic Transducers (40kHz)
- Arduino Uno
- Breadboard
- Jump wires
- Wooden or cardboard platform
- A small Wind Tunnel (1'x1'x2')

Experimental Setup

The experimental setup is designed as to not disturb the wind flow and still get the accurate readings. For this purpose, two ultrasonic transducers were fixed at L=31 cm from each other and their line of sight is at 45 degrees angle with the wind flow. The basic calculations involve finding the time of flight through relative speed concept.



$$t_{12} = \frac{L}{c + v * cos\,45}$$

$$t_{21} = \frac{L}{c - v * cos45}$$

$$\Delta t = t_{21} - t_{12} = \frac{L}{c - v * cos45} - \frac{L}{c + v * cos45}$$

$$c^2 - 0.5 * v^2 \approx c^2$$

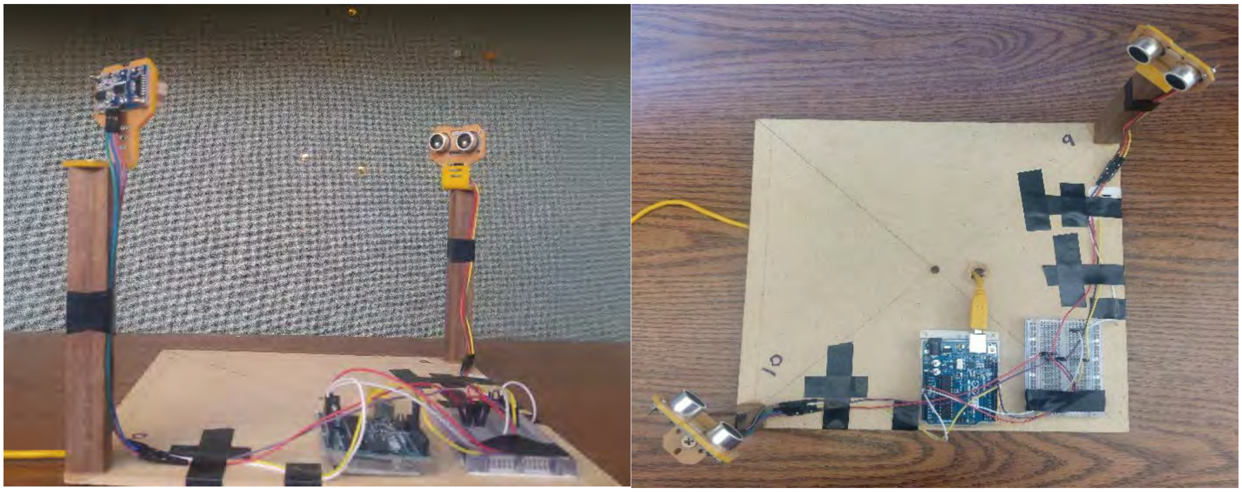
$$v = \frac{\Delta t \times c^2}{2L * cos45}$$

Where t_{12}= time it takes for ultrasonic signal from Sensor 1 to reach Sensor 2

& t_{21}= time it takes for ultrasonic signal from Sensor 2 to reach Sensor 1

The assumption being made here is that $c^2 - 0.5 * v^2 \approx c^2$ because the square of speed of sound is significantly larger than the half of wind speed squared.

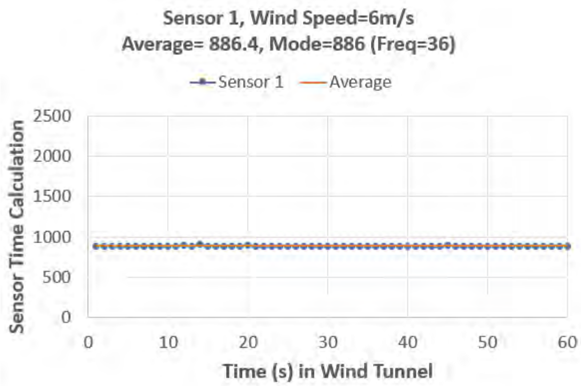
The two sensors are attached to the plywood board (11”x11”) with the help of two supporting wooden square rods. The rods were each 6” high and supported the ultrasonic sensor brackets on the top to mount the sensors. The plywood board serves as a platform to attach Arduino, breadboard and jump wires, and it was clamped in the wind tunnel through a nut & bolt. The ultrasonic sensor HC-SR04 operates on 40kHz frequency and its datasheet is easily available online.

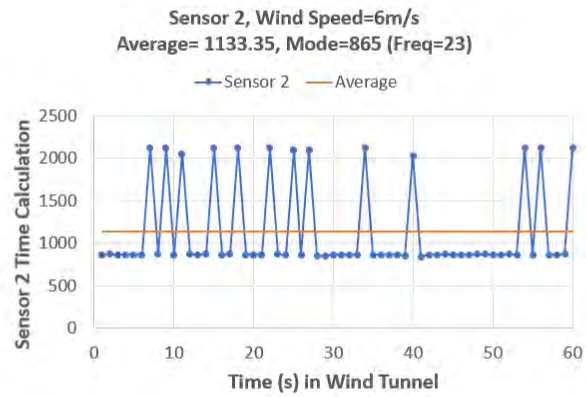


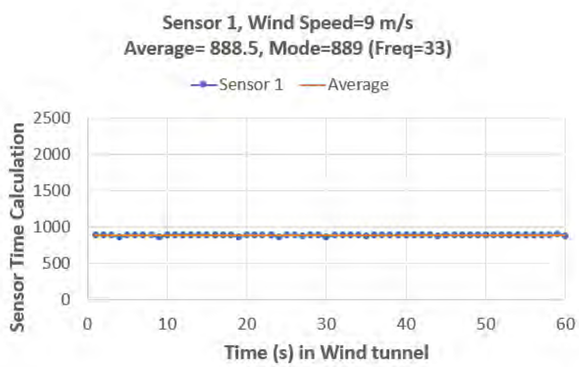
Arduino Uno with Arduino IDE is used as a microcontroller to control the transmitter and receiver signals and calculate the time of flight. The clock speed of Arduino Uno is 16MHz and provides considerably accurate results for our purposes. The *pulseIn()* function was used to calculate the time of flight it takes for the ultrasonic wave to reach the receiver on Sensor 2 from transmitter of Sensor 1 and vice versa. The calculated time is in microseconds and provide a good correlation with the change in wind speed as discussed in results. All the other calculations and analysis was performed by exporting the time of flight data to Microsoft Excel. The detailed code used for this purpose is included in Appendix.

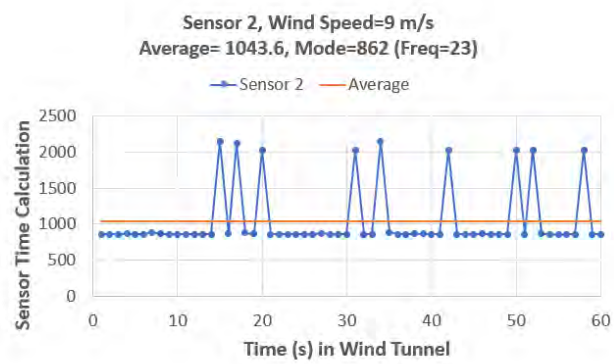
Results and Discussion

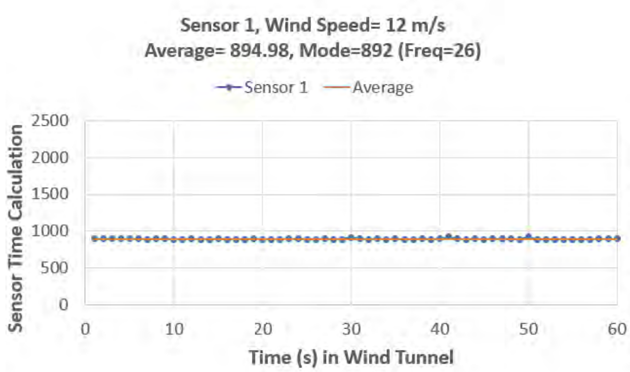
Time of Flights t_{12} and t_{21} were calculated through Arduino at seven different wind speeds from 6m/s to 24 m/s at an increment of 3m/s. A total of 60 readings were taken by each sensor and the mode (highest occurred reading) was selected to calculate wind speed. It was observed that while Sensor 1 provided fairly good results (Mode ≅ Mean), Sensor 2 had some aberrations in the results where some of the readings just sky rocketed. The probable reasons for this error could either be the electrical or signal noise received by the sensor. This error could be removed by using more sophisticated ultrasonic sensors and timing devices.

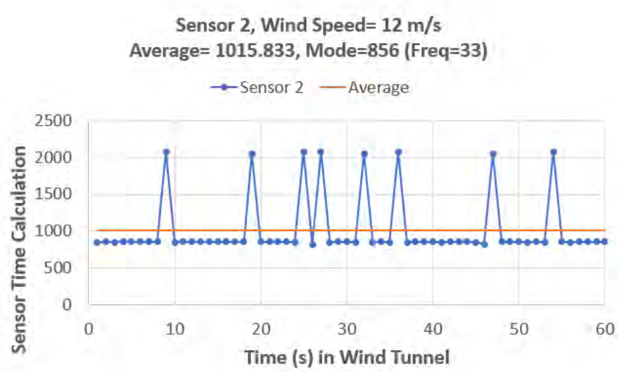


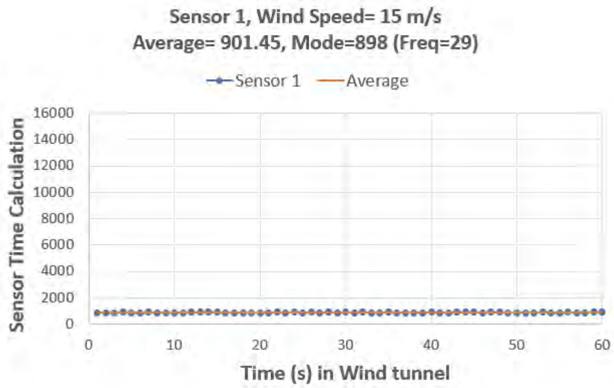


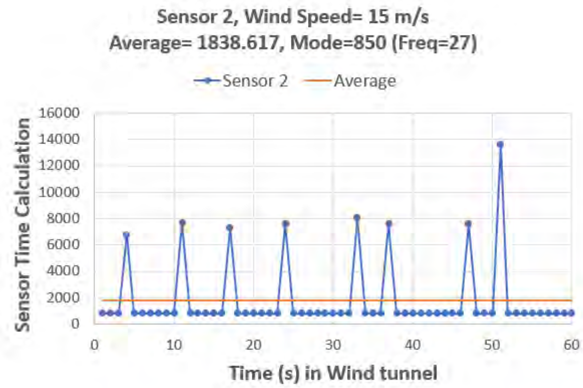


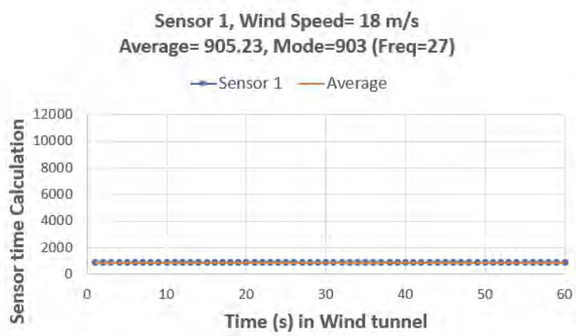


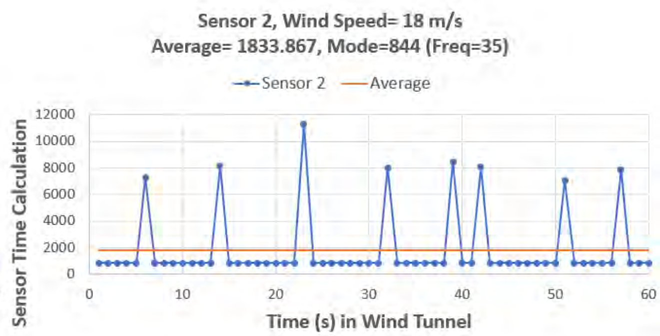


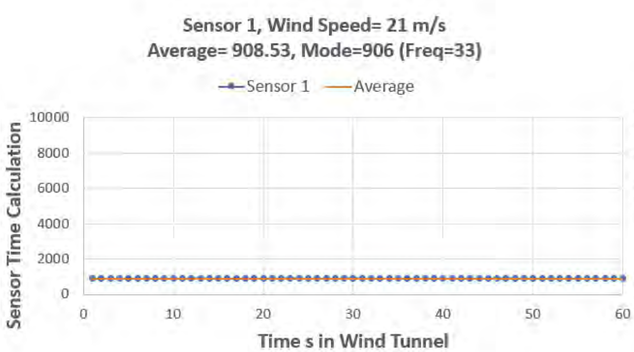


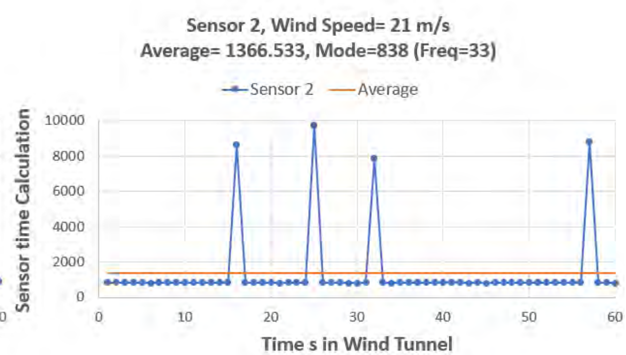

Sensor 1, Wind Speed= 15 m/s
Average= 901.45, Mode=898 (Freq=29)
Sensor 1
Average
Sensor Time Calculation
Time (s) in Wind tunnel

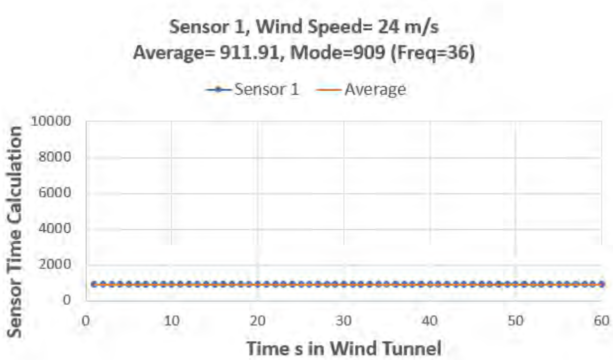

Sensor 2, Wind Speed= 15 m/s
Average= 1838.617, Mode=850 (Freq=27)
Sensor 2
Average
Sensor Time Calculation
Time (s) in Wind tunnel

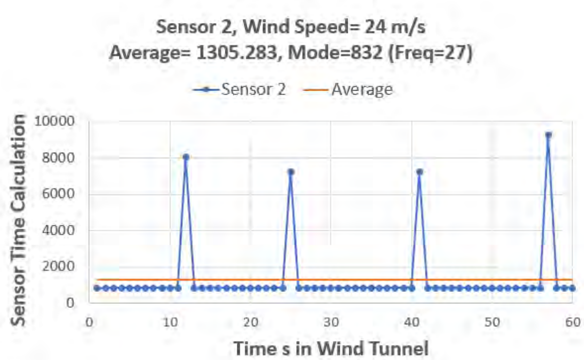

Sensor 1, Wind Speed= 18 m/s
Average= 905.23, Mode=903 (Freq=27)
Sensor 1
Average
Sensor time Calculation
Time (s) in Wind tunnel

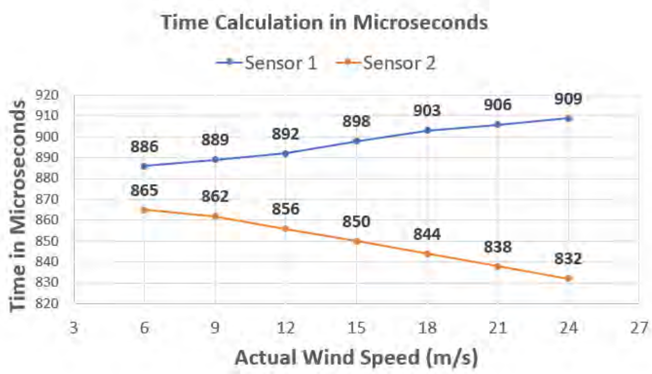

Sensor 2, Wind Speed= 18 m/s
Average= 1833.867, Mode=844 (Freq=35)
Sensor 2
Average
Sensor Time Calculation
Time (s) in Wind Tunnel

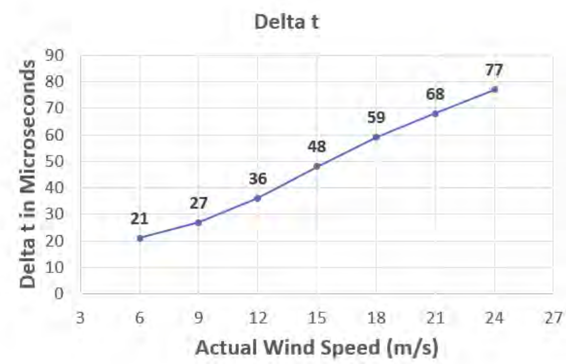

Sensor 1, Wind Speed= 21 m/s
Average= 908.53, Mode=906 (Freq=33)
Sensor 1
Average
Sensor Time Calculation
Time s in Wind Tunnel

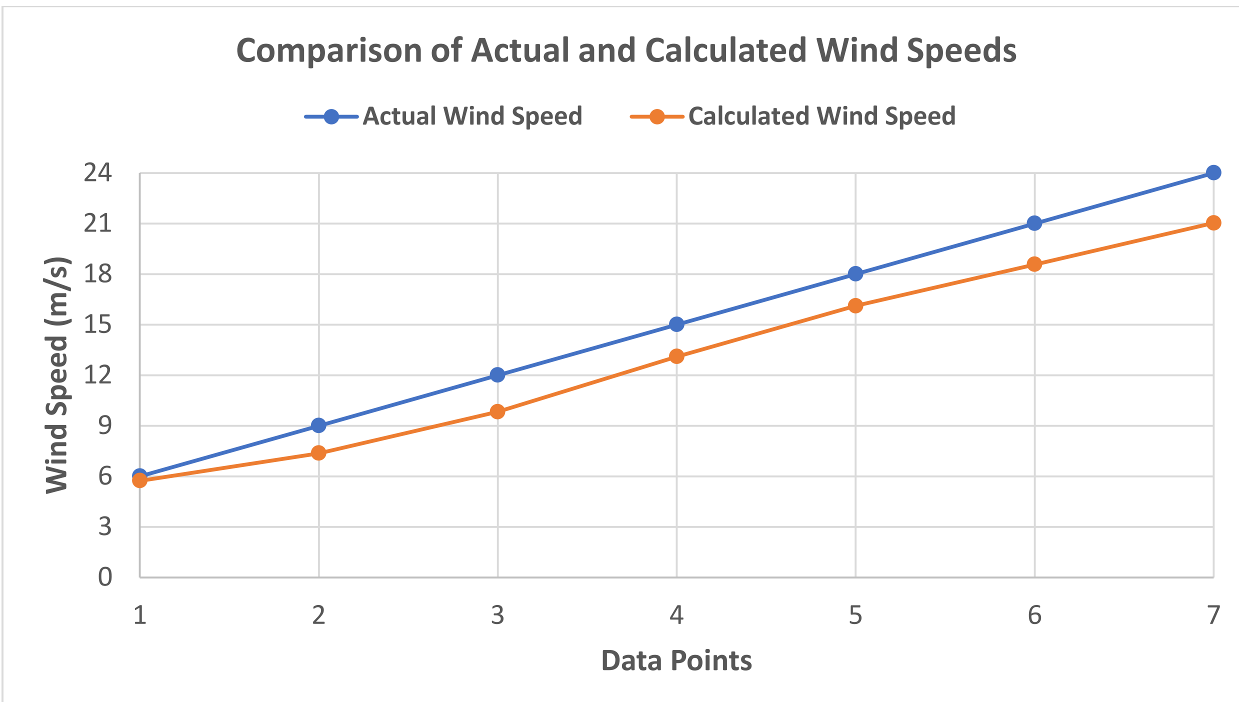

Sensor 2, Wind Speed= 21 m/s
Average= 1366.533, Mode=838 (Freq=33)
Sensor 2
Average
Sensor time Calculation
Time s in Wind Tunnel

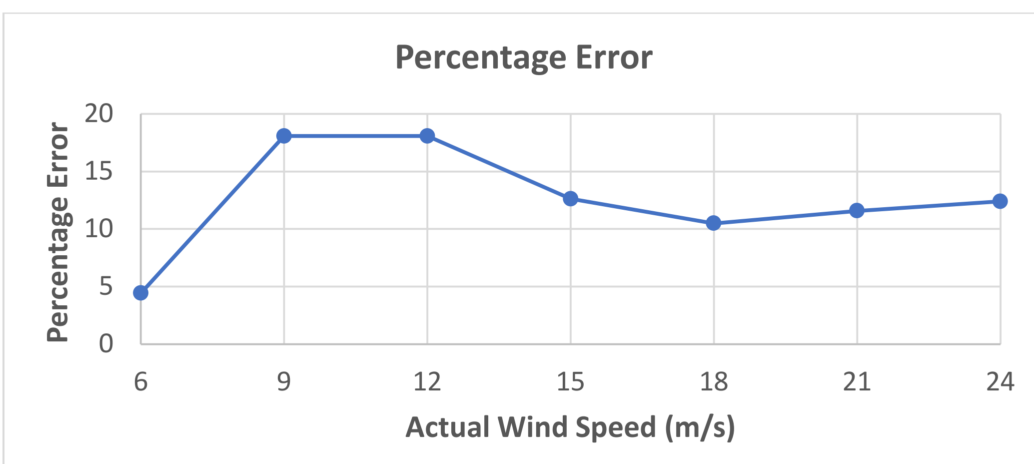

Sensor 1, Wind Speed= 24 m/s
Average= 911.91, Mode=909 (Freq=36)
Sensor 1
Average
Sensor Time Calculation
Time s in Wind Tunnel


Sensor 2, Wind Speed= 24 m/s
Average= 1305.283, Mode=832 (Freq=27)
Sensor 2
Average
Sensor Time Calculation
Time s in Wind Tunnel


Time Calculation in Microseconds
Sensor 1
Sensor 2
Time in Microseconds
Actual Wind Speed (m/s)
886
889
892
898
903
906
909
865
862
856
850
844
838
832


Delta t
Delta t in Microseconds
Actual Wind Speed (m/s)
21
27
36
48
59
68
77


Comparison of Actual and Calculated Wind Speeds
Actual Wind Speed
Calculated Wind Speed
Wind Speed (m/s)
Data Points


Percentage Error
Percentage Error
Actual Wind Speed (m/s)

Appendix

The code used in Arduino IDE is given below.

```
float duration1; //Duration 1= Time it takes for ultrasonic wave to
reach from Sensor 1 to Sensor 2
float duration2; //Duration 2= Time it takes for ultrasonic wave to
reach from Sensor 2 to Sensor 1
int trig=11; // Trigger Pin
int echo1=9; //Echo Pin for Sensor 1
int echo2=10; //Echo Pin for Sensor 2

void setup() {
Serial.begin(9600); // Start Serial monitor
pinMode(trig,OUTPUT);
pinMode(echo1,INPUT);
pinMode(echo2,INPUT);
}


void loop() {

digitalWrite(trig,LOW);
delayMicroseconds(2);
digitalWrite(trig,HIGH);//Sends the ultrasonic signal
delayMicroseconds(10);
digitalWrite(trig,LOW);
duration1=pulseIn(echo1,HIGH); // TOF recorded by sensor 1


delay(500);

digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);
duration2=pulseIn(echo2,HIGH); // TOF recorded by sensor 2

Serial.print(duration1);
Serial.print(",");
Serial.println(duration2); //Results can be transformed into a .csv
file

delay(500);
}
```