Oct 30th, 12:00 AM - 12:00 AM

Errors in Distance and Angle Measurements of Ultrasonic Sensor HC-SR04

Fidan Smaili
University for Business and Technology - UBT, fidan.smaili@ubt-uni.net

Plotdon Beqiri

Rinor Lubovci

Errors in Distance and Angle Measurements of Ultrasonic Sensor HC-SR04

Fidan Smaili[1*], Plotdon Beqiri[2], Rinor Lubovci[2]

[1]University for Business and Technology, Faculty of Mechatronics Management

[2]University of Prishtina, Faculty of Mechanical Engineering

e-mail: fidan.smaili@ubt-uni.net

Abstract. This project presents the measurement of distances and angles by means of Arduino Uno microcontroller and Ultrasonic Sensor HC-SR04. Accuracy on distance and angle measurements of the sensor has been investigated. The change of object distance in motion is displayed and monitoring of the change of the angle and position. Data measurements were compared with the real measurement unit of the meter to observe their deviation, 1.6% of the standard deviation is observed between real measurement and sensor data, meanwhile, at the angle data measurements were no deviation is observed.

Keywords: Measurement errors, Measurement uncertainty, Arduino Uno, Ultrasonic sensor.

1 Introduction

As ultrasonic sensors become more common, sufficient testing is imperative. The HC-SR04 is the most common type of sensor taught in schools, as it comes with every Arduino Kit, but there's a lack of testing done on this sensor. This sensor is based on the principle of measuring the time from sending and receiving an ultrasound chirp. [1]. This principle ensures accurate measurement of bodies, regardless of their colour or their surface type - this allows to measure the distance of transparent materials such as glass and water. Ultrasonic sensors are produced in different types; sensors for laboratory use are simpler, with only a transmitter and receiver together in the chassis, while for industrial use we have more robust types, with metal chassis and with multiple receivers and transmitters. Some types allow you to change the sensitivity of the sensor by means of a potentiometer. We also have sensors with a direct connection to a computer, where all sensor parameters can be changed, from the maximum amplitude of the waves up to the mode of delivery of measurements [2]. Ultrasonic waves have the same propagation characteristics as sound - all frequency sound waves higher than 20 kHz are classified as ultrasonic waves. [3]. The HC-SR04 ultrasonic sensor, also known as a sonar is an obstacle sensor. It measures up to 4 [m]. The sensor starts by sending an electrical charge from the microcontroller to the Transmitter, which lasts at least 10 [μs] [microseconds], in response to this electrical charge the sensor sends 8 ultrasonic pulses with a frequency of 40 [kHz]. This transmission method enables the pulse sent by the sensor to be distinguished from ambient noise, and to not receive the wrong ultrasonic pulse. After receiving 8 ultrasonic pulses, the Receiver switches to HIGH, and starts waiting for the return pulses for 38 [μs]. If the pulses do not return within 38 [μs], this means that at 4 meters that the sensor can measure, there is no obstruction and the Receiver switches to LOW. If the pulses return, the Receiver switches to LOW immediately and produces an electrical charge depending on the time elapsed since the electrical charge was sent by the Transmitter.

To measure distance, we take the distance formula:

$$Distance = \frac{Velocity}{Time} \tag{1}$$

by substituting the electric charge received in microseconds [μs] in time, and constant speed of sound in the air $0.034\ \left[\frac{cm}{\mu s}\right]$ is found the total distance from sending the signal to receiving it. To find the distance from the obstacle to the sensor you must divide the total distance by 2 [4]. Ultrasonic sensor pins should be connected as follows, pin VCC, connects to Arduino VCC. Supplies the sensor with 5 [V] Pin Trig, or Trigger, is used to "shoot" ultrasonic waves from the sensor. Pin Echo, produces an electrical charge when the reflected signal is received. The signal amplitude is proportional to the time elapsed since the signal was sent and received. Pin GND, connects to the GND pin on the Arduino [4]. Our code is template, as it consists of only an ultrasonic sensor, a notable part is the Newping library, which increases our accuracy:

```
#include "NewPing.h"
#define TRIGGER_PIN 9
#define ECHO_PIN 10
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
float duration, distance;
int iterations = 5;

void setup()
{
        Serial.begin(9600);
}

void loop()
{
        // Send ping, get distance in cm
        duration = sonar.ping_median(iterations);
        distance = (duration / 2) * 0.0343;

        Serial.print("Distance = ");

        if (distance >= 400 || distance <= 2)
        {
                Serial.println("Out of range");
        }
        else
        {
                Serial.print(distance);
                Serial.println(" cm");
        }
        delay(500);
}
```

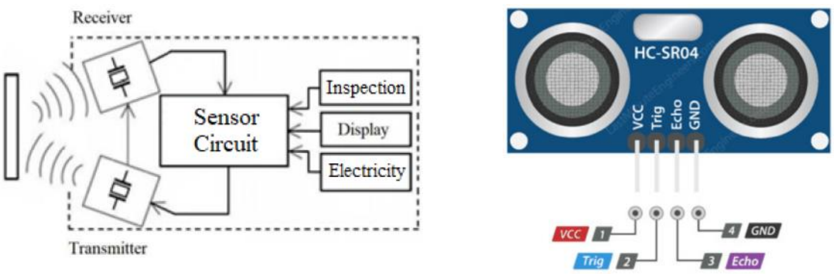

Fig. 1. Sensor block diagram and HC-SR04 ultrasonic sensor (Source: https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/)

We wanted to compare the results we got from our Arduino IDE system to something more practical, so we choose LabVIEW since it is very commonly taught and used in industrial settings. Our virtual instrument base was LINX, as it made our setup very simple.

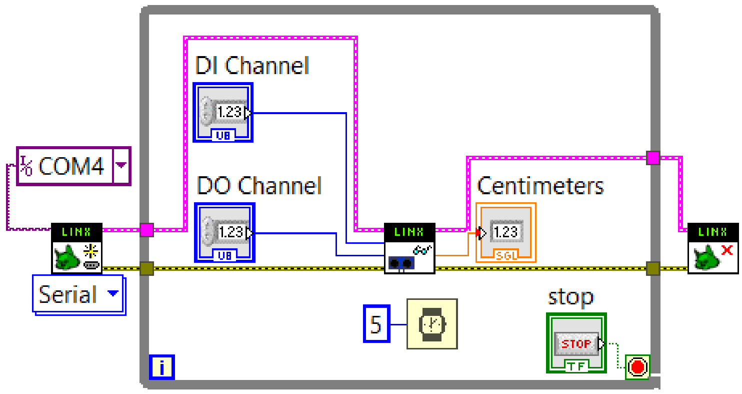

Fig. 2. Block diagram prepared in LabVIEW

2 Research method

We decided on 200 cm being our maximum distance, as beyond that our readings were becoming too erratic.. The sensor was placed on a horizontal position, by facing only the wall while moving away from it to reduce large fluctuations in measurements due to the sensor not sensing a smaller object. Measurements were done in room temperature, respectively 18˚C. Instead of only measuring the wall, an opaque object, we also chose a sheet of glass, as it was transparent, to see if there's a difference.

We proceeded to move away from the wall every 25 cm. All these measurements were done in reference to a Class II European tape-meter. The following table shows the maximum error of possible for each distance value according to the standard European class II tape meters.

Table 1. Meter measurement errors

Maximum errors by standard					
1m	2m	3m	5m	8m	10m
±0.5	±0.7	±0.9	±1.3	±1.9	±2.3

Fig. 3. Taking the reference distance from the meter using a smaller object such as a notebook

3 Results

Table 2. Rinor's sensor data for distance testing. All measurements in [cm]

Distance	Meter Tolerance	Arduino IDE	LabVIEW	Error ARD	Error LabVIEW	Total ARD	Total LabVIEW
25	0.5	24.88	24.603	0.12	0.397	0.62	0.897
50	0.5	49.1	48.47	0.9	1.53	1.4	2.03
75	0.5	74	73.196	1	1.804	1.5	2.304
100	0.5	98.02	97.71	1.98	2.29	2.48	2.79
125	0.7	123.39	122.89	1.61	2.11	2.31	2.81
150	0.7	148.16	146.7	1.84	3.3	2.54	4
175	0.7	173.06	171.32	1.94	3.68	2.64	4.38
200	0.7	198.17	195.43	1.83	4.57	2.53	5.27

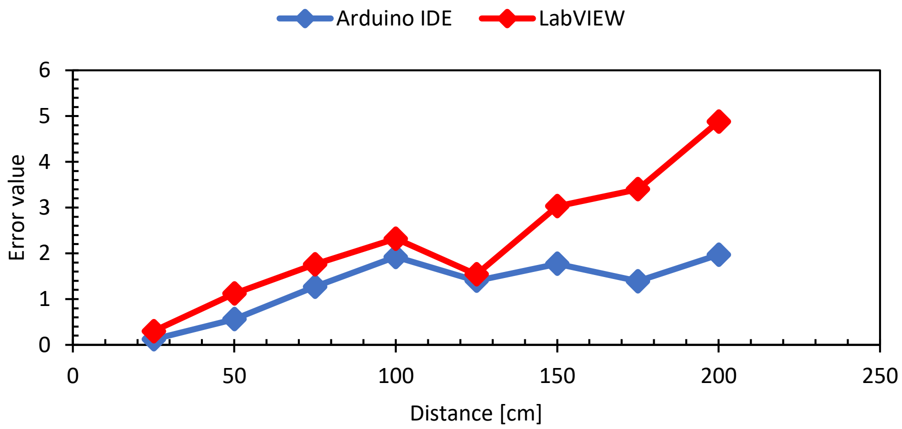

Fig. 4. Rinor's sensor errors, graphed and compared for both systems

Table 3. Plotdon's sensor data for distance testing. All measurements in [cm]

Distance [cm]	Meter Tolerance	Arduino IDE	LabVIEW	Error ARD	Error LabVIEW	Total ARD	Total LabVIEW
25	0.5	24.88	24.7	0.12	0.3	0.62	0.8
50	0.5	49.44	48.88	0.56	1.12	1.06	1.62
75	0.5	73.73	73.24	1.27	1.76	1.77	2.26
100	0.5	98.08	97.68	1.92	2.32	2.42	2.82
125	0.7	123.6	123.46	1.4	1.54	2.1	2.24
150	0.7	148.23	146.97	1.77	3.03	2.47	3.73
175	0.7	173.61	171.6	1.39	3.4	2.09	4.1
200	0.7	198.03	195.12	1.97	4.88	2.67	5.58

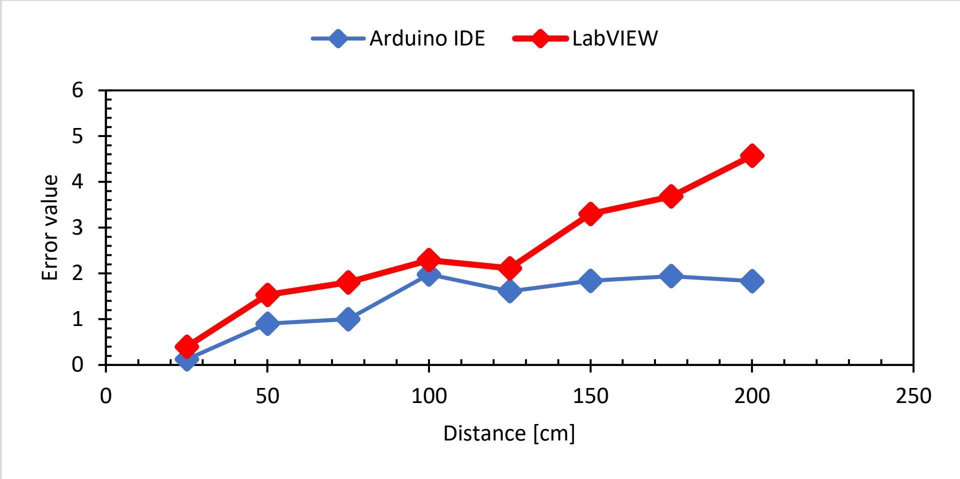

Fig. 5. Plotdon's sensor errors, graphed and compared for both systems

Table 4. Transparent body measurements. All measurements in [cm]

Distance	Meter Tolerance	Arduino IDE	LabVIEW	Error ARD	Error LabV	Total ARD	Total LabV
50	0.5	48.14	48.5	1.86	1.5	2.36	2
100	0.5	98.74	97.42	1.26	2.58	1.76	3.08
150	0.7	148.16	146.43	1.84	3.57	2.54	4.27
200	0.7	197.62	195.23	2.38	4.77	3.08	5.47

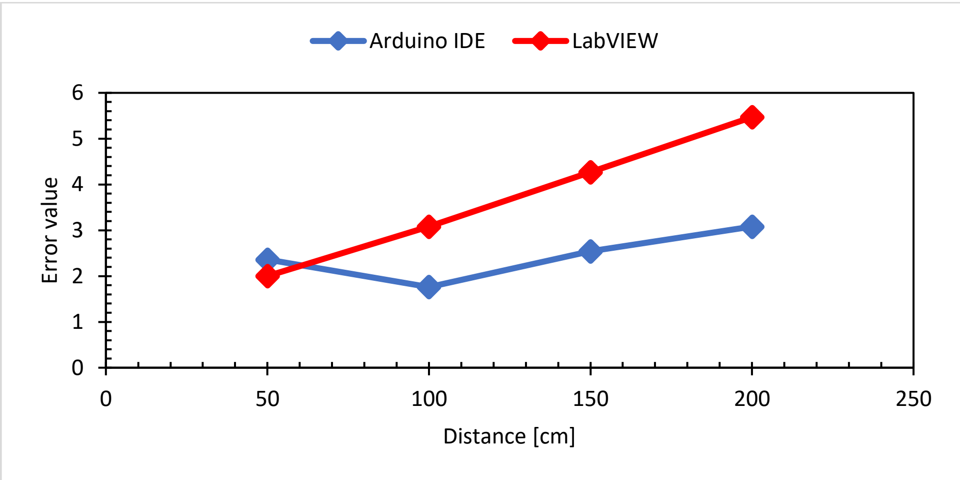

Fig. 6. Transparent body errors, graphed and compared for both systems

We read the corresponding tape-meter tolerance, and we take the maximum deviation of up to 2 meters which is obtained by taking the maximum total error and dividing by our maximum distance:

$$P_S = \frac{\max(|\Delta_S| + |\Delta_M|)}{300} * 100\%$$

For Rinor's sensor it came out to be 1.32 % for the Arduino IDE system, and 2.635 % for the LabVIEW system. For Plotdon's sensor it came out to be 1.335 % for the Arduino IDE and 2.79 % for LabVIEW. Meanwhile for our transparent object 1.54 % for the Arduino IDE and 2.73 % for LabVIEW.

Angle measurements were compared to manufacturers data, that is, 15˚. To test this feature, we took measurements from a set distance , it being 100 cm away from the ultrasonic sensor.

Then by means of a measuring object, in our case a sheet of metal, we moved from left-right, and right-left until the sensor captured the signal. The moment we received the data from the sensor, we measured the hypotenuse using the tape-meter. By simple trigonometry, we find the closing angle between sensor measurement and hypotenuse taken from the meter.

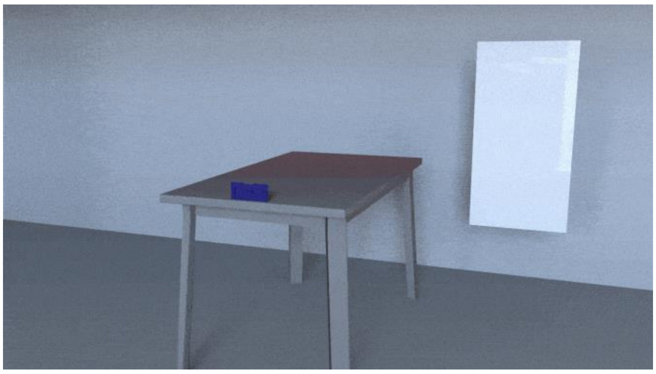

Fig. 7. The measuring object on the left and the right side

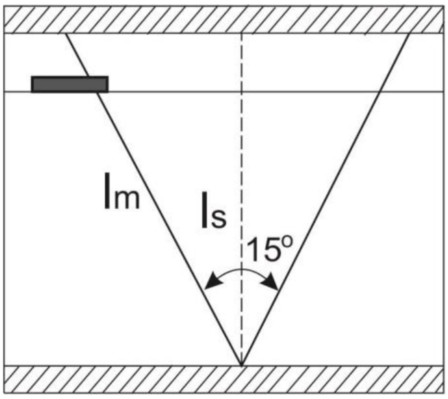

Fig. 8. The principle of measurement https://www.sciencedirect.com/science/article/pii/S2405896316326623

Table 5. Distances and angle calculated for Rinor's sensor

	Hypotenuse	Adjacent	Angle
Right	100.3	100	5.71
Left	101	100	8.069
			≈ 13.779

Table 6. Distances and angle calculated for Plotdon's sensor

	Hypotenuse	Adjacent	Angle
Right	101.05	100	8.26
Left	101.6	100	10.18
			≈ 18.44

4 Conclusion

Practical measurements reveal that the HC-SR04 sensor measures distance with high accuracy. In measurements of up to 200 [cm], the accuracy of measurements was up to 1.54 [%] for our transparent object, which fits with the manufacturers data. The accuracy of measurements was generally dependent on the distance to the object. Our LabVIEW system was much worse compared to our Arduino IDE system. For larger distances, Plotdon's sensor was more accurate than Rinor's sensor. Compared to the documentation, our angle differed by 2-3˚, which is acceptable. For accurate measurements, avoid LabVIEW, reduce air flow and avoid extreme temperatures.

5 References

[1] J. Smoot, CUI Devices, 1 April 2021. [Online]. Available: https://www.cuidevices.com/blog/the-basics-of-ultrasonic-sensors. [Accessed 12 11 2021].

[2] E. Kevin, Dittman & Greer, Inc., 2007. [Online]. Available: http://www.dittman-greer.com/news/2007-winter/3-tech_ultrasonic.htm. [Accessed 12 11 2021].

[3] R. E. (. Berg, Britanica, 6 10 2017. [Online]. Available: https://www.britannica.com/science/ultrasonics. [Accessed 10 11 2021].

[4] L. M. Engineers, 18 December 2020. [Online]. Available: https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/.