

## Research Article

### Improvement of ultrasonic sensor-based obstacle avoidance system in drones

 Fatih Alpaslan Kazan <sup>1\*</sup>,  Haydar Solak <sup>2</sup>

<sup>1</sup> Department of Aviation Electrical and Electronic, School of Civil Aviation, Selcuk University, Konya, Türkiye

<sup>2</sup> Electrical-Electronics Technology, 100th Year Vocational Training Center, Konya, Türkiye

**Received**  
March 8, 2023

**Revised**  
June 6, 2023

**Accepted**  
June 6, 2023

#### **Keywords**

Arduino  
Drone  
Mission Planner  
Obstacle Avoidance  
Pixhawk  
Ultrasonic Sensor

#### **ABSTRACT**

Although drone users have received the necessary training, the reflexes of making decisions against a sudden natural event such as wind and avoiding a nearby obstacle may not be sufficient. Therefore, whether drones fly autonomously or under user control, they must sense and act accordingly for an uninterrupted mission. In this study, a drone design and application for obstacle detection and obstacle avoidance were carried out. In the designed drone, Pixhawk was used as the flight control board, ultrasonic sensors were used to detect obstacles, and Arduino Uno was used as a microcontroller to obtain avoidance commands. The sensors used in obstacle detection and their performance are the most decisive factors in achieving the targeted goal. Because obstacle detection sensors are affected by electrical noises, the success of detecting obstacles decreases. For this reason, first of all, the integration of these sensors into the system was investigated and the drone was developed accordingly. Then, an algorithm was developed using a software filtering method both to minimize communication instabilities and to increase the clarity of detection. Finally, the ability to evade obstacles both while the drone is suspended in the air and while it is in motion has been investigated. In the experiments carried out, it was found that the drone was able to successfully avoid obstacles up to a flight speed of 3.94 m/s.

Production and hosting by  
[Turkish DergiPark](https://dergipark.org.tr). This is an  
open access article under the  
CC BY-NC license  
(<https://creativecommons.org/licenses/by-nc/4.0/>).



\* Corresponding author, e-mail: [akazan@selcuk.edu.tr](mailto:akazan@selcuk.edu.tr)

#### **Authorship contribution statement for Contributor Roles Taxonomy**

**Fatih Alpaslan Kazan**, Writing, Investigation, Visualization, Supervision, Conceptualization, Methodology, Formal analysis.  
**Haydar Solak**, Investigation, Methodology, Review, Experimental studies.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Citation:** Kazan, F. A., Solak, H. 2023. Improvement of ultrasonic sensor-based obstacle avoidance system in drones. *International Journal of Aeronautics and Astronautics*, 4(1), 9-35.



## 1. Introduction

Drones are used in many different fields. For example, security, health, search and rescue, cinema, agriculture, sports, marketing, and cargo transportation are just a few of them. However, no matter which of these they are used in, the common problem with drones is that the drone accidentally crashes into an obstacle and breaks. This means that both the current task is not fulfilled and when the sales prices of the drones are taken into account, undesirable consequences for the task and the user.

Although drone users have received the necessary training, their speed in making decisions against a natural event that may occur suddenly, such as wind, and avoiding an obstacle nearby may not be enough. Therefore, whether drones are flying autonomously or under user control, they must detect obstacles around them and act accordingly for a seamless mission.

There are currently commercially produced drones that can avoid obstacles. For example, DJI Mavic Air 2 is one of them. This mini drone, which has a total of 6 sensors, two at the bottom, front, and rear, has problems not working in low or high light. Again, due to the lack of an obstacle-detecting sensor on the sides, there is a possibility of hitting an obstacle on side flights.

Increasing its importance and usage area every passing day has also increased the number of studies conducted on drones. Some of the studies conducted on drones, their components, and obstacle avoidance in Turkey and the world are summarized below.

Design and implementation of a radio frequency jammer to block the flight of drones [1], control of brushless motors used in drones [2], object tracking in drones, [3], Investigation of the effects of vibrations originating from the engine and propeller on the inertial measurement units used in determining the direction and position of the drones [4], the use of LIDAR, LIDAR-Lite, and ultrasonic sensors in obstacle avoidance applications of unmanned surface vehicles [5], improving the measurement accuracy of ultrasonic sensors used in distance measurement [6], areas of use of drones and challenges encountered [7], the use of unmanned aerial vehicles and the difficulties encountered during use [8], investigation of balance point stability in drones [9], drone design with increased maneuverability and speed [10], precision landing application with image processing methods using Pixhawk, ArduPilot, Raspberry Pi, and a low-budget camera [11], directing the drone swarm to the target using different optimization algorithms [12], routing optimization for drones to be used in cargo transportation [13], use of drones in agricultural activities [14], creating the mathematical model of the quadcopter by obtaining the equations of motion and rotation according to Newton's laws [15], quadcopter design that can carry a fire-extinguishing ball [16], improving battery performance in drones [17], estimating the remaining flight time on drones [18, 19], creating a model for the battery [20], detecting the location of the drone in the face of mechanical failure and cyber-attack with an external system added to the drone's battery [21], development of an algorithm to reduce energy consumption in drones used in the commercial package works and performing autonomous flight [22], determination of battery capacity in lion batteries using artificial intelligence [23], recommendation of a system with adaptive speed and avoidance algorithm for drones [24], shortest path estimation for robot and unmanned aerial vehicles using artificial neural networks [25], automatic route determination and tracking in drones [26] are some of them.

In terms of similarity to this study, three remarkable studies were encountered in the literature.

In the first study presented in [27], a drone with a total weight of 1.4 kg with a payload was designed and it was stated that it could avoid obstacles. However, up to what speeds the drone can avoid obstacles and experimental studies conducted on this have not been shared.

Another study is the one given in the source numbered [28]. In the study presented in [28], the effect of ultrasonic sensors on drone noise and related measurement errors could not be eliminated.

The remarkable last study is the one given in [29]. In the study presented in [29], it was stated that the drone could only avoid obstacles at speeds below 3 m/s. The weight of the designed drone was also not mentioned at all.



As can be seen, preventing drones from avoiding obstacles and breaking down due to user error or environmental factors is still an issue that needs improvement, both academically and commercially. The aim of this experimental study is to make an improvement that will prevent drones from crashing into an obstacle while flying at speeds above 3 m/s, regardless of the reason.

## **2. Material and Method**

The sensors used and their performance are the most decisive factors in achieving the targeted goal. Because in academic research, it has been seen that obstacle detection sensors are affected by electrical noises, and as a result, the success of the study decreases. For this reason, these sensors and their integration into the system were researched as a priority and a suitable drone was designed accordingly. Then, software filtering methods were used and an algorithm was developed in order to both minimize communication instabilities and increase detection clarity. Finally, the drone's ability to avoid both while suspended in the air and motion was experimentally investigated.

### **2.1 Hardware components of the drone used in the experimental study**

Components such as propellers that will enable the drone to take off, brushless dc motors, drivers that will control the speeds of these motors, battery systems, and frames have been examined and the most suitable and compatible ones of them have been investigated. As a result of this research, the following components were selected:

- Pixhawk as the flight control board
- Arduino Uno as a microcontroller
- HC-SR04 ultrasonic sensors as obstacle detection sensors
- S500 frame
- 3300mAh LiPo battery
- 2212/920 model brushless DC motor and 9450 propeller set produced by DJI Company
- 40A electronic speed controller manufactured by Readytosky
- XROCK brand Radio V5 telemetry system
- FS-16X model remote control manufactured by Flysky Company
- NEO-M8N GPS module

### **2.2 Software components of the drone used in the experimental study**

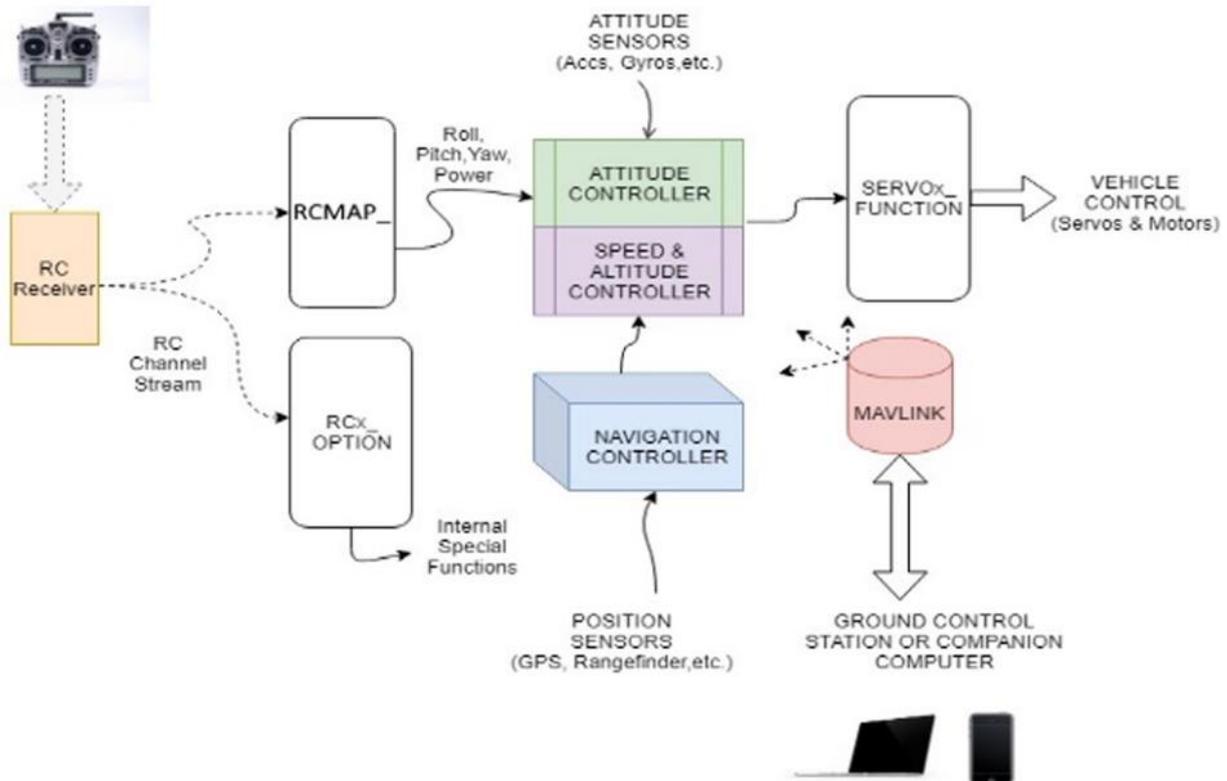
#### **2.2.1 MAVLink**

MAVLink is an abbreviation of Micro Air Vehicle Link and is a communication protocol. It has been developed to provide secure data exchange between the ground station and the aircraft. It is used to provide communication between Arduino, Raspberry Pi, and similar microcontrollers and aircraft. Messages sent in encrypted form are received only by the flight controller. Messages such as flight mode changes are also sent by the ground station or microcontroller. According to these messages sent via the MAVLink protocol, the flight controller directs the drone. In this study, MAVLink communication was used to send the codes written for Arduino Uno to the flight controller and to exchange data between the flight controller and the controller.



### 2.2.2 ArduPilot

ArduPilot is an open-source software package and is widely used for controlling both ground and air vehicles. While performing autonomous movement, it evaluates information from flight control cards and external sensors, as can be seen in the architecture given in Figure 1. ArduPilot was preferred in this study because it was developed by the same platform as Pixhawk.



**Fig. 1.** The basic structure of ArduPilot software architecture and sensor connections [30].

### 2.2.3 Mission Planner

The program called Mission Planner is the most preferred control platform of ArduPilot software. In addition to many flight modes, very detailed flight records can also be accessed on this platform. Figure 2 shows the interface of the Mission Planner program.

The reason for using Mission Planner in this study is to be able to examine the drone's speed and maneuvers by using the log records it provides so that it can clearly see to what extent and at what speed the drone can achieve obstacle avoidance.

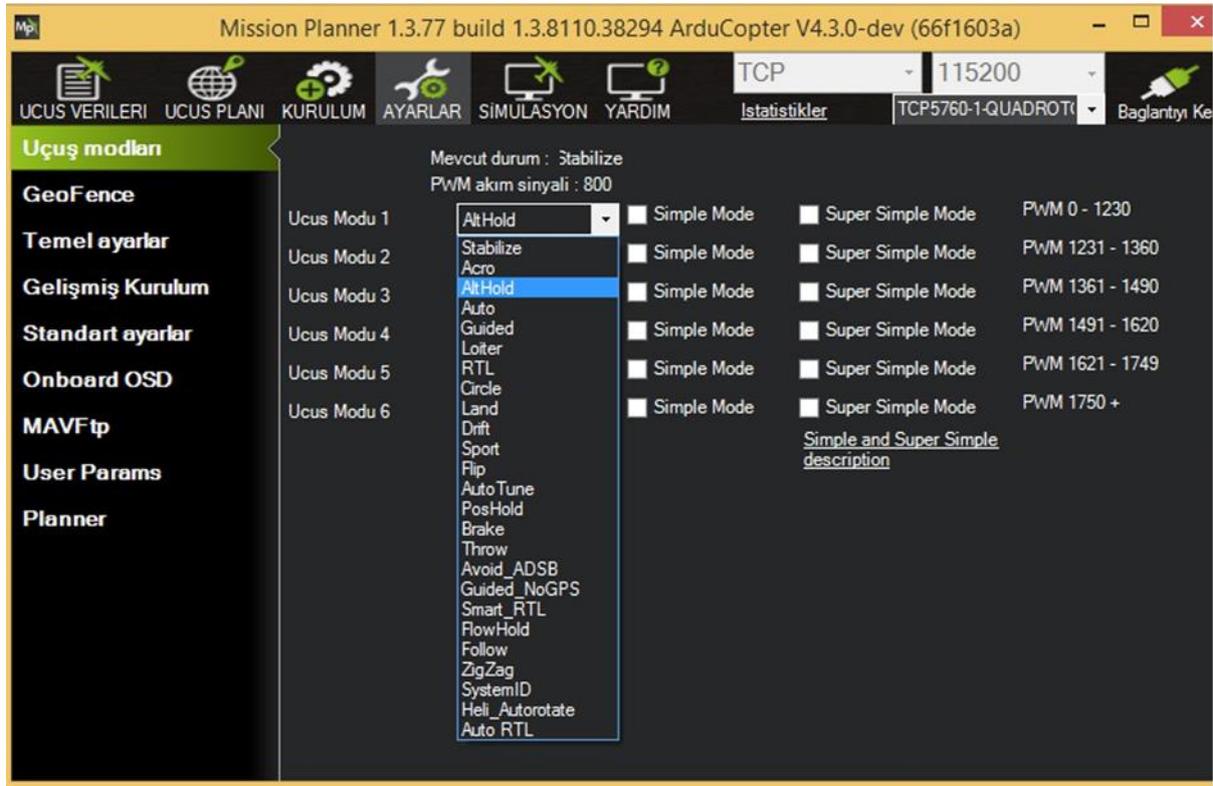


Fig. 2. The interface of the program called Mission Planner.

### 2.3 The sensor used for the drone to detect the obstacle

Many different sensors such as ultrasonic, infrared, and LIDAR (Light Detection and Ranging) are used in autonomous systems to detect obstacles. However, in this study, the HC-SR04 ultrasonic sensor was preferred due to its price/performance ratio and compatibility with Arduino Uno.

As is known, ultrasonic sensors are sensing elements used to detect the presence and distance of an object based on the time that the sound waves they send are reflected from the object and come back. Using these sensors, it is possible to detect objects at a distance of a little more than 3 m [31].

The angle of the detection cone in ultrasonic sensors varies with distance, as seen in Figure 3. This angle must be small in order to detect objects at greater distances.

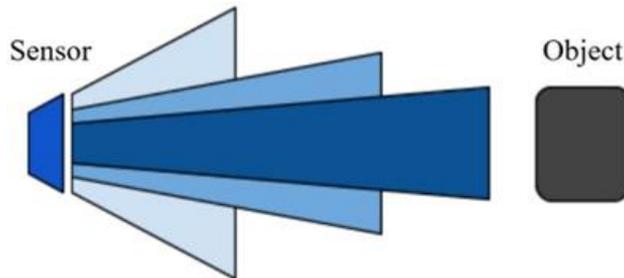


Fig. 3. Variation of the sensing distance according to the sensing cone in ultrasonic sensors [32].



The ability of the sensor to detect objects also depends on the size of the objects and their orientation to the sensor. For example, if the objects do not present a surface perpendicular to the sound signal sent by the sensor, the sound wave cannot return to the sensor. When the size of the object is too small, there will be no reflection at all. This situation is illustrated in Figure 4.

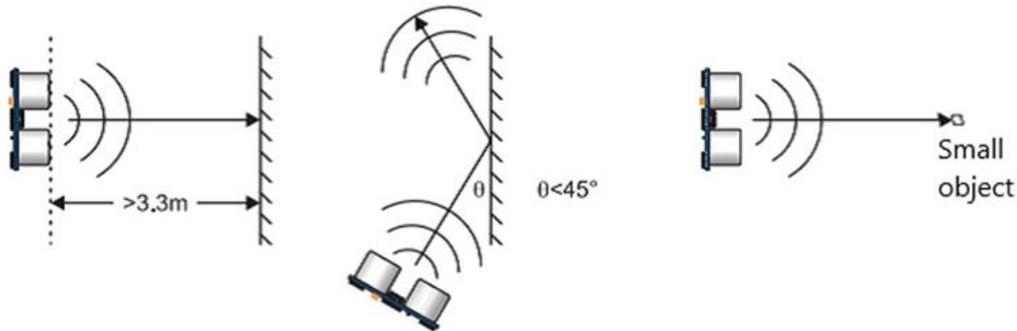


Fig. 4. Situations where the ultrasonic sensor cannot detect.

As can be seen in Figure 5, there are four pins on the HC-SR4 ultrasonic sensor, namely VCC, TRIG, ECHO, and GND. In order for the sensor to perform its function, a “Start pulse” of about 10µs is sent from the TRIG pin. The sensor generates a signal of 8 pulses at a frequency of 40 kHz and sends it to the transmitter component. The time difference between the moment this pulse is sent for 10µs and the time it reaches ECHO is used to calculate the distance.

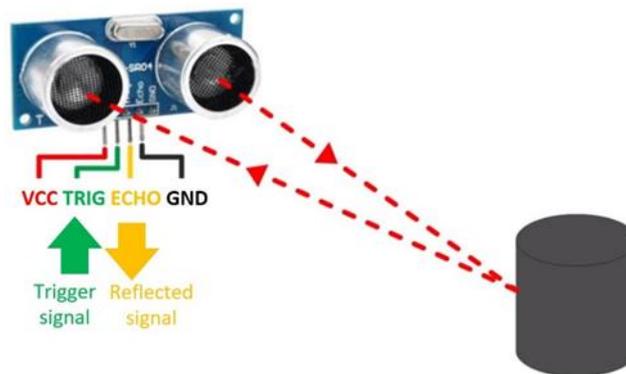


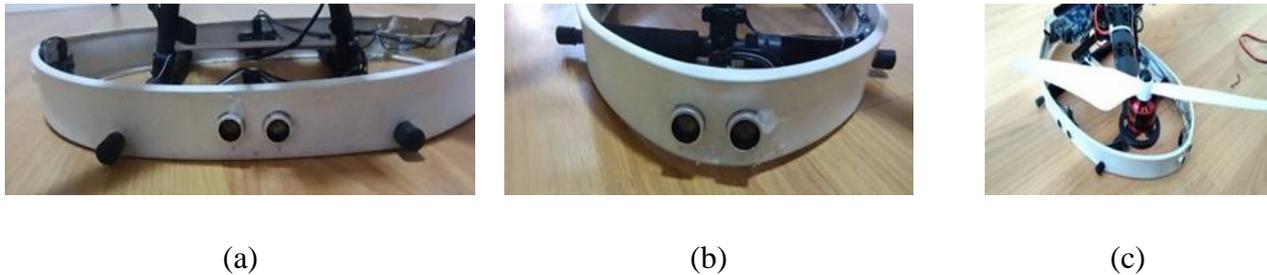
Fig. 5. Operation of the ultrasonic sensor.

In the process of calculating the distance, it is used that the speed of ultrasonic sound waves under certain atmospheric conditions is known. The product of half of the sound wave's traveling time (t) and the propagation speed of the sound (V) gives the distance (X) information. This calculation is seen in Equation 1.

$$X = \frac{t}{2} \cdot V \tag{1}$$



Since the obstacle detection distance will be large, precision at the mm level is not required. However, since the detection of the obstacle is made by means of sound waves, both a suitable placement should be planned and software measures should be taken so that the drone propellers do not affect these waves. For this purpose, the sensors are placed at the bottom of the drone, which is expected to be least affected by noise, as can be seen in Figure 6.



**Fig. 6.** Layouts of ultrasonic sensors (a) Front view, (b) Side view, (c) View from a different angle

The weight of the drone, which is completed by combining ultrasonic sensors and all other components, is 1232 gr without battery and 1568 gr with battery. The view of the drone from two different angles is given in Figure 7.



**Fig. 7.** View of the designed drone from two different angles

#### 2.4 Connection diagram between ultrasonic sensors, Arduino Uno, and Pixhawk

The connections of the Arduino Uno with the ultrasonic sensors were made via digital pins 3, 4, 5, and 6. The TRIG and ECHO terminals of the ultrasonic sensor can be connected to the microcontroller separately. However, combining these terminals and connecting them from a single place eliminates the problem that there are no pins to connect on the microcontroller when the number of sensors is increased. For this reason, as can be seen in Figure 8, the TRIG and ECHO terminals of all the ultrasonic sensors used were combined among themselves and connected to the microcontroller. Pins 10 and 11 of Arduino Uno were used for serial communication with Pixhawk and were connected to each other via the MAVLink protocol.

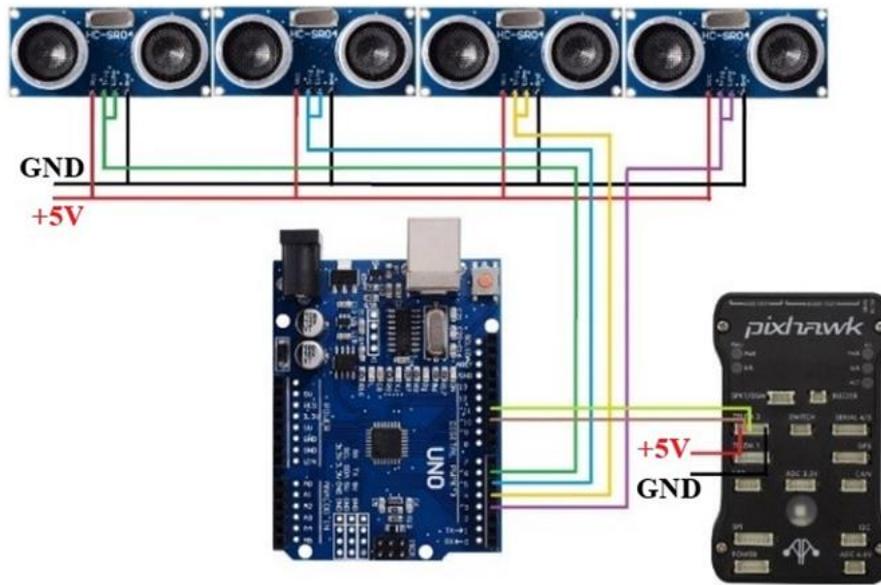


Fig. 8. Connection diagram between ultrasonic sensors, Arduino Uno, and Pixhawk.

### 2.5 Communication diagram between hardware

The Arduino Uno, which processes the information from the sensors, configures the flight controller according to the information obtained. If information is received from which sensor during this configuration process, the algorithm written specifically for that sensor is activated and the drone acts according to this algorithm. Therefore, in this process, the information coming from the controller is disabled. The remote control is connected to Pixhawk via MAVLink protocol. Arduino and Pixhawk are connected to each other via MAVLink protocol via I2C. The block diagram of this communication between the hardware is shown in Figure 9.



Fig. 9. Block diagram of communication between hardware.



## 2.6 Algorithm

When the drone detects an obstacle, it is necessary to be able to make the drone pitch and roll movements to enable it to escape from the obstacle. The flow diagram of the algorithm created for this purpose is given in Figure 10.

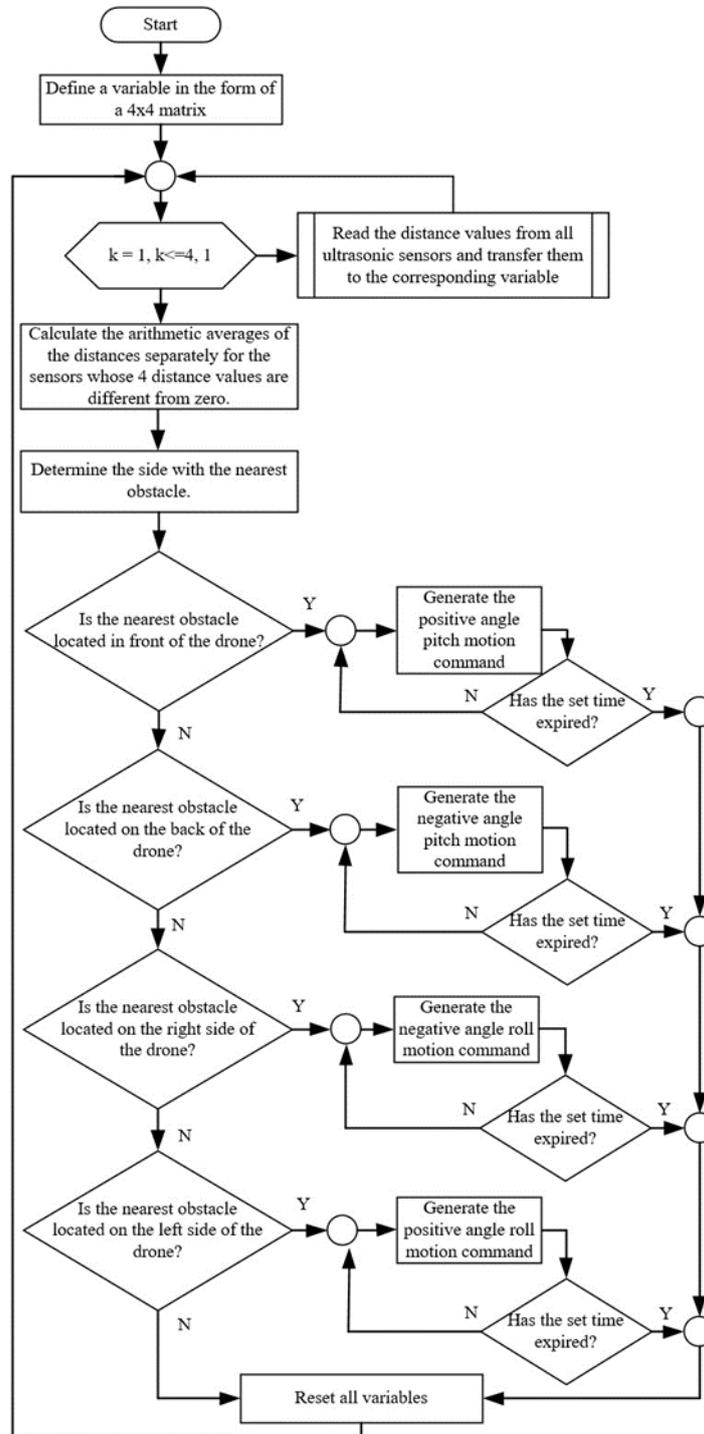


Fig. 10. Obstacle detection and avoidance algorithm.



As can be understood from the algorithm, according to the information received from any sensor, the drone is forced to move in the opposite direction to the side where the obstacle is located for a certain period of time. At the end of the specified time, it continues its flight again autonomously or under control.

As can be seen in Figure 10, a 4x4 matrix was first created in the written program. The first measurement values (a, b, c, d) taken from the sensors was transferred to the first column of this matrix as can be seen in Table 1.

**Table 1.** First measurement (The first values measured by the sensors were transferred to the 1st column).

Sensor	Value	Value	Value	Value
Front	<i>a</i>			
Right	<i>b</i>			
Rear	<i>c</i>			
Left	<i>d</i>			

Each time the loop was entered, the values read in the previous loop were transferred to the next column, and the values taken in that measurement were placed in the first column. Therefore, at the end of the 4th measurement, a filled 4x4 matrix was obtained by taking 4 pieces of data from each sensor at cycle time intervals. This data shifting process and the content change of the variable in the matrix form will be understood more clearly when Table 2 - Table 4 is examined.

**Table 2.** Second measurement (The first values measured by the sensors were transferred to column 2, new values were placed in column 1).

Sensor	Value	Value	Value	Value
Front	<i>e</i>	<i>a</i>		
Right	<i>f</i>	<i>b</i>		
Rear	<i>g</i>	<i>c</i>		
Left	<i>h</i>	<i>d</i>		

**Table 3.** Third measurement (The first values measured by the sensors were transferred to the 3rd column, the 2nd values to the 2nd column, the new values to the first column).

Sensor	Value	Value	Value	Value
Front	<i>i</i>	<i>e</i>	<i>a</i>	
Right	<i>j</i>	<i>f</i>	<i>b</i>	
Rear	<i>k</i>	<i>g</i>	<i>c</i>	
Left	<i>l</i>	<i>h</i>	<i>d</i>	

**Table 4.** Fourth measurement (The first values measured by the sensors were transferred to the 4th column, the second values to the 3rd, the third values the 2nd, and the new measurements were transferred to the first column).

Sensor	Value	Value	Value	Value
Front	<i>m</i>	<i>i</i>	<i>e</i>	<i>a</i>
Right	<i>n</i>	<i>j</i>	<i>f</i>	<i>b</i>
Rear	<i>p</i>	<i>k</i>	<i>g</i>	<i>c</i>
Left	<i>r</i>	<i>l</i>	<i>h</i>	<i>d</i>



Then, the arithmetic average was calculated separately for all the sensors whose consecutive 4 data were different from zero. If there is zero value in the matrix (that is, no obstacle is detected), the arithmetic average of the sensor to which that row belongs is set to zero. In this way, it was determined whether the data coming from each sensor was a reflected or parasitic signal from the obstacle. Using these final average values, the distance of the nearest obstacle, and which side it is on were determined. Then, the finalized distance value was transferred to the subprograms that determine which of the pitch or roll movements the drone will perform. The commands obtained from here were sent to Pixhawk via the MAVLink communication protocol and accordingly, the user control was disabled for a certain period of time and the drone was allowed to maneuver according to the incoming information. The positive and negative angle pitch and roll movements mentioned in the algorithm are shown in Figure 11 and Figure 12, respectively.

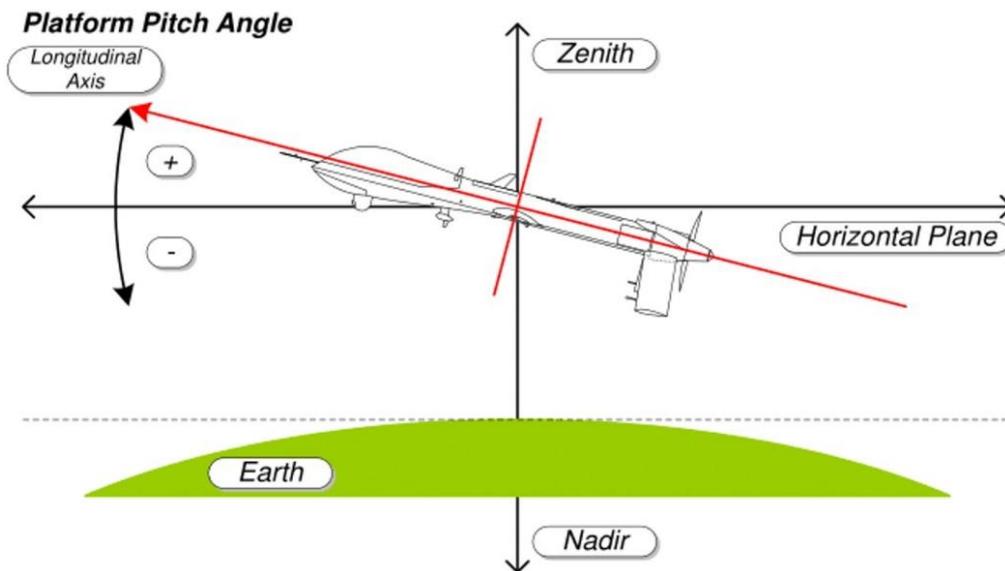


Fig. 11. Positive and negative pitch movement in the aircraft.

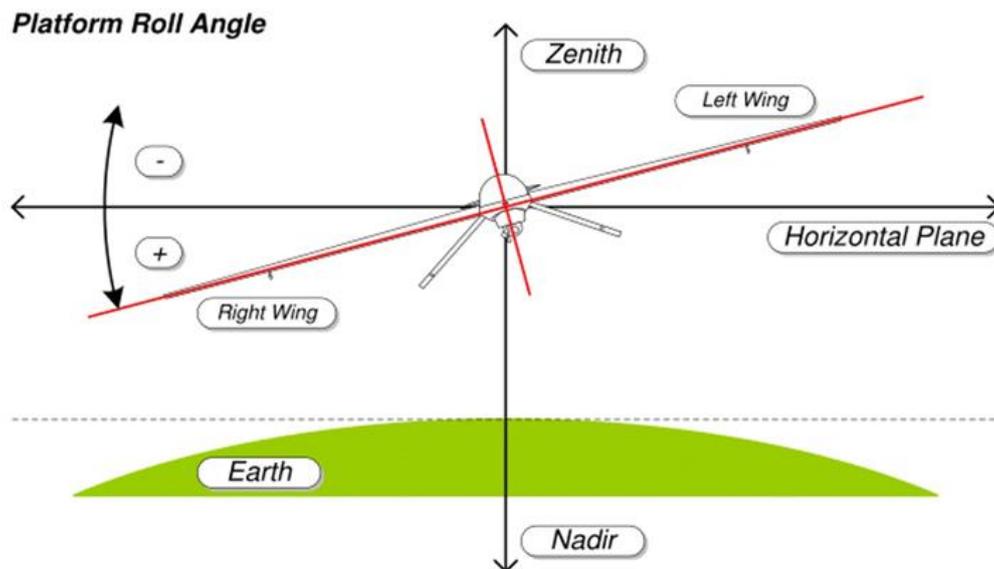


Fig. 12. Positive and negative angle roll movement in the aircraft.



### 3. Experimental Study

Experimental studies are presented in this section. Firstly, the sensors were disabled, flights were carried out with the basic flight modes and the most suitable flight mode was decided. Then, the placement of ultrasonic sensors, their integration into the system, and various flight tests were carried out in the specified flight modes. The tests were carried out by first fixing the drone to the ground and then flying it in different forms. Therefore, experimental studies will be examined gradually in order to provide a clearer understanding of the results obtained.

#### 3.1 Distance measurement tests while the drone is fixed to the ground

After the drone was fixed to the ground, tests were carried out both while the propellers were not working and while operating at very high speeds. The purpose of these initial tests is to observe whether interference occurs during data exchange between ultrasonic sensors, Arduino, and Pixhawk.

In this context, first, the drone was fixed to the ground, then numerous tests were carried out by placing objects at different distances on different sides of the drone. Tests were performed by placing obstacles on only the front, only the right, only the back, and only the left sides. Serial port screenshots of the data obtained from the tests were examined and it was examined whether the system works properly, whether the sensors could detect the distances of the obstacles correctly, and whether the interference was occurring.

In this examination, tests were first carried out by placing an obstacle in front of the ultrasonic sensor located on the front of the drone. In the tests carried out, the distance values produced by the ultrasonic sensor located on the front of the drone when the motors are not working at a certain time interval are given in Figure 13. The expression “61, 0, 0, 0 cm” in Figure 13 refers to the distance values measured by the front, right, rear, and left sensors, respectively. When Figure 13 is examined, it will be seen that in the measurement of the distance in cm of the fixed obstacle at a distance of 62 cm, the measured distance is generally correct, but sometimes it is measured as 61 cm and sometimes 63 cm with an acceptable error of  $\pm 1.61\%$ . It is understood from Figure 13 that no interference value is produced by the other 3 sensors that do not have obstacles in front of them.

```
COM5
17:54:55.008 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.008 ->
17:54:55.008 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.008 ->
17:54:55.055 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.055 ->
17:54:55.102 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.102 ->
17:54:55.102 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.102 ->
17:54:55.149 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.149 ->
17:54:55.149 -> RAM_VAL: 61, 0, 0, 0cm
17:54:55.196 ->
17:54:55.196 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.243 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.243 ->
17:54:55.243 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.243 ->
17:54:55.290 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.290 ->
17:54:55.290 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.336 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.336 ->
17:54:55.383 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.383 ->
17:54:55.383 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.383 ->
17:54:55.383 -> RAM_VAL: 63, 0, 0, 0cm
17:54:55.383 ->
17:54:55.430 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.430 ->
17:54:55.477 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.477 ->
17:54:55.477 -> RAM_VAL: 62, 0, 0, 0cm
17:54:55.477 ->
17:54:55.477 -> RAM_VAL:
```

Fig. 13. Serial port screenshot of the data received from the front sensor in a certain time interval when the motors are not working.



In the next test, the distance values produced by the ultrasonic sensor located on the right side of the drone were examined at a certain time interval when the motors were not working. When the values obtained from these tests and given in Figure 14 were examined, it was found that the measured values were generally correct when measuring the distance of a fixed obstacle at a distance of 55 cm. However, it will be seen that sometimes the distance is measured as 56 cm with an acceptable error of 1.8%, and no interference value is produced/ detected by the other 3 sensors that do not have obstacles in front of them.

```
17:51:14.122 ->
17:51:14.169 -> RAM_VAL: 0,55,0,0cm
17:51:14.169 ->
17:51:14.169 -> RAM_VAL: 0,55,0,0cm
17:51:14.169 ->
17:51:14.216 -> RAM_VAL: 0,55,0,0cm
17:51:14.216 ->
17:51:14.263 -> RAM_VAL: 0,55,0,0cm
17:51:14.263 ->
17:51:14.263 -> RAM_VAL: 0,55,0,0cm
17:51:14.263 ->
17:51:14.310 -> RAM_VAL: 0,55,0,0cm
17:51:14.310 ->
17:51:14.310 -> RAM_VAL: 0,55,0,0cm
17:51:14.310 ->
17:51:14.356 -> RAM_VAL: 0,55,0,0cm
17:51:14.356 ->
17:51:14.403 -> RAM_VAL: 0,55,0,0cm
17:51:14.403 ->
17:51:14.403 -> RAM_VAL: 0,55,0,0cm
17:51:14.403 ->
17:51:14.450 -> RAM_VAL: 0,56,0,0cm
17:51:14.450 ->
17:51:14.591 -> RAM_VAL: 0,56,0,0cm
17:51:14.591 ->
17:51:14.591 -> RAM_VAL: 0,56,0,0cm
17:51:14.591 ->
17:51:14.638 -> RAM_VAL: 0,56,0,0cm
17:51:14.638 ->
17:51:14.685 -> RAM_VAL: 0,56,0,0cm
17:51:14.685 ->
17:51:14.685 -> RAM_VAL: 0,55,0,0cm
17:51:14.685 ->
17:51:14.685 ->
17:51:14.826 -> RAM_VAL: 0,55,0,0cm
17:51:14.826 ->
17:51:14.872 -> RAM_VAL: 0,55,0,0cm
17:51:14.872 ->
17:51:14.872 -> RAM_VAL: 0,55,0,0cm
17:51:14.872 ->
```

Fig. 14. Serial port screenshot of the data received from the right sensor in a certain time interval when the motors are not working.

Serial port screenshots of the data received from the rear and left sensor at a certain time interval while the motors are not working are shown in Figure 15 and Figure 16, respectively. When these two figures are examined, it will be seen that no parasitic value is detected by the other 3 sensors that do not have an obstacle in front of them.

```
17:52:55.565 ->
17:52:55.612 -> RAM_VAL: 0,0,0,24cm
17:52:55.612 ->
17:52:55.658 -> RAM_VAL: 0,0,0,24cm
17:52:55.658 ->
17:52:55.658 -> RAM_VAL: 0,0,0,24cm
17:52:55.658 ->
17:52:55.705 -> RAM_VAL: 0,0,0,24cm
17:52:55.705 ->
17:52:55.705 -> RAM_VAL: 0,0,0,24cm
17:52:55.705 ->
17:52:55.752 -> RAM_VAL: 0,0,0,24cm
17:52:55.752 ->
17:52:55.752 -> RAM_VAL: 0,0,0,24cm
17:52:55.752 ->
17:52:55.799 -> RAM_VAL: 0,0,0,24cm
17:52:55.799 ->
17:52:55.846 -> RAM_VAL: 0,0,0,24cm
17:52:55.846 ->
17:52:55.846 -> RAM_VAL: 0,0,0,24cm
17:52:55.846 ->
17:52:55.893 -> RAM_VAL: 0,0,0,24cm
17:52:55.893 ->
17:52:55.893 -> RAM_VAL: 0,0,0,24cm
17:52:55.893 ->
17:52:55.940 -> RAM_VAL: 0,0,0,24cm
17:52:55.940 ->
17:52:55.940 -> RAM_VAL: 0,0,0,24cm
17:52:55.940 ->
17:52:55.987 -> RAM_VAL: 0,0,0,24cm
17:52:55.987 ->
17:52:55.987 -> RAM_VAL: 0,0,0,24cm
17:52:55.987 ->
17:52:56.033 -> RAM_VAL: 0,0,0,24cm
17:52:56.033 ->
17:52:56.080 -> RAM_VAL: 0,0,0,24cm
17:52:56.080 ->
17:52:56.080 -> RAM_VAL: 0,0,0,24cm
17:52:56.080 ->
```

Fig. 15. Serial port screenshot of the data received from the rear sensor in a certain time interval when the motors are not working.



```
COM5
17:52:55.565 -->
17:52:55.612 --> RAW_VAL: 0,0,0,24cm
17:52:55.612 -->
17:52:55.659 --> RAW_VAL: 0,0,0,24cm
17:52:55.659 -->
17:52:55.658 --> RAW_VAL: 0,0,0,24cm
17:52:55.658 -->
17:52:55.705 --> RAW_VAL: 0,0,0,24cm
17:52:55.705 -->
17:52:55.705 --> RAW_VAL: 0,0,0,24cm
17:52:55.705 -->
17:52:55.752 --> RAW_VAL: 0,0,0,24cm
17:52:55.752 -->
17:52:55.752 --> RAW_VAL: 0,0,0,24cm
17:52:55.752 -->
17:52:55.799 --> RAW_VAL: 0,0,0,24cm
17:52:55.799 -->
17:52:55.846 --> RAW_VAL: 0,0,0,24cm
17:52:55.846 -->
17:52:55.846 --> RAW_VAL: 0,0,0,24cm
17:52:55.846 -->
17:52:55.893 --> RAW_VAL: 0,0,0,24cm
17:52:55.893 -->
17:52:55.893 --> RAW_VAL: 0,0,0,24cm
17:52:55.893 -->
17:52:55.940 --> RAW_VAL: 0,0,0,24cm
17:52:55.940 -->
17:52:55.940 --> RAW_VAL: 0,0,0,24cm
17:52:55.940 -->
17:52:55.987 --> RAW_VAL: 0,0,0,24cm
17:52:55.987 -->
17:52:55.987 --> RAW_VAL: 0,0,0,24cm
17:52:55.987 -->
17:52:56.033 --> RAW_VAL: 0,0,0,24cm
17:52:56.033 -->
17:52:56.080 --> RAW_VAL: 0,0,0,24cm
17:52:56.080 -->
17:52:56.080 --> RAW_VAL: 0,0,0,24cm
17:52:56.080 -->
```

Fig. 16. Serial port screenshot of the data received from the left sensor in a certain time interval when the motors are not working.

In the second stage, the drone was still fixed to the ground again, but tests were carried out when the motors were working at very high speeds. As an example of the tests carried out in this context, the distance values produced by the ultrasonic sensors on the front and right sides of the drone at a certain time interval were taken given in Figure 17 and Figure 18, respectively. When both screenshots are examined, it will be seen that no parasitic values caused by high-speed motors were produced and the measured distance values remained stable at 44 cm and 43 cm, respectively.

```
COM5
17:49:10.506 -->
17:49:10.553 --> RAW_VAL: 44,0,0,0cm
17:49:10.553 -->
17:49:10.553 --> RAW_VAL: 44,0,0,0cm
17:49:10.553 -->
17:49:10.600 --> RAW_VAL: 44,0,0,0cm
17:49:10.600 -->
17:49:10.647 --> RAW_VAL: 44,0,0,0cm
17:49:10.647 -->
17:49:10.647 --> RAW_VAL: 44,0,0,0cm
17:49:10.647 -->
17:49:10.694 --> RAW_VAL: 44,0,0,0cm
17:49:10.694 -->
17:49:10.694 --> RAW_VAL: 44,0,0,0cm
17:49:10.694 -->
17:49:10.741 --> RAW_VAL: 44,0,0,0cm
17:49:10.741 -->
17:49:10.741 --> RAW_VAL: 44,0,0,0cm
17:49:10.741 -->
17:49:10.788 --> RAW_VAL: 44,0,0,0cm
17:49:10.788 -->
17:49:10.788 --> RAW_VAL: 44,0,0,0cm
17:49:10.788 -->
17:49:10.835 --> RAW_VAL: 44,0,0,0cm
17:49:10.835 -->
17:49:10.835 --> RAW_VAL: 44,0,0,0cm
17:49:10.835 -->
17:49:10.881 --> RAW_VAL: 44,0,0,0cm
17:49:10.881 -->
17:49:10.928 --> RAW_VAL: 44,0,0,0cm
17:49:10.928 -->
17:49:10.928 --> RAW_VAL: 44,0,0,0cm
17:49:10.928 -->
17:49:10.975 --> RAW_VAL: 44,0,0,0cm
17:49:10.975 -->
17:49:10.975 --> RAW_VAL: 44,0,0,0cm
17:49:10.975 -->
17:49:11.022 --> RAW_VAL: 44,0,0,0cm
17:49:11.022 -->
17:49:11.022 --> RAW_VAL: 44,0,0,0cm
17:49:11.022 -->
```

Fig. 17. Serial port screenshot of the data received from the front sensor in a certain time interval while the motors are working at high speed.



```
COM5
17:56:25.695 -> RAW_VAL: 0,48,0,0cm
17:56:25.695 ->
17:56:25.732 -> RAW_VAL: 0,48,0,0cm
17:56:25.732 ->
17:56:25.779 -> RAW_VAL: 0,48,0,0cm
17:56:25.779 ->
17:56:25.779 -> RAW_VAL: 0,48,0,0cm
17:56:25.825 ->
17:56:25.825 -> RAW_VAL: 0,48,0,0cm
17:56:25.825 ->
17:56:25.825 -> RAW_VAL: 0,48,0,0cm
17:56:25.825 ->
17:56:25.872 -> RAW_VAL: 0,48,0,0cm
17:56:25.872 ->
17:56:25.872 -> RAW_VAL: 0,48,0,0cm
17:56:25.872 ->
17:56:25.872 -> RAW_VAL: 0,48,0,0cm
17:56:25.919 ->
17:56:25.919 -> RAW_VAL: 0,48,0,0cm
17:56:25.966 ->
17:56:25.966 -> RAW_VAL: 0,48,0,0cm
17:56:25.966 ->
17:56:25.966 -> RAW_VAL: 0,48,0,0cm
17:56:25.966 ->
17:56:26.013 -> RAW_VAL: 0,48,0,0cm
17:56:26.013 ->
17:56:26.060 -> RAW_VAL: 0,48,0,0cm
17:56:26.060 ->
17:56:26.060 -> RAW_VAL: 0,48,0,0cm
17:56:26.060 ->
17:56:26.060 -> RAW_VAL: 0,48,0,0cm
17:56:26.060 ->
17:56:26.107 -> RAW_VAL: 0,48,0,0cm
17:56:26.107 ->
17:56:26.107 -> RAW_VAL: 0,48,0,0cm
17:56:26.154 ->
17:56:26.154 -> RAW_VAL: 0,48,0,0cm
```

Fig. 18. Serial port screenshot of the data received from the right sensor in a certain time interval while the motors are working at high speed.

If the serial port screenshots in Figure 13-Figure 18 are also examined in terms of whether sensors that are not obstacles in front of them produce interference, it will be seen that all sensors that are not obstacles in front of them do not produce interference, so any distance values other than 0 are not read.

### 3.2 Obstacle avoidance tests while the drone is fixed to the ground

In the next stage, tests of obstacle avoidance maneuvers of the drone when it detects these obstacles were started. Firstly, the drone was fixed to the ground, then, tests were carried out while the drone's motors were not working and working at high speed. In these tests, the changes in pitch, roll, and throttle values depending on the direction in which the obstacle is perceived were examined.

In this context, first of all, when an obstacle is detected only from the front, the drone's escape from the obstacle by making a pitch movement with a positive angle (towards the back) was examined. Serial port screenshots of the test are shared in Figure 19. The pitch, roll, and throttle values are determined according to the distance data obtained from the ultrasonic sensors. When these values are examined, the following results will be reached:

- The roll value, which should not have been generated because an obstacle was detected from the front, was realized as expected. In other words, the parasitic value was not generated and the roll value remained zero.
- The pitch value varied depending on the distance of the obstacle.
- The throttle value is also different from zero. The reason for this is to make the drone escape from the obstacle in a shorter time by generating more thrust. Therefore, there is a larger pitch and throttle value when closer to the obstacle and a smaller one when moving away from the obstacle.

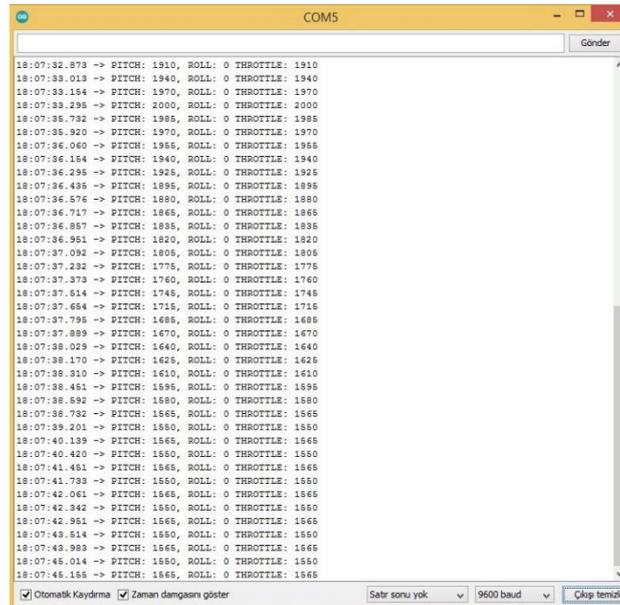


Fig. 19. Variation of pitch, roll, and throttle values in any time interval when an obstacle is detected in front of the motors while they are not working.

A serial port screenshot of the change in the roll value that will enable the drone to avoid the obstacle by lying to the left (making a negative angle roll movement) when an obstacle is detected only from the right side is given in Figure 20. When the pitch, roll, and throttle values in Figure 20 are examined, it is seen that the pitch value, which should not be generated at all because the obstacle is detected only from the right, is realized as expected (that is, no interference value is generated and the pitch value remains zero). It is a software preference that the throttle value, which receives a value other than zero when an obstacle is detected from the front, remains zero here. Therefore, it is left as zero except for the case of only front and only rear obstacle detection. This preference means that if the throttle value at the moment when the obstacle is detected is lost, the motors will continue to work with the same value and there will be no change in speed.

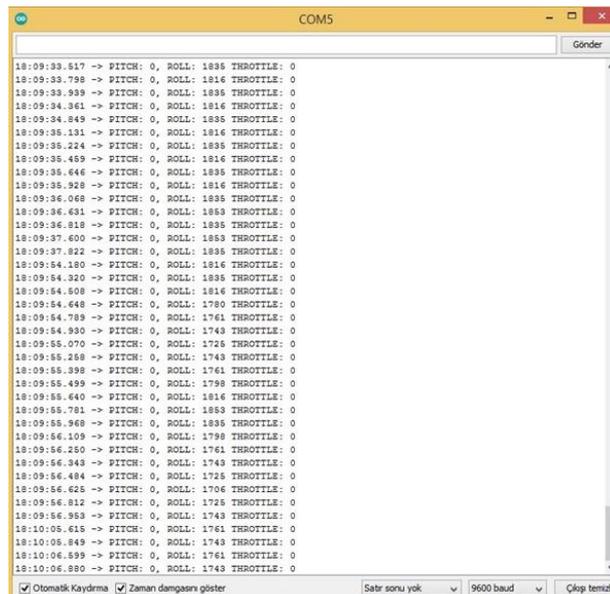
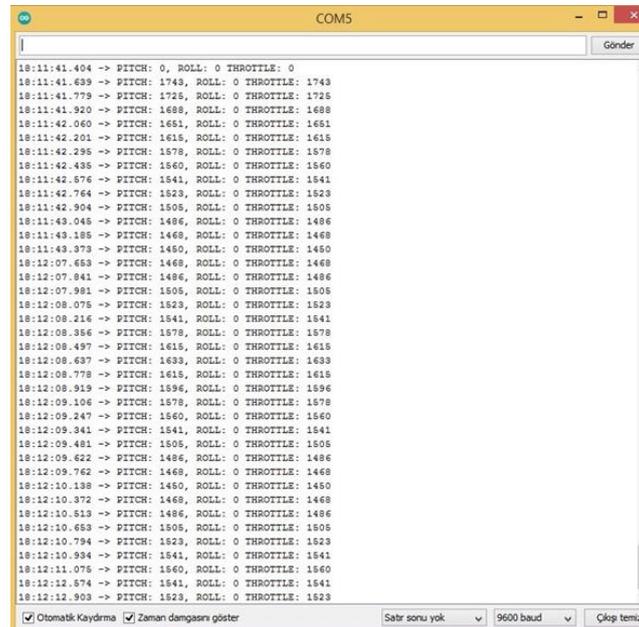


Fig. 20. Variation of pitch, roll, and throttle values in any time interval when an obstacle is detected on the right side while the motors are not working.



Figure 21 shows a serial port screenshot of the change that allows the drone to escape from the obstacle by making a forward (negative angle) pitch movement when an obstacle is detected only from the rear side.



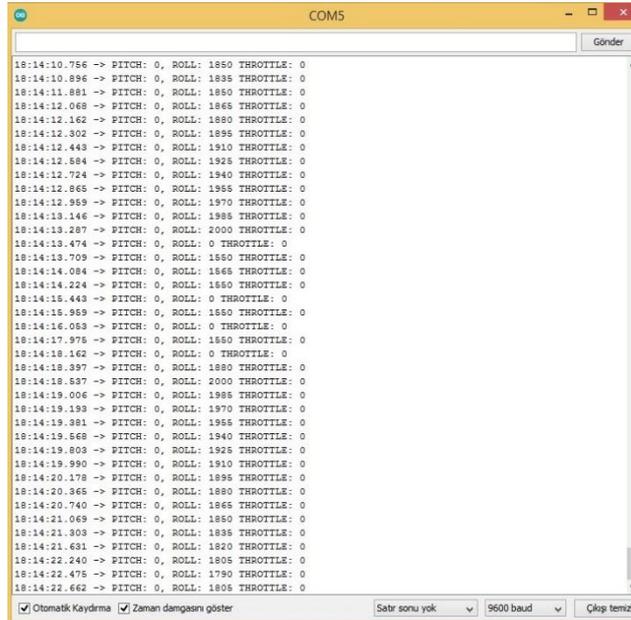


Fig. 22. Variation of pitch, roll and throttle values in any time interval when the motors detect an obstacle on the left side while they are not working.

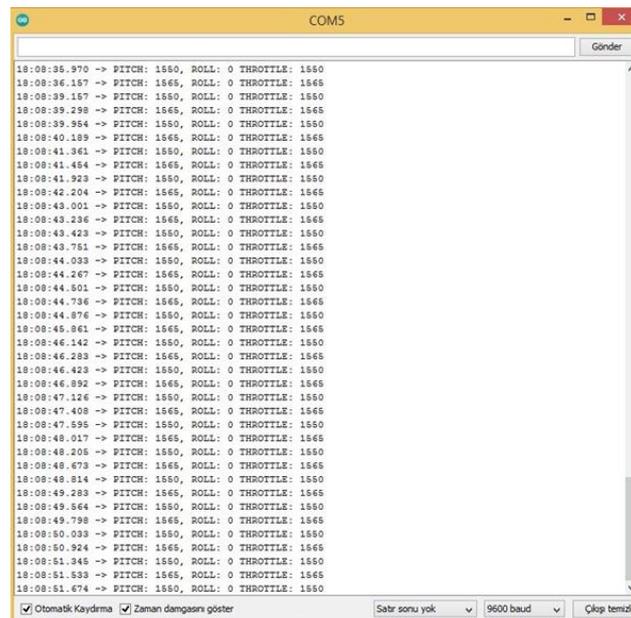


Fig. 23. The variation of pitch, roll and throttle values in any time interval when the motors are working at very high speeds and only when a fixed obstacle is detected in front.

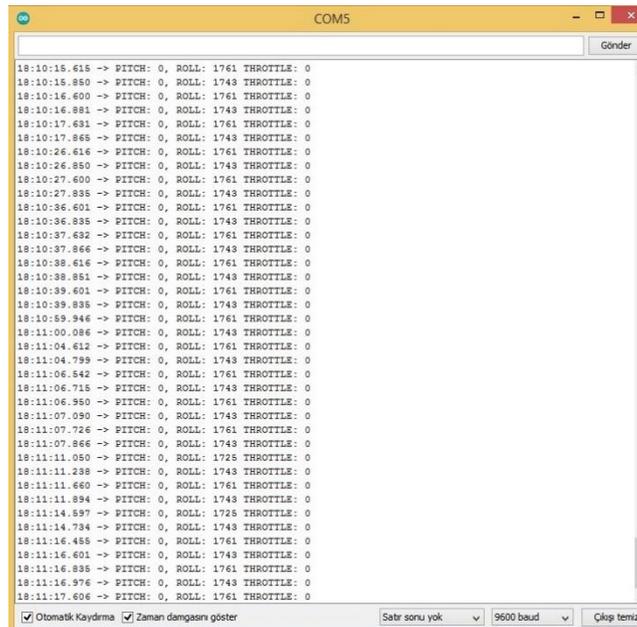


Fig. 24. Variation of pitch, roll and throttle values in any time interval when the motors are working at very high speeds and detect a fixed obstacle on the right side only.

However, when examining Figure 25, where only the rear obstacle state is given, and Figure 26, where only the left obstacle state is given, it will be seen that the values that should not be zero are momentarily zero several times due to the momentary interruption of the communication between Arduino and Pixhawk. This momentary communication interruption between Arduino and Pixhawk did not cause major problems in the field tests with the drone.

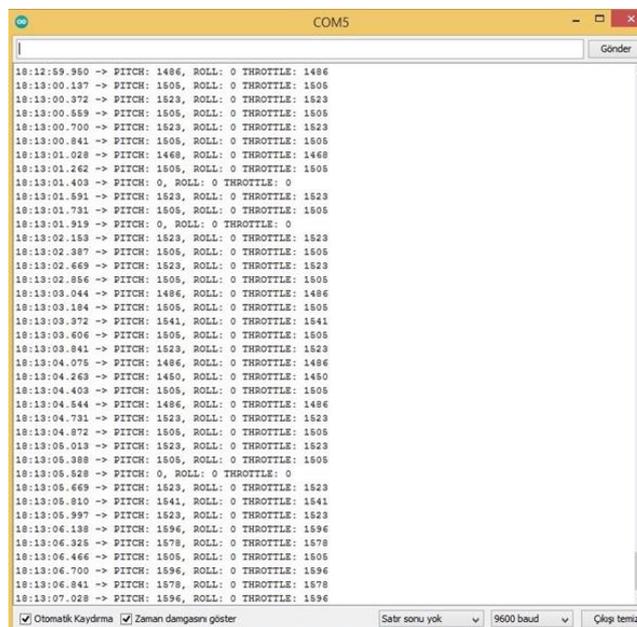


Fig. 25. Variation of pitch, roll and throttle values in any time interval when the motors are working at very high speed and only when an obstacle is detected at the rear.

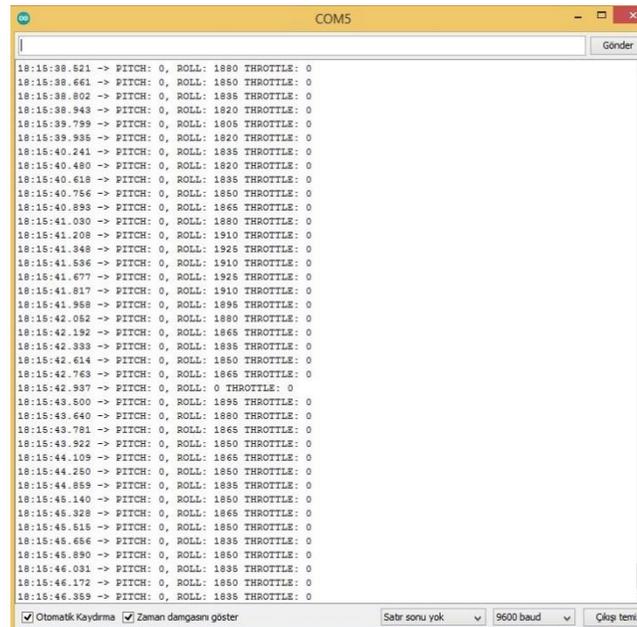


Fig. 26. Variation of pitch, roll and throttle values in any time interval when the motors are working at very high speeds and only detect an obstacle on the left side.

### 3.3 Obstacle avoidance tests performed in the air

After examining whether the drone can detect its distance from the obstacle, generate the necessary commands to avoidance from the obstacle and whether interference occurs due to different reasons in the parameters that should not produce values, flight tests were started with the drone in the field. In these tests, the speed of the drone, its maneuvers, pitch, roll, and throttle values were examined by using the log records provided by Mission Planner. In this way, it was investigated to what extent and up to what speed the obstacle avoidance process took place.

In the first of the tests carried out within the scope of this research, an obstacle was brought closer to the drone that horizontal and vertical speed is zero (suspended in the air). The speed-time and pitch angle-time graphs taken from the log records for the 5-second time interval of this test are shown on the same graph in Figure 27. The graph seen in Figure 27 in red is the speed-time graph, and the one seen in green is the pitch-time graph. When the obstacle was brought closer to the drone suspended in the air, the avoidance took place at a time of about 100 ms with the detection of the obstacle. During this time, the drone's speed increased from 0 m/s to 3.5 m/s (the point shown by 1 in the graph) and moved away from the obstacle. When it reached the allowable distance from the obstacle, its speed decreased to 0 m/s again. Immediately afterward, the obstacle was brought to a different distance and a test was conducted again. In the test, the drone speed increased from 0 m/s to 6 m/s (point 2 on the graph) in a time interval of about 100 ms and moved away from the obstacle. When the allowed obstacle distance was reached, the drone's speed decreased back to 0 m/s.

When the pitch-time graph in Figure 27 is examined, it is seen that the pitch angle has reached from 0 degrees to -15.88 degrees when the drone starts to perform the maneuver away from the obstacle, at the first moment when the obstacle is detected (point shown with 3 in the graph). When the drone reached the safe distance, it first realized a pitch angle of about +12 degrees (the point shown by 4 on the graph) in order to hang in the air again, and after this oscillation, the angle value decreased to 0 degrees again. The change of pitch values in the second and third obstacle tests conducted during this process was similar.



Fig. 27. Changes in the speed-time (red) and pitch angle-time (green) graphs of a drone suspended in the air, in the case of an obstacle approaching from the front, over a certain time interval.

The graphs obtained when an object is brought close from the side of the drone suspended in the air are shared in Figure 28. The green graph in Figure 28 shows the speed-time graph, and the red one shows the roll angle-time graph. Immediately after the moment of 20:17:10.000 on the graph (at the point indicated by 1 on the graph), the drone's speed reached from 0 m/s to 1.3 m/s (point indicated by 2 on the graph) in a very short time with the detection of the obstacle by the drone. The drone, which had a zero-degree roll angle until that moment, tried to avoid the obstacle by reaching a roll angle of -45.57 degrees (the point shown by 3 on the graph). When the obstacle remained away from the set distance, the drone stopped its roll motion, brought the roll angle to zero degrees (the point shown by 4 on the graph), and continued to suspend in the air. Therefore, at the moment when the roll movement ended, the speed of the drone became 0 m/s again (the point shown by 5 on the graph).

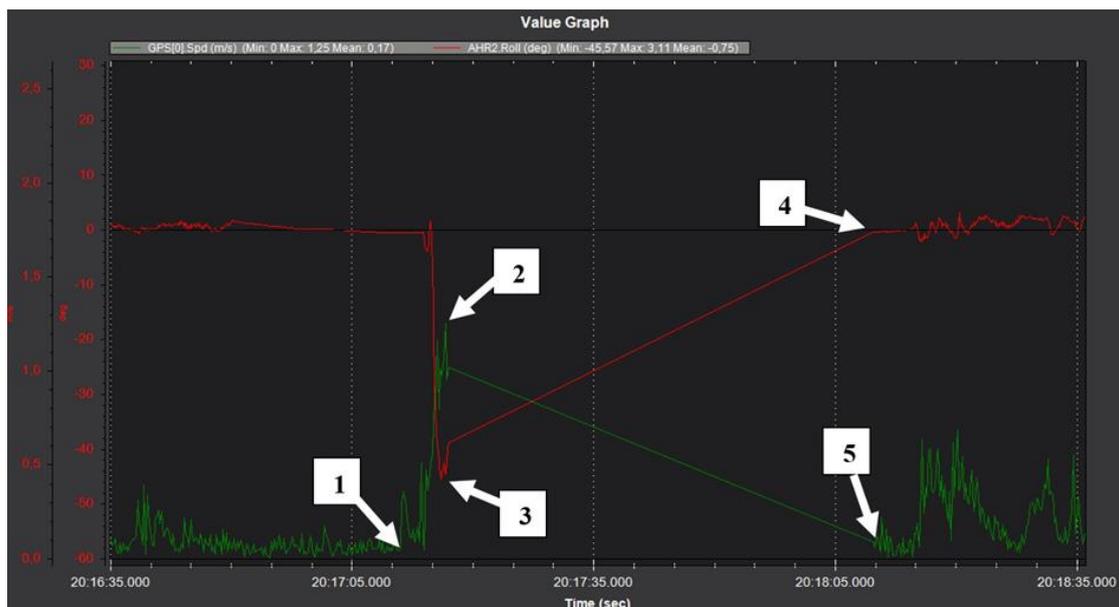
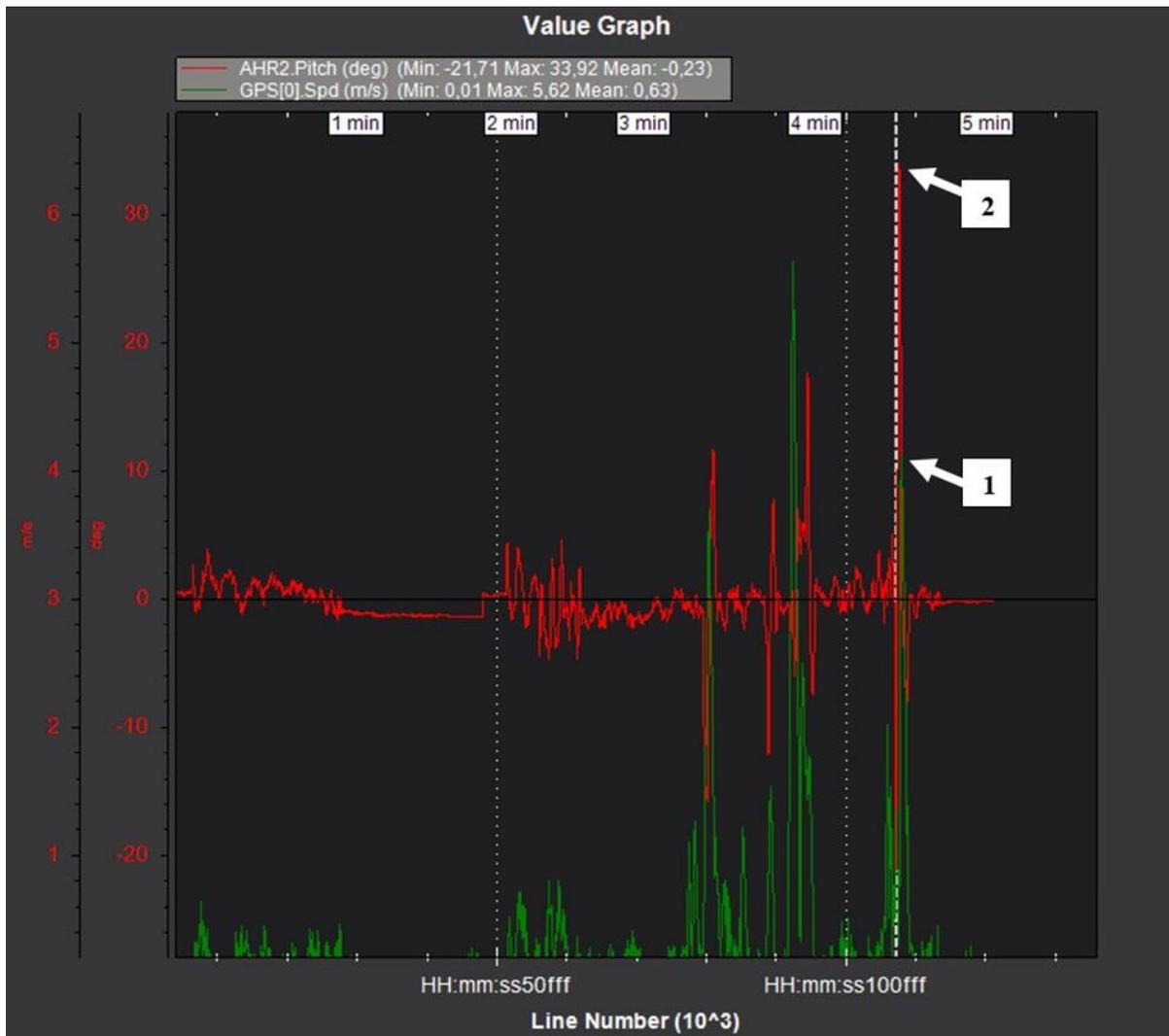


Fig. 28. Changes in the speed-time (green) and roll angle-time (red) graphs of a drone suspended in the air, in the case of an obstacle approaching from the side, over a certain time interval.



After examining the condition of approaching the obstacle to the drone hanging in the air, the behavior of the drone, which is moved from the front to the obstacle in the air at a low speed, was examined. The time-dependent change graphs of the pitch and speed values obtained from one of the tests performed within the scope of this review are presented in Figure 29. In Figure 29, the time-dependent change of the pitch angle is shown in red color, and the speed change is shown in green color graph. In Figure 30, GPS records of the drone's location and flight directions during this test are given.



**Fig. 29.** Graphs obtained when the drone flying through the air encounters an obstacle from the front (the green graph is the speed-time graph, and the red graph is the pitch angle-time graph).

As can be seen in the upper part of the graph in Figure 29, the average speed of the drone during this test is 0.63 m/s. Therefore, when the drone advancing at a speed of fewer than 1 m/s saw the obstacle, it reached a speed of 4.2 m/s (point 1 on the graph) in the opposite direction and escaped from the obstacle by making a pitch angle of 33.92 degrees (point 2 on the graph). The position at which it escaped from the obstacle and the change in the direction of the drone are indicated by the location symbol in the GPS records in Figure 30.



Fig. 30. GPS records of the location and flight directions of the drone flying through the air, including the test process in Figure 29.

The GPS records for a certain time period of another flight test conducted towards the obstacle are shared in Figure 31. The drone was flown at a higher speed in this test compared to the previous one. It is also understood from the points highlighted by 2 and 3 in the graphic section that the drone's speed at the position where it first saw the obstacle (highlighted with 1 in the figure) was 3.94 m/s. The angle of the pitch movement (27.57 degrees) made by the drone in the opposite direction in order to avoid the obstacle is also seen at the points marked by 4 and 5 in the graphic section of Figure 31. A detailed image showing that this maximum value of the pitch angle was reached at the time stamp 18:19:45.986 is given in Figure 32.

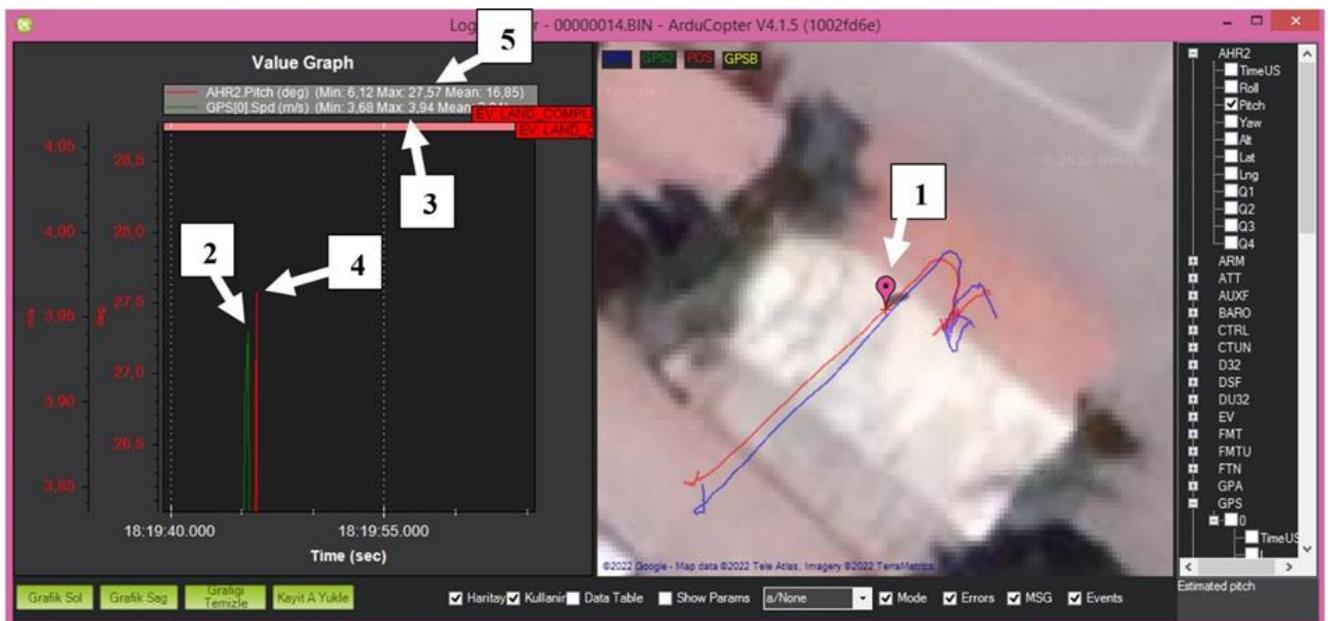


Fig. 31. The log records of the drone avoiding the obstacle at a speed of 3.94 m/s.

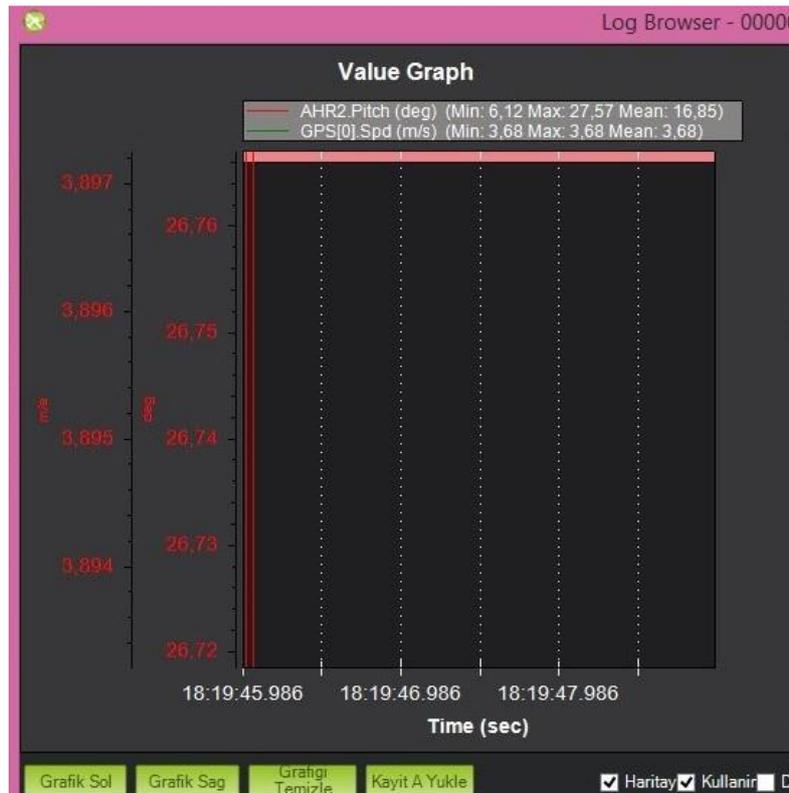


Fig. 32. Log recording that the drone made the pitch angle of 27.57 degrees at 18:19.45.986 instant.

#### 4. Conclusion and Discussion

The aim of this experimental study is to make an improvement that will prevent drones from hitting and crashing into obstacles while flying at speeds above 3 m/s, regardless of the reason. In order to achieve this goal, phased experiments were carried out on the ground and in the air with a drone designed with an ultrasonic sensor.

In the first stage, the drone was fixed to the ground. Then, the tests were started by placing objects at different distances on different sides of the drone. In these tests, when the motors are not running and running at high speed, it has been investigated whether the object's distance is measured correctly and whether the ultrasonic sensors with no obstacles in front of them produce interference. In this context:

- In the measurements made when the motors are not working, it has been observed that ultrasonic sensors generally measure the distance of the object precisely, but rarely, there are measurement errors ( $\pm 1.61\%$  and  $1.8\%$ ) below 2%.
- In the measurements made when the motors are working at high speed, it has been seen that the ultrasonic sensors do not produce any interference caused by the motors working at high speed, and the measured distance values coincide with the actual distance values.
- In the tests carried out when the motors are not working and working at high speed, it has been observed that all sensors that have no obstacles in front of them do not produce interference, so any distance value other than zero is not read.

In the next stage, the changes in the pitch, roll, and throttle values that the drone will produce when it detects obstacles were examined depending on the direction in which the obstacle was detected. In tests with the drone fixed to the ground and the motors not working:



- It has been observed that the roll value, which should not be produced any value because only a frontal obstacle is detected, is realized as expected (that is, no interference value is produced and the roll value remains zero), but pitch and throttle values are present.
- It has been observed that the pitch and throttle values, which should not be produced any value because only an obstacle is detected from the right, are realized as expected (that is, no parasitic value is produced and both values remain zero), but the roll value is present.
- It has been observed that the roll value, which should not be produced any value because the obstacle is detected only from the rear, is realized as expected (that is, no interference value is produced and the roll value remains zero), but the pitch and throttle values are present.
- It has been observed that the pitch and throttle values, which should not be generated at all because an obstacle is detected only from the left, are realized as expected (that is, no parasitic value is generated and both values remain zero), but the roll value, which should never be zero, is zero several times. It is envisaged that this happens due to a momentary interruption of communication between Arduino and Pixhawk.

In tests with the drone fixed to the ground again but the motors working at high speed, it has been seen that the pitch, throttle, and roll values generally take the values that should be. However, it has been observed that some values that should not be zero are zero several times due to the momentary interruption of communication between Arduino and Pixhawk.

In the final stage of the tests, flight tests were carried out with the drone in the field. In these tests, the speed of the drone, its maneuvers, pitch, roll, and throttle values were examined using the log records provided by Mission Planer, and it was investigated to what extent and up to which speed the obstacle avoidance process took place. In these three-stage tests:

- Firstly, the obstacle has been brought closer to the drone suspended in the air from different distances and from different sides. It has been observed that the speed of the drone reaches up to 6 m/s depending on the distance of the obstacle, and the pitch and roll angles made depending on the direction of the obstacle reach -15.88 and -45.57 degrees, respectively.
- In the second stage, the behavior of the drone, which was advanced towards the obstacle in the air at a low (average 0.63 m/s) speed and approaching the obstacle from the front side, was examined. In this test, the drone successfully evaded the obstacle by reaching a pitch angle of 33.92 degrees and a speed of 4.2 m/s.
- At the last stage, the behavior of the drone, which was advanced towards the obstacle in the air at a higher speed, was examined. At the end of this examination, it was seen that the drone, which had a speed of 3.94 m/s when it sensed the obstacle, made a 27.57 degree pitch movement and reduced its speed to zero, and safely avoided the obstacle by moving in the opposite direction without hitting the obstacle. Therefore, obstacle avoidance was successfully performed at a speed at least 31.3% higher than the speed ( $< 3$  m/s) obtained in the study shared in [29].

Since the sensors used are fixed to the drone, the pitch and roll angles that the drone makes when it sees the obstacle cause the other sensors to distort their viewpoints, that is, to look toward the ground. This makes it difficult to detect a new and dynamic obstacle that will appear in the escape direction during the escape from the obstacle. Therefore, a new mounting method can be designed that will ensure that the sensors are always positioned parallel to the ground so that the drone can better detect obstacles in the direction of movement, both while avoiding the obstacle and moving at high pitch and roll angles.

The obstacle avoidance system, designed using 4 ultrasonic sensors, can also be realized by using a sensor and a servo mechanism that will ensure that the sensor always faces the direction of the drone's movement.



There are communication delays in the commands to disable the control sent via MAVLink and to make the movement required by the obstacle. For this reason, the good adjustment of the parameter called HeartbeatTime between Arduino Uno and Pixhawk is a piece of useful information for other researchers who will work in this field.

### Acknowledgment

This study was supported by Selçuk University Scientific Research Projects Coordination Office with project number 22201010.

### References

- [1] Ayaz, M. Y., 2021."Dron için RF karıştırıcı tasarımı ve gerçekleştirilmesi," Yüksek Lisans, Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Konya, Türkiye.
- [2] Özdemir, Ş., 2013."İnsansız hava araçlarında kullanılan fırçasız DC motorların kontrolü," Yüksek Lisans, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [3] Sivritaş, Ş., 2020."Fast 3D Tracking Algorithm Implementations On Mini Quard-copters," Yüksek Lisans, Özyeğin Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [4] Çakıcı, E., 2019."Mini insansız hava araçları için bir fırçasız motor test sistemi geliştirilmesi," Yüksek Lisans, ESOGÜ, Fen Bilimleri Enstitüsü, Eskişehir, Türkiye.
- [5] Al-Tekreeti, M. M. A., 2019."Unmanned surface vehicle (USV) with obstacle avoidance system," Yüksek Lisans, Çankaya Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [6] Çalış, M., 2019."Ultrasonik mesafe ölçümü ve doğrulanması," Yüksek Lisans, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [7] Kale, A. A. 2021. Drone Technology and Its Applications, SSRN, doi: <http://dx.doi.org/10.2139/ssrn.3922787>
- [8] Venkatesh, S., Kundan, S., and Srija, A. 2021. Obsatacle Avoidance Robotic Vehicle Using Hc-Sr04 Ultrasonic Sensor, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 2358-2362.
- [9] Tayebi, A. and McGilvray, S. 2006. Attitude stabilization of a VTOL quadrotor aircraft, IEEE Transactions on control systems technology, 14(3), 562-571.
- [10] Abd Rahman, Y. A., Hajibeigy, M. T., Al-Obaidi, A. S. M., and Cheah, K. H. 2018. Design and fabrication of small vertical-take-off-landing unmanned aerial vehicle. MATEC Web of Conferences, 152(2018), 26 February, Selangor, Malaysia.
- [11] Wubben, J., Fabra, F., Calafate, C. T., Krzeszowski, T., Marquez-Barja, J. M., Cano, J.-C., and Manzoni, P. 2019. Accurate landing of unmanned aerial vehicles using ground pattern recognition, Electronics, 8(12), 1532.
- [12] Ekmen, M. İ., 2021."Drone sürüşü ile hedef takip optimizasyonu," Yüksek Lisans, Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Konya, Türkiye.
- [13] Yiğit, A., 2020."Adaptive Large Neighbourhood Search Heuristic On Vehicle Routing Problem With Drones And Time Windows," Yüksek Lisans, Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [14] Önler, E., 2012."Ultrasonik sensör yardımıyla belirlenen yaprak yoğunluğuna göre püskürtme miktarını ayarlayan sistemin tasarımı (akıllı ilaçlama makinası)," Yüksek Lisans, Namık Kemal Üniversitesi, Fen Bilimleri Enstitüsü, Tekirdağ, Türkiye.



- [15] Selim, E., Uyar, E., and Alcı, M. 2013. Quadrocopterin matematiksel modeli ve kontrolü, TOK2013-Otomatik Kontrol Ulusal Toplantısı, 548-553.
- [16] Köz, S., 2019."Yangın Topu Kullanılarak Yangın Söndüren Quadrocopter Tasarımı Ve Prototip İmalatı," Yüksek Lisans, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne, Türkiye.
- [17] Kim, J., Choi, Y., Jeon, S., Kang, J., and Cha, H. 2020. Optrone: Maximizing performance and energy resources of drone batteries, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11), 3931-3943.
- [18] Abdilla, A., Richards, A., and Burrow, S. 2015. Power and endurance modelling of battery-powered rotorcraft. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE, 28 September - 02 October, Hamburg, Germany, 675-680.
- [19] Maekawa, K., Negoro, S., Taniguchi, I., and Tomiyama, H. 2017. Power measurement and modeling of quadcopters on horizontal flight. *Fifth International Symposium on Computing and Networking (CANDAR)*: IEEE, 326-329.
- [20] Chen, Y., Baek, D., Bocca, A., Macii, A., Macii, E., and Poncino, M. 2018. A case for a battery-aware model of drone energy consumption. *IEEE International Telecommunications Energy Conference (INTELEC)*: IEEE, 07-11 October, Turino, Italy, 1-8.
- [21] He, L. and Pace, J. 2020. Estimating Altitude of Drones Using Batteries. *IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*: IEEE, 21-24 April, Sydney, NSW, Australia, 264-265.
- [22] Park, S., Zhang, L., and Chakraborty, S. 2017. Battery assignment and scheduling for drone delivery businesses. *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*: IEEE, 1-6.
- [23] Qaisar, S. M. and AbdelGawad, A. E. E. 2021. Prediction of the Li-Ion Battery Capacity by Using Event-Driven Acquisition and Machine Learning. *7th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*: IEEE, 22-25 June, Krakow, Poland, 1-6.
- [24] Kanarachos, S. A. 2009. A new method for computing optimal obstacle avoidance steering manoeuvres of vehicles, *International Journal of Vehicle Autonomous Systems*, 7(1-2), 73-95.
- [25] Zohaib, M., Pasha, M., Riaz, R., Javaid, N., Ilahi, M., and Khan, R. 2013. Control strategies for mobile robot with obstacle avoidance, *Journal of Basic and Applied Scientific Research*, 3(4), 1027-1036.
- [26] Gökçe, Ş., 2019."Dronlar için otomatik rota tayini ve takibi," Yüksek Lisans, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [27] Janardhan, K. S., Baba, S. S., Joe, P. A., and Babu, A. V. 2019. Design and fabrication of UAV for defence applications, *J. Mech. Contin. Math. Sci*, 14(6).
- [28] Suherman, S., Putra, R. A., and Pinem, M. 2020. Ultrasonic Sensor Assessment for Obstacle Avoidance in Quadcopter-based Drone System. *3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*: IEEE, 25-27 June, Medan, Indonesia, 50-53.
- [29] Rambabu, R., Bahiki, M. R., and Ali, S. A. M. 2015. Relative position-based collision avoidance system for swarming uavs using multi-sensor fusion, *ARPN J. Eng. Appl. Sci*, 10(21), 10012-10017.
- [30] ArduPilot, MAVLink Basics, 2022. <https://ardupilot.org/dev/docs/mavlink-basics.html> (15 June 2022).
- [31] Graff, K. F., 1981. A history of ultrasonics. *Physical acoustics*, 15, 1-97.
- [32] Morgan, E. J. 2014. HC-SR04 Ultrasonic Sensor [Online] Available: <https://datasheetspdf.com/pdf-file/1380136/ETC/HC-SR04/1>