MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
FACULTY OF AIR NAVIGATION, ELECTRONICS AND TELECOMMUNICATIONS
DEPARTMENT OF AVIONICS

APPROVED
Head of department
_____Yu.V.Hryshchenko
‘___’_____2023

# GRADUATION WORK

**(EXPLANATORY NOTES)**

**FOR THE DEGREE OF MASTER**

**SPECIALITY 173 ‘AVIONICS’**

## Theme: ‘ Altitude Control System for Remotely Piloted Aircraft Systems ’

Done by: _____AV-254m ,Y.Yu. Yakovlev_____
(student, group, surname, name, patronymic)

Supervisor: _____PhD, As. Prof., O.V. Kozhokhina_____
(scientific degree, academic rank, surname, name, patronymic)

Consultant on ‘Labor protection’: _____ _____
(signature)(surname, name, patronymic)

Consultant on ‘Environmental protection’: _____ _____
(signature) (surname, name, patronymic)

Standard controller: _____ V.V. Levkivskyi
(signature) (surname, name, patronymic)

Kyiv 2023

МІНІСТЕРСТВО ОСВІТИ І АУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА АВІОНІКИ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____Ю.В. Грищенко
«___»_____2023

# КВАЛІФІКАЦІЙНА РОБОТА

**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

ЗА СПЕЦІАЛЬНІСТЮ 173 «АВІОНІКА»

## Тема: «Система контролю висоти для дистанційно пілотованих авіаційних систем»

Виконав:_____АВ-254М,Я.Ю. Яковлев_____
(студент, група, прізвище, ім'я, по-батьвові)

Керівник: _____к.т.н, доц., О.В. Кожохіна_____
(наукова ступінь, вчене звання, прізвище, ім'я, по-батькові)

Консультант розділу «Охорона праці»: _____ _____
(підпис) (прізвище, ім'я, по-батькові)

Консультант розділу «Охорона навколишнього середовища»: _____ _____
(підпис) (прізвище, ім'я, по-батькові)

Нормоконтролер: _____ В.В. Левківський
(підпис) (прізвище, ім'я, по-батькові)

Київ 2023

NATIONAL AVIATION UNIVERSITY

Faculty of Air Navigation, Electronics and Telecommunications

Department of avionics

Specialty 173 ‘Avionics’

APPROVED

Head of department
_____ Yu.V.Hryshchenko
‘___’_____2023

**TASK**

**for executiongraduation work**

Y.Yu. Yakovlev

1. Theme: ‘Altitude Control System for Remotely Piloted Aircraft Systems’, approved by order №2040/ст of the Rector of the National Aviation University of 5Octoder 2023.

2. Duration of which is from 05 September 2022 to 30 November 2022.

3. Input data of graduation work: Different methods of determining flight atitude for different types of Remotely Piloted Aircraft Systems by analyzing open sources. Master's theses of other students with similar topics. Own experience in the construction and design of simplified Altitude Control System, which was acquired during the design and construction of our own demonstration model of the Altitude Control System. Scientific articles and books on the topic of digital filters and automatic control systems.

4. Content of explanatory notes: List of conditional terms and abbreviations, Introduction, Chapter 1, Chapter 2, Chapter 3, Chapter 4, Chapter 5, Chapter 6, Chapter 7, Chapter 8, Chapter 9, Conclusions, References,Appendix A, Appendix B.

5. The list of mandatory graphic material: figures, tables, graphs.

6. Planned schedule

| № | Task | Duration | Signature of supervisor |
|---|---|---|---|
| 1. | Validate the rationale of graduation work theme | 02.10-03.10 | |
| 2. | Carry out a literature review | 04.10-15.10 | |
| 3. | Develop the 1 and 2 chapter of graduation work | 16.10-20.10 | |
| 4. | Develop the 2-5 chapter of graduation work | 21.10-10.11 | |
| 5. | Develop the 5-7chapter of graduation work | 11.11-28.12 | |
| 6. | Develop the 8 and 9 chapter of graduation work | 29.12-06.10 | |
| 7. | Tested for anti-plagiarism and obtaining a review of the graduation work | 10.12 | |

7. Consultants individual chapters

| Chapter | Consultant (Position, surname, name, patronymic) | Date, signature | |
|---|---|---|---|
| | | Task issued | Task accepted |
| Labor protection | | | |
| Environmental protection | | | |

8. Date of assignment: '___ ' _____ 2022

Supervisor _____ O.V. Kozhokhina

The task took to perform _____ Y.Yu. Yakovlev

(signature) (surname, name, patronymic)

## ABSTRACT

Explanatory notes to graduation work ‘Altitude Control System for Remotely Piloted Aircraft Systems’ contained 90 pages, 60 figures, 2 tables, 44 references.

**Keywords:** ALTITUDE CONTROL SYSTEM, REMOTELY PILOTED AIRCRAFT SYSTEMS, SENSORS, DIGITAL FILTERS, AUTOPILOT, CONTROLLER, ERROR, AUTONOMOUS FLIGHT CONTROL SYSTEMS

**Theobject of the research** - process of functioning of altitude control system for remotely piloted aircraft systems’.

**The subject of the research**- altitude control mechanisms and systems within remotely piloted aircraft systems (RPAS).

**Purpose of graduation work -** is to analyze different methods of determining flight altitude for different types of remotely piloted aircraft systems and to investigate means of stabilizing the altitude control system.

**Research Method**– literature review,comparative analysis, experimental modeling, statistics theory, information theory.

**Scientific novelty** – recommendations on the selection of different sensors for height measurement for different types of remotely piloted aircraft systems; research on the change of errors when passing data through a Kalman filter from two different sensors. This research could potentially contribute to the development of more efficient, reliable, and safer RPAS.

**CONTENTS**

## LIST OF ABBREVIATIONS

| | |
|---|---|
| AC | Alternating Current |
| ARNS | Aeronautical Radio Navigation Service |
| BTT | Basic Training Target |
| CAA | Civil Aviation Authority |
| CPU | Central Processing Unit |
| DC | Direct Current |
| EASA | European Aviation Safety Agency |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EKF | Extended Kalman Filter |
| FMCW | Frequency Modulation Continuous Wave |
| GDOP | Geometric Dilution Of Precision |
| GNSS | Global Navigation Satellite System |
| GPS | Global Position System |
| $I^2C$ | Inter-Integrated Circuit |
| ICAO | Civil Aviation Organization |
| ICSP | In-Circuit Serial Programming |
| IIR | Infinite Impulse Response |
| LCD | Liquid Crystal Display |
| LED | Light-Emitting Diode |
| LIDAR | Laser Imaging, Detection, And Ranging |
| LVDT | Linear Variable Differential Transformer |
| MAV | Micro Aerial Vehicle |
| MEMS | Micro-Electro-Mechanical Systems |
| MLE | Maximum Likelihood Estimation |
| PID | Proportional Integral Derivative |
| PV | Process Variable |
| RA | Radio Altimeter |
| RALACS | Radar Altimeter Low Altitude Control System |
| RPAS | Remotely Piloted Aircraft System |
| RPV | Remotely Piloted Vehicles |
| SCL | Serial Clock |
| SD | Surveillance Drone |
| SDA | Serial Data |
| SP | Set Point |
| UAV | Unmanned Aerial Vehicles |

# INTRODUCTION

**Actuality.**In the contemporary technological landscape, the ubiquitous deployment of remotely piloted aircraft systems (RPAS) has become integral across a multitude of sectors, encompassing the commercial, entertainment, and military domains. These systems, characterized by their diverse morphologies and specifications, are strategically engineered to execute specific functional roles. A crucial element within these systems, pivotal to optimal flight performance, is the altitude control system.

This system is essential for providing accurate altitude data to the operator. In its automated configuration, the altitude control system collaborates synergistically with other subsystems, such as the attitude control system, to maintain predetermined altitude levels. The system employs a variety of methodologies for altitude determination, including a Global Positioning System (GPS), an array of pressure sensors, and radio altimeters. The integration of these disparate methods is designed to enhance the precision of altitude measurements.

The altitude control system incorporates two fundamental components: controllers and filters. Filters are instrumental in augmenting the system's stability, effectively attenuating potential disturbances caused by errors and environmental noise. This aspect is of paramount importance in RPAS altitude control systems, where inaccuracies in low-altitude readings can lead to severe system failures. Controllers, as implied by their designation, regulate key operational parameters. The selection and meticulous calibration of these controllers is critical for ensuring system efficacy.

Given the relatively nascent stage of development of these complex systems, there is a persistent imperative for the enhancement of existing technologies and the development of novel systems in this domain. Therefore, the primary aim of this thesis is to conduct a thorough investigation into the diverse altitude control systems utilized in RPAS. The objective extends beyond theoretical exploration, encompassing the practical application of this knowledge in the construction of a working model of an altitude control system. This initiative is undertaken with the intention of contributing to the progressive evolution of RPAS technologies, ensuring their sustained relevance and efficacy in the rapidly advancing technological milieu.

**Purpose of the work**is to analyze different methods of determining flight altitude for different types of remotely piloted aircraft systems and to investigate means of stabilizing the altitude control system

**Theobject of the research** - process of functioning of altitude control system for remotely piloted aircraft systems'.

**The subject of the research** -altitude control mechanisms and systems within remotely piloted aircraft systems (RPAS).

**Research Method** – literature review,comparative analysis, experimental modeling, statistics theory, information theory.

**Scientific novelty** – recommendations on the selection of different sensors for height measurement for different types of remotely piloted aircraft systems; research on the change of errors when passing data through a Kalman filter from two different sensors. This research could potentially contribute to the development of more efficient, reliable, and safer RPAS.

# CHAPTER 1
# HISTORY AND CLASSIFICATION

## 1.1 Developmentof remotely piloted aircraft system with altitude control system

The first remotely piloted aircraft systems were developed in the late 20s. At the heart of these aircraft were ordinary airplanes of those years, which were converted for control via radio waves. There was no altitude control system on these remotely piloted aircraft systems. The operator had to determine the height visually without using additional means. This imposed restriction on the flight range, because the operator had to constantly monitor the remotely piloted aircraft systems in order not to lose visual control over it. They were mainly used by the military for close reconnaissance and to practice shooting as flying targets.

Aerial Target (Figure 1.1), a British radio-controlled aircraft, became one of the first remotely piloted aircraft systems. Its remote-control components were designed by Dr. Archibald Low. He was an English consulting engineer, research physicist and inventor, and author of more than 40 books. During tests in March 1917, it became the first drone(common name remotely piloted aircraft system)that fly under control.


Figure 1.1. Aerial Target

Fairy Queen and de Havilland Queen Bee can be attributed to remotely piloted aircraft systems (or drones as they are called) of a similar type. The Fairy Queen and de

Havilland Queen Bee (Figure1.2) target aircraft were introduced in the 1920s and 1930s respectively. Subsequent models were similarly named, such as the Airspeed Queen Wasp and the Miles Queen Martinet. For the first time, drones began to fly during the First World War. They were launched using a catapult and flown using radio-controlled technologies. The use of these aircraft was constantly expanding. Reconnaissance drones were also heavily used during the Vietnam War. Common missions include decoy actions during missile launches, live combat, and dropping leaflets during psychological warfare.


Figure 1.2. Winston Churchill waiting to see the launch of a de Havilland Queen Bee

Another interesting example of early drones is the B-17 Flying Fortress "Drone" (Figure1.3). He is notable for making a then-record flight of 2,600 miles from Hilo Naval Air Station, Hawaii, to Muroc Army Air Field, California. The flight, which was led by Brig. Gen. William L. Richardson, chief of the Army Air Force's Guided Missile Division, took 14 hours and 55 minutes before the first plane landed at Moorok. The drone was controlled from a nearby flying plane. Azon radio remote-control equipment was added, with two television cameras fitted in the cockpit to allow a view of both the ground and the main instrumentation panel to be transmitted back to an accompanying aircraft. The drones were a veteran of the Bikini atomic bomb tests, during which they were delivered by remote control into the clouds of a radioactive bomb. [1]

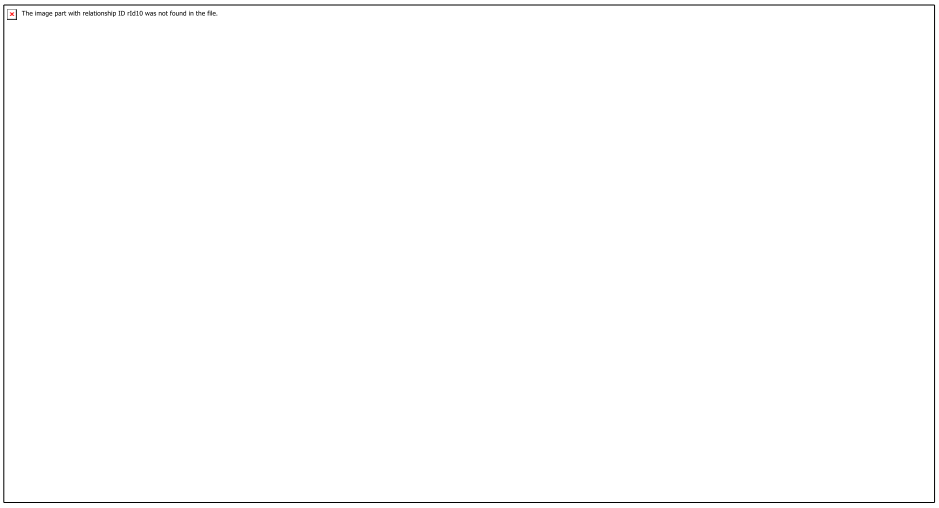



Figure 1.3 B-17 Flying Fortress "Drone"

All the above-mentioned remote-control aircraft were very limited functionally and could not fully reveal the potential of this class of aircraft. One of the first remote control devices that was really useful and did a good job was the MQM-57 Falconer drone (Figure 1.4).

It was built on the basis of the Radioplane BTT (Basic Training Target) drone, known as RP-71, produced by the Radioplane Company (later a division of Northrop). The RP-71 had built-in cameras (and additional television cameras) and was used by the US Army as the AN/USD-1 surveillance drone.


Figure 1.4 MQM-57 Falconer

The drone itself received the official designation OA-2343/USD-1, but was commonly referred to as either AN/USD-1 (although in reality this was the designation of the entire surveillance system of the drone, including ground equipment) or SD-1 (as an abbreviation of AN/USD -1, and SD is conveniently interpreted as "surveillance drone"). Serial production of the SD-1 for the US Army began in 1959.

The equipment that was installed on this remote-control aircraft consisted of: KS-54 or KS-61 camera system (including a KA-39A or KA-30 still picture camera), respectively flare ejectors, and an AN/DPN-32 (later AN/DPN-62(V)) radar beacon. The SD-1 was tracked on the ground using the AN/MPQ-29 radar system. AN/MPQ-29 was operating in X-Band and used the frequency-tunable magnetron type 2J51. The antenna used the conical scan mode for target tracking. This drone did not have an altitude control system, but it worked with the AN/MPQ-29 radar system, which allowed not only to adjust its position, but also to find out the altitude of the flight. However, the accuracy of this method was quite low. Due to this limitation and due to the fact that the drone could not provide images in real time, this piloted aircraft system could not perform its functions at low altitude due to the fact that it could collide with the ground[2,3].

One of the first drones to use a barometric altimeter based and radio altimeter-based altitude control system was the BQM-34 Firebee. (Figure1.5). This jet drone was used as a platform for testing various new systems for the remote-control system.


Figure 1.5 (BQM-34 Firebee)

Some examples were equipped with the RALACS (Radar Altimeter Low Altitude Control System). They were used to test these systems in action, because they were new. BQM-34s equipped with a TV system in the nose, were successfully flown by remote "pilots" watching the TV image in real time. Later, RALACS were installed on cruise missiles [4].

Commercial drones in their modern form for widespread use appeared not so long ago. But having absorbed all the experience in the development of military systems, these drones use a rather complex, but reliable altitude control system. It includes barometric sensors, ultrasonic sensors, a gps system and microprocessors for analyzing and comparing data from all sensors. A vivid example of such a system is the Parrot AR drone (Figure 1.6). [5]

Figure 1.6 Parrot AR drone

**1.2 Contribution of cruise missiles to development of the altitude control system**

The main contribution to the development of altitude control systems at the initial stages of the development of piloted aircraft systems were guided flying missiles, a forerunner of present-day cruise missiles. Unlike drones for aerial reconnaissance, these systems had to have a more advanced positioning system, and therefore an altitude control system, in order to accurately strike at specified coordinates.

The first aircraft of this type was the “Kettering Bug” (Figure1.7), which was manufactured by the Dayton-Wright Company.

Figure 1.7 Kettering Bug

Already in 1918, he had an inertial, automatic control system. After the start of the machine, the electric gyroscope ensured the stabilization of the "Beetle" in the air and directed it on a given course. The control system used a pneumatic/vacuum system, an electrical system and an aneroid barometer/altimeter. The flight height was previously set using a barometric altimeter.

Overall, Kettering's Beetle is interesting because it was the first unmanned aerial vehicle designed for military purposes that was at least partially successful. [6]

The first cruise missile in the modern sense of the word was the V-1 (Fau-1). The V-1(Figure 1.8) control system is an autopilot that keeps the projectile on the course and altitude specified at launch throughout the flight.

Figure 1.8. V-1 (Fau-1)

Stabilization in heading and pitch is carried out on the basis of the readings of a 3-degree (main) gyroscope, which are summed in pitch with the readings of a barometric altitude sensor, and in heading and pitch with the values of the corresponding angular velocities measured by two 2-degree gyroscopes (for damping oscillations of the projectile around its own center of mass). Targeting is carried out before launch using a magnetic compass, which is part of the control system. [7]

One of the first cruise missiles to receive a radio altimeter was the GAM-77A Hound Dog (or AGM-28 Hound Dog). The GAM-77A was a turbo-jet powered air-launched cruise missile deployed by the B-52(Figure1.9).

The rocket had four flight modes:

- High-altitude attack — the missile made the entire flight at an altitude of17,000 meters or more, and then dived on the target.
- Low-altitude attack - the missile flew at an altitude of about 1,500 meters. At the same time, the range was reduced to 640 kilometers (due to increased air resistance), but the missile was less visible to unpleasant radarsLow-altitude radar attack — the missile flew at an altitude of 900 to 30 meters above the surface, targeting the radar altimeter. The rocket could not bend the unevenness of the terrain, except by jumping over them with a vertical jump.
- Bow attack - the missile flew from the launch point to a pre-programmed position, then turned and headed for the target. Such a maneuver was supposed to put the enemy's weapons in jeopardy relative to the entire attack, and allow the missile to slip through. [8]


Figure 1.9 B-52 with two GAM-77A under the wing on pylons

### 1.3 Methodology and modern classification of remotely piloted aircraft systems

The International Civil Aviation Organization (ICAO) assigned the general name RPAS (Remotely Piloted Aircraft System) to remotely piloted aircraft. The definition given for this is systems “based on cutting-edge developments in aerospace technologies, offering advancements which are opening new and enhanced civil-commercial applications as well as improvements to the safety and efficiency of the entire civil aviation"[9]. This methodology is usually used by international aviation agencies: Eurocontrol, European Aviation Safety Agency (EASA), and the Civil Aviation Authority (CAA).

If we do not take into account by international aviation agencies, systems of this type are usually allocated and called Unmanned aerial vehicles (UAVs) and remotely piloted vehicles (RPVs). UAVs and RPVs may seem like the same thing, but there are some differences between them. It's important to know the difference for several reasons: first, if you're going to buy one of these vehicles, you'll want to make sure you're buying

one that fits your needs; and secondly, the rules for using them differ depending on the type of system you are using.

Unmanned aerial vehicles, or UAVs, are types of remotely piloted aircraft systems that are most associated with the name remotely piloted aircraft systems. UAVs are designed to fly autonomously using pre-programmed mission control software. They are usually used for photography, video recording and other entertainment purposes.

They usually have a wide array of sensors and cameras to perform tasks such as capturing aerial photographs or images. And while they've been around for a while, they continue to evolve and become more sophisticated every day.

The RPV must be constantly controlled by a person. It does not have an autopilot system. It has a limited set of sensors, which makes it cheaper than UAVs but reduces its functionality.[10]

One interesting subtype of UAVs is the MAV. A micro aerial vehicle (MAV) is a subclass of UAVs with decreasing size. Modern vehicles of this type can be flown in an autonomous way and, although new technologies promise to reach the insect-sized in the next future, today have a size of 5 cm.

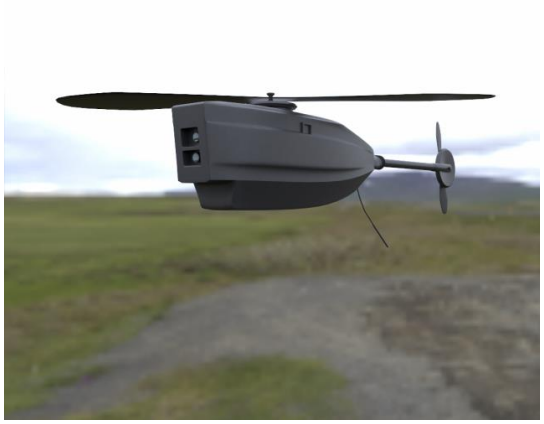


Figure 1.10 MAV “Black Hornet Nano”

Originally designed to support the military in combat zones, such as the Black Hornet Nano (Fig. 1.10), a 10 cm x 2.5 cm unmanned aerial vehicle helicopter designed for use in a combat zone, equipped with a camera that helps to quickly apply it and not be noticed due to its small size. [11]

Acquiring knowledge about the classification of remotely piloted aircraft systems is necessary within the scope of my work, because different types of remotely piloted

aircraft systems have an altitude control system of varying complexity or may not have it at all, which will allow me to better systematize and use information to build a working model.

# CHAPTER 2

# DIFERENT METHODS OF MEASHURE ALTITUDE

## 2.1. Barometric method

One of the most common methods of determining altitude is the barometric method. Diaphragm-based barometric sensors work according to this method. They were installed on the first airplanes, and therefore it is obvious that they were also used for the first altitude control systems for remotely piloted aircraft systems.

In addition to being used as a sensor for altitude measurement, diaphragm-based barometric sensors are used in various fields, in particular for measuring the pressure of liquids and gases, and have a similar structure. Diaphragms are made from round metal discs or flexible elements such as rubber, plastic or leather. Membranes are very sensitive to sudden changes in pressure. A membrane made of metal can measure a maximum pressure of approximately 7 MPa, and a membrane using an elastic type of material can measure extremely low pressures (0.1 kPa - 2.2 MPa) when connected to capacitive transducers or differential pressure sensors.The membranes are very versatile - they are usually used in very aggressive environments or in situations with extreme overpressures.

Having been installed in the vicinity of the altitude sensor, barometric sensors based on vicoristic diaphragms are used in various spheres, protecting the pressure of the environment and gases, and may resemble Budova.

The membranes are very versatile - they are usually used in very aggressive environments or in situations with extreme overpressures.

Changing the diaphragm size is converted into an electrical or pneumatic signal. Electrical sensors take mechanical input data from an elastic sensor and incorporate an electrical component, thereby increasing the sensitivity and applications of the sensors. The most common sensors of this type are capacitive, inductiveand piezoelectric sensors:

- Capacitive sensors

A capacitive sensor consists of parallel plates - capacitors - connected to a diaphragm, which is usually metal and is subjected to pressure from the forces involved in the process on one side and a reference pressure on the other side. The electrodes are attached to the membrane and are powered by a high frequency generator. The electrodes sense any movement of the diaphragm and this affects the change in the capacitance of the capacitor plates.

An example of a simplified design of a capacitive pressure sensor can be seen in Figure 2.1.It consists of insulated standoffs (A), diaphragm (B), capacitor plates (C), pressure port (D).

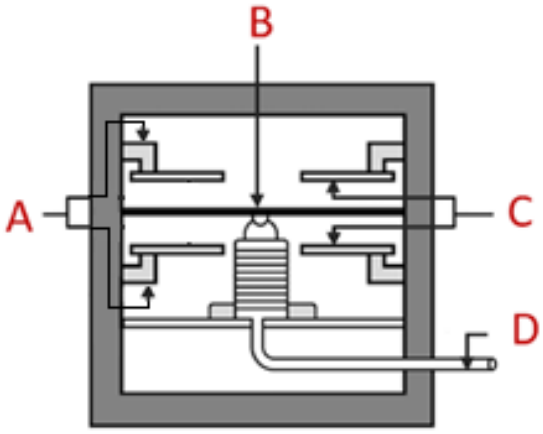



Figure 2.1. Construction of capacitive pressure sensor

- Inductive pressure sensor

Inductive pressure sensors in combination with a diaphragm or Bourdon tube. The ferromagnetic core is attached to the elastic element and has a primary and two secondary windings. Current is supplied to the primary winding. When the core is centered the same voltage will be induced to the two secondary windings. When the core moves under the influence of pressure, the voltage ratio between the two secondary windings changes. The voltage difference is proportional to the pressure change.

An advantage of the Pressure Inductive sensor type, as a Pressure capacitive sensor type, over the resistive type is that no moving contacts are present, thereby providing continuous resolution of the change, with no extra friction load imposed on the measuring system.

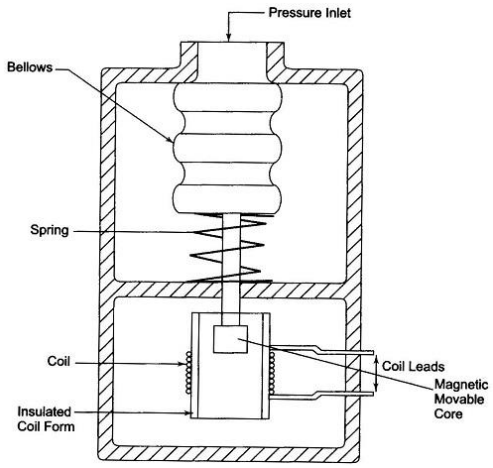



Figure 2.2. Construction of inductive pressure sensor

An example of a simplified design of an inductive pressure sensor can be seen in Figure 2.2.

- Piezoelectric sensors

Piezoelectric sensors use a crystal sensor. When pressure is applied to the crystal, it deforms and creates a small electrical charge. The measurement of electric charge is proportional to the change in pressure. This type of sensor has a very fast response time to constant pressure changes. Like a pressure sensor based on the principle of measuring magnetoresistance, the piezoelectric element is very sensitive, but responds much faster. Thus, if time is of the essence, the piezoelectric sensor will be a priority for use. The pressure range of this type of sensor is 0.021 - 100 MPa with a sensitivity of 0.1 MPa.[12]

An example of a simplified design of an inductive pressure sensor can be seen in Figure 2.3.

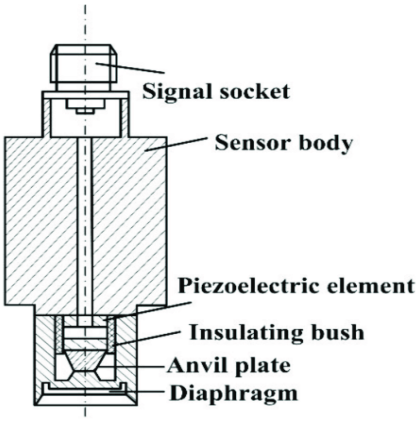



Figure 2.3. Construction of piezoelectric pressure sensors

The principle of operation of all barometric altimeter is based on the dependence of absolute pressure p from altitude H= $p=f_1(H)$ . You can see this dependence on Figure 2.3.

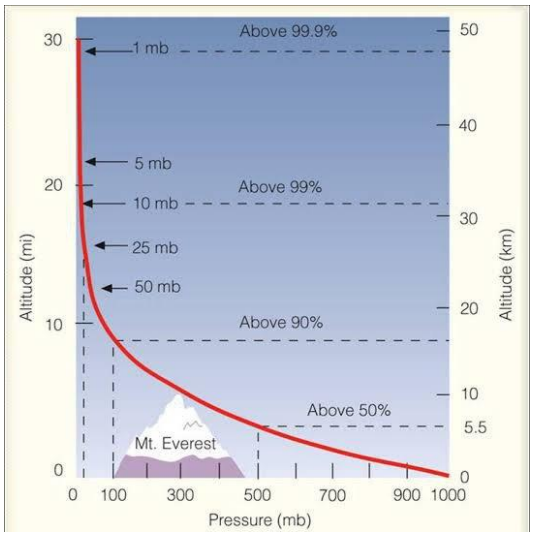



Figure 2.3. Dependence of atmospheric pressure on altitude

To determinecalibration formula, the dependences of density from altitude $\gamma=f_2(H)$ andabsolute temperature from altitude $T=f_3(H)$ are used too. These dependencies are presented in Figure 2.4. The dependences corresponding to standard atmospheric conditions and their variations are presented here. These dependencies are statistical, but not functional.

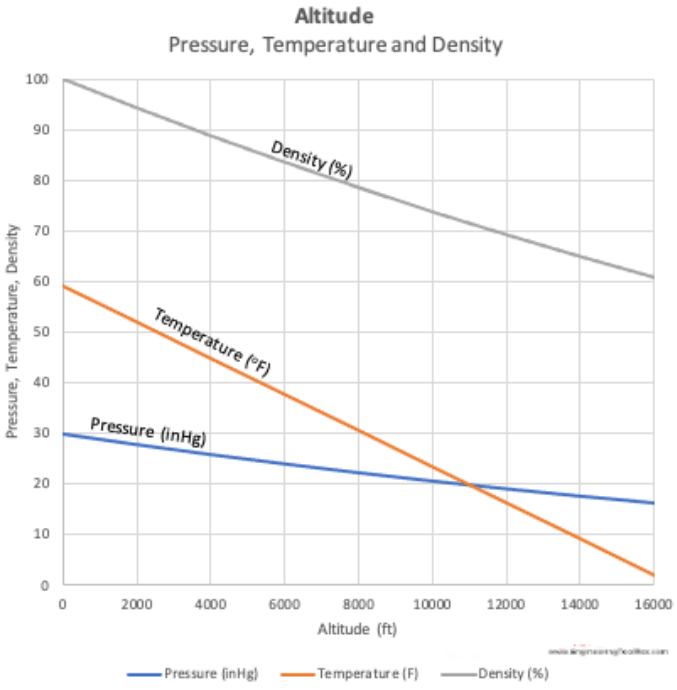



Figure 2.4. Dependence of pressure, temperature and air density on altitude

To obtain the dependence between atmosphere parameters and altitude H wewill consider a cylinder air column with square S at altitude H (Figure2.5).

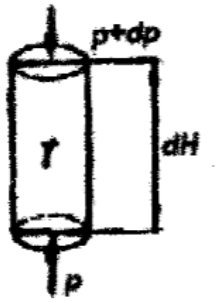



Figure 2.5. A cylinder air column

Taking into account balance of forces acting on column we will find

$$(p + dp)S + \gamma S dH = pS \qquad (2.1)$$

Using equation of state

$$\gamma = \frac{p}{RT}, \qquad (2.2)$$

where R is the gas constant, we will re-arrange the expression to the formula

$$\frac{dp}{p} = -\frac{dH}{RT}. \qquad (2.3)$$

It is known that the average atmosphere temperature up to the altitude 11 km is the linear function of the form

$$T = T_0 - \tau H, \qquad (2.4)$$

where $T_0$ is the average absolute temperature at the level sea equal to 288 $^oC$; $\tau$=6.5 degree/km – the temperature gradient. The temperature T at altitude 11 km is equal to 216.5 $^oK$(-56.5 $^oC$). The average temperature T is the same at the altitudes from 11 to 33 km. Having analyzed the information presented above, we will write a general expression for the average temperature, representing a mathematical expression

$$M[T]=\begin{cases}T_0-\tau H & \text{for } H\le 11\,km\\ T_{11} & \text{for } 11<H<33\,km\end{cases} \qquad (2.5)$$

Substituting two formulas and taking integral, we will obtain

$$P=P_o\left(1-\frac{\tau}{T_o}H\right)^{\frac{1}{R\tau}}, \qquad (2.6)$$

where 0p =760 mm of the mercury column.

This formula is called barometric formula.

Solving equation relative to H we will derive the so-called hypsometric formula

$$H=\frac{T_o}{\tau}\left[1-\left(\frac{p}{p_o}\right)^{R\tau}\right]. \qquad (2.7)$$

This is true up to altitude 11 km. For great altitudes we have another formula for pressure and altitude depends

$$p=p_{11}\exp\left(-\frac{H-H_{11}}{RT_{11}}\right) \qquad (2.8)$$

Solving this equation relative to H we will define the hypsometric formula for altitude H>11 km

$$H=H_{11}+\frac{RT_{11}}{m}\lg\frac{p_{11}}{p}, \qquad (2.9)$$

Where m=lg e=0,4343.[13]

These sensors have a number of disadvantages. Their main problem is that they can only show the True altitude of the flight, that is, the altitude above sea level. Therefore, systems that use atmospheric pressure to obtain the altitude value are often used in conjunction with other systems that are based on the radio method or ultrasound. Another problem is energy consumption. While many barometric pressure sensors are compact and lightweight, some applications may require even smaller sensors or specific form factors, which can be challenging to accommodate.

Over time, some barometric pressure sensors can experience drift, which can cause the sensor output to change even when the actual pressure remains constant. This can be caused by factors such as ageing, mechanical impact or temperature fluctuations. Choosing a sensor with good long-term stability can help mitigate this problem but not solve it.

Barometric pressure sensors can be sensitive to changes in temperature and humidity, which can affect their readings.

Although many barometric pressure sensors are compact and lightweight, some applications may require even smaller sensors or certain form factors that may be difficult to accommodate. As an example, you can take the SMC-630 Barometric Pressure Sensor. It has dimensions (W x D x H) 135 x 60 x 83mm and weight 1 kg. This is a great option for large to medium sized systems, but not for small ones like civilian copters.

Sensors of this type were used at relatively early stages of development of remotely piloted aircraft systems. Now their functions in altitude control systems are performed by mems pressure sensors, which are more technological and are a logical continuation of the use of atmospheric pressure to determine altitude.

However, now MEMS pressure is gaining more popularity for use in the construction of drones and small aircraft. Micro-Electro-Mechanical Systems (MEMS) are conventional mechanical systems on a small scale. They are often manufactured using silicon electronic chip technology, including nanometer etching and photolithography. These systems are not reduced mechanical systems, but change as linear dimensions increase.

The working principle of MEMS Pressure Sensors is identical to piezoelectric pressure sensors that we described above. Only the sizes differ. MEMS Pressure Sensors are smaller than conventional sensors, which allows them to be used in the smallest remotely piloted aircraft systems. Another advantage is the low price, which allows them to be used in civilian products. Thanks to the sensor's small size, low power and noise, high accuracy, and immunity to vibration, modern pressure sensors can detect an absolute altitude of <3 m, with a minimum detectable change in altitude of just a few cm. The disadvantages of these systems are also identical to those of atmospheric pressure sensors,

except for size. Sensors of this type are currently the most used barometric altitude sensors, so I chose this sensor as one of the two to build a working model of a modern commercial drone altitude control system.

Using the BMP180(Figure 2.6) as an example, we will consider this type of sensor in more detail

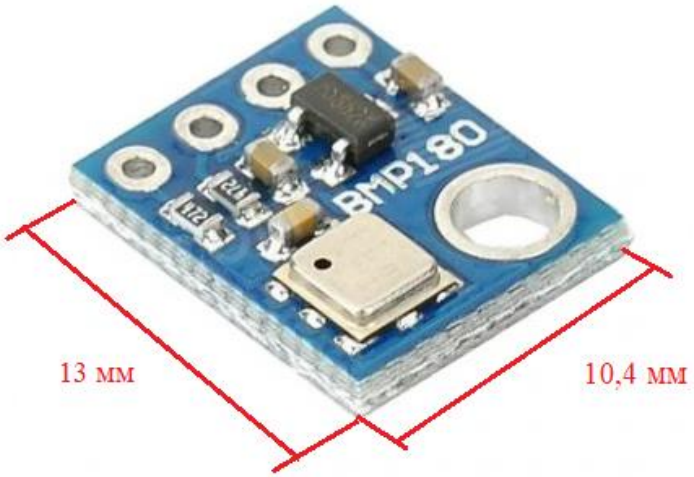



Figure 2.6. MEMS pressure sensor BMP180 with scales

The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with E2PROM and a serial I2C interface.

EEPROM (“Electrically Erasable Programmable Read-Only Memory’ also called E2PROM) stands for electrically erasable programmable read-only memory and is a type of non-volatile memory used in computers, usually integrated in microcontrollers such as smart cards and remote keyless systems, or as a separate chip device to store relatively small amounts of data by allowing individual bytes to be erased and reprogrammed.[16]

$I^2C$ (Inter-Integrated Circuit) alternatively known as I2C or IIC, is a synchronous, multi-master/multi-slave (controller/target), single-ended, serial communication bus invented in 1982 by Philips Semiconductors. It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication.[17]

The I2C interface protocol (Figure 2.7) has special bus signal conditions. Start (S), Stop (P) and binary data conditions are shown below. Atstart condition, SCL(Serial Clock) is high and SDA( Serial Data) has a falling edge. Then the slave address is sent.

After the 7 address bits, the direction control bit R/W selects the read or write operation. When a slave device recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL(ACK) cycle. At stop condition, SCL is also high, but SDA has a rising edge. Data must be held stable at SDA when SCL is high. Data can change value at SDA only when SCL is low. Even though VDDIO can be powered on before VDD, there is a chance of excessive power consumption (a few mA) if this sequence is used, and the stat of the output pins is undefined so that the bus can be locked. Therefore, VDD must be powered before VDDIO unless the limitations above are understood and not critical.

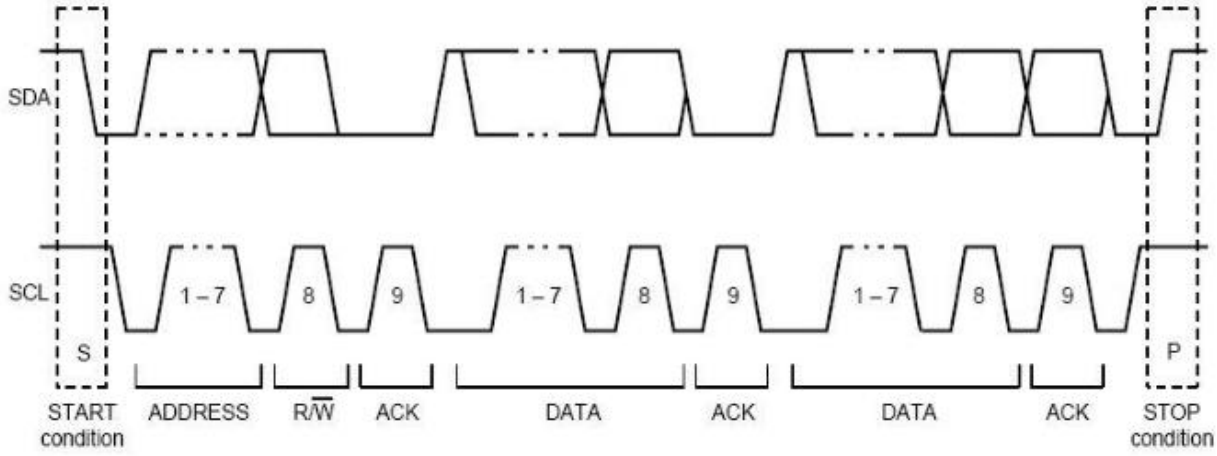



Figure 2.7. I2C protocol

The BMP180 delivers the uncompensated value of pressure and temperature. The E2PROM has stored 176 bit of individual calibration data. This is used to compensate offset, temperature dependence and other parameters of the sensor:

- UP = pressure data (16 to 19 bit)
- UT = temperature data (16 bit)

On the Fig 2.8 we can see typical application circuit of the BMP180

This sensor is combined, i.e. it can also measure temperature. The microcontroller sends a trigger signal to start measuring pressure or temperature. After the time conversion, the result (UP or UT respectively) can be read via the I2C interface. Calibration data must be used to calculate temperature in °C and pressure in hPa. These

constants can be read from the BMP180 E2PROM via the I2C interface during software initialization. [15]

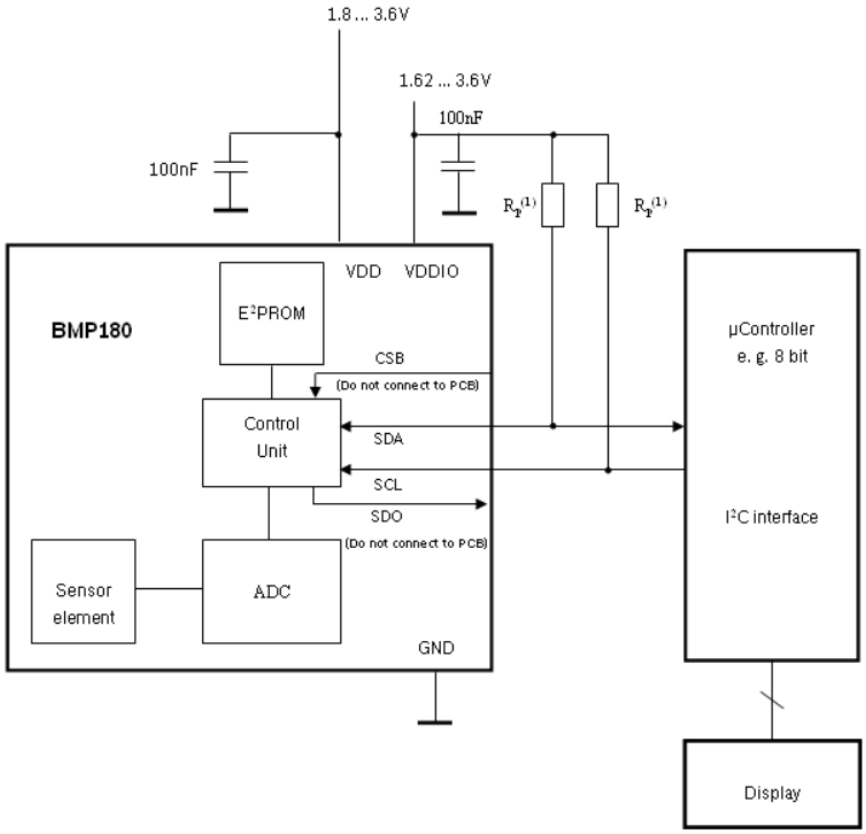



Figure 2.8. Typical application circuit

### 2.2 Ultrasound method

Another method of altitude determination is the use of ultrasound. This method is used by ultrasonic sensors. In 1914, Fessenden developed the first transducer, which is used in echo sounders, where it can find objects in the water, but not the direction of objects. The era of modern ultrasonics started about 1917, with Langevin's use of high-frequency acoustic waves and quartz resonators for submarine detection. [18]

There are basically two main elements: a transmitter and a receiver. Using piezoelectric crystals, the transmitter generates sound, and from there it travels to the target and returns to the receiver component.

To find out the distance between the target and the sensor, the sensor calculates the time it takes for sound to travel from the transmitter to the receiver. The calculation is performed as follows:

Sound travels at a speed of about 343 meters per second. This corresponds to approximately 29.412 µs (microseconds) per centimeter. To measure the distance traveled by sound, we use the formula[19]:

$$l = \frac{t * v_{sound}}{2} \quad (2.10)$$

Where l=distance , t= the time it takes for the sound to reach the obstacle , $v_{sound}$= speed of sound.

To get the final formula, let's substitute the speed of sound in the formula above:

$$l = \frac{t * 343\text{m}/s}{2} \quad (2.11)$$

For a more detailed introduction to this type of sensor, consider the structure and principle of operation of the HC-SR04( Fig 2.9), as it is the most common ultrasonic sensor on the market.

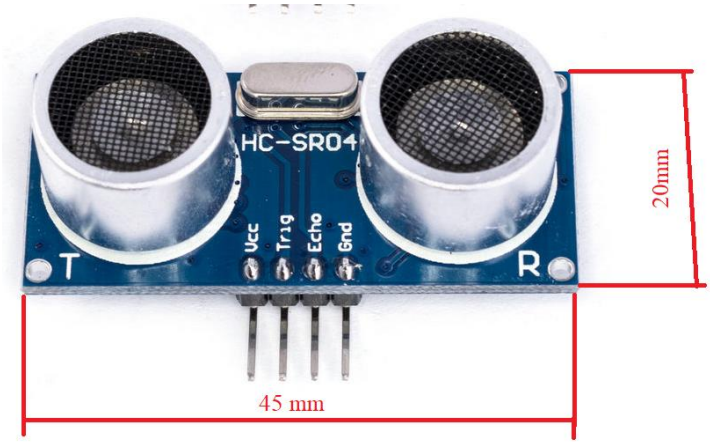



Figure 2.9. Ultrasonic Sensor with scales

On the sensor, the transmitter and receiver are designated as T and R, respectively. The rangefinder generates sound waves at a frequency of 40 kHz. The sound wave is reflected from the object and returns to the receiver, the sensor gives information about the time it took for the sound wave to propagate from the sensor to the object and back. This process is described in Figure 2.10.

The ultrasonic signal is propagated by a wave directed at an angle of 30°. The direction of propagation of the ultrasonic signal from the transmitter is shown in Figure

2.11. The most effective measuring angle is 15°. External objects falling under this measurement angle interfere with the determination of the distance to the desired object.

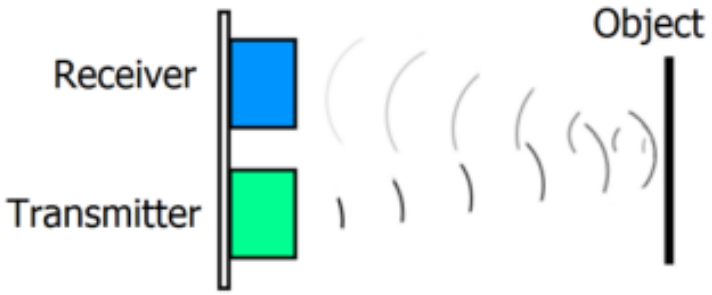



Figure 2.10. Movement of the ultrasonic signal from the transmitter to the receiver

An ultrasonic wave is reflected from almost any surface, even a transparent one, but it can be difficult to determine the distance to fluffy or small objects.

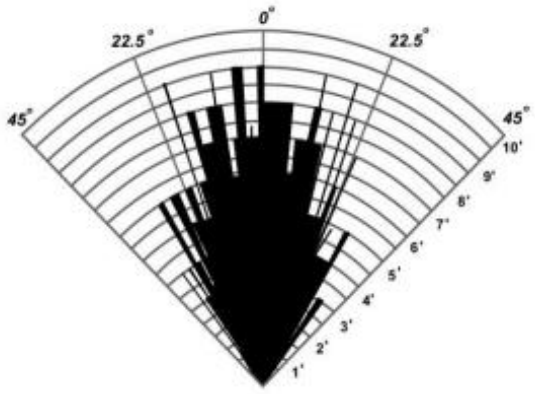



Figure 2.11

Another disadvantage of this sensor is that the reading is affected by the angle of incidence of the wave. If the sensor is directed perpendicular to the object, the measurements will be most accurate. In addition, if the angle of incidence is too large, then the wave reflected from the object does not reach the receiver at all, which will lead to a large discrepancy between the received data and the real distance(Fig2.12).

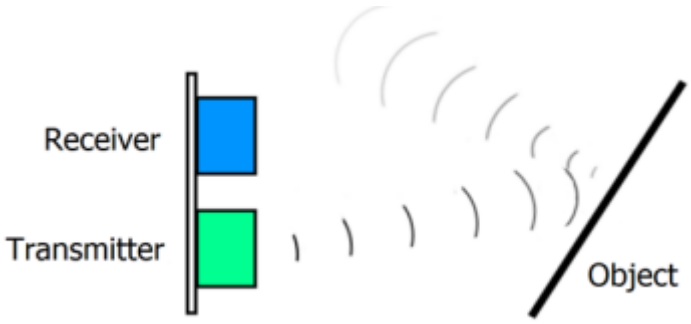



Figure 2.12. Movement of the ultrasonic wave from the transmitter to the receiver at an angle

HCSR04 has four pins: VCC, GND, TRIG and ECHO. The VCC and GND pins are for powering the HCSR04. There is a single control contact: TRIG. The TRIG pin is responsible for sending the ultrasonic signal. This pin must be set HIGH for 10 µs, after which the HCSR04 will send a tone packet of eight cycles at 40 kHz. After the beep is sent, the ECHO pin will go HIGH. The ECHO pin is the data pin used for distance measurement.

The HCSR04 can be triggered to send an ultrasonic burst by setting the TRIG pin to HIGH. After the packet is sent, the ECHO signal automatically goes HIGH. This pin will remain HIGH until the burst is latched by the sensor. You can calculate the distance to the object by monitoring how long the ECHO pin remains HIGH.

The time that ECHO remains HIGH is the time spent moving the packet. Briefly, this process can be described as follows:

1.Set TRIG to HIGH;

2.Set a timer when ECHO goes to HIGH;

3.Keep the timer running until ECHO goes to LOW;

4.Save that time;

5.Use equation to determine the distance travelled.

Visual representation can be seen in the Figure2.13 [20, 21]

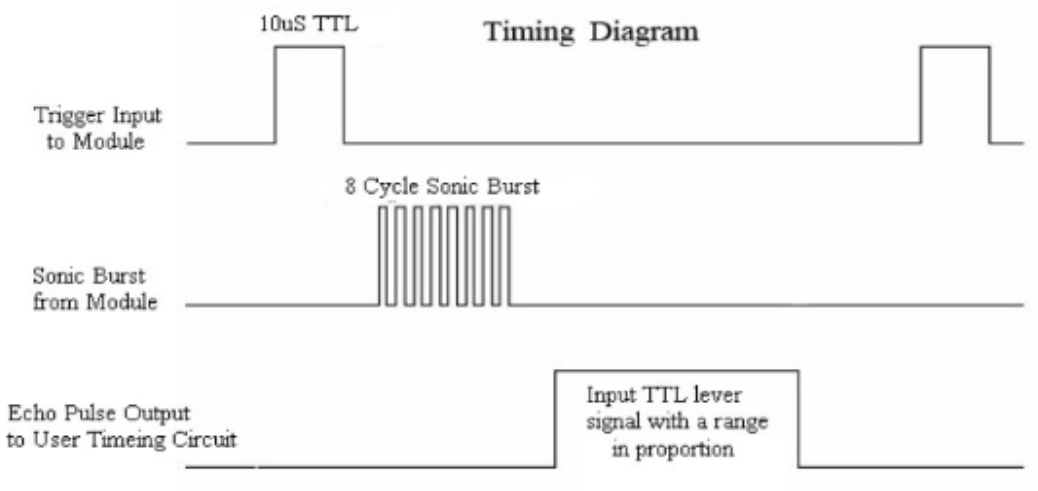



Figure 2.13 Visual representation of taking distance measurements by HC-SR04

The advantages of this type of sensor are its small size and weight, which allows it to be installed on small systems. Another plus is the low price, and therefore availability. All of the above advantages make Ultrasonic Sensors the most suitable for use in commercial civilian quadcopters. Another significant advantage is measurement accuracy. For example, the HC SR04 provides non-contact measurement with an accuracy of up to 3 mm. The disadvantages include the fact that this sensor can measure only the true altitude. Another disadvantage is that these sensors can take measurements at low altitudes. For example, HCSR04 can take measurements only at a altitude of no more than 4 meters. Due to the above disadvantages, Ultrasonic Sensors are used in conjunction with other types of sensors, such as MEMS Pressure Sensors or GPS module, and are typically used during the landing of remotely piloted aircraft systems or for safe low-altitude flight.

### 2.3 Radio method

One of the effective methods of altitude determination for remotely piloted aircraft systems is the method of radio wave reflection. This method uses radio altimeter. A radio altimeter (RA) is an airborne electronic device capable of measuring the altitude of the aircraft above terrain immediately below the aircraft.

Lloyd Espenshed (Fig 2.14) first created a radio altimeter in 1924. Espenshid was born in St. Louis, Missouri, and graduated from Pratt Institute with a degree in electrical engineering. He was interested in wireless and radio communications and worked for telephone and telegraph companies.

Figure 2.14. Lloyd Espenshed

Besides the altimeter, he was also the co-creator of the coaxial cable, an important component of television and long-distance telephone service. He held over 100 patents in communications technology. But only 14 years later, Bell Labs was able to adapt the radio altimeter of that time for use in airplanes, and later in remotely piloted aircraft systems. [22]

There are two types of radio altimeters in use today. One type utilizes Frequency Modulation Continuous Wave (FMCW) modulation, the second utilizes pulsed modulation. RAs with FM are designed to measure low altitudes from 0 to 1500 m in the systems of approach, automatic landing and over flight of obstacle sat low and extremely low altitudes. Modern pulsed RA sallow measuring almost the entire range of flight altitudes from 50 to 30,000 m and can be used for both pre-landing maneuver and for landing aircraft or remotely piloted aircraft system.

FMCW (or RA with FM) radio altimeters operate with Tx/Rx working in conjunction with separate transmit/receive antennas. They used a frequency method of distance measurement. RAs with FM are widely used in air navigation to measure low altitudes, so they are called low-altitude radio altimeters.Radio Altimeters operate in an

Aeronautical Radio Navigation Service (ARNS) spectrum allocation in the 4.2-4.4 GHz frequency band.

The principle of operation of RA with FM is as follows. The high-frequency continuous frequency-modulated signal generated by the RA transmitter is radiated towards the ground. Part of the high-frequency energy falling on the earth's surface is reflected from the ground, received by the receiving antenna and enters the receiver. Due to frequency modulation, the frequency of the direct signal will change and differ from the frequency of the reflected signal during the propagation of the signal to the ground and back. In the RA receiver, a signal is emitted, the frequency of which is the subtraction of the reflected frequency from the direct one (frequency difference signal). According to the given FM law, the frequency of this signal (frequency difference) is a function of altitude and can serve as its measure.

Let us consider as an example the case of FM according to the law of a symmetrical saw (Figure 2.15a). the sawtooth law of frequency modulation is characterized by the modulation period:

$$Tm = 1/Fm \qquad (2.12)$$

Where Fm – modulation frequency.

The reflected signal is changed relative to the direct one for some time, which characterizes the propagation of the signal we sent and the reflected signal

$$\tau = 2H/s \qquad (2.13)$$

Where H – thealtitude of the aircraft, s – the speed of radio waves.

The difference between the frequencies of the direct and reflected signals corresponds to the differential frequency signal Fr(Figure 2.15b). From the similarity of triangles, you can make a relationship

$$\frac{cF_r}{2H} = \frac{4\Delta f}{T_м} \qquad (2.14)$$

Then accordingly

$$H = \frac{c}{8\Delta f F_м} Fr \qquad (2.15)$$

Where Δf- frequency deviation;

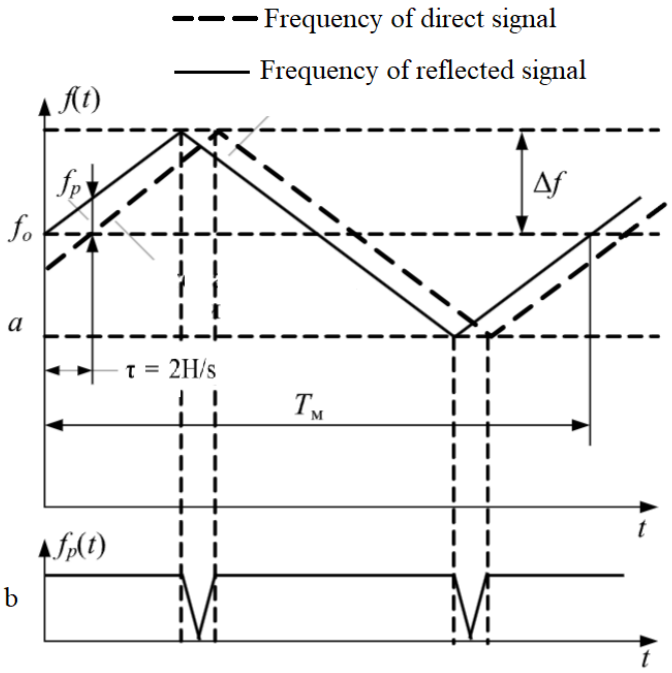



Figure 2.15.Sawtooth law of frequency modulation for radioaltimeter

Therefore, by measuring the difference frequency, you can determine the flight altitude of the remotely piloted aircraft system.

Now consider the structure of a typical RA with FM. The Figure2.16. shows a simplified structure in the form of a block diagram.

The SHF(Super High Frequency) generator generates continuous frequency-modulated oscillations (direct signal) which radiation by the transmitting antenna towards the ground. The signal reflected from the ground surface (reflected signal) are received by the receiving antenna RA and enter on one of the input so f the balance mixer (BM). At the second input of BM is a weakened direct signal. At the output of BM, a differentialfrequency (DFS) signal is output, which is fed to the differential frequency amplifier (FA) with a specially selected amplitude-frequency characteristic, and then to the pulse generating device (PGD).A sequence of pulses of constant amplitude and duration, the frequency of which is equal to the difference frequency, is fed to the PGD output. In

the pulse counter, this pulse sequence is converted into a DC voltage or a digitally signal, the value of which is proportional to Fr, and hence to the measured altitude.

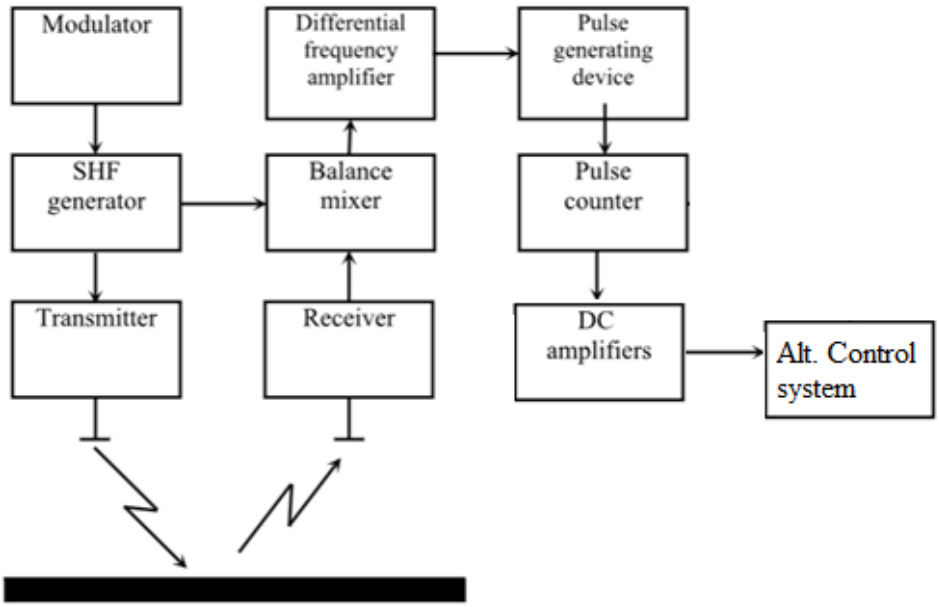



Figure 2.16. Block diagram of the RA with FM

Radioaltimeter have several main characteristics. One of them it is a signal strength. If the ground is smooth, the signal strength at the input of the receiver RA can be determined by the formula:

$$P_{\text{пр}} = pS_{\text{еф}} \qquad (2.16)$$

Another main characteristic it is maximum altitude:

$$H_{\max} = \frac{G\lambda_0 R}{8\pi}\sqrt{z} \qquad (2.17)$$

were G - coefficients of directional action of receiving antenna, R - coefficient thatcharacterizes the intensity of reflection of radio waves from the earth's surface,$\lambda_0$ – is the wavelength corresponding to the average value of the frequency of radiated oscillations, $z = \frac{P_{\text{вип}}}{P_{\text{пр min}}}$ – coefficient called the total sensitivity of RA($P_{\text{вип}}$ - power of the signal emitted by the transmitter, $P_{\text{прmin}}$ signal power at the input of the receiver.

Summarizing the information presented above, we can conclude that the method of altitude measurement by radio waves is quite accurate and with a small error, moreover, it

allows measuring altitude at significant altitudes for remotely piloted aircraft systems. However, like ultrasonic sensors, the radio altimeter measures only true altitude and the altitude measurement must be perpendicular or at a slight angle to the surface. In addition, radio altimeters are heavy and rather large products. In the Figure 2.17you can see the Garmin GRA 5500 radio altimeter with antennas. Its overall characteristics are as follows:

- Unit Altitude (with mounting rack): 3.99" (101.3 mm)
- Unit Width (with or without mounting rack): 3.02" (76.7 mm)
- Unit Depth (with mounting rack and connector): 11.62" (295.2 mm)
- Unit Depth (with mounting rack, connector and configuration module): 12.08" (306.9 mm)
- Unit Weight (with mounting rack): 3.5 lbs (1.6 kg)

As we can see, the large weight and dimensions indicate that the radio altimeter can be installed on remotely piloted aircraft systems of a fairly large size and with a powerful power plant.

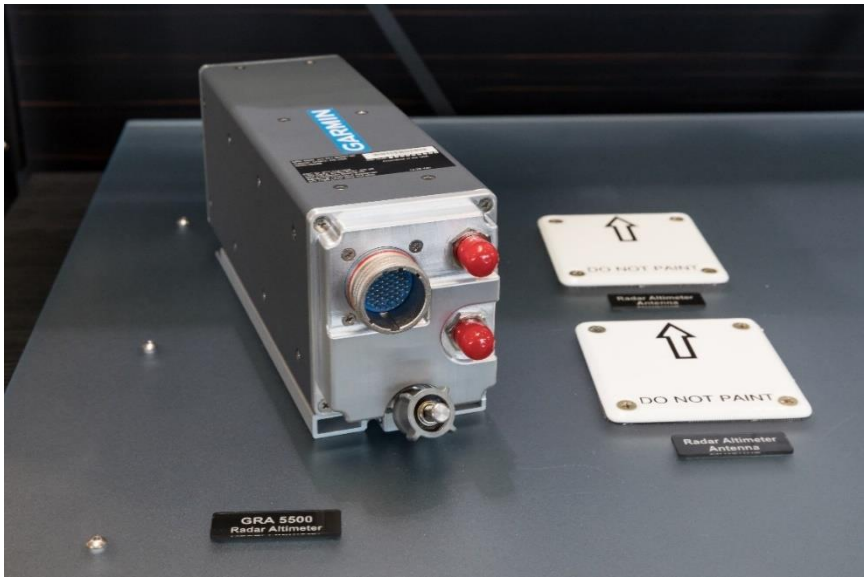



Figure 2.17. Garmin GRA 5500

But recently, radio altimeters of smaller size and weight began to be manufactured especially for remotely piloted aircraft systems of small sizes. Of course, these altimeters

have reduced the maximum altitude at which you can correctly measure altitude. One example of this type of radio altimeters is the US-D1(Figure 2.18).

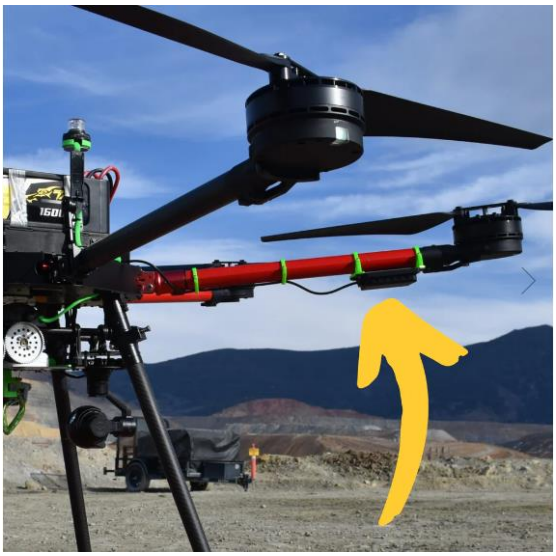


Figure 2.18. Radio altimeter US-D1

It has the following characteristics:

- Minimum Operating Altitude 0.5 meters
- Maximum Operating Altitude 50 meters
- Altitude Precision 6.0cm (< 1m), 4.0cm (> 1m)
- Field of View 43 ° x 30 °

Another disadvantage is the price. The Garmin GRA 5500 costs about $13,000, and the US-D1 costs $600.

Because of their price and specific characteristics, radio altimeters are often used in military drones and cruise missiles of various types.

### 2.4 GPS method

Another way to determine altitude for remotely piloted aircraft systems is to determine altitude using a GPS module.

The Global Navigation Satellite System (GNSS) or GPS (Global position system) relies on a constellation of satellites positioned approximately 25,000 kilometers above Earth's surface. These satellites are equipped with highly precise atomic clocks and

continuously broadcast signals that include both their respective positions and the current time. In contrast, GPS receivers have relatively less accurate quartz clocks. Essentially, a GPS receiver intercepts signals from the visible satellites, determines the time the signal was transmitted and the time it was received, and employs the speed of light to compute the distance to each satellite. With information from at least three satellites, the GPS receiver can establish its position relative to the satellites in a two-dimensional space. When it receives signals from four or more satellites, it can accurately determine its three-dimensional position, and additional satellite data can further enhance its accuracy.

This method has several significant measurement errors. There are a number of potential errors (Figure 2.19):

•The signal from the satellite can be distorted as it it passes through the ionosphere due to the charged

particles present – error about ±5m

•The signal path can be distorted by the troposphere due variations inpressure, humidity, temperature etc – ± 0.5m

•Ephemeris errors (deviation of the satellite from the intended path) –±2½m

•Satellite clock errors,– ±2m

•Receiver clock and other errors. Though the receiver clock is pretty poorpart of the clever bit of the system is that the error can be corrected forby using the satellites atomic clocks – ±4m

•Uncorrected multipath errors (see below) ± 1m

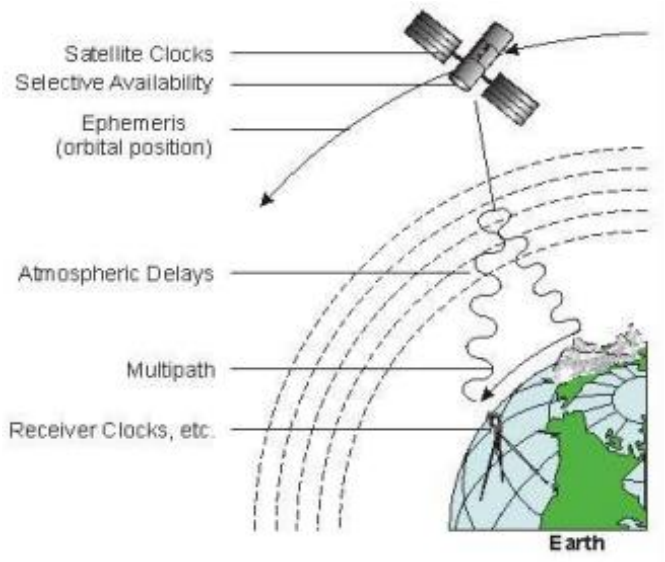



Figure 2.19. GPS error sources

This results in an approximate error of ±15 meters in the horizontal plane. When conducting horizontal trilateration, the receiver can utilize signals from satellites in any direction. However, when determining elevation, it can only rely on satellites positioned above it, which significantly increases the error, roughly two to threefold for altitude measurement, resulting in an error range of ±30-45 meters.

The accuracy of GPS positioning depends on the distribution of satellites in use. A spread-out arrangement of satellites yields greater accuracy, while clustering them in the same area leads to reduced precision. The receiver is aware of the satellite positions and can estimate the reliability of the calculated position, a value often provided to the user, known as GDOP (Geometric Dilution of Precision (Figure 2.20)).

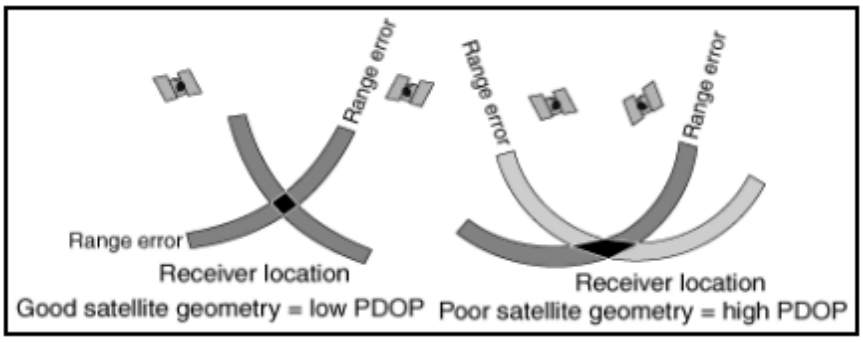



Figure 2.20. GDOP Errors

Another complexity arises from the uneven distribution of Earth's mass, affecting the planet's gravitational pull and thus sea level, resulting in variations across the globe, referred to as the geoid(Figure 2.21). [24]

The advantage of this method is that the GPS module is quite small(Figure 2.22).

But after analyzing the information written above, we can conclude that the use of this method of height determination is not competitive with others due to its large error. It is significantly inferior to the barometric method in accuracy when a MEMS pressure sensor is used.

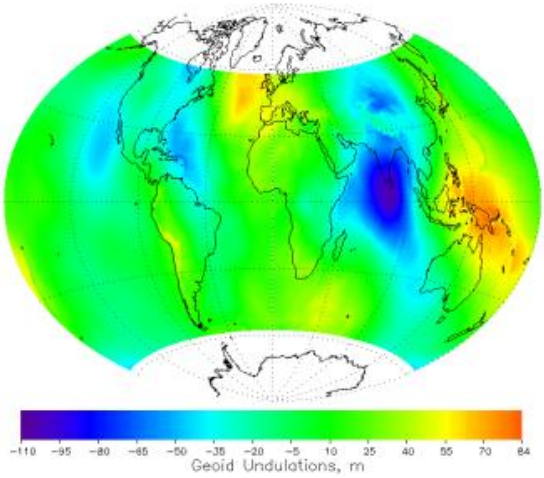



Figure 2.21. World geoid map

And since the size of the module is bigger than the MEMS sensor, the use of the GPS module to determine the height is unprofitable. However, it should be noted that the main task of the GPS module is to determine the position of remotely piloted aircraft systems relative to the earth's surface, and determining the altitude is an additional task.

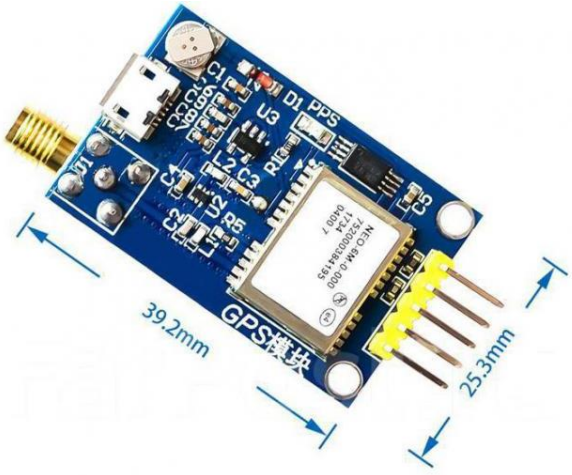



Figure 2.22. Scales of GPS module NEO-8M NANO

## 2.5 Laser method

One of the methods that appeared relatively recently, compared to others, for measuring the height and distance to objects in general is a method that involves the use of a laser beam. This method is used in laser distance measurement sensors.

A laser distance sensor is an electrical device that uses a focused beam of light to measure the distance to an object.

The light emitted by the laser sensor is a small bright point. The laser beam used in the sensor must be powerful enough to be visible in daylight. This increases accuracy and makes it easier to work with the sensor.[25]

Laser light is produced in a special way. In the normal state, most of the atoms in the laser are in the stable low energy level E1. Under the influence of the appropriate frequency of external light, atoms at a low energy level absorb photon energy for excitation and transition to a high energy level E2. Photon energy:

$$E = e2\text{-}e1 = h^* \upsilon \qquad (2.19)$$

where h is Planck's constant and υ is the frequency of the photon. On the contrary, when the frequency of light is υ, the atom in level E2 will jump to the low energy level to release energy and emit light, which is called stimulated radiation (Figure 2.23).

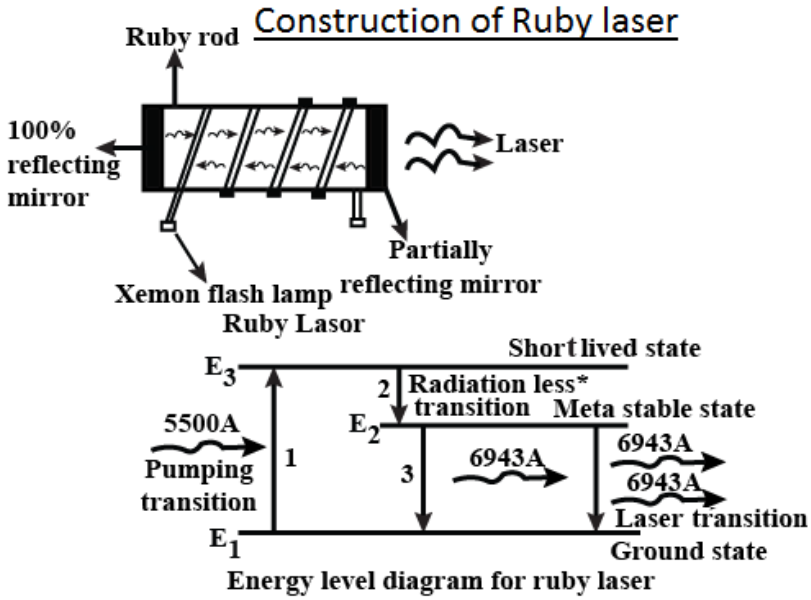



Figure 2.23. Construction and principle of work of ruby laser

The working principle of the laser is to make the atoms of the working materials abnormally at a high energy level, which can make the stimulated emission process dominant, so that the induced light with a frequency of υ can be amplified, and a large stimulated emission light can be generated through the avalanche amplification of the parallel reflector. which is called laser light for short.

Different types of substances are used in the manufacture of lasers:

- Solid-state laser;

- Gas laser;

- Liquid laser;

-Semiconductor laser.

The principle of operation of a laser rangefinder is the same as that of a radio altimeter and an ultrasound sensor. That is, the length or height is measured by measuring the time it takes for the ray to travel the distance there and back, and then the distance of the circular ray is determined by multiplying the speed of light. Due to the advantages of laser such as high directivity, high monochromaticity and high power, they are very important for measuring long distance, determining the orientation of the target, improving the signal-to-noise ratio of the receiving system and guaranteeing the accuracy of measurements. Developments based on the laser rangefinder LiDARsensor ("Laser Imaging, Detection, And Ranging") can be determined not only by position, but also by azimuth, speed and acceleration of the target.I distinguish several types of lasers based on the principle of length measurement:

- Pulsed laser distance sensor. A pulsed laser beam with a very short duration is emitted by a pulsed laser, and after reaching a measured target after a measured distance, part of the energy is reflected back.

- Phase laser distance sensor. The phase change of the modulated laser signal is used to calculate the distance, that is, an indirect method of measuring the phase delay is used. The accuracy of this method can be close to the millimeter level.

- Triangulation laser distance sensor. This measurement principle is that the light emitted by the laser is focused on the surface of the object to be measured after being focused by the collecting lens, and the receiving lens receives the scattered light from the

incident light spot and reflects it on the surface. The resolution of triangulation laser location is very high, which can reach the order of microns.[26]

To make sure of the good characteristics of dpn rangefinders, let's take the LW20 LiDARsensor (Figure 2.24) as an example, which has the following parameters [27]:

- Measuring range: 0.2 … 100 m (LW20/C) or 0.2 … 50 m (LW20/B)
- Size: 30 mm x 20 mm x 43 mm
- Weight: 20g
- Measuring speed: 48 … 388 readings per second (configurable)
- Interfaces: Serial or I2C and servo driver
- Cost: +- $299.
- Power supply voltage: 4.5 V … 5.5 V
- Power supply current: 100 mA
- Outputs & interfaces: Serial and I2C (3.3 V)
- Resolution: 1 cm
- Accuracy: ±10 cm
- Beam divergence:<0.5°
- Operating temperature: -10 ... +50°C

Figure 2.24. The LW20 LiDARsensor

From the above characteristics, we can conclude that for relatively little money (that is, low production costs), we can get a reliable and very accurate product in a small body and with a small weight, which can be powered from 5 V and has an accuracy of 0.1

m with a maximum measurement distance of 100 m. That is, laser rangefinders of this type are one of the best, if not the best devices for measuring height for remotely piloted aircraft systems of all sizes. The only significant disadvantage is that these rangefinders may not work well in difficult weather conditions, such as fog, heavy rain and snow.

# CHAPTER 3
# FILTERS FOR DATA FROM SENSORS

## 3.1 General information

During the dimming period, the sensors transmit very noisy data, which will negatively affect the productivity and stability of the heating system. In my opinion, the height control system also uses sensors that transmit noise, such as MEMS Pressure Sensors. In order to reduce the amount of noise, use different filters. To filter data from sensors that are used in the height control system, digital filters are used.

The term digital filter refers to a hardware or software implementation of a mathematical algorithm, the input of which is a digital signal, and the output is another digital signal, the form and/or amplitude and phase characteristics of which are modified in a special way. In analog systems, a filter is a linear device with a special frequency response that converts an input signal into an output suppressing or, conversely, amplifying certain frequencies in the spectrum of the input signal. The output signal is a convolution of the input signal and the impulse response of the filter. By analogy with an analog filter, a digital filter converts a sequence of counts of an input signal into a numerical sequence of an output signal. There are a large number of digital filters, but let's consider the main ones.

## 3.2 Median Filter

The median filter is based on comparisons, which do not require CPU-intensive operations such as multiplications or divisions. The center value of an odd number of samples is determined by a sequence of comparisons.

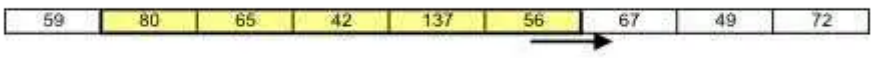
The Median filter smooths a signal by removing spikes. The median filter can be used both on images and on 1-dimensional signal problems. The median is calculated by sorting a list of numbers and finding the number in the middle. The median filter is based on comparisons, which do not require CPU-intensive operations such as multiplications or divisions.

Figure 3.1. Use of a median filter to improve an image severely corrupted by defective pixels

The center value of an odd number of samples is determined by a sequence of comparisons. In Figure 3.1, you can see how this filter can be used to remove defective pixels from an image.

The median filter operates as a sliding window over a set of data samples, sharing a conceptual similarity with a moving-average filter. While a moving-average filter calculates the mean of the window's values, making it susceptible to the influence of large-amplitude outliers, the median filter, on the other hand, identifies the median value as its output. In essence, median filtering essentially sorts the data window each time a new element is incorporated. The output corresponds to the value situated precisely in the middle of all the values within the window.

Suppose we configure the data window to encompass five elements (typically an odd number to ensure a clear median). In this example, let's assume the data window (highlighted in Figure 3.2) contains the following samples:




3.2. A Moving 5-Element Window Used For Median Filter

The median of the filter's window elements stands at 65, as within the window, there exist two values greater than 65 (80 and 137) and two values less than 65 (42 and 56). As the window shifts to the right, the median filter consistently selects the middle value among the five for its output. It's important to note that the outlier in this instance, 137, will never be chosen as the median within any of the groupings formed by the data window, as it surpasses all other values with which it is grouped.

The choice of the data window's length is typically guided by the expected width of impulse noise pulses. For scenarios where noise spikes are brief, such as one or two samples wide, a 5-element window suffices. However, in cases where noise spikes may be broader, a longer data window becomes necessary.

In Figure 3.3, we can see how the medial filter works with noisy data.

The median filter can be written mathematically as follows:

$$\hat{x}(m) = y_{\text{med}}(m) = \text{median}\,[y(m-K), \ldots, y(m), \ldots, y(m+K)] \quad (3.1)$$

Where $\hat{x}(m)$- output of a median filter; $y_{\text{med}}(m)$- input of a median filter with median; $y(m-K)$, ... , $y(m+K)$ - median window of length 2K+1 samples;

y(m) – median.

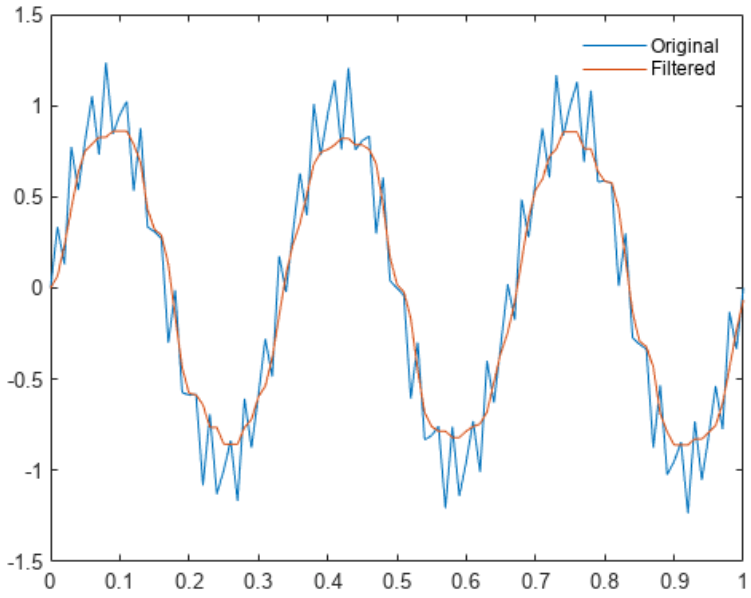



Figure 3.3 Medial filter filtering noisy data

Median filters are ill-equipped to handle "true" impulsive noise, which frequently extends beyond just one or two samples. Additionally, they introduce significant processing distortion by altering legitimate signal samples that are erroneously identified as impulsive noise. Enhancing the effectiveness of median filters can be achieved by implementing an adaptive threshold. This means that a sample is replaced by the median

only when the difference between the sample and the median surpasses the threshold value.[28,29]

### 3.3 Infinite Impulse Response (IIR) Filters

Infinite Impulse Response (IIR) filters are a class of filters that are feedback-based, i.e., the previous output plays a role in the current output. The implementation is simple but versatile: High and low-pass filters, notches and band-pass filters, even gain can be implemented with a rather small set of coefficients.

At the heart of every digital filter lies a mathematical algorithm employed to process a signal and attain a desired outcome. In the case of an Infinite Impulse Response (IIR) filter, this algorithm involves a feedback (Figure 3.4) mechanism that links the filter's output back to its input.

An eminent advantage of IIR filters is their capacity to deliver robust filtering performance while demanding relatively modest computational resources.

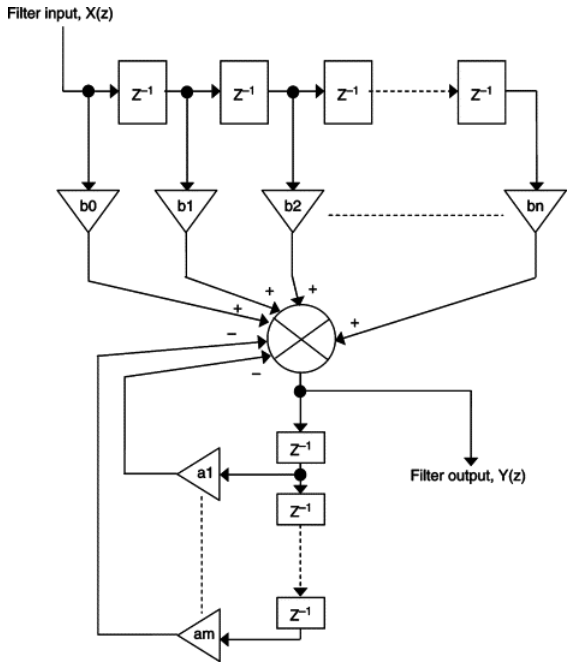



Figure 3.4. Typical architecture of an IIR filter

IIR filters find extensive application in digital control systems, where they play a pivotal role in executing diverse control algorithms such as Proportional-Integral-Derivative (PID) control, lead-lag compensation, and state feedback control. These control

strategies necessitate the incorporation of IIR filters to handle sensor data and produce control signals, ultimately stabilizing the system and achieving the desired performance goals.

An IIR filter can be described by the differential equation:

$$y[n] = \sum_{k=0}^{M-1} b_k x[n-k] - \sum_{k=1}^{N-1} a_k y[n-k] \quad (3.2)$$

The implementation we have is an example of a third-order IIR filter.In the initialization phase, the $b_{0-2}$ and $a_{0-2}$ are set based on the type of filter, e.g., Low-Pass Filter. When running the filter, $x_{n-k}$ and $y_{n-k}$ values are updated continuously by the current signal, while $a_k$ and $b_k$ are constants.

Since the IIR filter is a class of filters and not a separate filter, it includes various filters with different characteristics. The complete list of supported filters is: Low-Pass Filter, High-Pass Filter, Band-Pass Filter, Notch Filter, Peaking Band EQ Filter, Low Shelf Filter, High Shelf Filter, etc.

For example, consider the Low-Pass Filter.

A low-pass filter is designed to permit signals below a specified frequency threshold, denoted as "freq," while diminishing or rejecting signals outside that range. In the context of filters with distinct high and low discrimination, the term "bandwidth" typically refers to this frequency threshold. Think of a low-pass filter as essentially a "band-pass filter from zero to the bandwidth," while a high-pass filter operates as a "band-pass filter from the bandwidth to infinity." In this scenario, the variable "bandwidth" influences the transition between high and low regions.

In the graphical (Figure3.5) representation, the green trace represents the filtered signal, while the blue trace represents the original signal. The graph clearly illustrates that the low-pass filter effectively eliminates high-frequency components. [28, 30]

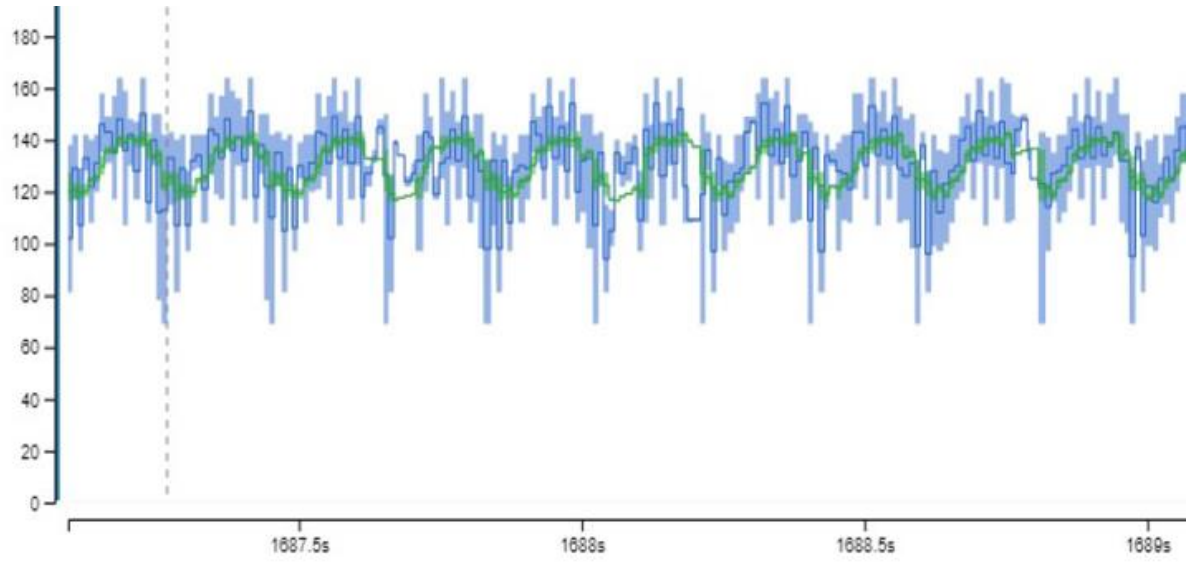



Figure3.5. Representation of IIR Low-Pass Filter filter work

### 3.4 Kalman Filter

Numerous contemporary systems employ multiple sensors to infer concealed or unknown states by analyzing a sequence of measurements. Take, for example, a GPS receiver, which can deduce location, altitude and velocity as the concealed states. The measurements, in this case, are provided by the differential time of signal arrival from satellites.

Among the foremost challenges encountered in tracking and control systems is the provision of precise and accurate estimations for these concealed states, particularly when uncertainty is prevalent. GPS receivers, for instance, contend with measurement uncertainties stemming from external factors such as thermal noise, atmospheric effects, minor shifts in satellite positions, receiver clock precision, and more.

The Kalman filter, as depicted in Figure 3.6, represents a recursive method for minimizing the least square error when estimating a signal that has been distorted during transmission through a channel and is observed amidst noise. Kalman filters are versatile and can be applied to processes that exhibit both time-varying and time-invariant characteristics. The theoretical foundation of the Kalman filter is rooted in a state-space framework, where a state equation captures the dynamics of the signal generation process,

while an observation equation describes the observation signal, which is subject to noise and distortion.

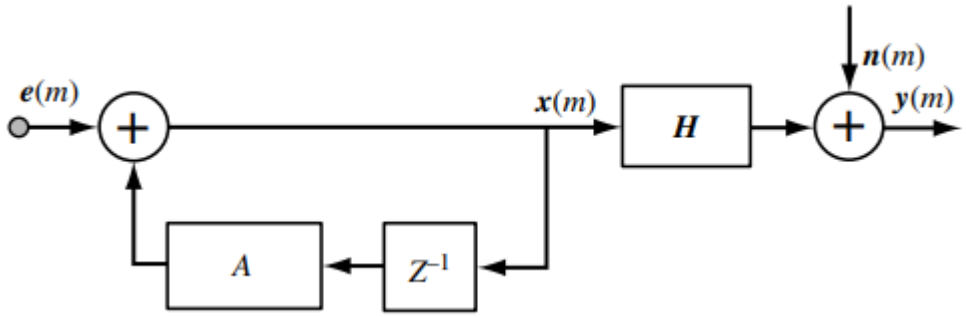



Figure 3.6. Illustration of signal and observation models in Kalman filter theory

For a signal x(m) and noisy observation y(m), equations describing the state process model and the observation model are defined as

$$\boldsymbol{x}(m) = A\boldsymbol{x}(m-1) + \boldsymbol{e}(m) \tag{3.3}$$

$$\boldsymbol{y}(m) = H\boldsymbol{x}(m) + \boldsymbol{n}(m) \tag{3.4}$$

where x(m) is the P-dimensional signal, or the state parameter, vector at time m; A is a P ×P dimensional state transition matrix that relates the states of the process at times m−1 and m; y(m) is the M-dimensional noisy and distorted observation vector; H is the M ×P dimensional channel distortion matrix; nm is an M-dimensional noise vector, also known as measurement noise – n(m) is a normal (Gaussian) process; and R is the M ×M dimensional covariance matrix of nm. [31]

For a better understanding, describe the operation of the filter in the form of an algorithm (Figure 3.7).[32]

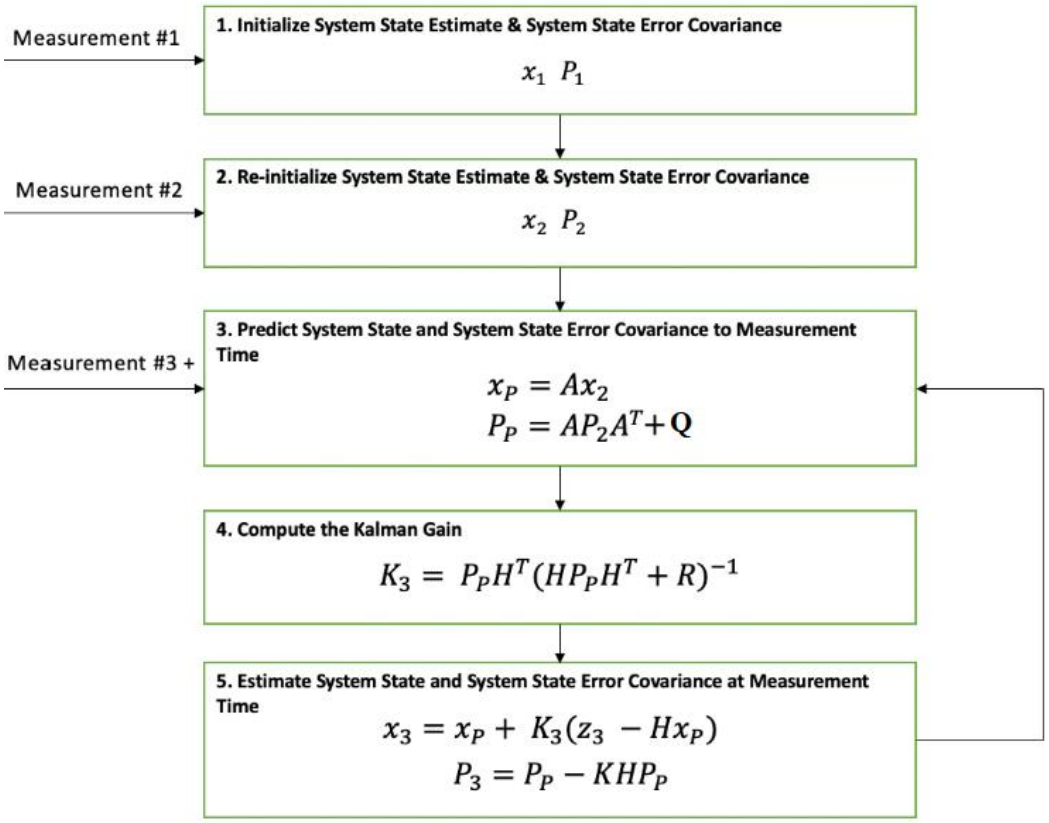



Figure 3.7. Kalman filter algorithm

From this representation of Kalman Filter Algorithm we can describe the following variables:

1. x (n x 1 column vector ) - state variable ;
2. P (n x n matrix)-state covariance matrix (estimate variance );
3. z (m x 1 column vector)- measurement;
4. A ( n x n matrix) - state transition matrix;
5. H (m x n matrix)- state-to-measurement matrix;
6. R (m x m matrix)- measurement covariance matrix (measurement variance);
7. Q ( n x n matrix)- process noise covariance matrix (or just our noise);
8. K (n x m matrix)-Kalman Gain;

The algorithm described above is a multivariate(multidemenshion)Kalman filter, which is suitable for processing a GPS signal or a radar signal. That is, where the signal consists of several components That's why they use matrices. To measure one variable that does not have the state-to-measurement matrix, H, use the Kalman Filter in one dimension.

If we don`t have state transition, Kalman filter algorithm (Figure 3.8) has the following form:

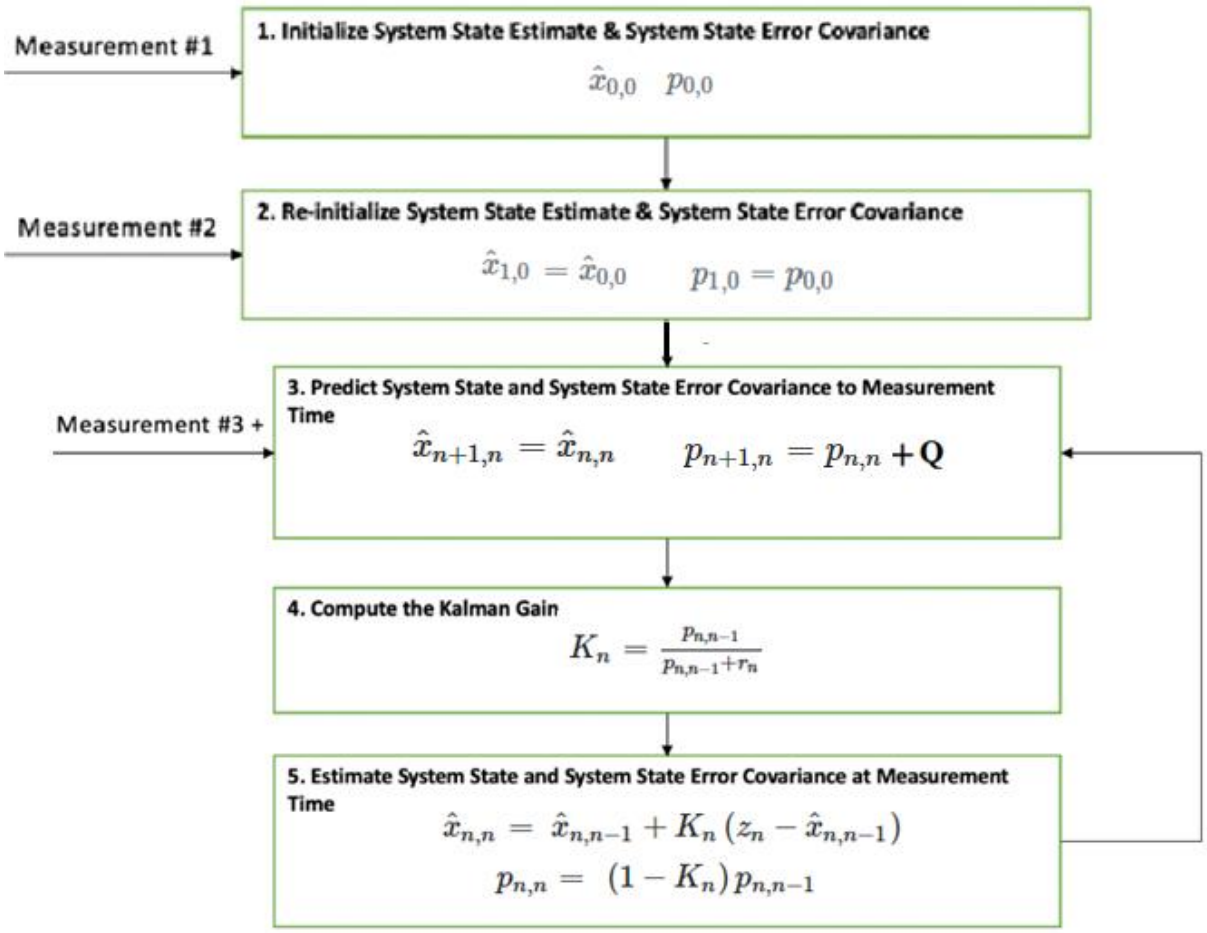



Figure 3.8. One dimensionKalman Filter without dynamic model

For better visualize this algorithm make experiment. Formulas I take from book "Kalman filter from the ground up" that was written by Alex Becker[33]. Using the Arduino UNO printed circuit board and BME 280 barometric sensor, we measure (Figure 3.9) thealatitude with a frequency of 1 second 12 times (Table 1).The sensor is doesn`t move. Let us assume that the noise is equal to 0.01 m. In our case, measurement variance of BME 280 was not publicly available, but we can roughly calculate it using the formula:

$$\text{rp}= \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 \quad (3.6)$$

Where $\sigma^2$- mean squared error

Where $\mu=\frac{1}{12}(172.31+172.45+172.49+172.38+172.60+172.43+172.78+172.60+172.63+172.54+172.31+172.57)=\pm172.5.$

So by substituting the variables we get:

$$r_p=\sigma^2=\frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)^2=\frac{1}{12}((172.31-172.5)^2+(172.45-172.5)^2+\ldots$$

$$+(172.31-172.5)^2+(172.45-172.5)^2)=\frac{1}{12}((-0.19)^2+(-0.05)^2+(-0.01)^2+(-0.12)^2+0.1^2+(-0.07)^2+0.28^2+0.1^2+0.13^2+0.04^2+(-0.19)^2+0.7^2)=0.018$$

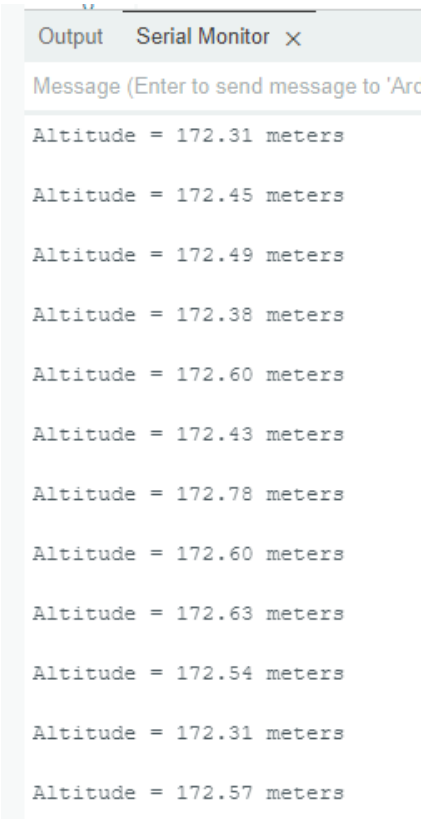



Figure 3.9. Meashure of our BME 280 sensor in a SerialMonitor in Arduino IDE

Table 1

| Number of measure | Measured altitude |
|---|---|
| $z_1$ | 172.31 |
| $z_2$ | 172. 45 |
| $z_3$ | 172.49 |
| $z_4$ | 172.38 |
| $z_5$ | 172.60 |
| $z_6$ | 172.43 |
| $z_7$ | 172.78 |
| $z_8$ | 172.60 |
| $z_9$ | 172.63 |
| $z_{10}$ | 172.54 |
| $z_{11}$ | 172.31 |
| $z_{12}$ | 172.57 |

So, after that calculation we can start work with One-dimensionKalman Filter

1. Assume that we would like to estimate the altitude using an imprecise altimeter any class. For example, the altimeter shows a 172.8 m. That's ours $\hat{x}_{0,0}$. Now we shall initialize the estimate variance. Let us assume that estimation error (standard deviation) of this altimeter is 5 m., Consequently the variance is $\sigma^2(p_{0,0}) = 5^2 = 25$ м.

2. Since our system's Dynamic Model is constant, so our variablesis not change($x_{1,0} = x_{0,0}, p_{1,0} = p_{0,0}$).

3. Add a noise to the $p_{1,0}$=25+0.001=25.001 m.

4. After that we can calculate Kalman gain for the first measure:

$$K_1 = \frac{p_{1,0}}{p_{1,0} + r_p} = \frac{25.001}{25{,}001 + 0.018} = 0.999$$

5. After calculation of Kalman gain we can calculate first current estimate variance

$x_{1,1}= x_{1,0}+K_1(z_1+ x_{1,0})=172.8+0.999(172.31-172.8)=172.31(m)$

6. Calculation of the extrapolated estimate variance:

$p_{1,1}=(1- K_1)* p_{1,0}=(1-0.999)*25.01=0.025$

7. Add a noise to the $p_{1,1} = 0.025 + 0.001 = 0.026$

8. Make a prediction: $x_{2,1}= x_{1,1}=172.311$, $p_{2,1}= p_{1,1}=0.026$

9. Now repeat points from 4 to 8:

$$K_2 = \frac{p_{2,1}}{p_{2,1} + r_p} = \frac{0.026}{0.026 + 0.018} = 0.591$$

$X_{2,2}= x_{1,0}+K_1(z_1+ x_{1,0})=172.31+0.591\ (172.45-172.31)=172.392(m)$

$p_{2,2}=(1- K_1)* p_{1,0}=(1-0.591)*\ 0.026=0.01063$

$$p_{3,2} = 0.01063 + 0.001 = 0.01163$$

$x_{3,2}= x_{2,2}=172.392$

Further calculations from 3 to 12 wemake by using a MathCad (Appendix A).After all calcullation we obtain a data, that'sare placed in Table 2.

Analyzed a $K_n$ and $p_{n,n}$ data from Table 2, we can make a conclusions, that these values ? decrease over time and their graph will resemble one part of a hyperbola. To analyse data of our estimate of the state, make a graph (Figure 3.9).

| | $K_n$ | $x_{n,n}$ | $p_{n,n}$ |
|---|---|---|---|
| 0 | - | 172.800 | 25 |
| 1 | 0.999 | 172.310 | 0.02501 |
| 2 | 0.591 | 172.392 | 0.01163 |
| 3 | 0.393 | 172.430 | 0.00807 |
| 4 | 0.310 | 172.415 | 0.00657 |
| 5 | 0.267 | 172.464 | 0.00581 |
| 6 | 0.244 | 172.456 | 0.00539 |
| 7 | 0.230 | 172.531 | 0.00515 |
| 8 | 0.222 | 172.546 | 0.00500 |
| 9 | 0.217 | 172.564 | 0.00491 |
| 10 | 0.214 | 172.559 | 0.00485 |
| 11 | 0.212 | 172.506 | 0.00482 |
| 12 | 0.211 | 172.520 | 0.00480 |

Table 2

Figure 3.9. Graphical Representation of Measurement Values and Their Refinement Through a One-Dimensional Kalman Filter as Processed by MathCad

The graphical data presented in Figure 3.11 offers insightful evidence regarding the efficacy of the Kalman filter in mitigating noise in sensor data. This visualization enables us to deduce that the Kalman filter is proficient in filtering out noise components from the sensor readings. Consequently, this proficiency underpinned the decision to employ the Kalman filter for data refinement in this study.

In the practical application phase of this project, the Arduino UNO printed circuit board was utilized alongside the BME 280 barometric sensor. To demonstrate the filter's

functionality in a real-world scenario, the Serial Port Plotter software was employed. The specific code implementation, detailed in Appendix B, was central to this process. To determine the measurement variance, Microsoft Excel Office was used, particularly leveraging its STANDARD DEVIATION function, a robust tool for statistical analysis in this context.

The Serial Port Plotter played a crucial role in visualizing the data from the Arduino. It was adeptly used to transmit both filtered and unfiltered signals into the program for comparative analysis. Figure 3.10 presents the resultant graph, which displays two distinct curves. The red curve represents the unfiltered signal, illustrating the raw data as received from the sensor. In contrast, the yellow curve depicts the filtered signal, showcasing the effectiveness of the Kalman filter in enhancing the clarity and reliability of the sensor data. This comparative graphical representation serves as a compelling testament to the Kalman filter's capacity to significantly improve data quality, an aspect critical to the accuracy and reliability of measurements in remotely piloted aircraft systems.

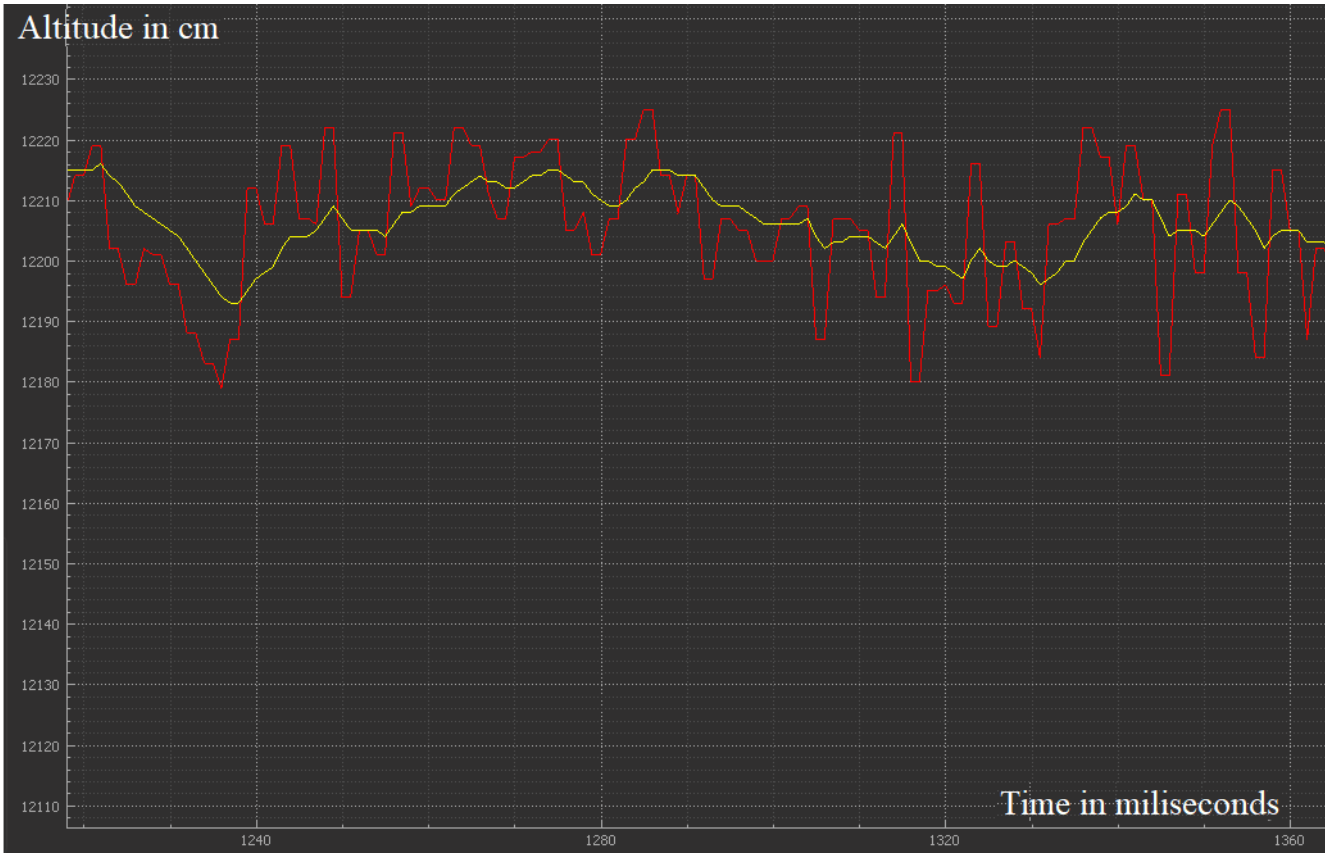



Figure 3.10. Representarion of work of the simple Kalman filter

In the realm of height measurement for remotely piloted aircraft systems (RPAS), the incorporation of vertical velocity is a standard practice. For enhanced precision, a two-dimensional Kalman filter, which integrates both vertical velocity and vertical acceleration, is often employed in the measurement matrix. The rationale behind this approach stems from the limitations observed with a one-dimensional Kalman filter when it operates devoid of dynamic changes. Specifically, a one-dimensional filter tends to exhibit significant inertia during movement, which can adversely affect the system's accuracy and responsiveness. This is deemed unacceptable for the optimal functioning of altitude control systems.

To calculate vertical speed, the same sensor utilized for altitude measurement is employed. The formula for determining speed is based on the fundamental equation:

$$v=d/t \qquad 3.6$$

where d-distance, t-time.

Additionally, a prevalent technique in the accurate estimation of a quadcopter's velocity involves the fusion of data from accelerometers and rate gyroscopes, typically through the use of a Kalman or complementary filter. Accelerometers are tasked with measuring both the acceleration due to gravity and any supplementary linear accelerations. Concurrently, rate gyroscopes are dedicated to gauging the rate of rotation. By aggregating and analyzing data on both acceleration and angular velocity over time, it becomes feasible to estimate the velocity of RPAS in various planes, including vertical velocity, which directly correlates to altitude.

In the context of RPAS, Micro-Electro-Mechanical Systems (MEMS) accelerometers are frequently chosen for acceleration data acquisition. The preference for MEMS technology in drones is primarily due to their advantageous attributes of minimal weight and compact size. These characteristics are crucial in the drone industry, where optimizing payload capacity and maintaining a streamlined design are paramount for efficient and effective operation.

In summary, the integration of advanced Kalman filtering techniques and the strategic use of MEMS accelerometers and gyroscopes are pivotal in achieving accurate

and reliable height measurement in RPAS. These technologies collectively enhance the system's ability to adapt to dynamic changes in movement, ensuring precise altitude control and overall system stability.

# CHAPTER 4

# CONTROL OF THE ALTITUDE OF A REMOTELY PILOTED AIRCRAFT SYSTEM BY AN AUTOMATIC PILOT

## 4.1 General information

Altitude control systems are part of the overall autopilot system, which includes various subsystems and the work of which relies on many sensors and algorithms. These subsystems are closely related to each other, because the work of each subsystem of the system will affect the work of another. They also usually use data from the same sensors. For example, the altitude control system can take the altitude parameter from the attitude control system for more accurate correction.

In the early days of aviation, an airplane required the pilot's constant attention in order to maintain a safe flight. Since the size of the plane increased, which allowed to fly for many hours, constant attention led to serious fatigue. The operator of remotely piloted aircraft systems also often faces this problem. The autopilot is designed to perform some of the pilot's tasks. The first autopilot for an airplane was developed by the Sperry Corporation in 1912[34] (Figure 4.1).

Figure 4.1. Lawrence Sperry (left) test his airplane with gyroscopic autopilot

In an autopilot, a gyroscopic heading indicator and air horizon were connected to hydraulically operated elevators and a rudder. (The ailerons were not connected because

the dihedral wing was designed in advance to provide the necessary roll stability.) It allowed the aircraft to fly straight and level on a compass course without the need for pilot intervention, greatly reducing the strain on the pilot. In part, I told about the history of the introduction of autopilots for remotely piloted aircraft systems in Chapter 1.

### 4.2 Feedback control loops

In general, the basis of any autopilot is a feedback loop. Feedback loops are based on the analysis of output data and this allows the system to regulate its work on its own without external intervention. The schematic structure of the feedback control loop is presented in Figure 4.2.

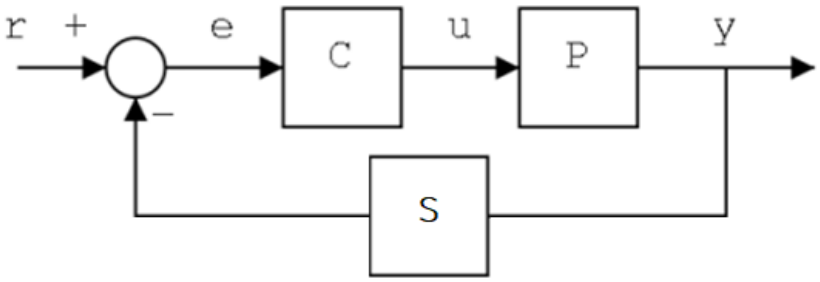



Figure 4.2. Basic Feedback Structure

This structure includes some components, such as:

- Plant

The plant is a system given to us, which (without the help of a controller or compensator) needs to improve its performance. Plants are usually given "as is" and are not subject to change. It is denoted by the letter P.

- Controller

The controller is an additional system that is added to the installation to control the operation of the installation. The system can have several compensators in order to compensate for various different parameters, they can be placed anywhere, regardless of whether before or after the plant. It is denoted by the letter C. The detailed analysis of the controllers is described in the next subsection.

- Sensor

A device that reads the necessary information about the state of the system. It is denoted by the letter S.

- Sum

A sum is a symbol on a system diagram (indicated above by parentheses) that conceptually adds two or more input signals and produces a single, final output signal.

- Selection node

The sensor node is a split wire.

- The way forward

The forward path in the feedback loop is the path after the Sum that goes through the plant to the output signal of the system.

- The way back

The return path is the path after the selection node that returns to the beginning of the system. This is also known as the "feedback path.

Blocks P and C are usually understood as transitional functions.

The transfer function serves as a convenient representation for expressing the input-output ratio of a linear, time-invariant system. It is derived by applying the Laplace transform to the differential equations that describe the dynamics of the system, assuming zero initial conditions. Alternatively, in the absence of these equations, the transfer function can be approximated using input-output data obtained by measuring. The main advantage of transfer functions is their ability to allow engineers to use simple algebraic equations instead of dealing with complex differential equations when analyzing and designing systems. [35]

There are two types of feedback control loop that are used in the design of autopilots:

• Open loop: This loop provides a control signal (right, left, up, down) without sensor data feedback.

• Closed loop: This loop provides sensor feedback to adjust behavior such as stability

### 4.3 PIB controller

Controllers are an integral part of all automatic systems, including unmanned aerial vehicles and aircraft in general. They are of various types: mechanical, electromechanical or electronic devices. In our case, the controllers control various flight parameters of the aircraft, such as roll, yaw, pitch, position of the aircraft in the three-dimensional system of coordinates (x ,y ,z ) ect. In our case, we are interested in the position of the aircraft in the three-dimensional system of coordinates along the z axis, because this is our altitude. One of the most common controllers used to control altitude in remotely piloted aircraft systems, as well as other flight parameters in general, is Proportional Integral Derivative (or simply PID) controllers.

A PID controller is primarily a digital device used in control systems to regulate various parameters. These controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable of their counterparts.

The main principle of operation of this controller is to calculate the error value as the difference between the desired setpoint and the erroneous process variable, and adds or removes corrections based on proportional, integral and derivative terms. If we describe all this in the form of a mathematical formula, then we will get the following formula:

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\,\mathrm{d}\tau + K_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t} \qquad (4.1)$$

Where

- $K_p$- non-negative coefficient for the proportional terms. Account for present values of error (the bigger the number the harder controller pushes);
- $K_i$ - non-negative coefficient for integral terms. Account the past value of error (the smaller the number the more quickly controller reacts to load changes, but greater the risk of oscillation);
- $K_d$ - non-negative coefficient for the proportional derivative terms. Accounts for possible future values of the error, base of current rate of chance (the bigger the number, the more controller dampens oscillation);
- u(t)- our control variable;
- e(t)- error value.

The transfer function of a PID controller we can represented as:

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \tag{4.2}$$

A block diagram of a PID controller we can see in Figure 4.3. In this Figure r(t) is the desired process variable (PV) or setpoint (SP), y(t) is the measured PV

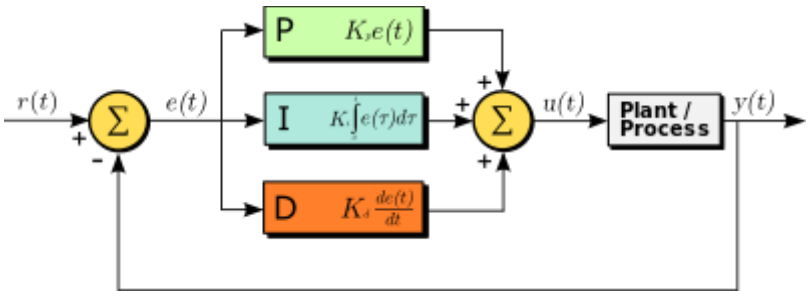



Figure 4.3. A block diagram of a PID controller in a feedback loop; r(t) is the desired process variable (PV) or setpoint (SP), and y(t) is the measured PV.

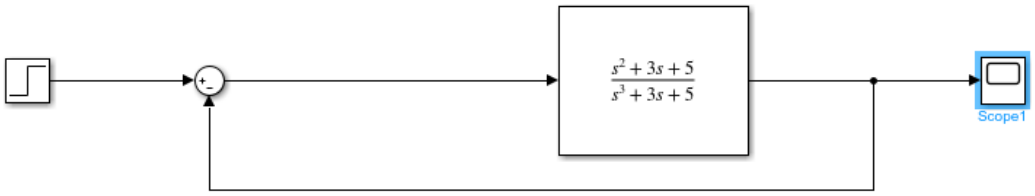
The dynamics of a PID controller can be effectively simulated using the Simulink software platform. This simulation facilitates an in-depth analysis of how the PID controller's tuning parameters, namely the proportional (Kp), integral (Ki), and derivative (Kd) coefficients, influence the system's output response. Such a simulation provides valuable insights into the controller's behavior under various conditions and is instrumental in optimizing control strategy for desired performance [Refer to Source 11].

For the purposes of our experimental study, a model was constructed within Simulink. This model encompasses several key components: a step function (with a final value set to 1), a summation block representing the negative feedback mechanism, a transfer function to model the system dynamics, and a scope for visualizing the system's response. The step function acts as a simple yet effective input signal to test the PID controller's response to a sudden change in the setpoint.

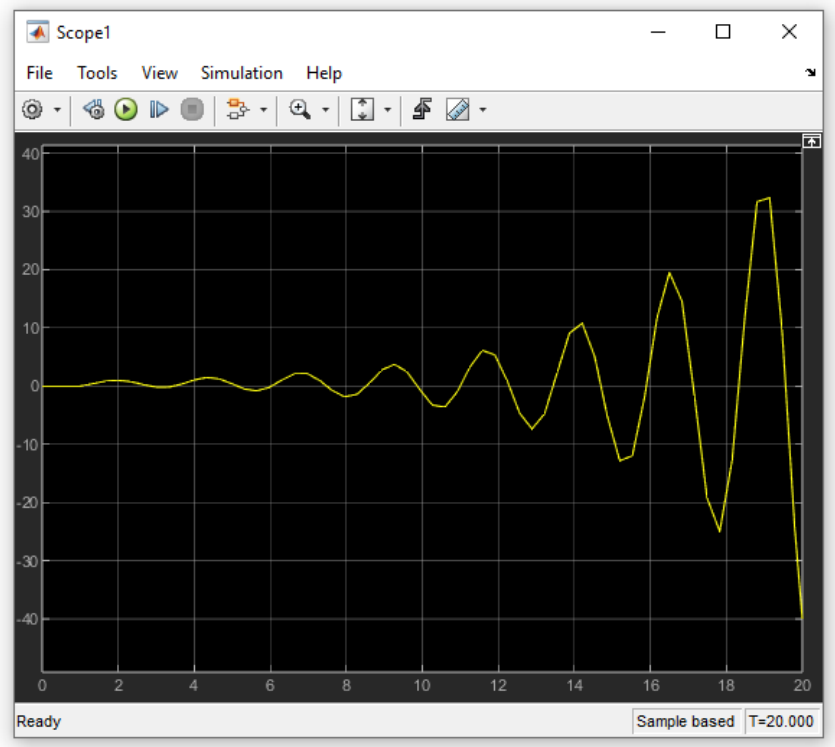
The schematic layout of this simulation model is detailed in Figure 4.4 (a), providing a clear visual guide to the arrangement and interconnection of the various components within the model. Additionally, the response of the system as modulated by

the transfer function is graphically represented in Figure 4.4 (b). This graph is crucial for interpreting the system's transient and steady-state behaviors under the influence of the PID controller.

Through this simulation, we aim to comprehensively understand the impact of each PID coefficient on the system's ability to reach and maintain the desired setpoint, while minimizing overshoot and ensuring stability. This understanding is critical in the field of control systems, especially in applications where precision and rapid response to changing conditions are paramount.



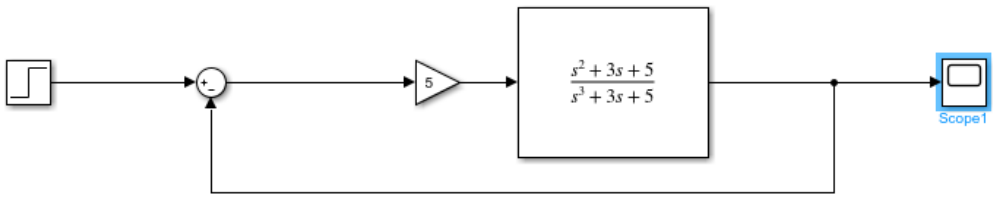

(a)



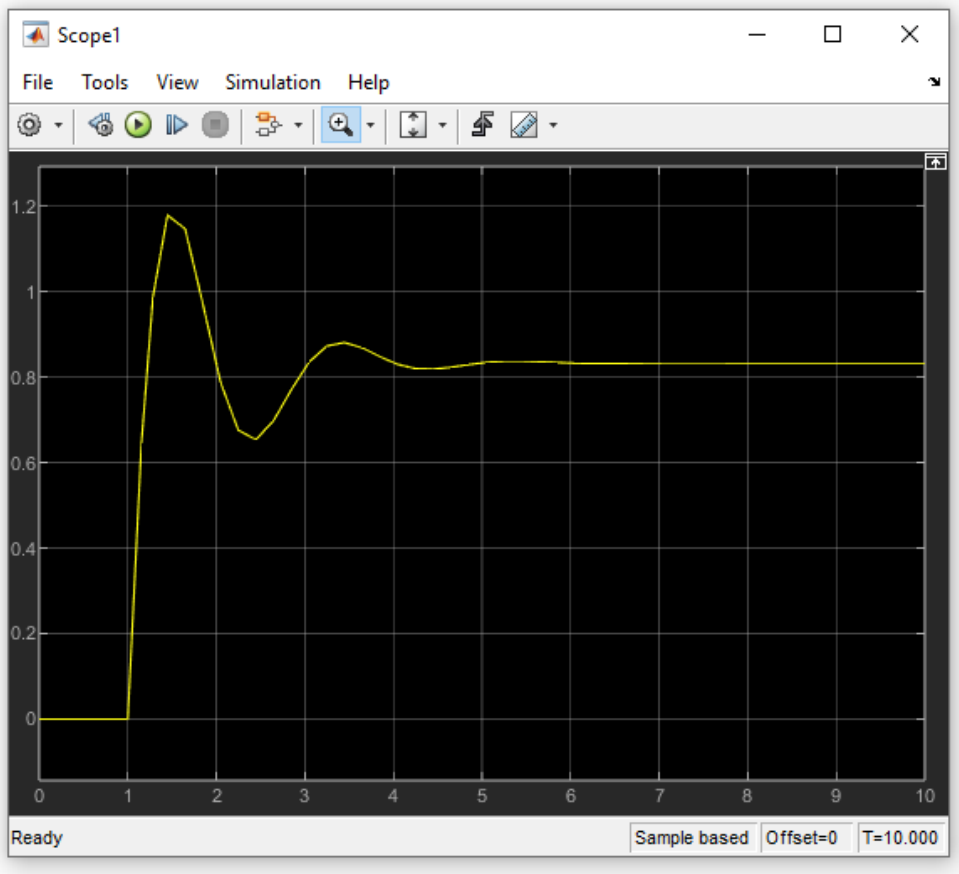

(b)

Figure 4.4. The block diagram of my simulation model (a), and the graph of the transfer function (b).

Now add one gane, that simulate our P coefficient and after that start our simulation (Figure 4.5).



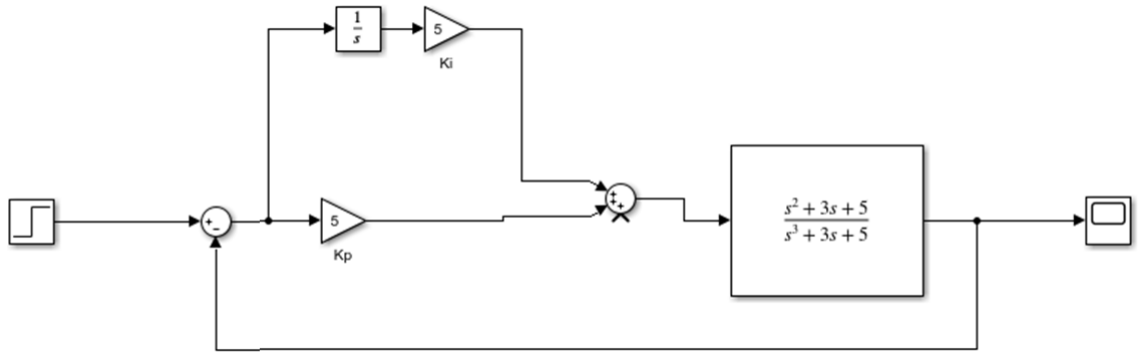

(a)



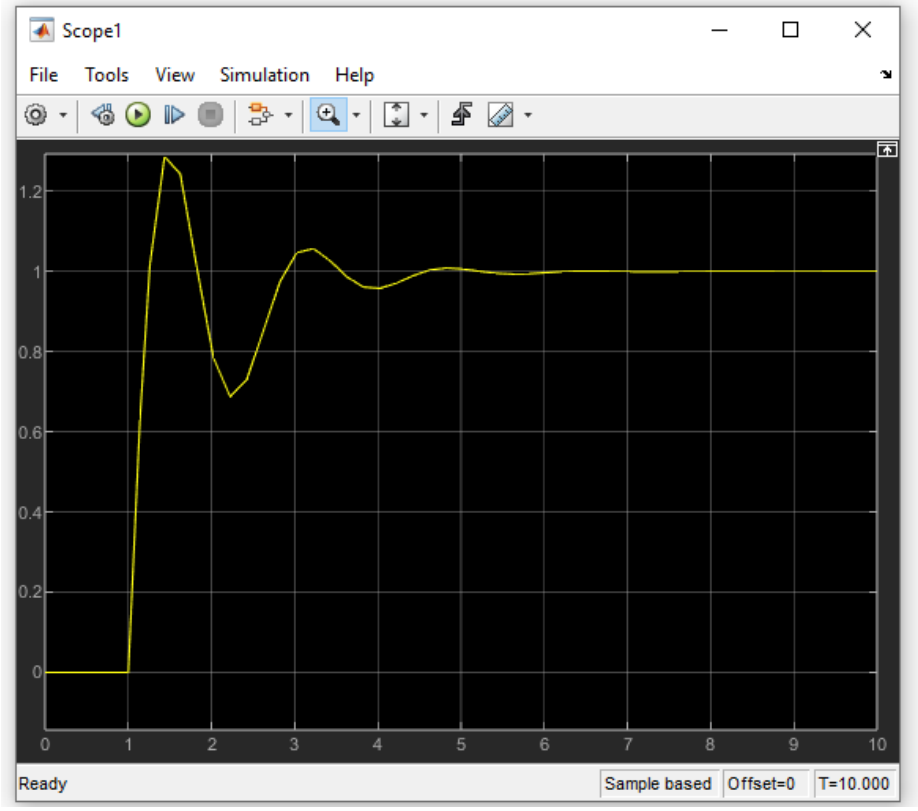

(b)

Figure 4.5. The block diagram of my simulation model with P coefficient (a), and the graph of the transfer function (b).

On the graph, that we can observe on Figure 4.5 (b), we can see, that transfer function after two oscillations, that continued +-5 seconds, had stabilized. But we still have two oscillations (overshoot) and our output is stabilized on 0.8, not on 1(Steady state error). So we can add I coefficient. For simulation I coefficient we take a integrator and gain (Figure 4.6 (a)). Add one sum to summarize our two coefficient P and I. On the Figure 4.6 (b) shown that two oscillations are still remain in our function, but final value is become 1.



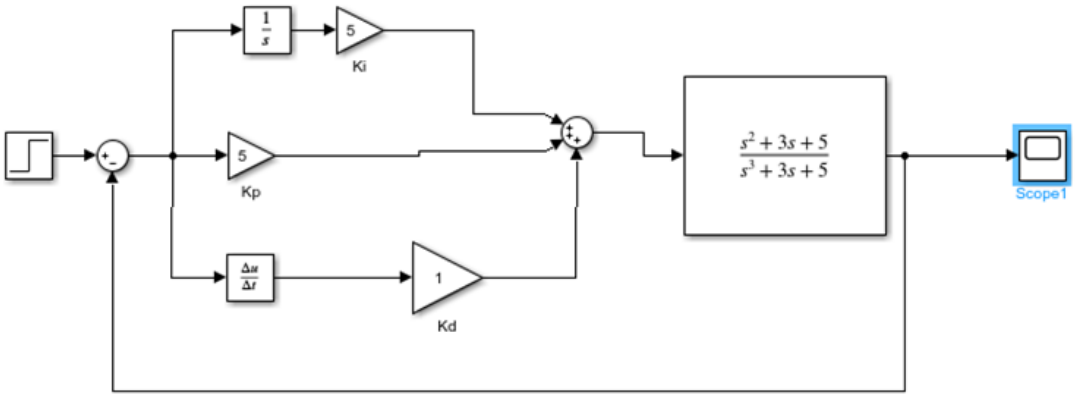

(a)



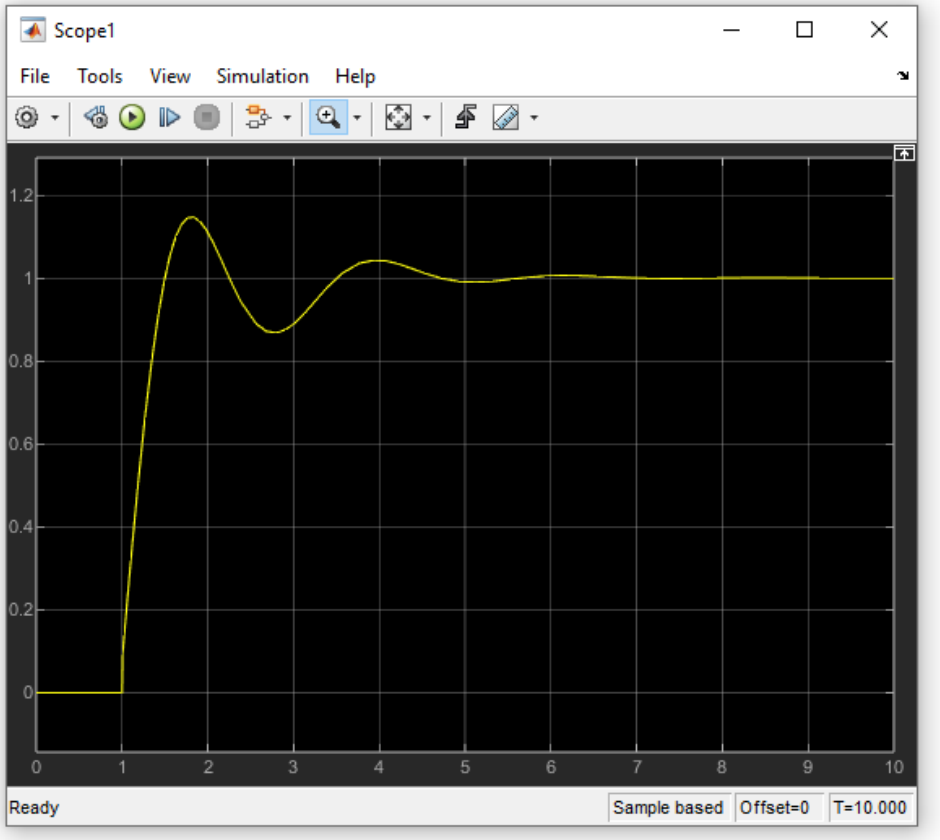

(b)

Figure 4.6. The block diagram of my simulation model with P and I coefficients (a), and the graph of the transfer function (b).

Now, add our derivative (D) coefficient to decrease our overshoot. To simulate D coefficient, we need to add a derivative block and gain and connect it to our sum block (Figure 4.7 (b)). The final graph of simulation we can see on the Figure 4.7 (b).




(a)




(b)

Figure 4.7. The block diagram of my simulation model with P, I, and D coefficients (a), and the graph of the transfer function (b).

After analyzing the graph, we will see, that our overshoot is become smaller. By changing (tuning) the parameters P, I, and D we can get the output, that we need.

# CHAPTER 5

# PROJECT HARDWARE

## 5.1Arduino UnoRev 3

The Arduino Uno Rev 3 is an advanced microcontroller board based on the ATmega328, an 8-bit microcontroller. This board is equipped with a diverse array of features that cater to a wide range of applications in electronics and robotics. The notable elements of the Arduino Uno Rev 3 include:

- Digital I/O Pins: It houses 14 digital input/output pins, offering considerable flexibility. Notably, 6 of these pins are capable of providing Pulse Width Modulation (PWM) output, enabling them to control devices like servo motors or adjust LED brightness.
- Analog Inputs: There are 6 analog input pins, allowing the board to interface with various analog sensors and devices.
- Oscillator: The presence of a 16 MHz crystal oscillator ensures precise timing and control over processes.
- Connectivity: For interfacing with computers or other devices, the board is equipped with a USB connector. Additionally, a power jack allows for external power sources like AC/DC adapters or batteries to be connected easily.
- Programming Interface: The inclusion of an In-Circuit Serial Programming (ICSP) connector facilitates the uploading of programs to the microcontroller.
- Reset Button: A reset button is included for easy rebooting of the system.

One of the key strengths of the Arduino Uno lies in its versatile communication capabilities. The ATmega328 microcontroller at its heart features a UART transceiver, enabling effective serial communication through digital pins 0 (RX) and 1 (TX). This feature is vital for data transmission between the Arduino and computers, other Arduino boards, or different microcontrollers. Additionally, the microcontroller supports other

serial communication protocols, including the I2C (TWI) and SPI interfaces, further broadening its range of applications in various digital communication contexts [36].

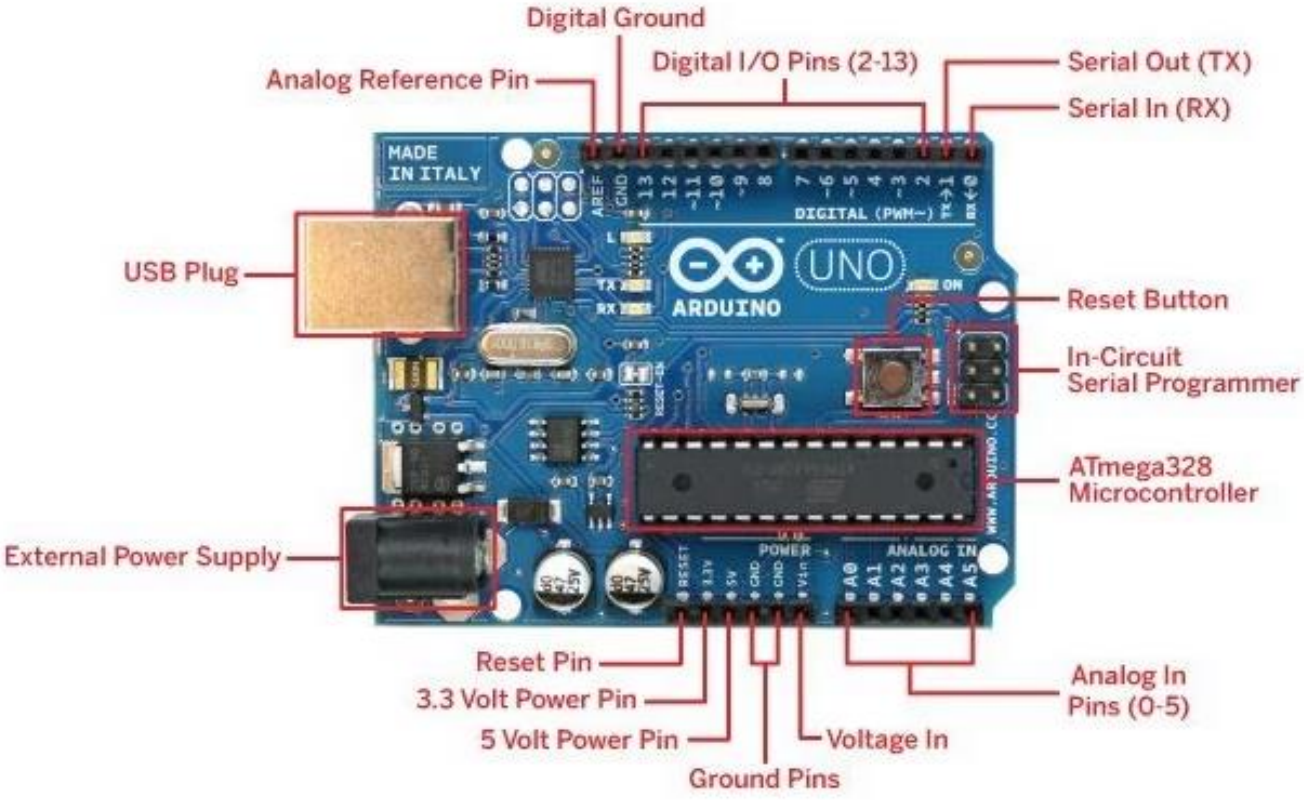



Figure 5.1. Arduino Uno with Rev 3

### 5.2 LCD 2004 display

LCD 2004 (Figure 5.2) just a simply LCD display for output text information. The display has a green backlight. Have 4 lines of text of 20 characters each. It has a contrast control and HD44780 compatible interface. I use as an additional device I2C LCD adapter to simplify data transfer and reduce the number of pins involved.

Figure 5.2. LCD 2004

### 5.3 Servo drive SG90

Small servo drive with torque of 2 kg/cm. Some characteristic of servo drive:

- no-load speed :0.12 sec/60 deg. at 4.8V power supply
- temperature range: 0 to +50 °C
- current consumption in motion: 50-80 mA
- rotation angle 120°
- dimensions: 3.3 cm x 3 cm x 1.3 cm
- weight: 9


Figure 5.3 Servo drive SG90

### 5.4 MEMS pressure sensors BMP180 AND BME280

To obtain altitude readings using atmospheric pressure, I used MEMS pressure sensors BMP180 and BME280. I have already described the operation of the sensor BMP180 in chapter 2. Unlike the BMP180 sensor, the BME280 sensor has somewhat expanded capabilities, because it can also measure air humidity. This sensor is often used to build small home weather stations. Like BMP180, BME280 uses a I2C bus to transmit data.

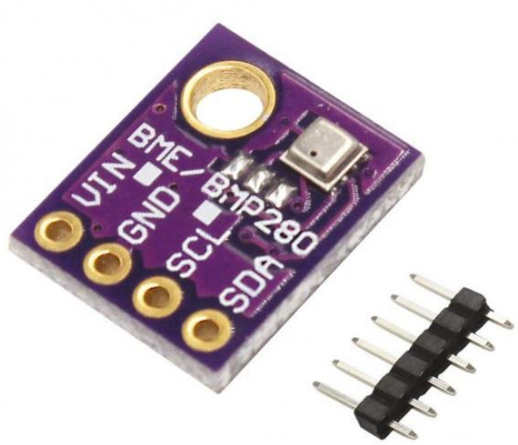



Figure 5.4 BME280

### 5.5 Ultrasonic sensorHC-SR04

I have already described the operation of the sensor HC-SR04 in chapter 2. Since it is quite accurate at close range, I use it as a minimum altitude control tool.

### 5.6 Additional components

Additional components of my project:

- red and yellow LEDs;
- accumulator with 5V output voltage (special for power servo drive);
- battery 6F22/9V (Krona) (for Arduino and other component power supply);
- 2 resistors with 1000-ohm resistance each, 1 resistor with 220 ohms;
- switch;
- connecting wires;
- Power Jack Cable (one side - Adapter 9V Krona, another side - Power jack plug).

### 5.7 Electrical connection scheme

A demonstrative electrical connection scheme is presented in Figure 5.5.

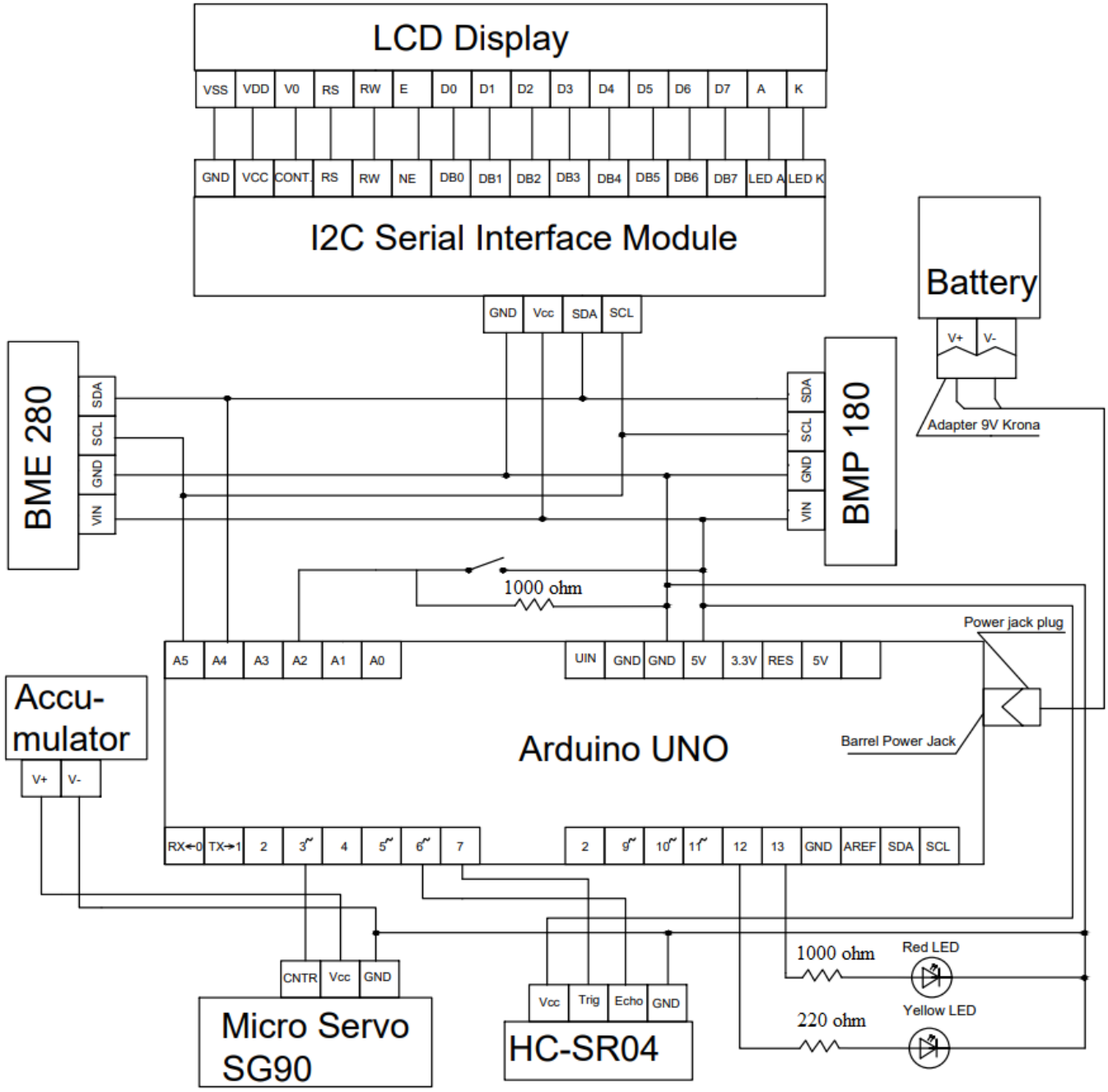



Figure 5.5. Electrical connection scheme of my project

# CHAPTER 6

## PROJECT SOFTWARE

In the development of the code, the Arduino Integrated Development Environment (IDE) was utilized. This IDE is a comprehensive development platform available for Windows, MacOS, and Linux operating systems. Developed primarily in C and C++ languages, the Arduino IDE is tailored for the creation and deployment of programs to not only Arduino-compatible boards but also to hardware from various other manufacturers.

For the programming language, C++ was selected, leveraging the structured coding framework that the Arduino IDE facilitates. C++ is renowned for its versatility in supporting multiple programming paradigms, including procedural, object-oriented, and generic programming. Its widespread applications span the development of operating systems, a diverse range of application software, device drivers, system utilities, high-performance server applications, and even complex computer games.

During the coding process, a selection of libraries was incorporated to enhance functionality. These included LiquidCrystal_I2C for LCD interfacing, Wire for communication over a data bus, Adafruit_BME280 and Adafruit_BMP085 for environmental sensing, and the Servo library for controlling servo motors. These libraries provided a robust foundation for the development and execution of the desired functionalities in the project. The entirety of the developed code, including detailed annotations and descriptions, is comprehensively documented in Appendix C of this document [Refer to Source 37 for detailed reference]. This appendix serves as a valuable resource for understanding the implementation specifics and the underlying logic of the code.

# CHAPTER 7

## DESCRIPTION OF THE OPERATIONAL MECHANISM OF THE ALTITUDE CONTROL SYSTEM

The altitude control system developed in this project is capable of acquiring both absolute and relative altitude data. This capability is achieved through the integration of two atmospheric pressure sensors, which are instrumental in determining the true altitude. However, it is important to acknowledge that sensor data inherently possesses a degree of noise, leading to potential inaccuracies in precision measurements. According to the technical specifications detailed in the datasheet of the BMP180 sensor, it can function as an altimeter within an accuracy range of approximately ±1 meter. The BME280 sensor exhibits similar accuracy characteristics in terms of altitude measurement.

To enhance the precision of these sensors and mitigate the impact of their inherent noise and error margins, a decision was made to employ a data fusion approach using a Kalman filter. This technique was informed by the comprehensive study presented in the conference paper by J.Z. Sasiadek and P. Hartana, titled “Sensor Data Fusion Using Kalman Filter” [38]. Their research offers a detailed exploration of several schematic models for the fusion of two sensors employing the Extended Kalman Filter (EKF). These models are elaborated upon in Figure 7.1 of their paper.

The rationale behind adopting the Kalman filter in this system is rooted in its ability to effectively process and integrate the data from both sensors, thereby yielding a more accurate and reliable altitude measurement. The Kalman filter algorithm operates by continuously estimating the state of a dynamic system (in this case, the altitude) and updating these estimates based on the incoming sensor data. By applying the principles and models outlined in Sasiadek and Hartana's paper, the system is able to reconcile the data from the BMP180 and BME280 sensors, achieving a more refined and dependable altitude reading. The implementation of this fusion technique is a critical aspect of the system’s operation, ensuring that the altitude data provided is both accurate and robust, suitable for applications where precision is paramount.

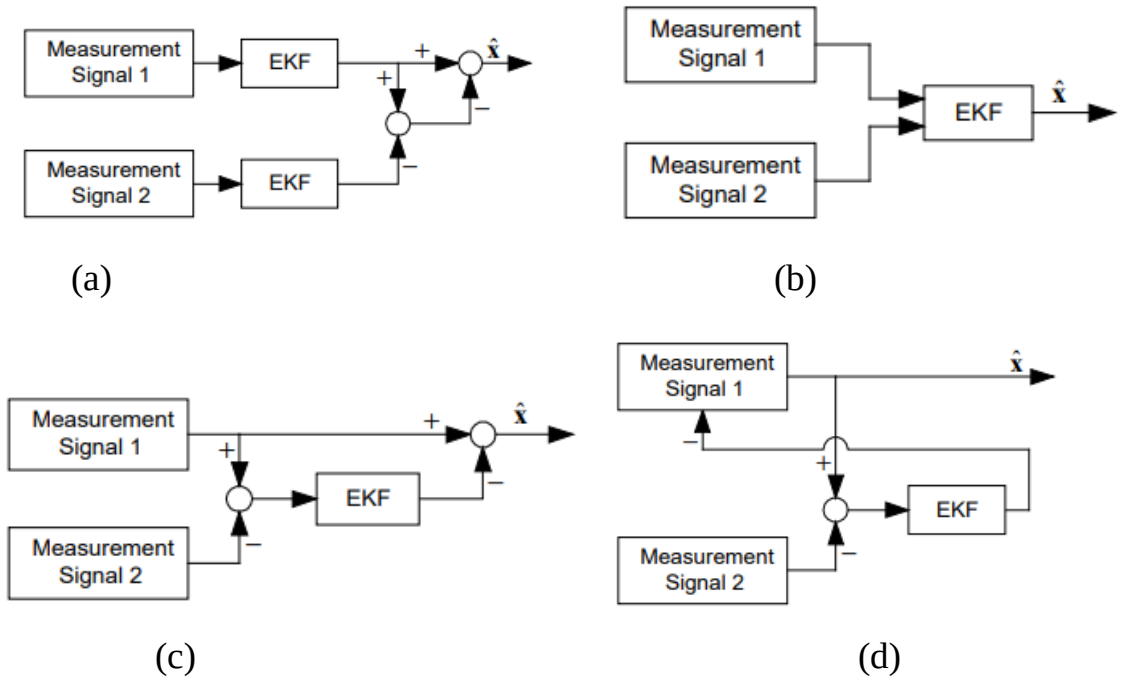



Figure 7.1: Schematic Representation of Various Sensor Data Fusion Methods

This figure illustrates distinct methodologies for the fusion of sensor data, including (a) Direct Pre-filtering scheme, (b) Direct Extended Kalman Filter (EKF) scheme, (c) Indirect Feedforward EKF scheme, and (d) Indirect Feedback EKF scheme.

Detailed Description of Sensor Data Fusion Methods:

Direct Pre-filtering Method: In this approach, the measurement signals from both sensors undergo a preliminary filtration process prior to their comparison. The primary objective here is to reduce the error in each signal before they are integrated. Following this filtration, the cumulative error derived from the signal correction is utilized to generate the final processed data. This method emphasizes initial signal purification to enhance overall data accuracy.

Direct EKF Method: The Direct EKF method is characterized by its ability to refine the measurement signal continuously, even if only a single measurement signal is available for correction. Over time, this process leads to an incrementally more precise measurement signal. However, a notable limitation of this method is the requirement for an established dynamic model of the sensor, which is crucial for accurate prediction of the sensor's position.

Indirect Feedforward EKF Method: This method involves a comparison of the signals before they are inputted into the EKF. The estimation error generated from this

comparison is then fed forward into one of the measured signals. This technique is particularly effective in aligning and correcting the data from the two sensors before it undergoes the EKF process.

Indirect Feedback EKF Method: The Indirect Feedback EKF method is similar in principle to the Indirect Feedforward EKF approach. The primary distinction lies in the application of the estimate error, which, in this case, is applied to the sensor's previous measurement. This method focuses on adjusting past data based on current error estimations to refine future measurements.

For the project, the Indirect Feedforward EKF Method was selected for fusing data from the two sensors. This choice was influenced by the method's relative independence from the state equation, which aligns well with the project's requirements, given that the state equation is not utilized in this context. The fusion process begins with the application of Maximum Likelihood Estimation (MLE) for combining data from the two sensors. MLE is a statistical method used for estimating the parameters of a model. In this context, MLE is employed to derive the most probable values of the parameters based on the observed data, thereby maximizing the likelihood function. The effectiveness of MLE in this setting is grounded in its ability to provide a robust statistical foundation for the fusion of data from the two distinct sensor sources, ensuring that the final altitude measurements are both precise and reliable. The estimation of the parameter by the maximum likelihood method is determined by the ratio:

$$\hat{X} = (H^T R_z^{-1} H)^{-1} H^T R_z^{-1} Z, \quad (7.1)$$

where $H = \begin{vmatrix} I \\ I \end{vmatrix}$ and $R_z = \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}$ correlation matrix of measurement errors for two sensors; $Z = \begin{vmatrix} Z_1 \\ Z_2 \end{vmatrix}$ matrix of our meashurement.

In non-matrix form, this expression has the following form:

$$\hat{X} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} Z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} Z_2 , \quad (7.2)$$

Where $\sigma^2$- mean squared error.

In my analysis, the determination of the mean squared error (MSE) was conducted in a manner analogous to the procedure delineated in Chapter 3 of this thesis. For the BMP180 sensor, the MSE, denoted as $\sigma^2$, was calculated to be 0.11, whereas for the BME280 sensor, it was found to be notably lower, at 0.06. This disparity in values leads to the inference that the BME280 sensor exhibits a reduced error margin compared to the BMP180.

In the pursuit of evaluating the mean, the mean squared error was also computed for the combined altitude value obtained from both sensors, resulting in a value of 0.07. Subsequently, this combined altitude value was subjected to a refinement process using the Kalman filter. During this phase, a state transition, specifically the vertical speed in this context, was incorporated. The computation of the vertical speed was executed in accordance with Formula 3.6. This formula necessitates the definition of distance as the differential between two successive combined altitude measurements, and a time interval was set at 0.3 seconds. The derived vertical speed was then also refined using the Kalman filter.

Following this stage, a comparative analysis was conducted between this filtered value and the data from the BME280 sensor, employing the methodology described previously. The decision to utilize the BME280 sensor for this comparison stems from its superior accuracy. Post-comparison, the mean squared error for the filtered value was calculated, yielding a result of 0.011.

Subsequently, the refined altitude value was then displayed on an LCD screen. Figure 7.2 in the thesis illustrates three distinct graphical representations of the signals: the red graph representing measurements from the BME280 sensor, the yellow graph depicting data from the BMP180 sensor, and the green graph indicating the final estimated values. An examination of these graphs reveals that the adopted method effectively mitigates noise interference and successfully amalgamates data from the two sensors. Notably, the final data tends to align more closely with readings from the BME280 sensor. This alignment is attributed to the implementation of Maximum Likelihood Estimation (MLE), which inherently biases the correction towards the more accurate sensor, in this case, the BME280. Furthermore, this bias is reinforced by the decision to utilize data from

the BME280 in the final comparative stage. A limitation of the current setup, as acknowledged, is the employment of the Kalman filter for data refinement without incorporating acceleration parameters.

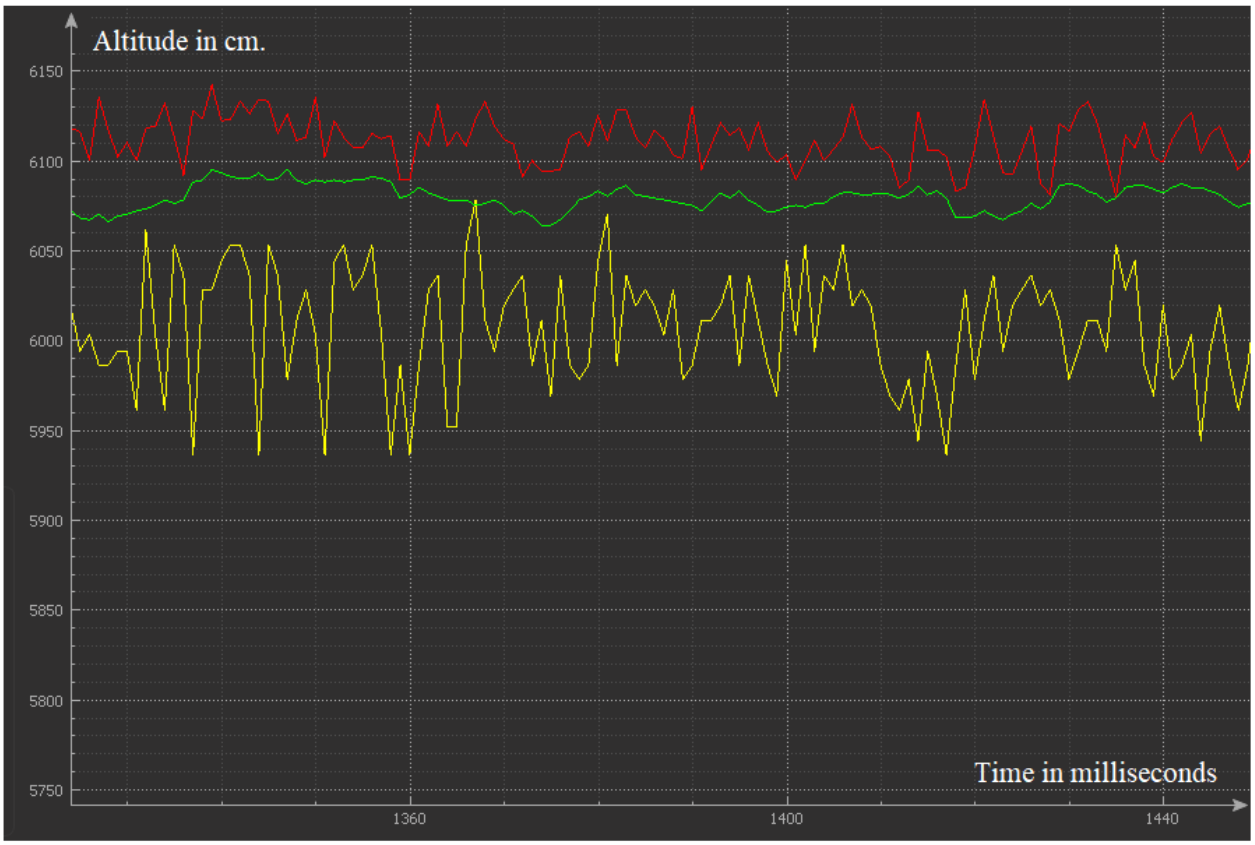



Figure 7.2. Result of Indirect feed forward EKF method of fusing sensor

In this project, an ultrasonic sensor was employed to ascertain the absolute altitude. Additionally, leveraging this sensor, a rudimentary autopilot system was developed, tailored specifically for an engaging and demonstrative presentation within an academic setting. The system's operational mechanics are such that when the ultrasonic sensor perceives the altitude to be below one meter, it activates a yellow LED. Further descent below half a meter triggers a red LED and actuates the servo motors, which respond proportionally to the decreasing altitude. This response mechanism is governed by a Proportional (P) controller. Moreover, the functionality to deactivate the autopilot system was also integrated into the design.

For visual representation of altitude data, an LCD display was utilized. However, it was observed that the computational limitations of the ATmega328 microcontroller

hindered the simultaneous display of both absolute and relative altitude measurements; specifically, it caused the servo motor to operate erratically. To address this issue, an automatic transition mechanism between the absolute and relative altitude readings was implemented. Upon system activation, both absolute and relative altitude measurement systems commence operation concurrently. When the absolute altitude measurement system detects an altitude lower than 1.6 meters, it automatically deactivates the relative altitude measurement system, and the display then solely presents the altitude data from the ultrasonic sensor. Conversely, when the altitude exceeds or equals 1.6 meters, the display reverts to showing the relative altitude.

This methodological approach has its merits and limitations. A notable drawback is the continuous operation of the ultrasonic sensor, which imposes additional demands on the power source. An alternative considered was to switch the altitude readings based on data from the barometric sensor. However, this method faces a challenge that the current system adeptly addresses: in scenarios involving flight over hilly or uneven terrain, there's a potential for altitude variations exceeding the switching threshold, which could result in the system failing to transition to the ultrasonic sensor readings that the autopilot relies on. This concern is particularly pertinent for systems designed to obtain absolute altitude data from sensors with limited measurement capabilities. In contrast, employing laser or radio altimeters, which offer significantly higher maximum measurement altitudes (e.g., ±50, 100 meters, or more), would render the option of switching from a barometric altimeter more advantageous and efficient.

# CONCLUSION

In conclusion, this thesis represents a comprehensive exploration into the development of an advanced altitude control system for remotely piloted aircraft systems (RPAS). The culmination of this research not only signifies a significant contribution to the field of aerial technology but also marks a step forward in the practical application of sophisticated control mechanisms in unmanned aerial vehicles.

Throughout this study, a multifaceted approach was adopted, encompassing both theoretical and practical aspects of altitude control in RPAS. The investigation began with an in-depth analysis of the historical evolution of altitude control systems, tracing their transition from bulky mechanical setups to the current state-of-the-art digital and compact systems. This historical context laid the foundation for understanding the complexities and challenges inherent in designing efficient and reliable altitude control systems for modern RPAS.

Central to this research was the design and implementation of a prototype altitude control system. This system was meticulously engineered, incorporating two Micro-Electro-Mechanical Systems (MEMS) pressure sensors and an ultrasonic sensor to ascertain altitude with high precision. The innovative use of the Indirect Feedforward Extended Kalman Filter (EKF) method to fuse data from these sensors addressed the critical issue of noise and error minimization, leading to more accurate altitude readings.

The implementation of a partial autopilot system, governed by a digital Proportional-Integral-Derivative (PID) controller, further demonstrated the practical applicability of the theoretical concepts explored in this study. The successful integration of these components underscores the potential of advanced control systems in enhancing the functionality and safety of RPAS.

Furthermore, this research contributes to the ongoing discourse in the field of RPAS technology by highlighting the importance of continuous innovation in sensor technology and data filtering methods. The findings of this study suggest that as sensors become more accurate and compact, and as new filtering techniques are developed, the potential for advancements in altitude control systems remains vast and promising.

In summary, the development and implementation of the altitude control system presented in this thesis not only achieves its intended objectives but also opens avenues for future research and development in RPAS technology. The knowledge and experience gained through this project are invaluable and contribute meaningfully to the broader goal of advancing the capabilities and applications of remotely piloted aircraft systems in various fields, ranging from commercial endeavors to critical military operations. As the demand for more sophisticated and reliable RPAS continues to grow, the findings of this thesis will undoubtedly play a pivotal role in shaping the future trajectory of this dynamic and rapidly evolving field.

**LIST OF REFERENCES**

[1]A remote-controlled drone prototype [Electronic resource] https://www.iwm.org.uk/history/a-brief-history-of-drones

[2] Information about SD-1 (1 source) [Electronic resource] https://www.designation-systems.net/dusrm/m-57.html

[3] Information about SD-1 (2 source) [Electronic resource] https://en.wikipedia.org/wiki/Radioplane_BTT

[4] Information about BGM-34 Firebee [Electronic resource] https://www.designation-systems.net/dusrm/m-34.html

[5] Information about Parrot AR.Drone [Electronic resource] https://ru.wikipedia.org/wiki/Parrot_AR.Drone

[6] Information about Kattering Bug [Electronic resource] https://en.wikipedia.org/wiki/Kettering_Bug

[7] Information about V-1 [Electronic resource] https://en.wikipedia.org/wiki/V-1_flying_bomb

[8] Information about North American GAM-77/AGM-28 Hound Dog [Electronic resource] https://www.designation-systems.net/dusrm/m-28.html

[9] Definition from ICAO Dog [Electronic resource] https://skybrary.aero/articles/introduction-remotely-piloted-aircraft-systems-rpas

[10] Clasification of Remotely Piloted Aircraft SystemDog [Electronic resource]https://www.droneswithoutborders.org/post/whats-in-a-name

[11] Eros Giorgi; Politecnico di Torino. Master Thesis Implementation of an autopilot for UAV/UGV systems based on Android smartphone (2019).

[12] Pressure sensor [Electronic resource]http://kontech-system.com.ua/articles/datchiki-davlenija-tipy-harakteristiki-osobennosti-podbor/

[13] Vorobiev V.G. and others - Aviation instruments, information measuring systems and complexes (1992).

[14] Disadavantages of pressure sensor [Electronic resource] sensors https://www.eastsensor.com/blog/barometric-pressure-sensor/

[15] Information about Bmp180 https://pdf1.alldatasheet.com/datasheet-pdf/view/1132068/BOSCH/BMP180.html

[16] I²C [Electronic resource] https://en.wikipedia.org/wiki/I%C2%B2C

[17] EEPROM [Electronic resource] https://en.wikipedia.org/wiki/EEPROM

[18] History of ultrasonic sensor [Electronic resource] https://www.ob-ultrasound.net/ultrasonics_history.html

[19] Formula for ultrasonic sensor [Electronic resource] https://www.bananarobotics.com/shop/HC-SR04-Ultrasonic-Distance-Sensor

[20] Information about HC-SR04 [Electronic resource] (1 source) https://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032189/pdf

[21] Information about HC-SR04 [Electronic resource] (2 source) https://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032189/pdf

[22] History of radio altimeter [Electronic resource]https://www.aviationpros.com/engines-components/aircraft-airframe-accessories/cabin-communications/article/10387134/radio-altitude-the-instrument-of-choice

[23] GPS [Electronic resource] https://www.advancednavigation.com/tech-articles/global-navigation-satellite-system-gnss-and-satellite-navigation-explained/

[24] GPS errors https://ggc.org.au/docs/operations/knowledge/Altitude%20Measurement.pdf

[25] General about LIDAR[Electronic resource] https://www.realpars.com/blog/laser-sensor

[26] Classification and formulas for laser rangefinder [Electronic resource] https://www.apogeeweb.net/electron/The-Function-And-Principle-Of-Laser-Sensor.html

[27] Lidar characteritics [Electronic resource] https://www.documents.lightware.co.za/LW20%20-%20LiDAR%20Manual%20-%20Rev%2012.pdf

[28] Filters general https://onlinedocs.microchip.com/pr/GUID-2607C185-2590-41FF-8911-0DAD5D1E3F69-en-US-5/index.html?GUID-CA58615E-6E62-497A-84BB-E1F02848A720

[29] Median filter https://www.edn.com/median-filters-an-efficient-way-to-remove-impulse-noise/

[30] Infinite Impulse Response (IIR) Filters: Ian Grout, in Digital Systems Design with FPGAs and CPLDs, (2008).

[31] Kalman general: Saeed V. Vaseghi, book “Advanced Digital Signal Processing and Noise Reduction” (2009).

[32] Kalman filter algorithme https://thekalmanfilter.com/kalman-filter-explained-simply/

[33] Kalman filter formals: Alex Backer, book “Kalman filter from the ground up” (2023).

[34] History autopilot https://en.wikipedia.org/wiki/Autopilot

[35] Feedback_Loops https://en.wikibooks.org/wiki/Control_Systems/Feedback_Loops

[36] Arduino Uno https://doc.arduino.ua/ru/hardware/Uno

[37] C++ https://en.wikipedia.org/wiki/C%2B%2B

[38] Conference Paper authored by J.Z. Sasiadek and P. Hartana “Sensor Data Fusion Using Kalman Filter”(2000).

[39] EASA PART 66 Module 5

[40] Council Directive 89/656 / EEC of 30 November 1989 on the minimum safety and health requirements for the use of personal protective equipment by employees at work (third separate Directive within the meaning of Article 16 (1) of Directive 89/391 / EEC).

[41] Order No. 1804 of 11/29/2018 is valid from 01/15/2019 published by Ministry of Social Policy of Ukraine. On approval of the Minimum Safety and Health Requirements when employees use personal protective equipment at the workplace.

[42] José LOPES ESTEVES, doctoral dissertation “Electromagnetic interference and information security:characterization, exploitation and forensic analysis” (2023)

[43] Information about electromagnetic effect on marine animals https://web.uri.edu/offshore-renewable-energy/ate/how-do-electromagnetic-fields-affect-marine-animals/

[44] JAR145.25(c) and JAR145 AMC145.25

Estimate 3

$r := 0.018$

$z := 172.49$

$p\_previous := 0.01163$

$x\_previous := 172.392$

$K := \frac{p\_previous}{p\_previous + r} = 0.393$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.43$

$p\_estimate := (1 - K) \cdot p\_previous = 7.065 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 8.065 \times 10^{-3}$

Estimate 4

$r := 0.018$

$z := 172.38$

$x\_previous := 172.43$

$p\_previous := 0.00807$

$K := \frac{p\_previous}{p\_previous + r} = 0.31$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.415$

$p\_estimate := (1 - K) \cdot p\_previous = 5.572 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 6.572 \times 10^{-3}$

Estimate 5

$r := 0.018$

$z := 172.60$

$x\_previous := 172.483$

$p\_previous := 0.00657$

$K := \frac{p\_previous}{p\_previous + r} = 0.267$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.464$

$p\_estimate := (1 - K) \cdot p\_previous = 4.813 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 5.813 \times 10^{-3}$

Estimate 6

$r := 0.018$

$x\_previous := 172.464$

$z := 172.43$

$p\_previous := 0.00581$

$K := \frac{p\_previous}{p\_previous + r} = 0.244$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.456$

$p\_estimate := (1 - K) \cdot p\_previous = 4.392 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 5.392 \times 10^{-3}$

Estimate 7

$r := 0.018$

$x\_previous := 172.456$

$z := 172.78$

$p\_previous := 0.00539$

$K := \frac{p\_previous}{p\_previous + r} = 0.23$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.531$

$p\_estimate := (1 - K) \cdot p\_previous = 4.148 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 5.148 \times 10^{-3}$

Estimate 8

$r := 0.018$

$x\_previous := 172.531$

$z := 172.60$

$p\_previous := 0.00515$

$K := \frac{p\_previous}{p\_previous + r} = 0.222$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.546$

$p\_estimate := (1 - K) \cdot p\_previous = 4.004 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 5.004 \times 10^{-3}$

Estimate 9

$r := 0.018$

$x\_previous := 172.546$

$z := 172.63$

$p\_previous := 0.005$

$K := \frac{p\_previous}{p\_previous + r} = 0.217$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.564$

$p\_estimate := (1 - K) \cdot p\_previous = 3.913 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 4.913 \times 10^{-3}$

Estimate 10

$r := 0.018$

$x\_previous := 172.564$

$z := 172.54$

$p\_previous := 0.0049$

$K := \frac{p\_previous}{p\_previous + r} = 0.214$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.559$

$p\_estimate := (1 - K) \cdot p\_previous = 3.852 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 4.852 \times 10^{-3}$

Estimate 11

$r := 0.018$

$x\_previous := 172.559$

$z := 172.31$

$p\_previous := 0.00485$

$K := \frac{p\_previous}{p\_previous + r} = 0.212$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.506$

$p\_estimate := (1 - K) \cdot p\_previous = 3.821 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 4.821 \times 10^{-3}$

Estimate 12

$r := 0.018$

$x\_previous := 172.506$

$z := 172.57$

$p\_previous := 0.00482$

$K := \frac{p\_previous}{p\_previous + r} = 0.211$

$x\_estimate := x\_previous + K \cdot (z - x\_previous) = 172.52$

$p\_estimate := (1 - K) \cdot p\_previous = 3.802 \times 10^{-3}$

$p\_predict := p\_estimate$

$p\_predict := p\_estimate + 0.001 = 4.802 \times 10^{-3}$

# Appendix B

```
#include<Wire.h>
#include<Adafruit_Sensor.h>
#include<Adafruit_BME280.h>
#defineSEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme; //
float varVolt=0.06;// average deviation
float varProcess=0.001 ;//speed of reaction to changes
float Pc=0.0;
float K=0.0;
float P=0.1;
float Xp=0.0;
float Xe=0.0;
voidsetup(){
  Serial.begin(9600);
  if(!bme.begin(0x76)){
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while(1);
  }
}
voidloop(){

  float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
  Serial.println(altitude);
  float fil_alt=filter(altitude);

  Serial.println("$");
  Serial.print(altitude);
  Serial.println(" ");
  Serial.println(fil_alt);
  Serial.println(";");
delay(50);
}
  floatfilter(floatalttotal){
  Pc=P+varProcess;
  K=Pc/(Pc+varVolt);
  P=(1-K)*Pc;
  Pc=P;
  Xp=Xe;
  Xe=K*(alttotal-Xp)+Xp;
  return Xe;
 }
```

# Appendix C

```
#include<LiquidCrystal_I2C.h>//0x27 0x76 0x77
#include<Wire.h>
#include<Adafruit_BME280.h>
#include<Adafruit_BMP085.h>
#include<Servo.h>
constint trigPin = 7; // Trigger Pin of Ultrasonic Sensor
constint echoPin = 6; // Echo Pin of Ultrasonic Sensor
constint Ledpin1 = 12;
constint Ledpin2 = 13;
int servocontr;
float yled=1;
float rled=0.5;
long duration;
int distance;
Servo servo1;
float varalt2=0.11;
float varalt1=0.06;
float varcom=0.07;
float varfilt=0.011;
float noise=0.001;
float Pc=0.0;
float K=0.0;
float P=0.1;
float Xp=0;
float Xpr=100.00;
float Xe=0.0;
float Pc1=0.0;
float K1=0.0;
float P1=0.1;
float Xp1=0;
float Xe1=0.0;
int angle1=0;
int angle2=80;
float pr_grnd=1013.00;
#defineSEALEVELPRESSURE_HPA (pr_grnd)
Adafruit_BMP085 bmp;
Adafruit_BME280 bme;
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
int switchpin=A2;
int sw;


voidsetup(){
  pinMode(Ledpin1, OUTPUT);
  pinMode(Ledpin2, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(switchpin, INPUT);
```

```
  Serial.begin(9600); // Starting Serial Terminal
  lcd.init();
  lcd.backlight();
  servo1.attach(3);
  if(!bme.begin(0x76)){ // BME280 address may be different (0x76 or 0x77) depending on
your module
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while(1);
  }

 if(!bmp.begin(0x77)){
    Serial.println("Could not find a valid BMP085 sensor, check wiring!");
    while(1){}
  }
}

voidloop(){

  delay(2);
  sw=analogRead(switchpin);

    //Ultrasonic sensor
    digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.34 / 2;
  // Prints the distance on the Serial Monitor
 float alt=distance*0.001;
 delay(5);
  if  (alt<=1.75){
     // LCD
  lcd.setCursor(0, 0);
  lcd.print("Absolute altitude");
  lcd.setCursor(3, 1);
  lcd.print(alt);
  lcd.setCursor(7, 1);
  lcd.print("   ");
  lcd.setCursor(10, 1);
  lcd.print("m.");

  if(sw<500){
  //indication
  int ledval;
  ledval=(255./1023.)*850.;
```

```
   if(alt<yled)analogWrite(Ledpin1, ledval);
   if(alt>yled)analogWrite(Ledpin1, 0);
   if(alt<rled)analogWrite(Ledpin1, 0);
   if(alt<rled)analogWrite(Ledpin2, ledval);
   if(alt>rled)analogWrite(Ledpin2, 0);
   delay(2);
//Servodrive

   int ralt=rled*1000;
   int servocontr1=(distance);
   servocontr1=map(servocontr1, 0, ralt, angle1, angle2);
   if(distance>ralt)servo1.write(angle2);
   if(distance<ralt)servo1.write(servocontr1);
  delay(2);
  }else{digitalWrite(Ledpin1, 0);
  digitalWrite(Ledpin2, 0);
  servo1.write(angle2);
  lcd.setCursor(0, 2);
  lcd.print("                   ");
  lcd.setCursor(3, 3);
  lcd.print("       ");
  lcd.setCursor(10, 3);
  lcd.print("      ");
  delay(10);
  }

  }else{
    //Pressure sensor1
    float altitude1=bme.readAltitude(SEALEVELPRESSURE_HPA);
   //Pressure sensor2
    float pr_grnd1=pr_grnd*100-100;
    float altitude2=bmp.readAltitude(pr_grnd1);
    delay(300);

    float altitude11=bme.readAltitude(SEALEVELPRESSURE_HPA);
   //Pressure sensor2
    float pr_grnd2=pr_grnd*100-100;
    float altitude22=bmp.readAltitude(pr_grnd2);

   // Indirect feed forward KF method of fusing sensor data
   float
comalt=(varalt2/(varalt2+varalt1))*altitude1+(varalt1/(varalt1+varalt2))*altitude2;
   float
comalt2=(varalt2/(varalt2+varalt1))*altitude11+(varalt1/(varalt1+varalt2))*altitude22;
   float versp=(comalt2-comalt)/0.3;
   float versp_filt=filtervel(versp);
   float filt_com=filter(comalt2)+versp_filt*0.3;
   float
alttotal=(varalt1/(varalt1+varfilt))*filt_com+(varfilt/(varfilt+varalt1))*altitude1;
   delay(200);
```

```
   Serial.println("$");
  Serial.println(altitude1);
  Serial.println(" ");
  Serial.println(altitude2);
  Serial.println(" ");
  Serial.println(alttotal);
  Serial.println(";");
   // LCD
  lcd.setCursor(0, 0);
  lcd.print("True altitude          ");
  lcd.setCursor(3, 1);
  lcd.print(alttotal);
  lcd.setCursor(10, 1);
  lcd.print("m.");
  delay(10);
  lcd.setCursor(0, 2);
  lcd.print("Vertical speed");
  lcd.setCursor(3, 3);
  lcd.print(versp_filt);
  lcd.setCursor(10, 3);
  lcd.print("m/s.");
  delay(10);
}
}
//sismplest Kalman
floatfiltervel(floatversp){
  Pc1=P1+noise;
  K1=Pc1/(Pc1+varalt2);
  P1=(1-K1)*Pc1;
  Pc1=P1;
  Xp1=Xe1;
  Xe1=K1*(versp-Xp1)+Xp1;
  return Xe1;
 }
    floatfilter(floatcomalt2){
  Pc=P+noise;
  K=Pc/(Pc+varcom);
  P=(1-K)*Pc;
  Pc=P;
  Xp=Xe;
  Xe=K*(comalt2-Xp)+Xp;
  return Xe;
 }
```