

DEPARTMENT OF INFORMATION TECHNOLOGY AND  
ELECTRICAL ENGINEERING

Spring Semester 2023

# Commissioning of a Novel Radar Speed Sensor for a Hyperloop Prototype

Bachelor Thesis

Mohamed Sadfi  
msadfi@student.ethz.ch

March 2023

Supervisors: Dr. Michele Magno, michele.magno@pbl.ee.ethz.ch

Professor: Prof. Dr. Florian Dörfler, dorfler@ethz.ch

# Acknowledgements

I would like to express my sincere gratitude to the entire PBL team for providing the administrative framework that facilitated the completion of my thesis. I am especially thankful to Dr. Michele Magno for his guidance and invaluable advice throughout my bachelor project.

I am immensely grateful to the entire Swissloop team, ranging from the dedicated focus students to the esteemed board and engineering leads. The opportunity to undertake my bachelor thesis within this organization has been transformative, enabling me to acquire invaluable skills and knowledge. I would like to extend my heartfelt appreciation to Marvin Steinkellner for his initial assistance in familiarizing me with the Swissloop environment, Philip Wiese for his numerous advice and unwavering support that helped me overcome obstacles and push the boundaries of my project, Carl Brander for his outstanding organizational skills and guidance throughout the thesis process, Hanno Hiss for designing the highly functional PCB that played a crucial role in my research, and all the other team members who warmly welcomed me into the Swissloop family.

The support, mentorship, and collaboration from each of these individuals and the wider Swissloop team have played an instrumental role in the successful completion of my bachelor thesis. I am truly grateful for their contributions, which have not only enriched my academic journey but have also inspired me to pursue further growth and excellence in my future endeavors.

# Abstract

This thesis addresses the integration of a novel radar speed sensor into the vehicle control unit of a hyperloop prototype. The hyperloop technology requires highly accurate speed measurement with minimal latency, making it crucial to develop a firmware solution that overcomes these challenges. The firmware captures the analog frequency signal from the Doppler sensor and converts it into relative speed information, while simultaneously digitizing the accelerometer's analog voltage signal for accurate acceleration measurements. A Kalman filter is implemented to fuse these data and eliminate noise, resulting in precise speed readings. The successful implementation of the firmware demonstrates the sensor's viability for integration into the hyperloop prototype. However, further improvements are needed to optimize the computational efficiency of the microcontroller, address resource constraints, and enhance noise filtering and accuracy. This work lays the foundation for integrating the novel radar speed sensor, showcasing its potential to revolutionize transportation. Future research and development will focus on optimizing computational efficiency and advancing the capabilities of radar-based speed sensing technologies in hyperloop systems and beyond.

# Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor. For a detailed version of the declaration of originality, please refer to Appendix [B](#)

Mohamed Sadfi,  
Zurich, March 2023

# Contents

<b>List of Acronyms</b>	<b>ix</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Swissloop . . . . .	2
1.3. Objective . . . . .	2
1.3.1. Outline . . . . .	3
<b>2. Related Work</b>	<b>4</b>
2.1. STM32 Microcontroller Unit . . . . .	4
2.2. Speed Sensors . . . . .	5
2.2.1. Electromagnetic Sensors . . . . .	5
2.2.2. Hall-Effect Sensors . . . . .	6
2.2.3. Accelerometers . . . . .	8
2.2.4. Laser Sensors . . . . .	9
2.2.5. GPS sensors . . . . .	10
2.3. Swissloop case . . . . .	11
2.3.1. VCU speed sensing on the pod . . . . .	12
2.4. Comparative Analysis . . . . .	13
<b>3. Theory</b>	<b>16</b>
3.1. Nucleo H723ZG . . . . .	16
3.1.1. STM32CubeMX . . . . .	17
3.1.2. Analog to Digital Converter . . . . .	17
3.1.3. Timers . . . . .	18
3.2. Doppler Effect . . . . .	19
3.3. Kalman Filter . . . . .	21

## Contents

<b>4. Hardware / Firmware / Algorithm Implementation</b>	<b>24</b>
4.1. Hardware	24
4.1.1. DOPRail Sensor	24
4.1.2. Phase Sensor PCB	27
4.1.3. GAM900 Accelerometer	28
4.2. Firmware	29
4.2.1. Sensors Reading	30
4.2.2. Sensors Fusion via Kalman Filtering	32
<b>5. Results</b>	<b>34</b>
5.1. Firmware Achievements	34
<b>6. Discussion</b>	<b>37</b>
6.1. Issues with the current design	37
6.1.1. Integration on the Vehicle Control Unit	37
6.1.2. Firmware efficiency	38
6.2. Improvements	38
6.2.1. Computational efficiency	38
6.2.2. Hardware	39
6.2.3. Integration of other sensors in the Kalman Filter	39
<b>7. Conclusion and Future Work</b>	<b>41</b>
<b>A. Task Description</b>	<b>43</b>
<b>B. Declaration of Originality</b>	<b>49</b>
<b>C. File Structure</b>	<b>51</b>
<b>Glossary</b>	<b>52</b>

# List of Figures

2.1. Electromagnetic Sensor . . . . .	6
2.2. Illustration of the Hall Voltage[1] . . . . .	7
2.3. RPM detector [2] . . . . .	8
2.4. Lidar Detector[3] . . . . .	10
2.5. GPS [4] . . . . .	11
2.6. Swissloop Sensors . . . . .	12
2.7. Light gates position measurement on the track . . . . .	13
3.1. NUCLEO H723ZG [5] . . . . .	16
3.2. Digital conversion of a sinus signal with different resolutions[6] . . . . .	18
3.3. Illustration of the Doppler Effect[7] . . . . .	19
3.4. Schematic of the DOPRail on the pod . . . . .	20
3.5. Kalman Filter flowgraph[8] . . . . .	23
4.1. DOPRail Sensor . . . . .	25
4.2. Setup for the DOPRail testing . . . . .	25
4.3. Reflecing surface : velcro strap 65 mm, height : 9 cm, Angle : 75° . . . . .	26
4.4. Reflecing surface : Carpet, height : 9 cm, Angle : 75° . . . . .	26
4.5. Phase sensor PCB . . . . .	27
4.6. Triggered signal . . . . .	28
4.7. GAM900 sensor[9] . . . . .	29
5.1. Linear data . . . . .	35
5.2. Parabolic Data . . . . .	36
6.1. Swissloop Kalman Architecture . . . . .	40

# List of Tables



# List of Acronyms

ADC	. . . . .	Analog to Digital Converter
API	. . . . .	Application Programming Interface
CAN	. . . . .	Control Area Network
CPU	. . . . .	Central Processing Unit
GPS	. . . . .	Global Positioning System
HAL	. . . . .	Hardware Abstraction Layer
IC	. . . . .	Integrated Circuit
IIS	. . . . .	Integrated Systems Laboratory
MCU	. . . . .	Micro Controller Unit
OpAmp	. . . . .	Operational Amplifier
PBL	. . . . .	Center for Project-Based Learning
PCB	. . . . .	Printed Circuit Board
PDF	. . . . .	Portable Document Format
PWM	. . . . .	Pulse Width Modulation
RPM	. . . . .	Rotations Per Minute
VCU	. . . . .	Vehicle Control Unit

# Introduction

## 1.1. Motivation

In the ever-evolving landscape of transportation, hyperloop technology stands out as a transformative innovation with the potential to revolutionize how we travel. With its promise of unparalleled speed and energy efficiency, hyperloop systems have captured the imagination of researchers and industry experts worldwide. However, the successful realization of hyperloop prototypes necessitates the integration of advanced sensor technologies to address critical operational challenges.

One such challenge is the accurate and reliable measurement of speed within the hyperloop environment. As the speed of the passenger pods reaches extraordinary levels, precise speed monitoring becomes paramount for ensuring passenger safety, system efficiency, and optimal performance. Traditional speed measurement methods may prove insufficient in this unique and demanding context.

This is where the significance of this thesis comes into play. Focusing specifically on the commissioning of a novel radar speed sensor for a hyperloop prototype, this research addresses the pressing need for an advanced sensing solution tailored to the hyperloop environment. By developing and implementing this state-of-the-art radar speed sensor, the thesis aims to overcome the existing limitations and provide a reliable and accurate speed measurement solution for hyperloop systems.

The outcomes of this research will hopefully pave the way for far-reaching advancements. By solving the speed measurement problem, the thesis contributes to the broader goal of advancing hyperloop technology, paving the way for its practical implementation. The novel radar speed sensor could not only enhance the safety and efficiency of hyperloop prototypes but also serve as a stepping stone towards the realization of functional

## 1. Introduction

hyperloop transportation systems on a larger scale. Ultimately, this research strives to accelerate the progress of hyperloop technology and usher in a new era of high-speed, sustainable transportation.

### 1.2. Swissloop

The Swissloop student group, affiliated with the Swiss Federal Institute of Technology (ETH) in Zurich, is a remarkable team of passionate and talented individuals dedicated to advancing innovative technologies in the field of transportation. Comprised of students from various disciplines, including engineering, computer science, and business, Swissloop aims to push the boundaries of transportation systems through cutting-edge research and development. With a strong focus on hyperloop technology, the group has participated in prominent international competitions, demonstrating their expertise and commitment to driving progress in sustainable and high-speed transportation. Through collaboration, creativity, and a relentless pursuit of excellence, the Swissloop student group continues to inspire and pave the way for the future of transportation.

### 1.3. Objective

Of course, aiming for breakthrough findings that will revolutionize the hyperloop industry is ambitious in a context of a bachelor's thesis. The main goal of this project is to commission a working speed sensor that will be later integrated on the Swissloop's vehicles to improve the velocity measurements' accuracy.

This thesis was split into four main phases to reach a well functioning speed radar.

1. The first objective was to get familiar with all pieces of hardware needed for the thesis, which include the **DOPRail** sensor from Haslerrail, the **GAM900** accelerometer from Baumer, the **Nucleo H723ZG** development board from STM32, and the **inverting Schmitt-trigger PCB**.
2. The next step was to develop the firmware for the DOPRail and the GAM900 sensors, in particular to process the analog signals coming out of them and get the corresponding velocity and acceleration values. The computation would be done on the STM32 H723ZG MCU to be compatible with the MCU used by Swissloop.
3. Once the speed and acceleration measurements were obtained, we could start implementing the **Kalman filter** to combine the sensors in a dynamical linear model and filter out the noise inherent to the sensors.
4. The final phase mainly focused on testing the developed firmware and code and adapting them to ease the integration of the thesis on the Swissloop's pods over the next few years.

## *1. Introduction*

### **1.3.1. Outline**

As we split the project into two main parts:

1. Reading the analog signals of the **DOPRail** and the **GAM900**
2. Implementation of the Kalman filter to improve the speed measurement's accuracy

# Chapter 2

## Related Work

The first task basically consisted of getting familiar with all the tools that were going to be used during the thesis. A number of research studies have investigated the use of radar speed sensors for hyperloop applications, with a focus on developing novel sensor configurations, evaluating sensor performance under different operating conditions, and analyzing the impact of sensor errors on hyperloop performance. Therefore, there were a plethora of sources that helped me understand the speed measurements using the radar sensor.

Furthermore, in addition to investigating the speed sensors, some additional research about the development tools, on which the implementation of the software needed to be done, was necessary, namely all the embedded systems and the specific tools linked to them.

In this section, I am going to summarize some of the related work that was studied beforehand to get a better understanding of the subject and also present what has already been done in terms of speed sensing in the context of hyperloop technology.

Additionally, I will investigate what the Swissloop's teams have been using for speed measurements on the previous pods.

### 2.1. STM32 Microcontroller Unit

A **MicroController Unit** (MCU) can be viewed as a compact computer that enables us to do various computational tasks. In the past decades, a growth in interest in them has been noticeable due to major improvements in their energy efficiency and computational power. We often use microcontrollers in embedded applications, more specifically in automatically controlled applications where there is no need for a user's interaction. Compared to microprocessors, MCUs are much cheaper and smaller. The key difference between a microcontroller and a microprocessor is that a microcontroller integrates the

## 2. Related Work

CPU, memory, and peripheral functions on a single chip, while a microprocessor requires external chips to handle peripheral communication. This makes microcontrollers more suitable for low-power, low-cost, and real-time applications, while microprocessors are better suited for high-performance computing tasks.

The components that can be found in a microcontroller chip can differ depending on which type of microcontroller you choose, but they all have some common architecture. It typically includes a CPU, memory, and various peripherals, such as input/output ports, timers, and analog-to-digital converters [10].

There are a plethora of microcontroller types depending on the characteristics specific to each use case. If the task needs to wait for an event for a long time to perform a relatively simple task, one would prefer low-power MCUs with low clock frequencies to minimize the power consumption. If the application consists of computationally expensive operations and real-time signal processing, we would opt for a high-clock-frequency microcontroller to improve the computational capacity. There are still plenty of parameters that we can change in a microcontroller chip to adapt it at its best to our specific use cases; that is what makes the microcontroller really powerful and flexible.

In this thesis, the STM32 nucleo development boards were used. The rich documentation of [ST Microelectronics](#) on those boards and the many functionalities already integrated on them were very useful to get started with programming the MCU[11].

## 2.2. Speed Sensors

The [Doprail sensor](#) uses the properties of the Doppler effect to measure the speed of the vehicle. Alternatively, there are other types of speed sensors that rely on other physical phenomena to get velocity measurements. To stay as relevant as possible, we will mainly focus on the sensors used in the context of railway applications. Speed sensing can also be done indirectly by calculating the time it takes to travel a specific distance. While other sensors will calculate the speed from the position, some sensors depend on direct speed measurements. In the following subsections, we are going to explore some alternative methods for velocity measurements used in the railway and hyperloop industries.

### 2.2.1. Electromagnetic Sensors

Electromagnetic sensors rely on the fluctuations in the magnetic field to determine the speed of a rotating part of the vehicle. That could refer to a wheel or a gear, for example. Specifically, the sensor includes a coil wrapped around a magnet. As the gear's teeth pass the sensor, they cause fluctuations in the magnetic current, which generate a corresponding voltage in the coil. This voltage's frequency is directly proportional to the gear's rotational speed.

## 2. Related Work

Figure 2.1.: Electromagnetic Sensor

### 2.2.2. Hall-Effect Sensors

Hall effect sensors are named after Edwin Hall, an American physicist who made the discovery that electricity and magnetism can interact to induce motion in objects. These sensors leverage this principle by converting magnetically encoded information into electrical signals[12].

#### Hall Effect

The Hall effect basically describes the interactions between the moving electrons in a conductor and the magnetic field. Let's assume that we have a coil that conducts current. When no magnetic field is applied, the electron's path is not affected. Now if we apply a magnetic field  $\mathbf{B}$  across the conductor, the resulting Lorentz force, added to the force due to the electric field generated by the moving charges, will affect the trajectories of the electrons, and the electrons will tend to accumulate on one side of the conductor, therefore producing a voltage called **Hall emf**.

The Lorentz force will only appear when charge carriers are moving through a magnetic field. For excited electrons placed in a magnetic field  $\mathbf{B}$ , the total force  $\mathbf{F}$  acting on them is given by

$$\mathbf{F} = q * (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.1)$$

where  $q$  is the total charge,  $\mathbf{E}$  the electric field,  $\mathbf{v}$  the speed of the electrons.

From figure 2.3, we can notice that the Hall voltage will initiate a force on the electrons as opposed to the magnetic force. The build-up of charges on one side of the conductor will result in an increase in the Hall electric field magnitude until the electric force caused by the Hall emf completely cancels the Lorentz force, therefore reaching a steady state

## 2. Related Work

for the electrons. By setting  $\mathbf{F} = \mathbf{0}$  in equation 2.1 and knowing that

$$V_H = E_H * d$$

, where  $V_H$  is the Hall voltage,  $E_H$  is the electric field induced in the conductor due to the build-up of negative charges on one side, and  $d$  is the width of the conductor, we get the following expression for the Hall voltage[13]

$$V_H = v * B * d \quad (2.2)$$

In the figure below, we can see how a conducting plate will generate a voltage when put in a magnetic field  $\mathbf{B}$  orthogonal to it.

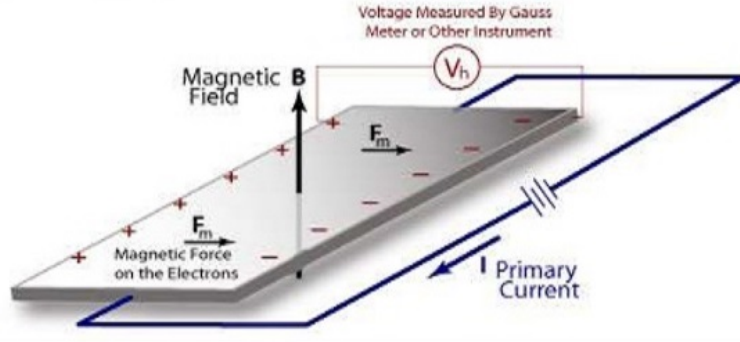


Figure 2.2.: Illustration of the Hall Voltage[1]

Hall sensors have found diverse applications in numerous industries and technologies, ranging from automotive and consumer electronics to industrial automation and medical devices. One of its application examples is speed sensing devices, also known as RPM detectors.

RPM detectors basically measure the rotational speed of rotating parts, for example, gears in a motor or wheels on a vehicle. The following schematic illustrates a simplified view of a RPM detector.



## 2. Related Work

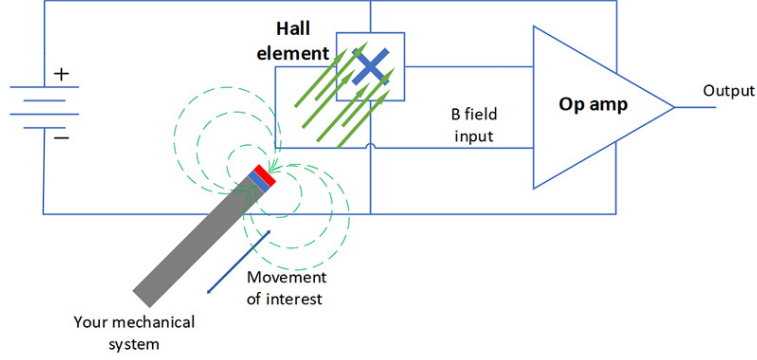


Figure 2.3.: RPM detector [2]

The Hall element can be roughly seen as the conductive plate of figure 2.3. It is going to detect the change in the magnetic field due to the movement of our mechanical system and generate a corresponding voltage, which we will amplify and output. This voltage can then be used in different ways to describe the movement of interest. Often magnets are mounted on the mechanical system, for example, on the teeth of a gear, to detect each time a tooth passes by and then get the velocity of the gear[14].

For the hyperloop technology, Hall effect sensors are not commonly used because of the lack of rotating parts on most of the pods. Still, we could still think of a way to use them, for example, by placing magnets at regular intervals on the track and then measuring the time between two high signals. Hall-effect sensors are often used on hyperloops for other measurements. Magnetic fields are omnipresent on hyperloops for the levitation of the pods as well as their propulsion. Therefore, we need sensors to measure those magnetic fields, and Hall effect sensors can also be used for this purpose.

### 2.2.3. Accelerometers

An accelerometer measures the rate of change of the velocity (acceleration) of an object. It can detect the acceleration in various direction. To measure the acceleration, the accelerometer will measure the force exerted on a mass during a change of the velocity of the object and then derive the value of the acceleration from that measurement.

Although there are many various kinds of accelerometers, the MEMS (Micro-Electro-Mechanical Systems) accelerometer is one that is frequently seen in consumer devices and it is the one that we are using in this thesis.

The proof mass, a relatively small mass in use in MEMS accelerometer sensors, is attached to a frame by flexible beams or springs. The proof mass often stays stationary during acceleration because of inertia, while the frame and the rest of the device move. The proof mass is moved away from its equilibrium location by this relative motion.

## 2. Related Work

The accelerometer makes use of the *Piezoelectric Effect* or *Capacitance Sensing* to measure this displacement:

- *Capacitance Sensing*: The proof mass and the frame serve as a capacitor's plates in a capacitive accelerometer. The capacitance between the plates varies as the proof mass travels. This change in capacitance can be electrically detected and is proportional to the device's acceleration or displacement.
- *Piezoelectric Effect*: The proof mass of a piezoelectric accelerometer contains piezoelectric crystals. When the proof mass is moved, the crystals are subjected to mechanical stress, which results in an output voltage proportionate to the applied force or acceleration.

The accelerometer's output of displacement or voltage is then transformed into a useful electrical signal by an electronic circuit. This signal may be further processed or interpreted by the gadget to deliver acceleration data, orientation data, or to initiate particular actions based on movement patterns.

The acceleration caused by gravity (1 g) is commonly measured by accelerometers, unless it is corrected for. This is vital to keep in mind. The velocity and position changes of the object being measured can be calculated by examining the acceleration data over time, and it is also feasible to identify particular gestures or movements[15].

### 2.2.4. Laser Sensors

Laser speed sensors are devices that use laser technology to measure the speed and velocity of objects. They work by emitting a laser beam that reflects off the target and returns to the sensor. By measuring the distance and the time between two round trips, we can estimate the speed with high accuracy. Some laser sensors also use the **Doppler Effect** to calculate the speed by measuring the frequency shift between the emitted light wave and the received wave.

Laser sensors emit light waves of extremely high frequency, typically in the range of  $10^{14}$  Hz. With such high frequencies, high data rates can be achieved. The working principle has its roots in the classical Doppler sensing technique. A laser beam is directed at a small spot. When it hits the spot, the reflected light beam will be detected via a photo detector, which transforms the light into a voltage. The difference between the emitted frequency and the received frequency gives us the speed of our moving object[16].

Another way of using lasers for speed detection is the well-known Lidar detector, which is mostly used for traffic monitoring and also in the railway industry. Lidar speed sensors work by emitting a beam of light, typically in the form of a laser, and measuring the time it takes for that light to bounce back after it reflects off of an object in its path. By measuring the time it takes for the light to travel to the object and back, the sensor can determine the distance to the object. When the sensor is mounted on a moving vehicle, it can use this distance information to calculate the speed of the vehicle relative to the

## 2. Related Work

object it is measuring.

For example, if the sensor is pointed at a stationary object and the vehicle is moving towards that object, the distance measured by the sensor will decrease over time. By analyzing the change in distance over time, the sensor can calculate the speed of the vehicle. Lidar speed sensors can be very accurate and are commonly used in autonomous vehicles, robotics, and other applications where precise distance and speed measurements are required[3].

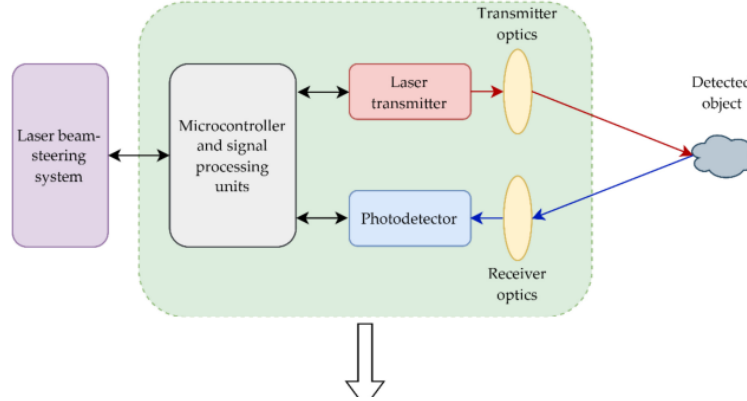


Figure 2.4.: Lidar Detector[3]

The technical characteristics of a laser detector can vary a lot. The most basic laser speed sensors have a range of some tens of meters, and the most advanced ones used in traffic monitoring can reach a target a couple of hundred meters away. The field of view is often pretty wide, sometimes  $360^\circ$ . Depending on the application, the field of view can be narrowed to improve the performance and accuracy of the sensor.

### 2.2.5. GPS sensors

Another widely used speed detection and position technology is the GPS (**G**lobal **P**ositioning **S**ystem). The U.S. Department of Defense (DoD) created GPS as a space-based system for positioning, navigation, and timing. It came about in the late 1960s and early 1970s when the Navy and Air Force combined their programs for timing and space-based navigation. GPS is commonly known as a satellite positioning or navigation system that offers signals for determining the location of people, vehicles, and other objects. It enables safe and efficient movement, measurement, and tracking from the earth's surface to geosynchronous orbit in space[4].

GPS sensors achieve extremely precise position detection and can also determine speed by measuring the distance traveled over a time interval. A GPS sensor will communicate with each GPS satellite. Those satellites will send their positions to the sensor. After

## 2. Related Work

the sensors get all the positions, it can determine its own position on earth. The image below gives an overview of how GPS works.

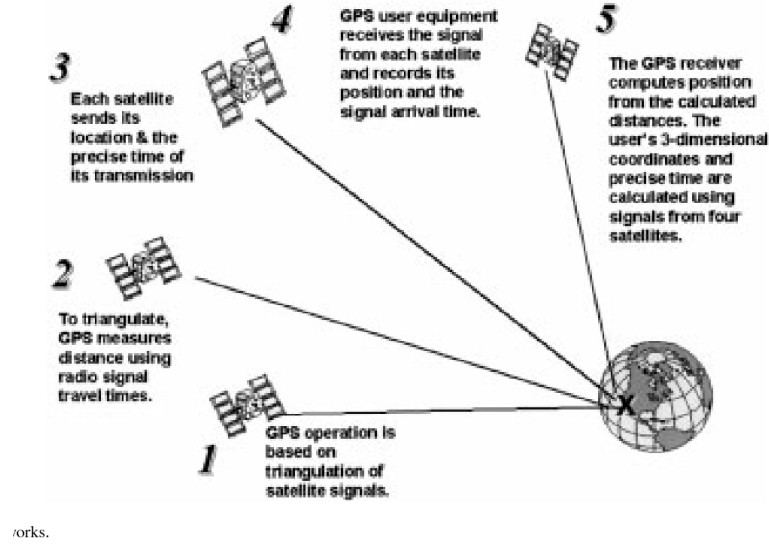


Figure 2.5.: GPS [4]

### 2.3. Swissloop case

The Swissloop's pod's sensor suite consists of a plethora of sensing devices. From magnetic field detection to battery management, they use a variety of sensors to keep track of every small event when running the pod. Each of those sensors uses different physical phenomena. Speed sensing is a crucial part of smooth functioning. Swissloop combines several speed and position sensors to measure the velocity of their pod as accurately as possible.

## 2. Related Work

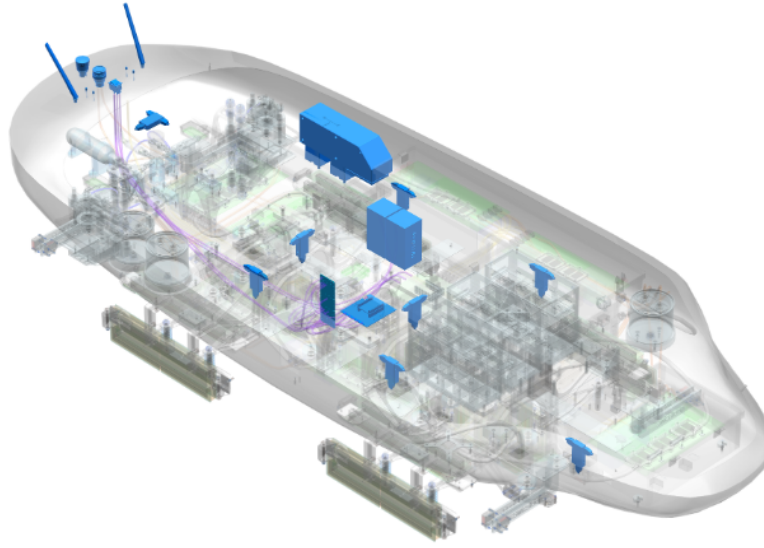


Figure 2.6.: Swissloop Sensors

The schematic above shows all the sensors that are connected to the vehicle control unit , including the inertial measurement units. In the next section, we are going to investigate those further.

### 2.3.1. VCU speed sensing on the pod

There are three speed sensors on the 2023 Swissloop prototype: the CORRail 2000 by Haslerrail, the STM accelerometer, and the O200 light gates. The light gates are, strictly speaking, position sensors. They measure the position of the pod on the track. The light gates are pointing towards the toothed surface of the track and are able to recognize the difference between the trough and crest of the surface. When we pass a tooth, we get a rising edge signal. When we pass a crest, we get a falling edge signal. The distance between each tooth and crest is constant; therefore, we can estimate the traveled distance thanks to the signal received. The figure below illustrates the light gates.

## 2. Related Work

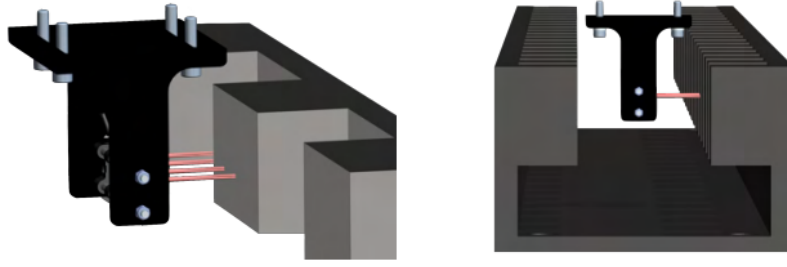


Figure 2.7.: Light gates position measurement on the track

We can get the speed by simply measuring the traveled distance over a time period. The CORRail 2000 is a speed sensor. It uses laser technology to measure the velocity of the pod. Its inconveniences come from its heavy weight and its inherent latency. In contrast to last year, this year's Swissloop uses an STM32 accelerometer, replacing the **GAM900**. The STM32 accelerometer is way smaller and lighter. It fits on the VCU PCB.

### 2.4. Comparative Analysis

All the inertial sensors mentioned above have advantages and disadvantages. The table below lists some of those pros and cons. The **DOPRail sensor** and the **GAM900** accelerometer, which were commissioned in this thesis, are also compared to the other sensors currently in use on the Swissloop's pod.

## 2. Related Work

Inertial Measurement Unit		
Sensor type	Advantages	Inconveniences
Electromagnetic Sensors & Hall effect Sensors	<ul style="list-style-type: none"> <li>-Accurate Measurements</li> <li>-Compact and Light</li> <li>-Cheap</li> </ul>	<ul style="list-style-type: none"> <li>-Possible interferences with the magnetic fields of the coils and the track</li> <li>-Not adapted for high velocities</li> </ul>
Accelerometers	<ul style="list-style-type: none"> <li>- Suitable for measuring dynamic changes in speed</li> <li>-Accurate Acceleration Measurements</li> <li>-Easy to implement</li> <li>-Widely used and researched</li> </ul>	<ul style="list-style-type: none"> <li>-Require Calibration</li> <li>-Very sensitive to vibrations, noise, etc. which can affect accuracy -</li> </ul>
Laser Sensors	<ul style="list-style-type: none"> <li>-Works well in a wide range of environments</li> <li>-Offers fast response time</li> </ul>	<ul style="list-style-type: none"> <li>-Higher cost compared to some other sensors</li> <li>-Requires a clear line of sight to the target object</li> <li>-Not suitable for measuring speed on non-reflective surfaces</li> </ul>
GPS Sensors	<ul style="list-style-type: none"> <li>-Provide accurate speed measurements based on satellite positioning.</li> <li>-Doesn't require physical contact with the ground or other objects</li> </ul>	<ul style="list-style-type: none"> <li>-Requires a clear line of sight to multiple GPS satellites for accurate measurements</li> <li>-Can be affected by signal interference or obstruction in urban environments or dense foliage</li> <li>-Might be affected by poor GPS reception</li> <li>-May have some delay in speed acquisition</li> </ul>

In the hyperloop industry, the laser sensors and the accelerometers may be more suited

## 2. Related Work

for accuracy and good latency. The magnetic and Hall effect sensors may be affected by the magnetic fields of the magnets on the vehicle, and the GPS is not really adapted because most hyperloops are designed to run underground in a vacuum tube.

We still have to compare the new sensors we are using in this thesis to the sensors that are in use this year on the pd

Swissloop's inertial sensors		
Sensor	Advantages	Inconveniences
CORRail 2000	<ul style="list-style-type: none"><li>-Wide range of measurement</li><li>-Good at low velocities and high velocities</li><li>-Direction detection</li></ul>	<ul style="list-style-type: none"><li>-Heavy weight</li><li>-Bad latency</li></ul>
Light Gate Trigger Sensor O200	<ul style="list-style-type: none"><li>-Accurate Measurements</li><li>-Easy to implement</li><li>-Latency sweet spot can be fined tuned</li></ul>	<ul style="list-style-type: none"><li>-Non-negligible latency at high velocities</li></ul>
STM32 Accelerometer	<ul style="list-style-type: none"><li>-Light weight</li><li>-Fits on the VCU PCB</li><li>-Much less noise</li></ul>	<ul style="list-style-type: none"><li>-Fragile</li></ul>
GAM900 Accelerometer	<ul style="list-style-type: none"><li>-Clean output due to integrated filter</li><li>-Analog or CAN FD output signal</li><li>-Three axis acceleration measurements</li></ul>	<ul style="list-style-type: none"><li>-Heavy weight</li><li>-Sensitive to noise and vibrations</li></ul>
DOPRail	<ul style="list-style-type: none"><li>-Good constant latency</li><li>-Very good accuracy</li><li>-Wide range of measurement</li></ul>	<ul style="list-style-type: none"><li>-Reflection on the track may vary</li><li>-Inaccurate at low velocities</li></ul>



# Theory

## 3.1. Nucleo H723ZG

The Nucleo H723ZG is a development board designed by STMicroelectronics that features an Arm Cortex-M7 **microcontroller**, specifically the STM32H723ZG. This board is part of the Nucleo family, which is a series of low-cost development platforms intended for prototyping and evaluating microcontroller-based designs. The Nucleo H723ZG board offers a range of features, such as a range of interfaces, including Ethernet, CAN, USB, and UART, making it an ideal platform for a wide variety of applications. The board also includes an ST-LINK debugger/programmer, which allows for easy debugging and programming of the microcontroller[5].

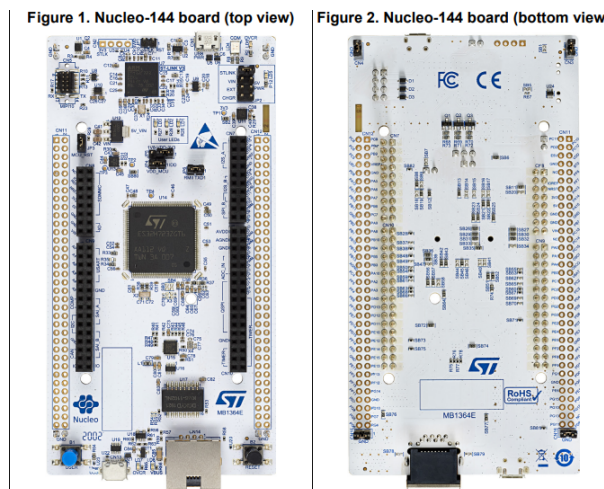


Figure 3.1.: NUCLEO H723ZG [5]

### 3. Theory

In addition to the many functions offered by the board, ST electronics provide the user with low-level drivers and APIs to ease the interaction and programming of the microcontroller. The HAL (Hardware Abstraction Layer) library is an embedded software layer that provides a higher level of abstraction for hardware peripherals. It is designed to maximize portability across different microcontroller families. STMicroelectronics provides a HAL library for its STM32 microcontroller families that offers high-level and feature-oriented APIs with a high-portability level. The HAL hides the MCU and peripheral complexity from the end-user[17][18].

As described above, the **Nucleo developer board** has a plethora of functionalities and strong computation power. In this thesis, we only used a small set of the features offered by the MCU. The main features used were the analog-to-digital converters (ADC) on the board and the timers. In the following sections, those devices are going to be investigated further, and the graphical software configuration tool STM32CubeMX will be briefly presented.

#### 3.1.1. STM32CubeMX

STM32CubeMX is a graphical software configuration tool developed by STMicroelectronics. It allows users to easily configure STM32 microcontrollers and microprocessors through a step-by-step process. The tool generates initialization C code for the Arm® Cortex®-M core or a partial Linux® Device Tree for the Arm® Cortex®-A core. This code can be used to initialize the microcontroller or microprocessor and set up its peripherals and middleware components. STM32CubeMX also provides pinout configuration, clock tree configuration, and power consumption calculation features[19].

The configuration process with CubeMX is guided and step-by-step, making it user-friendly. The project's microcontroller must be chosen first, of course. It is also possible to select nucleoboards by installing a few library extensions. By doing this, the configuration is completely adjusted to the kind of board we are using and all of its capabilities. The clock configuration and other MCU settings are, thanks to that, typically not a concern for us[19].

#### 3.1.2. Analog to Digital Converter

An ADC, or Analog-to-Digital Converter, is a device that takes an analog signal and converts it to a digital signal that can be processed and analyzed by digital circuits. The conversion process involves two main steps: sampling and quantization.

Sampling involves taking measurements of the analog signal at regular intervals, known as the sampling rate. The sampling rate is typically specified independently of the resolution and determines how often the ADC takes a sample of the analog input signal and converts it to a digital value. The more frequently the signal is sampled, the more accurately it can be reconstructed in the digital domain.

### 3. Theory

Quantization involves assigning a digital value to each sample based on its amplitude. The resolution of the ADC determines the number of bits used to represent each sample, which turn determines the number of possible digital values. For example, a 4-bit ADC can represent 16 different values, ranging from 0 to 15.

However, if the input voltage range is between 0 V and 2 V, and the ADC can only represent 15 voltage levels, there will be a loss of signal resolution. The output of the ADC will not be able to perfectly represent the input signal, and there will be some loss of information. This is a fundamental limitation of ADCs, and it is one of the factors that determine the overall performance of an ADC.

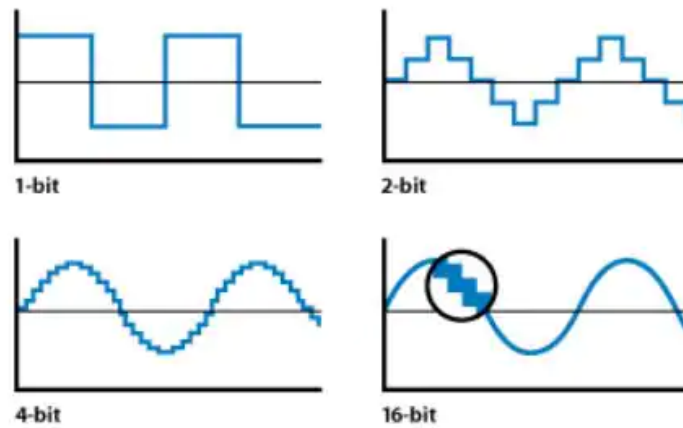


Figure 3.2.: Digital conversion of a sinus signal with different resolutions[6]

To minimize this loss of resolution, it is important to choose an ADC with a sufficient sampling rate and resolution for the specific application. Additionally, other factors such as noise and distortion can also impact the accuracy and reliability of the ADC, and these must be carefully considered when selecting and designing ADC circuits[20].

The **Nucleo H723ZG** provides us with 2 16-bit ADCs and one 12-bit ADC, which can respectively represent up to 65'536 and 4096 sample values.

#### 3.1.3. Timers

The multiple timer units present on the STM32 microcontroller can be either used internally, for example as a trigger signal or a counter, or externally for wave generation or frequency measurement. The STM32 timers offer a range of modes, including basic timer, general-purpose timer, and advanced-control timer.

The basic timer modules are primarily used for generating simple timing events or triggers with a limited number of channels and basic functionality. On the other hand, the general-purpose timers provide more comprehensive features, such as multiple input and

### 3. Theory

output channels, configurable prescalers, and advanced control modes. These timers are widely utilized for various tasks, including event counting, pulse generation, and input capture.

The STM32 timers can operate in different modes, such as up-counting, down-counting, or up/down-counting, depending on the specific requirements of the application. Furthermore, these timers support interrupt generation, allowing for efficient event handling and synchronization with other peripherals. Interrupts can be triggered based on various events, such as timer overflow, compare match, or input capture[21].

#### 3.2. Doppler Effect

The **Doppler Effect** or **Doppler shift**, named after the renowned Austrian physicist Christian Johann **Doppler** (1803–1853), basically describes how the frequency of a wave changes when the distance between the receiver and the emitter of a wave is not constant over time. The most common example to illustrate this change in frequency is the one with the moving ambulance, shown in the cartoon below.

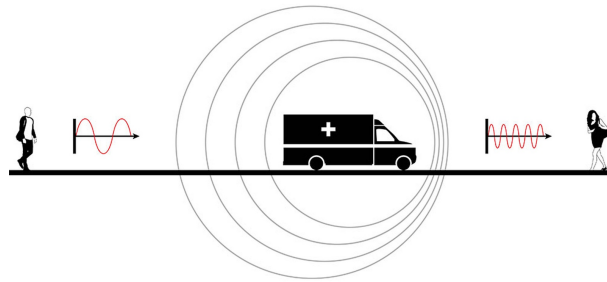


Figure 3.3.: Illustration of the Doppler Effect[7]

The receiver on the right will observe a higher pitched horn (higher frequency) and the other receiver will get a lower pitched horn (lower frequency), even though the ambulance is producing a constant frequency horn sound. The intuition behind those shifts of frequency is pretty simple. As depicted in the figure 3.3, the truck's motion in the direction of sound wave propagation causes a reduction in the wavelength, leading to an increase in the frequency observed by the right receiver. Conversely, when the truck moves in the opposite direction to the sound wave propagation, an increase in wavelength occurs, resulting in a decrease in the observed frequency[22].

We can obtain the formulas for the observed frequency by analyzing figure 3.3. Suppose the emitter is moving towards the receiver and emitting sound at a frequency  $f_s$ . The

### 3. Theory

receiver observes a frequency  $f_o$ . The wavelength of the sound wave emitted by the source is  $\lambda_s$ . The emitter is moving with a speed of  $v_s$ , and before emitting another wave, it will have traveled a distance of  $\Delta x$ . The wavelength perceived by the receiver is obtained by subtracting the traveled distance  $\Delta x$  from the original wavelength  $\lambda_s$ .

$$\lambda_o = \lambda_s - \Delta x = \lambda_s - v_s * T_s \quad (3.1)$$

Where  $T_s$  is the period of the wave. We can substitute  $T_s = \frac{\lambda_s}{v}$ , where  $v$  is the wave speed.

$$\lambda_o = \lambda_s - v_s * \frac{\lambda_s}{v} \quad (3.2)$$

Using  $\lambda = \frac{v}{f}$ , we get for the observed frequency  $f_o$

$$f_o = f_s * \frac{v}{v - v_s} \quad (3.3)$$

The general formula for the Doppler effect can be derived similarly.

$$f_o = f_s * \frac{v \pm v_o}{v \mp v_s} \quad (3.4)$$

In radar, we want to determine the speed of a moving object by measuring the received and sent frequencies. If we set the  $v_s = v_o$ , which simply means that the receiver is the also the emitter, we get the following equation[22]

$$\Delta f = f_o - f_s = \frac{2 * v_s}{v} * f_s \quad (3.5)$$

Where we used the approximation  $v \gg v_s \Rightarrow v - v_s \approx v$ .

When dealing with a moving vehicle, it is not feasible to position the reflective surface in front of it, as doing so would eventually lead to a collision. To address this issue, one can direct a sensor towards the ground at an angle of attack  $\alpha$ , as it is illustrated in the figure below

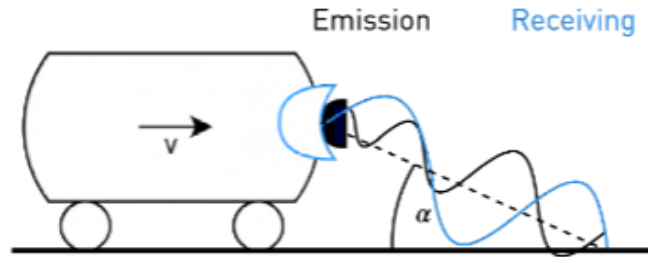


Figure 3.4.: Schematic of the **DOPRail** on the pod

### 3. Theory

The resulting alteration in frequency is expressed through the subsequent formula

$$\Delta f = \frac{2 * v_s}{v} * f_s * \cos(\alpha) \quad (3.6)$$

Solving for  $v_s$ , we get the following formula for the speed of the pod

$$v_s = \frac{\Delta f * v}{2 * f_s * \cos(\alpha)} \quad (3.7)$$

### 3.3. Kalman Filter

In control theory, the Kalman Filter, named after the electrical engineer [Rudolf E. Kálmán](#), is a filtering algorithm that is used to filter and predict the states of a dynamic system that is subject to uncertain and noisy measurements. It has become a widely used tool in a variety of applications, including control systems, signal processing, robotics, and financial modeling, among others. It finds applications in many different domains, especially in guidance, control, tracking, and navigation[23].

The Kalman Filter operates by recursively estimating the state of the system based on its past observations as well as a model of the system's behavior. By incorporating both the measurement and the system model, the Kalman filter can effectively suppress the noise and provide a reliable estimate of the system state. The Kalman filter process is composed of two main steps: the prediction step and the update step.

Let's consider a dynamical system whose true state is described by the following vector :

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T \quad (3.8)$$

The components of  $\mathbf{x}$  can be chosen according to the system we want to describe. The whole purpose of the Kalman filter is to try to get a good estimation of the true state  $\mathbf{x}$  of the time-varying system based on some measurements.

In the prediction step, the Kalman filter uses the model of the system to predict its state at the next time step, based on its current state and a control input, if available. We first need to predict the state of the system in the next iteration.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_{k-1} \quad (3.9)$$

where:

### 3. Theory

$\hat{\mathbf{x}}_{k|k-1}$  is the predicted state at time k based on measurements up to time k-1

$\mathbf{F}$  is the state transition matrix that models the system's behavior

$\mathbf{B}$  is the control input matrix

$\mathbf{u}_{k-1}$  is the control input at time k-1

We also need to predict the covariance matrix at each iteration. This matrix can be seen a measure of the estimated accuracy of the state estimate :

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \quad (3.10)$$

where:

$\mathbf{P}_{k|k-1}$  is the predicted covariance matrix based on measurements up to time k-1

$\mathbf{Q}$  is the process noise covariance matrix, which represents the uncertainty in the model.

In the update step, the Kalman filter uses the measurements obtained at time k to update its estimate of the system's state :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (3.11)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \quad (3.12)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \quad (3.13)$$

where:

$\mathbf{K}_k$  is the Kalman Gain matrix that determines how much the measurements should influence the estimate

$\mathbf{H}_k$  is the measurement matrix that maps the state vector to the measurements

$\mathbf{R}_k$  is the measurement noise covariance matrix, which represents the uncertainty in the measurements

$\mathbf{z}_k$  is the measurement vector at time k.

The Kalman filter is initialized with an initial estimate of the state vector  $\hat{\mathbf{x}}_0$  and the covariance matrix  $\mathbf{P}_0$ . At each time step, the Kalman filter performs the prediction and update steps to estimate the state of the system. The output of the Kalman filter is the estimated state vector  $\hat{\mathbf{x}}_k$  and the covariance matrix  $\mathbf{P}_k$ , which represent the best estimate of the system's state and the uncertainty in the estimate at time k.

An overview of the whole Kalman filtering process is illustrated in the flowgraph below.

### 3. Theory

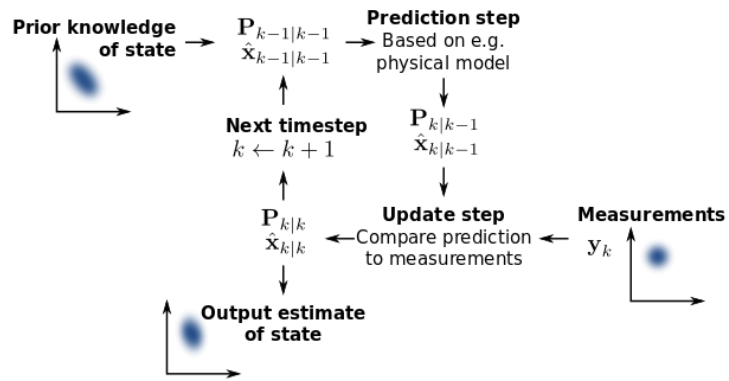


Figure 3.5.: Kalman Filter flowgraph[8]



# Chapter 4

## Hardware / Firmware / Algorithm Implementation

In this chapter, I will provide a detailed account of the implementation aspect of my bachelor's thesis. This involved the use of specialized hardware components, such as custom PCBs and microcontroller units. In addition to the hardware, firmware was developed to enable the sensors to provide real-time speed measurements. This chapter will delve into the specifics of the hardware components used in the project and the development of the firmware to enable the sensors to operate effectively in the hyperloop environment.

### 4.1. Hardware

There were three main hardware components, besides the [Nucleo H723ZG](#), that were used throughout this thesis. The upcoming sections will provide a more detailed examination of how they operate.

#### 4.1.1. DOPRail Sensor

The DOPRail sensor version, used in this thesis, is a custom version of the DOPRail 1000 sensor that was built and designed by [HASLERRail](#) for Swissloop. This sensor belongs to the family of speed sensors that use the [Doppler effect](#) to get speed measurements. The DOPRail emits a 122 GHz wave and then detects the wave that was reflected. Once the reflected wave is detected, it sends out a sinusoidal signal whose frequency will be equal to the difference in frequencies between the emitted wave and the reflected wave.

#### 4. Hardware / Firmware / Algorithm Implementation

Once we are able to effectively measure the frequency of the sensor's output signal, we can use the formula 3.6 to get the relative speed of the vehicle relative to the ground.



Figure 4.1.: DOPRail Sensor

The material of the reflecting surface will strongly influence the amplitude of the analog output signal of the DOPRail sensor. Some materials will have a better reflection coefficient than others and therefore reflect more of the emitted wave. Prior to my thesis, some testing was done with the DOPRail to actually compare some materials as reflection surfaces by comparing the amplitude of the signal sent by the sensor.

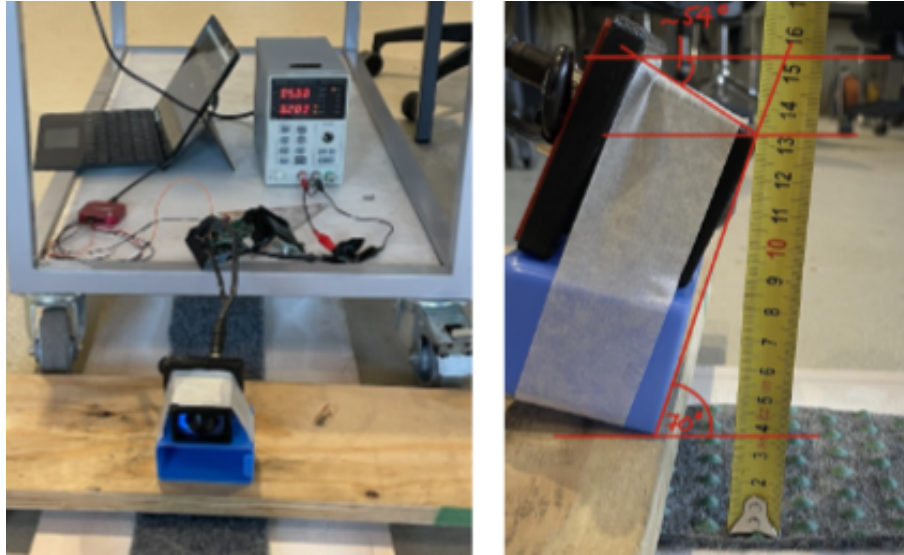


Figure 4.2.: Setup for the DOPRail testing

The figures below present some of the signals captured through testing with different

#### 4. Hardware / Firmware / Algorithm Implementation

configurations.

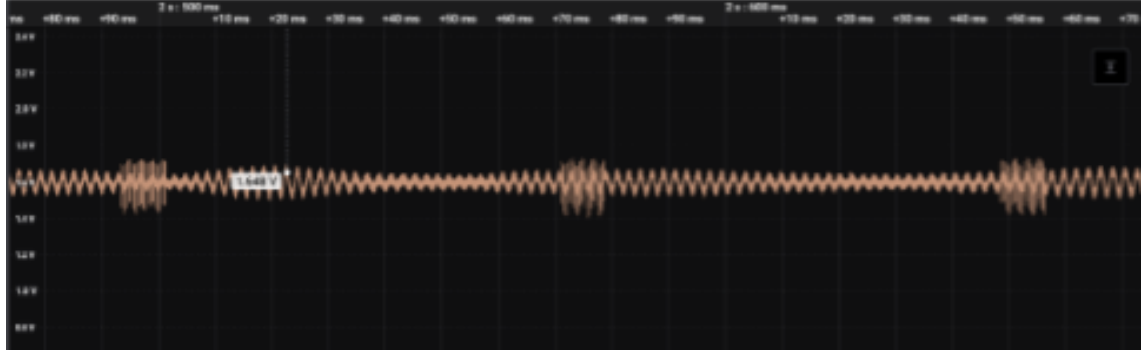


Figure 4.3.: Reflecing surface : velcro strap 65 mm, height : 9 cm, Angle :  $75^\circ$

We notice that the signal is pretty weak, which means that the transmission coefficient of the velcro strap is high, implying a bad reflecting coefficient. The next test was conducted using carpet as the reflecting surface and keeping the same angle and height.

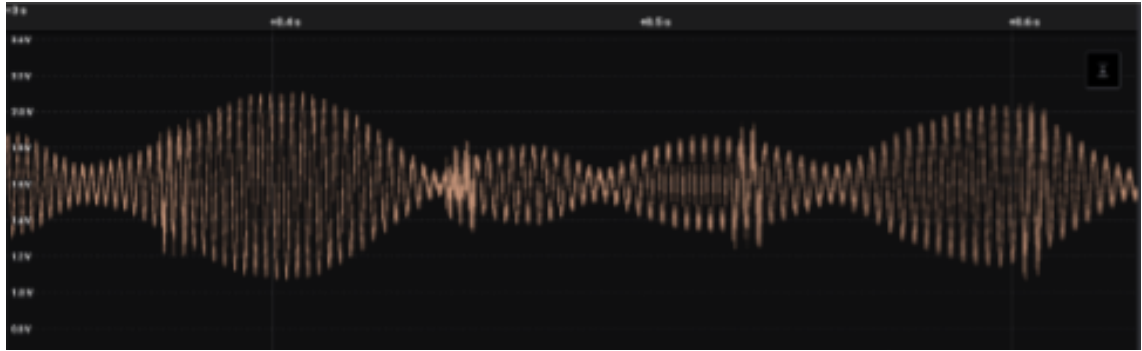


Figure 4.4.: Reflecing surface : Carpet, height : 9 cm, Angle :  $75^\circ$

The signal is much stronger when reflected on carpet. The DOPRail's wave reflects the majority of it back to the sensor.

To find the optimal angle for the sensor, a set of different angles had to be investigated. In the figure below, the raw signal of the sensor is shown from three different angles. The sensor has a pretty small range. The screenshots above were taken with the sensor positioned 19 cm above the reflecting surface. When we compare the signal in 4.4, we see that the signal with a 19 cm height is much weaker.

This circuit incorporates several components to achieve its functionality. At its core, it utilizes a buffer, which is an electronic device that isolates and strengthens an input signal. The buffer makes sure that the characteristics of the earlier stages of the circuit do not affect the later stages.

The next component is a second-order RC (resistor-capacitor) low-pass filter. This filter is designed to attenuate high-frequency components of the signal while allowing lower frequencies to pass through. It helps remove noise and unwanted disturbances from the input signal, enhancing the overall quality and reliability of the circuit's output.

Additionally, the circuit incorporates an inverting Schmitt trigger, which is implemented using an operational amplifier (OpAmp). An OpAmp is a high-gain amplifier that can perform various mathematical operations on an input signal. In this case, the OpAmp is configured as an inverting Schmitt trigger, which is a specific type of comparator. It compares the input voltage with predefined threshold levels and produces a binary output based on whether the input voltage exceeds these thresholds. This helps convert the analog input signal into a digital signal, making it easier to process and analyze.

Finally, the circuit includes a precision rectification circuit. Rectification is the process of

#### 4. Hardware / Firmware / Algorithm Implementation

converting an alternating current (AC) signal into a direct current (DC) signal. The precision rectification circuit ensures accurate and efficient rectification, minimizing losses and maintaining the integrity of the signal.

In short, this circuit uses a buffer to isolate and strengthen the input signal, a second-order RC low-pass filter to get rid of high-frequency noise, an inverting Schmitt trigger with an operational amplifier to turn the analog signal into a triggered signal, and a precision rectification circuit to accurately change the signal from AC to DC.

The provided figure illustrates a comparison between the raw sinusoidal signal captured from the sensor (channel 1) and the triggered signal obtained from the PCB (channel 0).

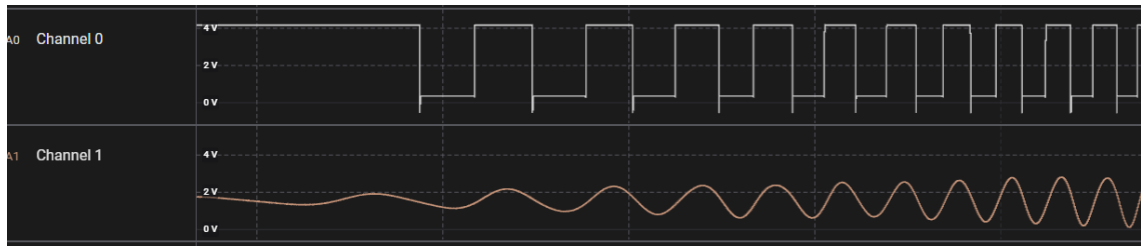


Figure 4.6.: Triggered signal

The PCB takes 24 volts as power. The triggered signal is then sent via the IX connector. The connector can be connected to any processing unit, for example, a vehicle control unit, to be processed.

##### 4.1.3. GAM900 Accelerometer

Baumer designed and constructed the GAM900 accelerometer which is MEMS based. It can measure three axis of acceleration as described in the picture below



Figure 4.7.: GAM900 sensor[9]

The acceleration is encoded as a current between 4 and 20 mA. 4 mA corresponds to the negative maximal acceleration, and 20 mA relates to the maximal positive acceleration. The sensor needs a 24-volt power supply. The measurement range is  $\pm 2g$  and we can configure it depending on the maximum acceleration of the vehicle. We decided to set the range to the maximal range provided by the accelerometer because the maximum acceleration that the Swissloop's pod can take is around 1.5 g. A smaller measurement range would improve the accuracy of the measurement when reading the signal with the ADC.

In addition to the analog current signal, the sensor can also send the information about the acceleration along each axis through the CAN (Controller Area Network) communication protocol. We choose to read the acceleration through the analog signal because it achieves the best latency and it was the most straight forward way to do it with the development board[9].

## 4.2. Firmware

The implementation of the firmware on the microcontroller consisted of two main tasks: reading the analog signals from the sensors to get the speed and acceleration measurements and combining the measurements in a **Kalman filter** to get rid of the noise of the sensors and improve the accuracy of the speed estimation. The firmware is designed to run on the **Nucleo H723ZG** microcontroller unit, which boasts various functionalities and circuits that greatly aid in implementing the firmware. Additionally, the STM Microcontroller provides a hardware abstraction language (HAL) library, which enhances the user experience by simplifying interaction with the microcontroller and enabling us to leverage its full performance potential.

The subsequent sections will delve into a comprehensive exploration of the firmware implementation. Our investigation will entail an in-depth examination of the primary tasks outlined earlier, with a particular emphasis on intricate details. By conducting a thorough analysis, we aim to deepen our understanding and shed light on the finer aspects of the firmware implementation.

### 4.2.1. Sensors Reading

#### Frequency measurement with timer interrupts

For the **DOPRail sensor's** firmware, the idea to get the measured speed out of the analog signal is to measure the frequency of the output signal and then use the formula 3.7 to get the velocity of the pod. The conversion of the raw sensor signal into a triggered signal, as depicted in Figure 4.6, significantly simplifies the reading process with the microcontroller board.

To measure the frequency of the triggered signal, we had to first measure the period of the signal between two rising edges and then invert it. To do so, we used the **timer interrupt** functionality of the microcontroller. The timer lets us count until some predefined value. By saving the values of the timer between two rising edges of the signal, we could easily get the period and then the frequency of the input signal. We configured the timer interrupt to trigger at each rising edge, which means that we jump into the callback function every time the interrupt triggers. To distinguish between the two rising edges when calculating the period, we used a variable that we set to 0 when the first rising edge came and then to 1 when the second came. Then we simply had to check if the value of the variable was true or false to distinguish the edges.

The algorithm is described with the pseudo code below

**Algorithm 1:** DOPrail Reading**Data:** Triggered Analog Signal**Result:**  $Velocity \geq 0$ 


---

```

1 if RisingEdge then
2    $f_s \leftarrow 122 \text{ GHz}$ ;
3    $v \leftarrow c$ ;
4   if Edge == 0 then
5      $TimeAtFirstEdge \leftarrow TimerValue$ ;
6      $Edge \leftarrow 1$ ;
7   else
8      $TimeAtSecondEdge \leftarrow TimerValue$ ;
9      $Period \leftarrow TimeAtSecondEdge - TimeAtFirstEdge$ ;
10     $Frequency \leftarrow TIMERFREQUENCY / Period$ ;
11     $Velocity \leftarrow Frequency * v / (2 * f_s * \cos(\alpha))$ ;
12     $Edge \leftarrow 0$ 
13  end
14 end

```

---

As mentioned in 4.1.1, the sensor emits a wave at 122 GHz, which travels at the speed of light. The timer frequency can be chosen during the configuration depending on our needs. The maximum value that the timer can take before resetting can be configured to a desired value as well in the CubeMX programming interface. One thing that had to be taken into account as well was the overflow of the timer. If we set the maximal count value of the timer too low, there is a risk that the timer will overflow between the two rising edges. To solve this issue, we just decided to give the timer maximum value a high value to omit the risk of overflowing between the two rising edges.

**Analog Signal Reading with the 16-bit ADC**

As mentioned in section 4.1.3, the information about the acceleration coming from the GAM900 is encoded in a current ranging from 4 mA to 20 mA. A current measurement of 4 mA corresponds to an acceleration of -2 g, whereas a current reading of 20 mA corresponds to an acceleration of +2 g. It is important to note that each current value within this range scales linearly between -2g and +2g.

To get the acceleration value out of the current, we first had to convert the current measurement into a voltage measurement because the Analog to Digital Converter on the development board needed a voltage. To solve this issue, each channel of the accelerometer was connected in parallel to a resistance of 110  $\Omega$  and we measured the voltage across this resistance with the ADC. The maximal voltage accepted by the ADC is 3.3 V, therefore, we chose a resistance of 110  $\Omega$  to keep the measured voltage under the limit and avoid damaging the board.



#### 4.2.2. Sensors Fusion via Kalman Filtering

After succeeding in getting the speed and acceleration measurements from the sensors, the next step consisted of combining them in a Kalman filter. The Kalman filter was implemented using the equations listed in Section 3.3. The utilization of the Swissloop STM23 H723ZG template facilitated C++ programming on the microcontroller by providing a comprehensive C++ API. This proved to be highly beneficial for implementing the Kalman filter, primarily due to the availability of various libraries that support efficient matrix operations in C++. The inclusion of these libraries greatly streamlined the implementation of the Kalman filter algorithm and enhanced its performance.

As enhanced in the section 3.3, the goal of the Kalman filter is to get an better estimate of the state of the system observed by using noisy measurements from the quantities we are interested in. In this thesis, we aimed to combine the speed and acceleration values sent from the DOPRail and the GAM900 accelerometer to filter out the noise and get clean speed values.

We defined the state vector  $\mathbf{x}$  of our filter as follow

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \quad (4.1)$$

where  $x_1$  is the speed and  $x_2$  is the acceleration of the observed object.

The observed measurements vector  $\mathbf{z}_k$  contains the values of both speed and acceleration sensors at each iteration  $k$ .

For the state transition matrix  $\mathbf{F}$ , we decided to use the simple cinematic relation between the speed and the acceleration of an object to predict the speed at each next iteration.

$$v_k = v_{k-1} + a * \Delta t \quad (4.2)$$

We assumed that the acceleration would stay constant in our prediction of the state, and we updated it in accordance with the accelerometer measurements.

Two matrices that needed to be estimated and initialized before starting the filtering process are the process noise covariance matrix  $\mathbf{Q}$  and the observation noise covariance matrix  $\mathbf{R}$ . The  $\mathbf{R}$  matrix gives an idea about the inherent error of the sensors used.

The DOPRail and the GAM900 have respectively an error of 4 % and 5 % on each measurement. We can use these percentages to estimate the variances of the measurements noise. Assuming that the errors are independent and identically distributed (which is a common assumption), we can calculate the variance  $\sigma$  of each sensor as follows

$$\sigma = error_i^2 * measurement_i \quad (4.3)$$

Since  $\mathbf{R}$  is a diagonal matrix, we can populate its diagonal elements with the calculated variance value for each measurement component.

Concerning the initialization of the process noise covariance matrix  $\mathbf{Q}$ , we did not find

an effective way to estimate it. After some investigation into what others did and testing, we found that the value of 0.5 for the diagonal elements of the matrix was a good choice. The estimation would have to be improved with further work on the Kalman filter.

The process of Kalman filtering consists of two main steps, as explained in Section 3.3: the next state prediction and the state update. We followed the same framework to combine the two sensors. We first initialized the Kalman filter with the parameters fitting our problem, then predicted the next state and the estimated covariance matrix  $\mathbf{P}$  using the transition matrix described above, and finally updated the estimated quantities with the measurements. The pseudocode below describes the process going on on the microcontroller.

---

**Algorithm 2:** Kalman filtering of the sensors

---

**Data:** *Noisy Velocity and Acceleration*

**Result:** *Filtered Velocity and Acceleration*

```

1 adc_init();
2 Kalman_init();
3 while True do
4   | adc_update();
5   | Kalman_predict();
6   | State  $\leftarrow$  Kalman_update(velocity, acceleration);
7 end
```

---

The infinite loop could be quite computation-heavy for the microcontroller. Therefore, it would have to be improved for energy efficiency.

In the following chapter, we are going to present the results we achieved in this thesis and some of the issues that could not be resolved. We are also going to describe the possible further work that has to be done to improve the current design to make it ready to be used on the Swissloop's pod.

# Chapter 5

## Results

As outlined in the [introduction](#), the aim of this thesis was to commission a working set of sensors to measure the speed accurately and possibly integrate it into this year's pod of Swissloop.

We reached the goal of implementing a working firmware that reads the speed and acceleration values out of the sensors and then combines them in a Kalman filter to get rid of the inherent noise of the measurements. Nevertheless, we were not able to integrate the developed sensor on the current Swissloop's pod due to various reasons explained in [6.1](#).

### 5.1. Firmware Achievements

#### DOPRail

Concerning the [DOPRAil sensor](#), the triggered signal from the [phase sensor PCB](#) arrives at the microcontroller timer's pin, and we are able to get the corresponding velocity by measuring the period of the signal. To make sure that the frequency calculated actually equaled the frequency of the signal, we generated a 100 Hz squared signal using the Pulse Width Modulation generation function of the STM32 [timers](#) and forwarded it to the input of the function that calculates the frequency. The output of the function was effectively 100 Hz, as expected.

#### GAM900

We also succeeded in reading the acceleration values from the [GAM900](#) by using the ADC of the microcontroller board. The latency achieved by the analog signal is the best

## 5. Results

that we could achieve, but we still have to consider the position of the sensor on the pod because the analog signal coming out of the sensor picks up more noise along its traveling paths. The inherent noise of the GAM900 is already pretty high and could affect the accuracy of the calculation if it increases.

### Kalman filter

The implemented Kalman filter achieved the desired functionality of filtering out the noise of some measurements. We were able to combine the speed from the DOPRail and the acceleration from the GAM900 to get accurate velocity values. As we can observe in the plot below, the Kalman filter actually filters most of the noise in a set of data.

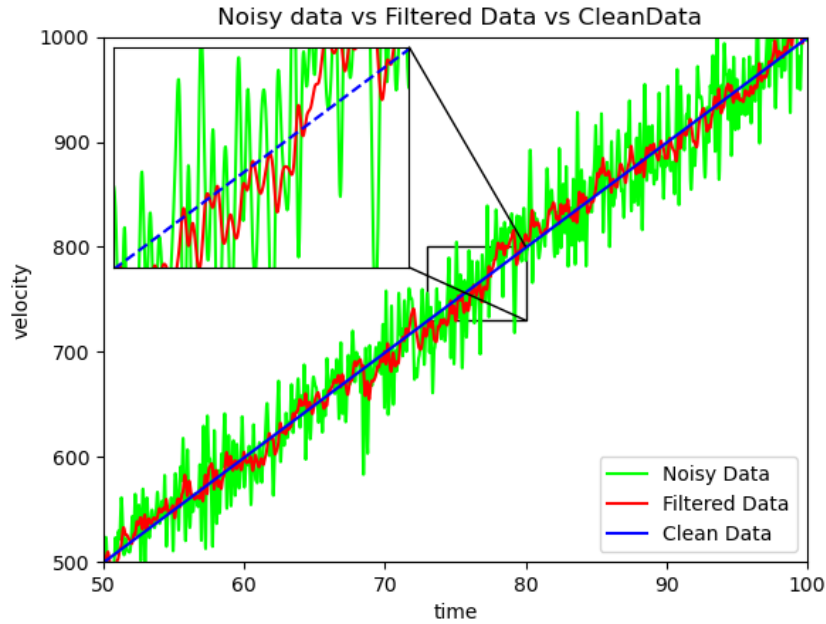


Figure 5.1.: Linear data

We added some normal-distributed noise with a standard deviation of 0.1 and a zero mean to the clean data to replicate the error of the sensors. By investigating the normalized errors of the noise and the filtered data from the clean data, we noticed that the filter also drastically reduced the errors.

## 5. Results

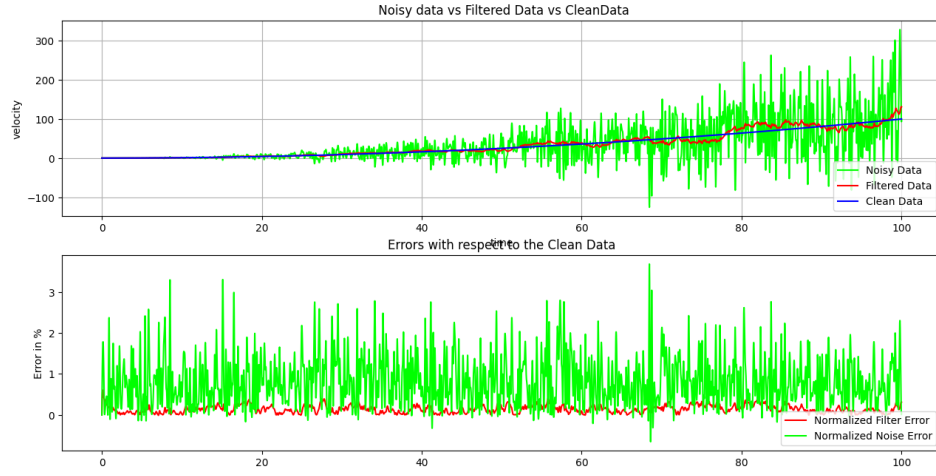


Figure 5.2.: Parabolic Data

Even though we achieved most of the tasks listed in the outline of the project, we could not yet reach a fully working design. There are still some issues that need to be looked into and solved to make the commissioned product useful on a hyperloop prototype. In the next chapter, we are going to present some of the problems with the current design and the further work needed to reach a fully working product.

# Chapter 6

## Discussion

In the previous chapter we presented the achievements we reached in this thesis. As pointed out, we could reach almost all of the planned functionalities. Nevertheless, we were not able to achieve a fully working device that could be already be used this year on this year Swissloop's pod.

In the following sections, we are going to explain more in details the issues encountered during the thesis that still remain and we are going to investigate the further work which need to done to solve those problems to reach an efficient and accurate radar speed sensor.

### 6.1. Issues with the current design

#### 6.1.1. Integration on the Vehicle Control Unit

The first issue encountered was the most problematic. The Swissloop team designs every year their own PCBs, including the vehicle control unit of the pod. The VCU is responsible for taking the signals from the sensors as its input and then do the needed computation to get the corresponding values for the speed, the battery levels, etc. In the design phase of the VCU in autumn 2022, my bachelor thesis was not set yet, meaning that the **DOPRail** was not planned to be on the pod. Therefore the VCU have no IX connector remaining so that I can connect the sensor to it. This was a pretty big set back for the project because the final goal was to integrate the sensor to the pod. Due to the time constraints, it was not possible to redesign a whole VCU during the bachelor thesis. We decided to only integrate the sensor on the pod of next year. The student responsible for the next VCU could therefore take the DOPRail into consideration when designing the VCU.

## 6. Discussion

### 6.1.2. Firmware efficiency

The readings of the sensors as well as the Kalman filtering of those measurements work fine. During the implementation of the firmware, we did not take into account the power efficiency of the process. Having a whole H723ZG microcontroller available for our computation, resources constraints was not really a topic that we investigated. The current code works fine but is a bit inefficient. Currently we are using approximately 24% of the flash size of the MCU. When we would integrate the firmware on the vehicle control unit, that use of the flash size is not viable. We still have to optimize the developed product to make it useful.

## 6.2. Improvements

As mentioned in the previous section, there is still plenty of room for improvement. Due to lack of time, we could not achieve everything we aimed for at the the beginning of this thesis. Next we will describe the various improvements that could be make to our design and how we could reach a useful product that could be part of an hyperloop.

### 6.2.1. Computational efficiency

As explained in [6.1.2](#), the current firmware that process the analog signals of the sensors and then combine them in a Kalman filter to get rid of the noise is computationally expensive. We can imagine the current design to lead to an excessive power consumption when integrate on the main microcontroller of the pod and could possibly impact and slow down the other tasks running on the MCU. A more efficient programming has to be considered.

Currently, we mainly use variables of type float in our Kalman filter. The float type is known to be expensive in terms of resources on the microcontroller due to its limited computational power compared to the CPU. One way to improve the performance would be to do the computation on the microcontroller with unsigned integers, which are way more efficient. We would have to scale the values to maintain decent accuracy for the calculations. By doing so, we would only need to cast the resulting velocity into a float at the end of the computations.

Another aspect we could investigate to achieve a power-efficient design is the timing of the execution of the various functions. In the [algorithm 2](#), the readings of the ADC and the Kalman filter process are both executed in the infinite while loop. This is pretty power-consuming. One way to improve that is to do the ADC readings and the update of the state of the Kalman filter in the timer interrupt callback function. We would wait until a new measurement from the DOPRail comes in to read the accelerometer value and

## 6. Discussion

update the Kalman state accordingly. We keep predicting the next state in the infinite loop because this task does not require a lot of computation power. Of course, all of this depends on the needs of the vehicle. It may need a faster updating rate for the velocity of the pod. In that case, updating the callback function may not be frequent enough.

### 6.2.2. Hardware

The DOPRail could not be on the 2022–2023 Swissloop’s vehicle because it could not be connected to the VCU. The vehicle control unit’s PCB will have to be redesigned next year to support the DOPRail. By putting an additional IX connector on the PCB, the analog signal coming out of the **phase sensor PCB** could be processed on the VCU’s microcontroller. Another piece of hardware that could be improved is the accelerometer. **GAM900** may be accurate and have a good latency, it is still pretty heavy (400 g). A better alternative for the acceleration sensor would be a STM32 accelerometer, which is a couple of millimeters wide and a lot lighter, therefore contributing to reducing the total weight of the pod, which is always a critical parameter for an hyperloop prototype. The small STM32 accelerometer has already been used on the current swissloop’s pod.

### 6.2.3. Integration of other sensors in the Kalman Filter

The current **Kalman filter** combines two inertial sensors to estimate the speed of the pod. The end goal of the filter is actually to include all the inertial measurement units of the vehicle to get a precise and clean speed estimation out of all sensors. The diagram below illustrates how all the inertial sensors would be combined on the pod.



## 6. Discussion

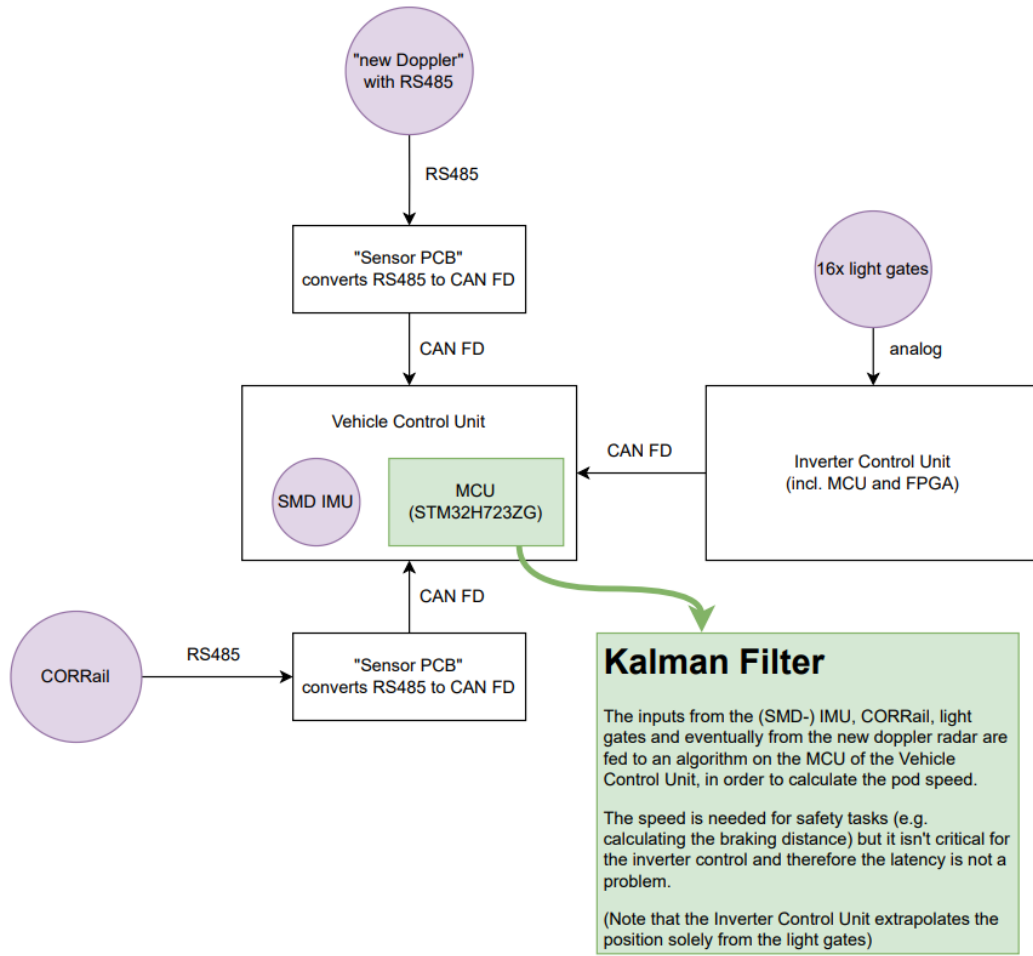


Figure 6.1.: Swissloop Kalman Architecture

The current design processes analog signals coming from the sensors. Although this method enables really good latency, the noise is pretty high. The sensors on the diagram use all the CAN communication protocols. It is way slower than an analog signal but less noisy. HaslerRail is currently developing a new DOPRail sensor that uses the CAN protocol. This new sensor will have to be investigated to see if it outperforms the current DOPRail sensor. Alternatively, we could use both on the pod. The analog DOPRail sensor could be connected to the inverter control unit because the inverter needs accurate velocity measurements with low latency, and the new DOPRail could be used on the VCU.

# Chapter 7

## Conclusion and Future Work

In conclusion, the software for a novel radar speed sensor that was at first intended to be integrated into the vehicle control unit of a Swissloop's pod was successfully implemented in this thesis on a microcontroller. The firmware managed to remove noise and delivered accurate speed data by integrating speed readings from the DOPRail sensor with acceleration measurements from the GAM900 in a Kalman filter. The radar speed sensor's viability for use in a hyperloop setting was shown by the operating firmware.

Yet, there is still space for improvement in the microcontroller's computational efficiency. The microcontroller's limited processing speed and memory added to the lack of time made it difficult to optimize computations of the firmware. To improve the computational efficiency, the firmware still needs to be updated and it will have to use as little resources as possible to fit on the whole vehicle firmware.

Moreover, some pieces of hardware have to be adapted to support the integration of the DOPRail on the vehicle. The vehicle control unit PCB will have to be connected to the sensor to be able to process its signal. Also the track design could be improved to maximize its reflection coefficient and therefore amplify the strength of the signal.

Additionally, we could integrate more functionality to the developed firmware. As discussed in the chapter 6, the Kalman filter should include, in its final version, all the inertial measurement units of the hyperloop prototype to improve drastically the accuracy of the speed value given by all the inertial sensors. The current filter will have to be improved and optimized to support more sensors. The control signals from the VCU could also be forwarded to the Kalman filter to get a better estimation of the state.

In summary, this thesis lays the foundation for integrating the novel radar speed sensor into the vehicle control unit of a hyperloop prototype. The achieved functioning firmware

## *7. Conclusion and Future Work*

showcases the potential of this sensor technology in revolutionizing the transportation industry. Addressing the problems for computational efficiency, noise filtering and accuracy will certainly be key in realizing the full potential of this innovative radar speed sensor in hyperloop systems and beyond.

Appendix	<b>A</b>
----------	----------

Task Description



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich



Task Description for a Bachelor Thesis on

# **Commissioning of a Novel Radar Speed Sensor for an Hyperloop Prototype**

at the Department of Information Technology and  
Electrical Engineering

for

**Mohamed SADFI**  
msadfi@student.ethz.ch

**Advisors:** Dr. Michele Magno, [michele.magno@pbl.ee.ethz.ch](mailto:michele.magno@pbl.ee.ethz.ch)

**Professor:** Prof. Dr. Florian Dörfler, [dorfler@ethz.ch](mailto:dorfler@ethz.ch)

**Handout Date:** 27.02.2023

**Due Date:** 05.06.2023

## 1 Project Goals

The global surge in demand for mobility brings current modes of transport to their limit. For that reason, sustainable and efficient ways of transporting goods and people are high in demand. Vacuum transportation represents an approach in that direction. By reducing the air-drag loss and other friction-based losses through levitation and a vacuum environment, so-called Hyperloop pods are envisaged to travel over large distances at high speeds.

ETH Zurich's Hyperloop Team Swissloop participates at the European Hyperloop Week Pod Competition years and ranks among the best teams worldwide. This year focuses on engineering an energy efficient levitation system and novel linear switched reluctance motor.

The technology regarding the sensors evolves every year and improves greatly their detection capability and its precision. The commissioning of a novel radar speed sensor for a hyperloop prototype is a crucial step towards ensuring the safe and efficient operation of the high-speed transportation system. The new radar sensor, the [DOPRail 1000](#) designed by HASLERRail, a partner of Swissloop, will be responsible for measuring the speed of the pod on the track accurately as it travels through the vacuum-sealed tube. The commissioning process will involve the thorough testing and calibration of the sensors to ensure that it is functioning correctly and providing accurate speed readings.

The main task of the project will be to implement a speed Kalman filter to process the measurements received from the sensor so that the speed of the pod can be derived with more accuracy. The testing will be done on a prototype, which will have to be built at the beginning of the project, running on the track to simulate the conditions of real travel. The filter will run on a MCU of type STM32H723 or STM32H725. If the testing part appears to be successful and depending on the time left, additional sensors will be integrated in the Kalman filter, and the newly commissioned radar will be optimized and integrated on the 2023 Swissloop pod. The success of the commissioning process will play a critical role in the overall success of the hyperloop prototype and pave the way for the widespread deployment of this innovative transportation technology and could be of great help for the future Swissloop teams. The sensor that will be used in this project is a prototype of the final version of the sensor coming out in spring 2023. Therefore, all developed software will have to be functional on the prototype and on the final version as well.

## **2 Tasks**

The project will be split into three phases, as described below:

### **Phase 1 (Week 1-4)**

1. Investigate how the existing sensor, CORrail 2000, on the pod work.  
Understand how it send data and how the receive data is processed.
2. Study and get used to the hardware and the tools to program, i.e.,  
microcontroller programming, communication, sensor data acquisition.
3. Investigate the position derivation from radar sensor measurements using a  
Kalman filtering. Figure out the guidelines for its implementation on the  
microcontroller unit.
4. Design and build a small prototype test vehicle for gathering sensor data on  
Swissloop's Prototype Track. The sensor will be mounted on this vehicle at  
first to acquire data, because the pod can't be used for testing purposes.

### **Phase 2 (weeks 5-9)**

1. Gather the sensor's data for different configurations, e.g., angle of  
incidence, surface condition of the rail, driving speed, etc.
2. Using the existing prototype sensor and an accelerometer, implement a  
velocity Kalman filter using embedded C on the MCU of type STM32H723  
or STM32H725 to improve measurement accuracy.
3. Test the sensor on the test vehicle and check if the speed can be determined  
accurately in real time. The sensor needs in the best case to be better than  
the current sensor used on the pod. The current sensor achieves a latency  
of 10 microseconds and an accuracy of 1/265 m.
4. Optimize the Kalman filter in order to attain the desired latency and  
accuracy.
5. Integration of additional sensors in the Kalman Filter. Most likely an IMU  
( inertial measurement unit).

### **Phase 3 (week 10-14)**

1. Comparing the results with other velocity/position measurements on the Hyperloop prototype regarding accuracy and latency of the newly commissioned radar sensor.
2. System integration on the 2023 Swissloop pod of the final version.
3. In-field test on the Swissloop's pod.
4. Write the report and preparation of the final presentation. The writing of the report will be done in parallel of the previously stated tasks.

### **Milestones**

The following milestones need to be reached during the thesis:

- Acquisition of the sensor's data and dataset.
- Design and built of the test vehicle.
- Velocity Kalman filter implementation on the microcontroller unit using embedded C.
- Final design and in-field test.
- Final report and presentation

## **3 Project Organization**

During the thesis, students will gain experience in the independent solution of a technical-scientific problem by applying the acquired specialist and social skills. The grade is based on the following: Student effort, thoroughness and learning curve; Results in terms of quality and quantity; final presentation and report; documentation and reproducibility. All theses include an oral presentation, a written report and are graded. Before starting, the project must be registered in myStudies and all required documents need to be handed in for archiving by PBL.

### **3.1 Weekly Report**

There will be a weekly report/meeting held between the student and the assistants. The exact time and location of these meetings will be determined within the first week of the project in order to fit the students and the assistants schedule. These meetings will be used to evaluate the status and document the progress of the project (required to be done by the student). Beside these regular meetings, additional meetings can be organized to address urgent issues as well. The weekly report,



along with all other relevant documents (source code, datasheets, papers, etc), should be uploaded to a clouding service, such as Polybox and shared with the assistants.

### **3.2 Project Plan**

Within the first month of the project, you will be asked to prepare a project plan. This plan should identify the tasks to be performed during the project and sets deadlines for those tasks. The prepared plan will be a topic of discussion of the first week's meeting between you and your advisers. Note that the project plan should be updated constantly depending on the project's status.

### **3.3 Final Report and paper**


PDF copies of the final report written in English are to be turned in. Basic references will be provided by the supervisors by mail and at the meetings during the whole project, but the students are expected to add a considerable amount of their own literature research to the project ("state of the art").

### **3.4 Final Presentation**

There will be a presentation (15 min presentation and 5 min Q&A for BT/ST and 20 min presentation and 10 min Q&A for MT) at the end of this project in order to present your results to a wider audience. The exact date will be determined towards the end of the work.

### **References:**

Will be provided by the supervisors by mail and at the meetings during the whole project.

Place and Date Zürich, 27.02.2023 Signature Student 

Appendix	<b>B</b>
----------	----------

## Declaration of Originality



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

COMMISSIONING OF A NOVEL RADAR SPEED  
SENSOR FOR AN HYPERLOOP PROTOTYPE

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

SADFI

**First name(s):**

Mohamed Ali

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Zürich, 05.06.2023

**Signature(s)**

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*

## File Structure

All the code written during this thesis was uploaded to a github repository. Please send a request to msadfi@ethz.ch to access it.

The project has 2 main folders. The SpeedSensorSTM32 folder which is the Swissloop's STM32 H723ZG template used which include the all the configurations files needed to code and program the MCU from Clion or VScode. The TestingKalmanFilter folder is a small VScode project used to test if the filter worked well and to plot with a python script the actual filtering that was done on a noisy set of data.

```

/
├── SpeedSensorSTM32 .....
│   ├── Src .....
│   │   └── Swissloop .....Code written for the Sensors Firmware
└── TestingKalmanFilter .....Code used for tests and plots

```

# Glossary

**API** An API (Application Programming Interface) is a set of rules and protocols that allows software applications to communicate and interact with each other. It provides a standardized interface for accessing and utilizing the functionalities and resources of a software system or service. APIs enable developers to integrate different applications, leverage third-party services, and build new software solutions..

**Hyperloop** A hyperloop is a proposed mode of high-speed transportation that uses sealed tubes or tunnels to transport passenger pods or cargo capsules at very high speeds. The concept involves creating a low-pressure environment inside the tube to minimize air resistance, allowing the pods to travel at near-supersonic speeds. Hyperloop systems typically employ magnetic levitation to lift and propel the pods, reducing friction and enabling smooth, efficient movement. This futuristic transportation technology has the potential to revolutionize long-distance travel by providing faster, more sustainable, and energy-efficient transportation options..

**Variance** In the context of the variance of the error of a sensor, variance refers to the statistical measure that quantifies the variability or spread of the errors produced by the sensor. It represents how much the individual measurements deviate from the true value or expected value. The variance of the error provides an indication of the consistency and precision of the sensor's measurements. A lower variance implies that the errors are relatively small and consistent, indicating higher accuracy and reliability of the sensor. Conversely, a higher variance suggests larger and more unpredictable errors, indicating lower precision and potential limitations in the sensor's performance..

**Vehicle Control Unit** The vehicle control unit is a central electronic module in a vehicle that oversees and coordinates the operation of different systems and components for optimal performance, safety, and efficiency. It receives data from sensors, processes it, and controls functions such as engine management, braking, and stability

## *Glossary*

control. Acting as the vehicle's brain, the VCU ensures effective communication and integration between various subsystems..

# Bibliography

- [1] E. E. 123, “Hall effect and hall sensors,” Unknown, [Online; accessed 14-April-2023]. [Online]. Available: <https://electricalengineering123.com/hall-effect-hall-sensor/>
- [2] M. Soltero, “What is a hall-effect sensor?” 2021, [Online; accessed 16-April-2023]. [Online]. Available: [https://e2e.ti.com/blogs\\_/b/analogwire/posts/what-is-a-hall-effect-sensor](https://e2e.ti.com/blogs_/b/analogwire/posts/what-is-a-hall-effect-sensor)
- [3] A. B. T. K. Nikola Lopac, Irena Jurdana, “Application of laser systems for detection and ranging in the modern road transportation and maritime sector,” 2022, [Online; accessed 20-April-2023]. [Online]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjh-qDAobj-AhUogP0HHX8ZDDMQFnoECBQQAQ&url=https%3A%2F%2Fwww.mdpi.com%2F1424-8220%2F22%2F16%2F5946%2Fpdf&usg=AOvVaw2bJnKtE68Xd7rHGmi8-x3W>
- [4] J. G. McNeff, *The Global Positioning System*. Cambridge University Press, 2002.
- [5] STMicroelectronics, *STM32H7 Nucleo-144 boards (MB1364)*, STMicroelectronics, 2020.
- [6] M. Gudino, “Analog-to-digital converters basics,” 2018, [Online; accessed 03-May-2023]. [Online]. Available: <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters>
- [7] Bitsche, “Wenn der schall staucht: Doppler effekt,” 2010, [Online; accessed 01-May-2023]. [Online]. Available: <https://www.bitsche-optik.at/blog/wenn-der-schall-staucht-doppler-effekt/>
- [8] P. Aimonen, 2021, [Online; accessed 11-May-2023]. [Online]. Available: [https://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Basic\\_concept\\_of\\_Kalman\\_filtering.svg/langfr-1920px-Basic\\_concept\\_of\\_Kalman\\_filtering.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Basic_concept_of_Kalman_filtering.svg/langfr-1920px-Basic_concept_of_Kalman_filtering.svg.png)
- [9] Baumer, *GAM900 acceleration sensor*, Baumer, Unknown 2022.

## Bibliography

- [10] E. A. LLC, “Microcontroller unit [mcu],” 2022, [Online; accessed 2-April-2023]. [Online]. Available: <https://embeddedartistry.com/fieldmanual-terms/microcontroller-unit/>
- [11] STMicroelectronics, “Stm32 nucleo boards,” Unknown, [Online; accessed 2-April-2023]. [Online]. Available: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>
- [12] RS, “The guide to hall-effect sensors,” Unknown, [Online; accessed 14-April-2023]. [Online]. Available: <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/hall-effect-sensors-guide>
- [13] A. K, “The hall effect and hall emf,” 2014, [Online; accessed 16-April-2023]. [Online]. Available: <https://www.youtube.com/watch?v=Ip43wws6FEw>
- [14] Honeywell, “Hall effect sensors,” Unknown, [Online; accessed 16-April-2023]. [Online]. Available: [https://win.adrirobot.it/sensori/sensore\\_magnetico/Esempi\\_applicativi\\_sensore-magnetico.pdf](https://win.adrirobot.it/sensori/sensore_magnetico/Esempi_applicativi_sensore-magnetico.pdf)
- [15] Wikipedia, “Accelerometer,” 2021, [Online; accessed 29-May-2023]. [Online]. Available: <https://en.wikipedia.org/wiki/Accelerometer>
- [16] C. P. Wang, *J. Quant. Spectrosc. Radiat. Transfer Vol. 40, No. 3, pp. 309-319*. Pergamon Press, 1988.
- [17] STMicroelectronics, *Description of STM32F4 HAL and low-layer drivers*, STMicroelectronics, 2020.
- [18] K. Magdy, “Stm32 hal library tutorial,” 2021, [Online; accessed 26-April-2023]. [Online]. Available: <https://deepbluembedded.com/stm32-hal-library-tutorial-examples/>
- [19] STMicroelectronics, *STM32CubeMX for STM32 configuration and initialization C code generation*, STMicroelectronics, Unknown 2023.
- [20] Unknown, “Analogue to digital converter,” Unknown, [Online; accessed 03-May-2023]. [Online]. Available: <https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html>
- [21] ST, “Stm32l4 timers,” Unknown, [Online; accessed 29-May-2023]. [Online]. Available: [https://www.st.com/resource/en/product\\_training/STM32L4\\_WDG\\_TIMERS\\_GPTIM.pdf](https://www.st.com/resource/en/product_training/STM32L4_WDG_TIMERS_GPTIM.pdf)
- [22] OpenStax, “The doppler effect,” Unknown, [Online; accessed 02-May-2023]. [Online]. Available: [https://phys.libretexts.org/Bookshelves/University\\_Physics/Book%3A\\_University\\_Physics\\_\(OpenStax\)/Book%3A\\_University\\_Physics\\_I\\_-\\_Mechanics\\_Sound\\_Oscillations\\_and\\_Waves\\_\(OpenStax\)/17%3A\\_Sound/17.08%3A\\_The\\_Doppler\\_Effect](https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_(OpenStax)/Book%3A_University_Physics_I_-_Mechanics_Sound_Oscillations_and_Waves_(OpenStax)/17%3A_Sound/17.08%3A_The_Doppler_Effect)



## Bibliography

- [23] S. Sanyal, “An intuition about kalman filter for computer vision,” 2021, [Online; accessed 10-May-2023]. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/10/an-intuition-about-kalman-filter/>
- [24] D. Hankerson, S. Vanstone, and A. Menezes, *Guide to Elliptic Curve Cryptography*, ser. Springer Professional Computing. Springer, 2004.