Design specification

Search and Rescue - Underwater

Version 1.0

Author: Alexander Roser
David Andersson
Hampus Frick
Isac Lundin
Ken Dahl
Oscar Holm
Oskar Philipsson

Date: December 19, 2022

Status

Reviewed	-	-
Approved	-	-

Project Identity

Group E-mail:	searchandrescueunderwater2022@gmail.com
Homepage:	http://www.grphomepage.se
Orderer:	Gustav Zetterqvist, ISY
	E-mail: gustav.zetterqvist@liu.se
Customer:	Andreas Gällström, Saab Dynamics
	E-mail: andreas.gallstrom@saabgroup.com
Customer:	Jonatan Olofsson, Saab Dynamics
	E-mail: jonatan.olofsson@saabgroup.com
Supervisor:	Daniel Bossér, ISY
	E-mail: daniel.bosser@liu.se
Supervisor:	Philip Andersson, Saab Dynamics
	E-mail: philip.e.andersson@saabgroup.com
Supervisor:	Erik Söderberg, Saab Dynamics
	E-mail: erik.soderberg@saabgroup.com

Group Members

Name	**Responsibility**	**E-mail**
Alexander Roser (AR)	Head of testing	alero223@student.liu.se
David Andersson (DA)	Head of documentation	davan957@student.liu.se
Hampus Frick (HF)	Project leader	hamfr643@student.liu.se
Isac Lundin (IL)	Head of hardware	isalu162@student.liu.se
Ken Dahl (KD)	Head of software	kenda091@student.liu.se
Oscar Holm (OH)	Head of information	oscho082@student.liu.se
Oskar Philipsson (OP)	Head of design	oskph717@student.liu.se

Document History

Version	Date	Changes made	Sign	Reviewer
0.1	2022-09-20	First draft.	DA, IL, OH, OP	DA
0.2	2022-10-03	Second draft.	DA, IL, OH, OP, AR, HF, KD	DA
0.3	2022-10-05	Third draft.	DA, IL, OH, OP, AR, HF, KD	DA
0.4	2022-10-14	Fourth draft.	DA, IL, OH, OP, AR, HF, KD	DA
1.0	2022-10-31	First version.	DA, IL, OH, OP, AR, HF, KD	DA

Contents

1 Introduction

This document is the design specification for the project Search and Rescue Underwater in the course TSRT10, Reglerteknisk projektkurs, CDIO, given at Linköping University.

1.1 Parties

In this project there are three parties involved. The customers, Andreas Gällström and Jonatan Olofsson at Saab Dynamics, the client Gustav Zetterqvist at Linköping University, an advisor Daniel Bossér at Linköping University and a project group working on the project. The members of the project group, their roles, and contact information can be found in at the beginning of the document.

1.2 Background information

One of Saab Dynamics' largest departments designs underwater crafts of various types. These are used for inspecting underwater infrastructure. When a submarine somehow gets stuck during a mission, an emergency pinger is activated to be able to find the submarine and retrieve it again. You often then have to go out by boat and lowering hydrophones to find out where the emergency ping sound is coming from. In stressful environments, it can take a very long time to search an area. Therefore, instead of searching an area manually, an autonomous UAS could complete the task.

1.3 Definitions

- UAS - Unmanned aircraft system, i.e., drone.
- GUI - Graphical user interface.
- ROS2 - Robot operating system.
- Session - The time span from which the UAS is turned on until the UAS is turned off.
- Mission - Contains all tasks produced by the task planner. Multiple missions can be executed in one session.
- UUV - Unmanned underwater vehicle.
- Emergency pinger - Transmitter of the emergency signal relayed by the UUV.
- Semi-autonomous mode - A mode where the user is coordinating where the UAS should go.

2 System overview

The full system is divided into multiple subsystems. These are; UAS, ground station and underwater craft, in which these subsystems also can be divided into smaller subsystems.

There is also a simulation environment that will be used for development and evaluation purposes.

The UAS contains of a flight controller which controls the UAS with additional sensors, a Raspberry Pi which takes in the information from the sound card and calculates/transmits the data to the ground station. A hydrophone which gathers the data (the ping) from the distressed UUV and sends the data to the sound card.

The UUV has an emergency pinger which it activates when it is in distress.

The ground station does the more advanced calculations that the Raspberry Pi onboard the UAS cannot do.

2.1 Hardware

This section contains descriptions of the general hardware and the components that are used.

2.1.1 Sensors

The UAS has different sensors that are used for the implementation of the system. They are as follows:

- GPS – Global Positioning System, see chapter 4 for more information.
- IMU – The combination of accelerometer, gyroscope and magnetometer.
- Barometer – Measures the current air pressure that the UAS is experiencing, see chapter 2.1.3 for more information.

2.1.2 Hydrophone

The hydrophone that will be used is called *AS-1 Hydrophone*, and is manufactured by Aquarian Scientific. [1] The hydrophone is attached to a nine-meter long cable that will be connected to the UAS. Information that may be of use for this project is listed below:

- Receiving Sensitivity: -208dBV re 1 μPa (40μV/Pascal)
- Linear range: 1Hz to 100kHz $\pm$ 2dB,
- Horizontal Directivity(20kHz): $\pm$ 0.2dB
- Horizontal Directivity(100kHz): $\pm$ 1dB
- Vertical Directivity(20kHz): $\pm$ 1dB
- Vertical Directivity(100kHz): + 6dB -11dB

2.1.3 Barometer

The barometer is of model MS5611. It has an altitude resolution of 0.065/0.042/0.027/0.018/0.012 mbar dependent on the oversampling ration, where the oversampling ratios are 256/512/1024/2048/4096. It has an accuracy of $\pm$1.5 mbar at 25°C and 750 mbar and the error band is at -20°C to +85°C and 450 to 1100 mbar: $\pm$2.5 mbar. Its full operating range is 10 to 1200 mbar at -40 to +85°C. The response time is 0.5/1.1/2.1/4.1/8.22 ms dependent on the same oversampling rations as earlier.

2.1.4 Sound card

A sound card is going to be mounted on the UAS to be able to take the analog data from the hydrophone and send it to the Raspberry Pi. The sound card is of the model *Zoom F3*. Over USB, it has support for sending 2-channel audio at 96 kHz. This is just enough to sample the roughly 45 kHz signal that the pinger is emitting. The sound card's settings can also be controlled over Bluetooth in case physical access to the sound card is not possible.

2.1.5 Raspberry Pi

The computer that is used is a *NanoPi Neo*. It was mainly chosen because of its low weight of 6.5 grams. It has Wi-Fi and Bluetooth support through an external antenna that is not included. It has GPIO for interfacing with a plethora of other devices, and importantly for the project, UART support.

2.1.6 Manual controller

When the UAS is to be driven outside, it must be possible to control manually, to be able to abort the session at any time [6]. Therefore, a handheld controller must be connected to the UAS and supervised by a person, during fly-time.

The control used is a *Radiomaster TX16S*. It has a transmission frequency in the range $2.4 - 2.48$ GHz and a range of 2 km [2].

When the manual controller is used to steer the UAS, it will bypass the Raspberry Pi and send signals directly to the flight controller.

2.1.7 UUV

The UUV will be transmitting a distress signal using the emergency pinger, see section 2.1.8 for more information about the emergency pinger. The UUV will be sending the distress signal when an accident has occurred and the UUV is immobile.

2.1.8 Emergency pinger

The emergency pinger is of model *LINDA UNDERWATER ACOUSTIC BEACON - MODEL C1A* and transmits on a frequency of 45 kHz. It is not clear if the transmitted signal propagates with spherical or cylindrical symmetry. This depends both on the model and the depth in the water. Therefore, it needs to be investigated, since it might affect strategies using the amplitude of the signal for distance estimation.

2.1.9 Base station

The base station consists of a laptop that will control the session. To do so, it will have the GUI installed so the end user can monitor and control the UAS while it executes the mission.
The heavy computations needed during the session will also be done by the base station to improve the speed of the calculations and also to reduce the amount of power the UAS needs and by that save battery life.
If some simulations need to be done during the session to aid some decisions, these will also be performed by the base station.

2.1.10 UAS

The body of the UAS is a *Holybro x500 v2*[3], equipped with a *Pixhawk 6C Flight Controller*, *PM02 V3-12S Power Module*, *M8N GPS Module* and a *SiK Telemetry Radio V3 433/915MHz* (range ∼ 300 m).

The UAS will carry the hydrophone and sound card. A Raspberry Pi will also be mounted on the UAS, acting as the central computer of the module. On the UAS-module, the signal processing and control algorithms will be implemented.

2.2 Communication

This section describes how communication between the different subsystems will take place. The communications are illustrated in Figure 1.

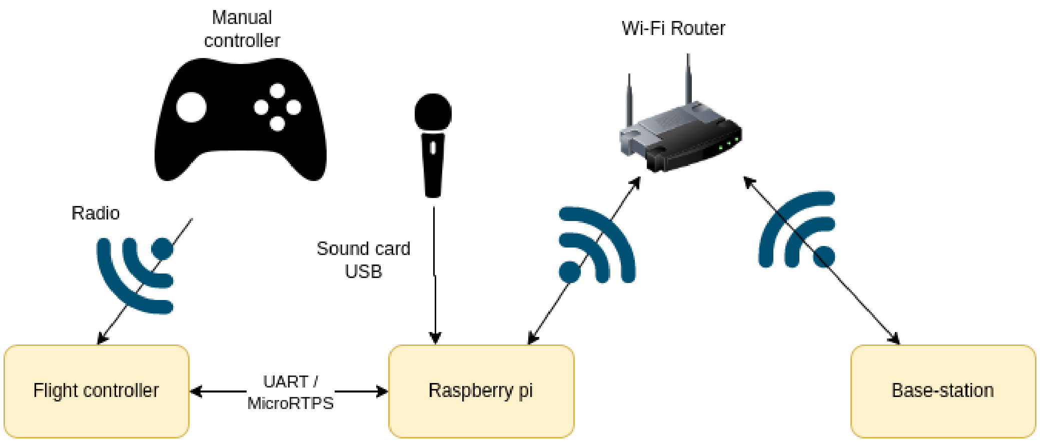

Figure 1: Communication between all subsystems in project

On the UAS there is one flight controller and a Raspberry Pi. They talk to each other over a UART. The actual protocol they talk over is called MicroRTPS, which is a publish / subscribe protocol, similar to ROS2 topics. Over this bus, sensor data from the flight computers sensors will be sent, as well as flight instructions to the flight controller from the Raspberry Pi. The Raspberry Pi also talks to the sound card over USB.

The base station and Raspberry Pi will communicate with a Wi-Fi router, which will allow communication within the local network. This will limit the range of the UAS by the range of the router.

For manual control, a classical radio controller is also used. With the controller, autonomous flying can be disabled and a human can directly control the drone.

2.3 Simulation

The simulations will run in *Gazebo11*, an open-source software library collection with physically realistic 3D-environment simulation capabilities. The existing environment is already fitted with a UAS and a basic simulation world. The virtual UAS can via a terminal receive and execute positional commands to fly to.

Extensions to the simulation world will be made to allow for water physics to be simulated. This will be essential for simulations regarding hydrophone submerging and under-water signal propagation. A model of the hydrophone itself will also be added to the simulation environment, as well as the emergency pinger signal.

With the added physics of the hydrophone, the control algorithm for the UAS might need to be modified to account for the extra weight and, most likely, swinging of the cable. Motion planning and task planning will be implemented and, at last, entire missions will be simulated where all parts of the simulated product will cooperate.

3 Software architecture

This project will use an advanced software architecture, which allows for software in the loop testing (SIL). This means that the same code that should be used on the real hardware, can run against a simulation environment where the hardware is replaced with a digital twin. The software should also be run in a reproducible environment so that dependency problems are eliminated. To achieve this, the software will be developed in Docker containers running on Linux. ROS2 will also be used as the main framework for the project, utilizing the already developed simulation tool Gazebo, and visualization tool RViz2.

Whether the code is running on the real UAS or simulation is selected by starting different docker containers and setting up the networks correctly.

The overview of the software architecture is shown in Figure 2. It shows that the majority of the code is identical for the simulation as well as when running on real hardware. The OR blocks select whether to simulate hardware or not.

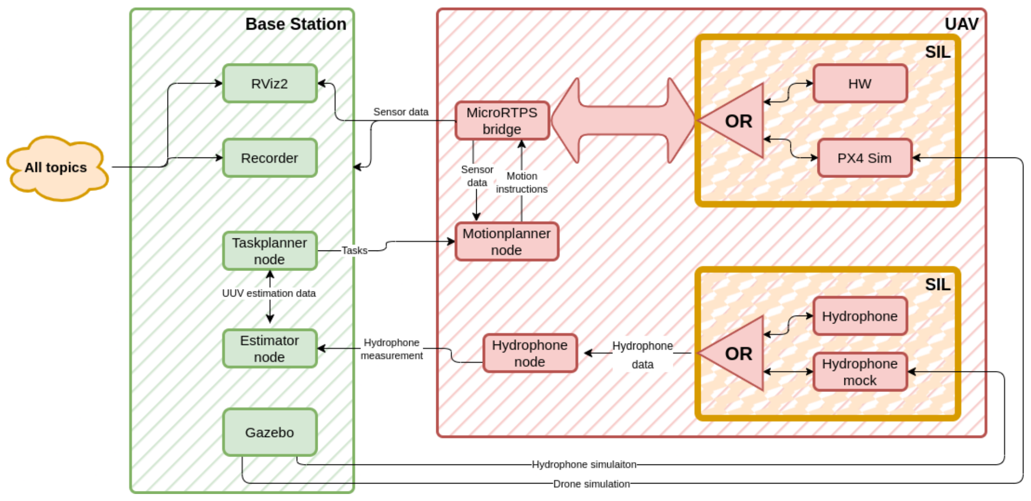

Figure 2: Software architecture overview

3.1 ROS2

As previously mentioned, the project will heavily utilize the tools from the ROS2 project. The project code will be divided up into the nodes shown in Figure 2. The nodes will communicate with each other through one of the three communications methods in ROS2: topics, services and actions. What data will be communicated between the nodes are also shown in the same figure.

The nodes that require a lot of computational power, will be placed on the base station and the nodes that need to interact with real hardware will be placed on the UAS.

There will also be a recorder node running on the base station to record all messages sent in ROS2. This is done in ROS2 bags. These bags can then later be used for replaying scenarios for debugging purposes, or replaying events from running on real hardware, in the simulation environment.

3.2 GUI

The GUI will be run on the base station and the purpose of it is to provide the user with information and the ability to send commands to the UAS. It will always display battery level, an estimate of remaining flying time and a visualization of the explored area. For the current session and mission, it will display elapsed time and flown distance. Commands include changing between manual, semi-autonomous and autonomous mode as well as mode-specific commands.

In semi-autonomous mode, the user will be able to send commands such as coordinate to destinations, descend to submerge the hydrophone, ascend to lift the hydrophone into the air, and land. Autonomous mode allows the user to start and cancel autonomous missions along with choosing motion planning algorithms.

A sketch on how the GUI might look like can be seen in Figure 3.

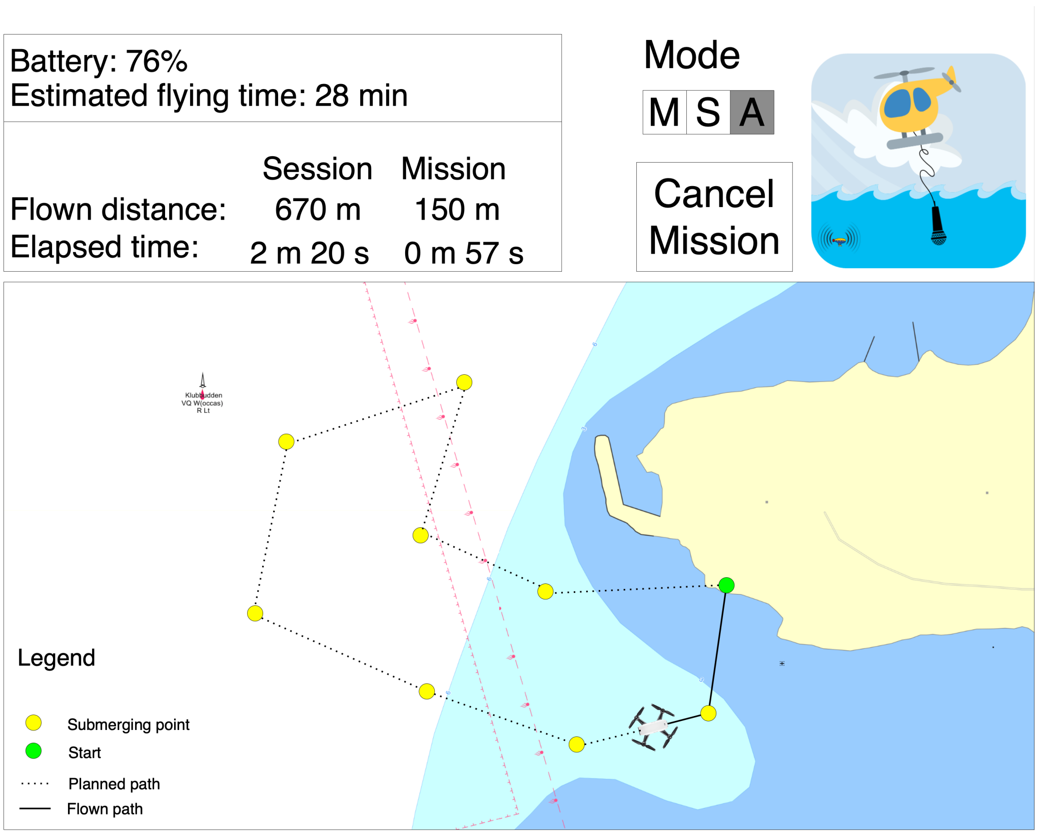

Figure 3: A sketch of the GUI for autonomous mode. The larger part of the GUI will consist of a nautical chart with planned and flown paths of the UAS. Information and buttons for sending commands will be available as well

4 Positioning

In order to determine its position, the UAS will be equipped with GPS.
The GPS has a NEO-M8N module, with a horizontal accuracy of 2,5 m (CEP 50%)[7]. The barometer will also be used to estimate the UAS:s altitude. For indoor use, the UAS will use *Qualisys* in visionen. The *Qualisys* positioning system is used to track objects in a 3D space. This system has twelve infrared cameras that emit infrared flashes with a frequency of 0-300 Hz. The cameras can then detect reflections and triangulate the position of the object. The UAS will need to be fitted with reflective balls for the system to work.

Measuring the UAS:s position is crucial for target estimation, task planning and the control algorithms.

5 Task planner

The primary goal for the task planner is to set up a plan for where the UAS should submerge the hydrophone in order to locate the UUV by using a static planner. An extended version of the task planner would be to set up this plan dynamically, based on collected data from the hydrophone.

The static part will have predetermined locations where it submerges the hydrophone and collects data. The data will be processed and trilateration will be used to determine the position of the UUV. Trilateration is a method to determine the intersection of three different circles, given that we have the center and the radius of them. Given a set of range estimations, $\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_K$, where $\hat{r}_k = \|x - p_k\|_2 + e_k$, $k = 1, 2, \ldots, K$. x is the position of the UUV and p_k is the location of the k:th measurement. e_k is an uncertainty in the estimation.

From there, the target position is estimated by minimizing

$$V(x) = \sum_{k=1}^{K} \|\hat{r}_k - \|x - p_k\|_2\|_2^2. \tag{1}$$

then, $\hat{x} = \arg\min V(x)$.

The minimization can be done in multiple ways. Some are Gauss Newton-search, grid search and gradient descend.

The dynamic method will use previous collected data and target estimation. In order to get an unambiguous estimation of the UUV:s location, multiple measurements are needed; four in a 3D perception and three in a 2D perception.

After an initial location p_1 and range estimation r_1, the UUV could be anywhere on a circle with radius r_1 and center p_1. The next measurement should be somewhere on that circle. With another range estimation r_2 at another location p_2, the location of the UUV is nearby one of the intersections of the two circles. To determine which of the two is most likely, the next measurement should be done at one of these estimated locations. With three range estimates from three different locations, the UUV:s position can be estimated. Figure 4 shows an example on where the first three measurements are done. From now on, the following measurements should be done at the most recent estimation of the UUV:s location. The task is considered done when the most recent measurement is within the desired precision.

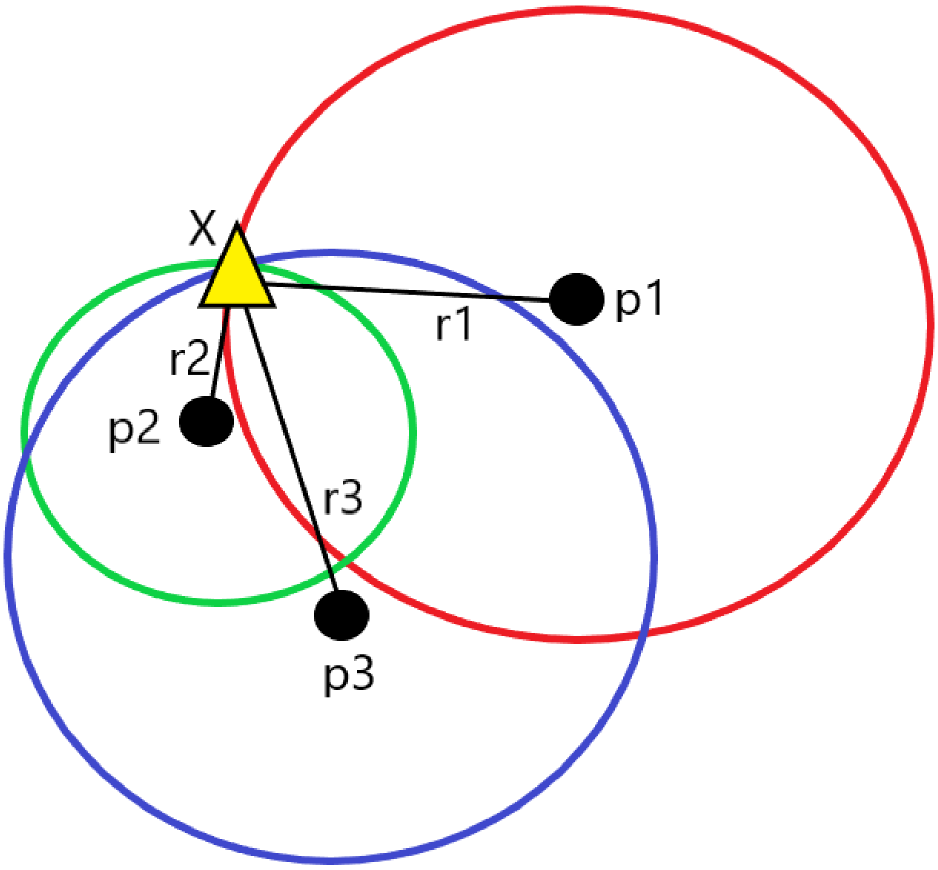

Figure 4: A sketch of how a dynamic trilateration "algorithm" can be done. The first measurement is done at p_1, the second is done at p_2 close to the red circle and the third one, p_3 is done near one of the intersections between the red and blue circles. Since the positioning is not accurate enough, the measurements are not done exactly on the circles.

In some cases, it might be more time efficient to make the first three measurements close to each other in such a way that the three locations make up a small triangle. From there, a direction to the UUV can be estimated. This can be useful if the first range estimation is large. If that were to be the case, the UAS would fly to the next measuring point, only to estimate that the UUV is on the opposite side of the initial point. This would waste battery-power to little value. The fact that the UUV will not be on land can be useful to cancel out possible locations.

5.1 Emergency protocol

The UAS will need to have a manual override if something would happen to it, and the control is then taken away from the automated system. Therefore, if someone uses the remote control, the UAS will abort its mission and the operator will have the control. In case of manual intervention, the UAS needs to be reset by the base station, in order to operate autonomously again.

When the UAS looses connection to the base station or when its battery level gets too low, the UAS should fly back to a predefined safety spot and successfully land. This is done, by aborting the mission, elevate until the hydrophone is above water, fly to the landing area and slowly descend to the ground. Since this maneuver may need to be carried out without connection to the base station, the UAS must have this functionality implemented by itself.

There should also be an emergency button on the controller and/or in the GUI. If this button is pressed, all motors should turn off. This is in case if the UAS is in direct danger to any person or object.

6 Signal processing of the emergency signal

To be able to determine the location of the UUV, relevant information is needed to be concluded from the signal emitted by the emergency pinger. This section describes how the signal received by the hydrophone might be dealt with.

6.1 Filtering

The emergency pinger emits only signals in a specific frequency band. If there are problems for the Raspberry Pi to process all the audio samples because the sampling rate is too high, one possible mitigation is to use hardware frequency mixers to shift down the signal in frequency. This, along with good hardware and software compensations filters, could result in a much lower sampling rate being needed and putting less computational stress on the Pi.

Another possible good idea is to experiment with an anti-alias filter if the sound card does not already include this. A digital filter could also be applied to compensate for the hydrophones characteristics if they are known, in order to improve the correctness of the measured signals.

Using a band pass filter around the known emitting frequency is also a good idea to reduce unwanted noise.

6.2 Signal estimation for stationary submersion

This section deals with different strategies for estimating the position of the UUV when the UAS is submerging the hydrophone at a stationary point.

6.2.1 Doppler effect - stationary submersion

The Doppler effect takes advantage of how the frequency is perceived while moving. Basically, the frequency of a signal will be observed differently if the observer is moving relative to the transmitter. The relationship between transmitted frequency f, and received f' is

$$f' = f\frac{v \pm v_R}{v \pm v_T}. \tag{2}$$

In equation (2), v is the velocity of the signal, v_R is the speed of the receiver relative to stationary and v_T is the same for the transmitter. The velocity v of the signal is the speed of sound in the present medium, in this case water. Since the UUV is the transmitter of the emergency signal here, it can be assumed that $v_T = 0$.

In a scenario when the UAS is submerging the hydrophone with constant velocity vertically, the Doppler effect may be used. For example, consider two different positions for the hydrophone, p_1 and p_2. Further, assume that the UUV is located stationary at position p_0 and that $||p_1 - p_0||_2 < ||p_2 - p_0||_2$. Then, when the UAS is submerging the hydrophone vertically with constant velocity, the relative velocity between p_1 and p_0 will be greater than between p_2 and p_0. Therefore, according to equation (2), the hydrophone will measure a higher frequency for the case when the hydrophone initially was located at p_1 compared to p_2. In this manner, one can tell which submerging point that was closest to the UUV.

6.2.2 Received time difference - stationary submersion

Figure 5 displays a simplified version of how soundwaves propagate through a medium, in this case water. The emitted signal from the emergency pinger is a soundwave; therefore, it will propagate in the shape of a sphere. Evidently, the further away from the emitter, the greater time the signal has been traveling. An illustration of this can be seen in figure 6. The distance U $\rightarrow$ A is longer than U $\rightarrow$ C and U $\rightarrow$ B.

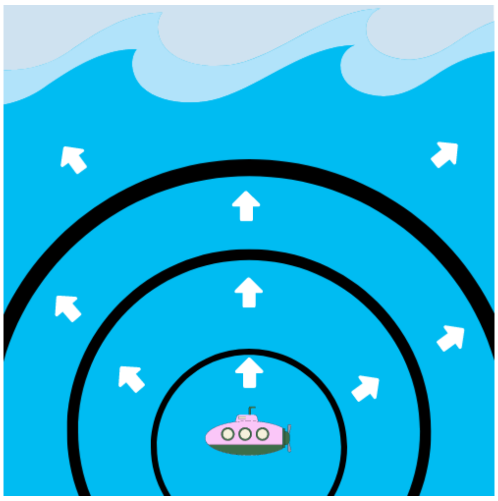
Figure 5: The propagation of soundwaves in water

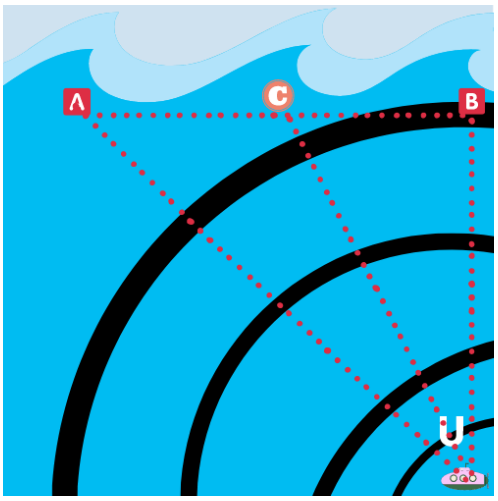

Figure 6: The propagation of soundwaves in water with distance

With the measurements A, B and C in figure 6, a phase shift of the static pulses emitted from the transmitter will arise. This is due to the increase of length traveled for the signal. If a positive phase shift occurs between two measurements, then the second measurement is further away from the origin. And the opposite if the phase shift is negative. An example of a phase shift between two different measurements can be seen in figure 7.

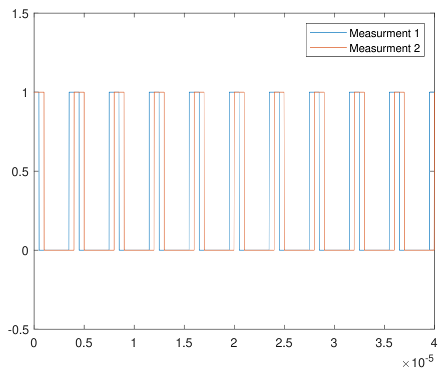

Figure 7: Example of a phase shift of two measurements

To be able to measure a change of phase for the emitted signal, a couple of things are needed to be fulfilled. The emergency pinger must emit its signal with a consistent and accurate rate of pulses. The receiving computer, a Raspberry Pi in our case, will need to keep track of its internal clock with a sufficient accuracy.

6.2.3 Received signal strength - stationary submersion

The idea of this approach is to measure the received signal strength (RSS). The RSS is determined by equation,

$$y = P^0 - \beta \log(\|x - p\|) + e. \qquad [5] \tag{3}$$

P_0 is emitted signal strength, β is the path loss coefficient, x is the target position, y, is the measurements from the sensor and e is the measurement noise.

A distance with equation (3) can be calculated between a measurement and the emergency pinger. If then several measurements are made, a triangulation can be configured to give an estimation of the transmitter's location. Figure 8 with the measurements: A, B and C depicts how a triangulation of signal strength can be made. The red dot represents the transmitter.

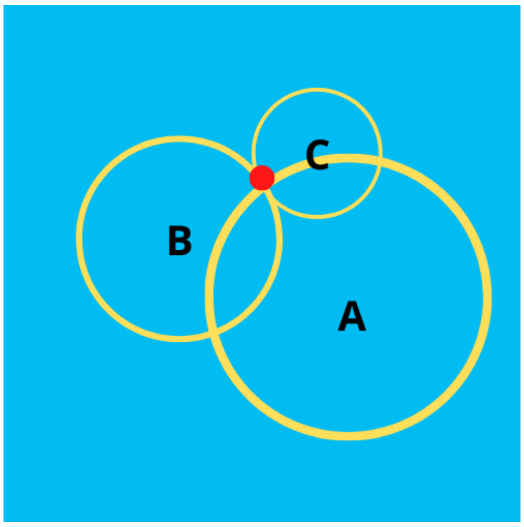

Figure 8: Triangulation of three measurements

6.3 Signal estimation for mobile submersion

The strategies outlined below require the UAS to be able to fly horizontally with the hydrophone tailing submerged. An explanatory picture of the UAS with the tailing hydrophone is seen in figure 9.

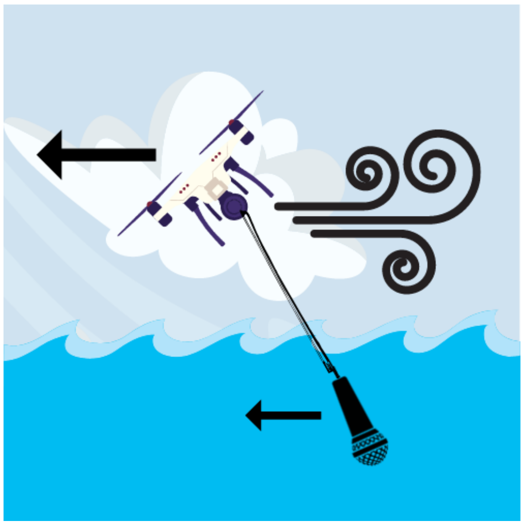

Figure 9: A sketch of the UAS flying with the hydrophone submerged

6.3.1 Doppler effect - mobile submersion

When the UAS fly horizontally with the hydrophone towed in the water, the Doppler effect may be used in a different, but similar approach compared to what was explained in section 6.2.1. Assume that the hydrophone is moving in some direction with a velocity v. Then, if the hydrophone is moving towards the UUV which transmits the emergency signal, the hydrophone will measure a higher frequency than the emitted signal according to equation (2), since v_r is positive in this case. In contrast, the hydrophone will measure a lower frequency if it is moving away from the UUV. Therefore, this strategy enables the UAS to calculate in which direction the UUV is located.

6.3.2 Received signal strength - mobile submersion

When the UAS has the hydrophone tailing submerged, the received signal strength explained in equation (3) can be used. The difference of the technique explained in section 6.2.3, is that the UAS can determine if it is moving towards the UUV or not depending on how the strength of the signal changes.

7 Motion planner

The motion planner is in control of how the UAS is supposed to fly in between the different submerging points, i.e., planning the route. It is also responsible for planning the motion when descending and ascending during hydrophone submersion. Since the environment is essentially free of obstacles, the motion between points will most probably be a straight line. The motion planning program will be run on the Raspberry Pi.

The motion planner will use the PX4's predefined mission mode, which calculates a straight line in between decided submerging points.
To calculate how the UAS is supposed to fly in between these points the mission mode uses a feed-forward PID controller where the position, velocity and accelerations are reference signals to the system and then calculates the heading, see Figure 10.
These points may be updated to the motion planner during the flight, and then the mission mode recalculates the route for the UAS.

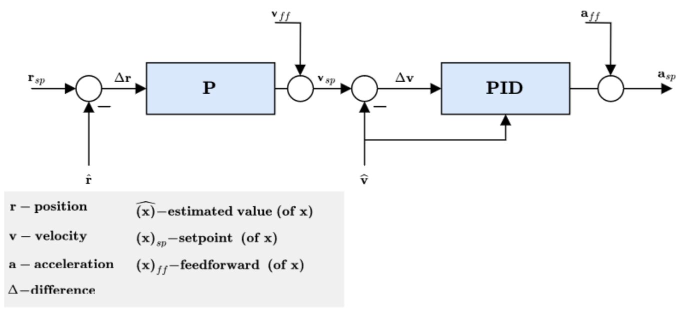

Figure 10: The figure describes the PID controller for the calculated path the UAS should fly in dependent on the position, velocity and acceleration and where the next predefined locations is at. Figure from [4].

8 Control algorithms

The purpose of the control algorithms is to make sure the UAS follows the trajectory given by the motion planner. It is also responsible for basic flying characteristics like; moving, starting, landing, keeping the UAS stable and also lowering the UAS, resulting in submerging the hydrophone in water. Its software will be run on the Raspberry Pi.

The existing control algorithm implemented on the control board for the UAS is a mix between P and PID controllers, see Figure 11. The position and angle are controlled by P controllers while the velocity and angular velocity are controlled by PID controllers. In these control loops, the constants can then be changed according to how the UAS is needed to be controlled for the new dynamics.

To be able to fly the UAS with the extra weight and dynamics of the cable with the hydrophone and sound card, the control algorithms for the UAS may need to be updated to suit the new situation. The existing control system will therefore be modified to be more restrictive to not lose control, since the hydrophone cable will be prone to swinging. If oscillations occur in the hydrophone cable, the derivative gain in the angular velocity controller (see Figure 12) will be decreased to decrease the oscillations. If this is not enough, the proportional gain will be decreased.

Lastly, if these solutions does not work, a new PID controller with new constants will be needed to be developed instead of only trying to tune the existing one. A step response for the angular velocity of the UAS with the hydrophone attached will be produced using the IMC by flying the UAS manually. This will be used to produce a simple model of the new system, see equation (4). The Internal Model Control (IMC) method will then be used to tune the PID controller, see equations (5) and (6).

$$G(s) = \frac{K_p}{sT+1}e^{-sL} \tag{4}$$

$$F(s) = K(1 + \frac{1}{T_i s} + T_d s) \tag{5}$$

$$K = \frac{T + L/2}{K_p(T_c + L)}, \quad T_i = T + L/2, \quad T_d = \frac{TL}{2(T + L/2)} \tag{6}$$

The control algorithms might also be extended to be able to fly with the hydrophone in the water, for position estimation of the UUV using the Doppler effect. In that case, the control system will have to be tuned to consider the dynamics of the resistance force from the water on the hydrophone. The methodology for this will be the same as the previous paragraph.

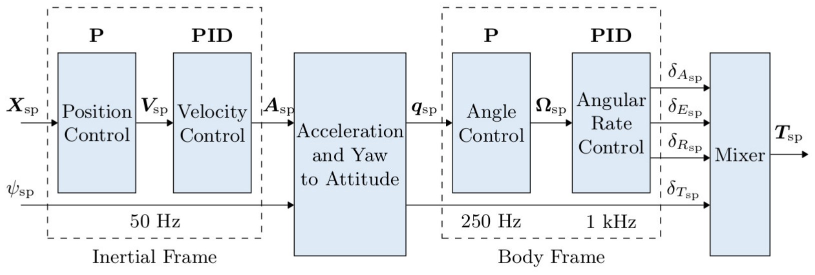

Figure 11: The preinstalled control architecture of PX4 multicopters. Figure from [4].

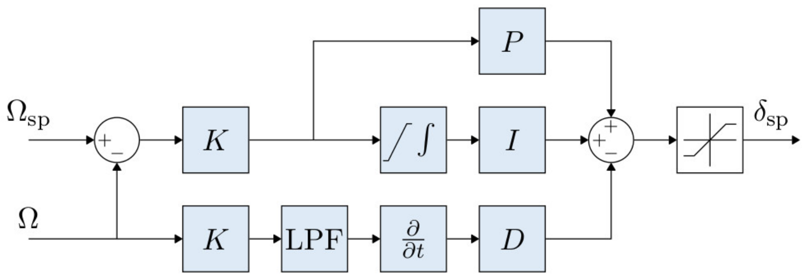

Figure 12: The controller used for controlling the angular velocity. Ω is the angular velocity and Ω_{sp} is its setpoint/reference. The output, δ_{sp} is the reference for the actuator. Figure from [4].

References

[1] As-1 hydrophone. URL https://www.aquarianaudio.com/as-1-hydrophone.html.

[2] Radiomaster radio comparison chart. URL https://www.radiomasterrc.com/blogs/radiomaster-press/radiomaster-radio-comparison-chart. Accessed: 22-10-03.

[3] X500 v2 full kit. URL http://www.holybro.com/product/x500-v2-kit/. Accessed: 22-10-05.

[4] Px4 user guide - controller diagrams. URL https://docs.px4.io/main/en/flight_stack/controller_diagrams.html.

[5] F. Gustafsson. *Statistical Sensor fusion.* Studentlitteratur, 2018.

[6] Transportstyrelsen. Drönare, 2022. URL https://transportstyrelsen.se/sv/luftfart/Luftfartyg-och-luftvardighet/dronare/. Accessed: 22-10-05.

[7] *NEO-M8 u-blox M8 concurrent GNSS modules Data sheet.* u-blox, 9 2021. R11.