Sensur av hovedoppgaver

Universitetet i Sørøst-Norge
Fakultet for teknologi og maritime fag

Prosjektnummer: **2020-06**
For studieåret: **2019/2020**
Emnekode: **SFHO3201-1 19H Bacheloroppgave**

Exhaust flow mapping
Kartlegging av strømninger i eksos

Utført i samarbeid med: Siemens AS

Ekstern veileder: Mahender Billam

Sammendrag: Vi fikk i oppdrag å designe en innovativ og effektiv måte å validere mønsteret av eksosgassen til en gassturbin. For å oppnå dette ble det brukt eksisterende teknologier basert på varmeveksler. Systemet produserer en kvalitativ beskrivelse av flytmønsteret. Arkitekturen av systemet er slik at man klarer å unngå å utsette elektriske komponenter for skadelige omgivelser.

Stikkord:
- Gas turbine
- Simulation
- Experiment

Tilgjengelig: JA

Prosjekt deltagere og karakter:

Navn	Karakter
Evelina Anita Iversen	
Helge Sondre Ulberg	
Ole Henrik Flesaker	
Stian Bergstrøm	
Sondre Fuhre Orsten	

Dato: 15. juni 2020

____
Jamal Attaya Safi
Intern Veileder

____
Karoline Moholth
Intern Sensor

____
Øivind Eidsmoen
Ekstern Sensor

FFA
Fluid Flow Analytics

Bachelor's thesis documentation

Evelina Anita Iversen
Helge Sondre Ulberg
Ole Henrik Flesaker
Stian Bergstrøm
Sondre Fuhre Orsten

06.01.2020 - 25.05.2020

Revision History

Revision	Date	Author(s)	Description
1.0	10.01.2020	Stian B.	Document created
2.0	29.01.2020	Group	Documentation for first presentation
3.0	24.03.2020	Group	Documentation for second presentation
4.0	25.05.2020	Group	Documentation for third (final) presentation

Abstract

We, Fluid Flow Analytics, came up with an innovative and effective way of confirming the fluid flow pattern within the exhaust of a gas turbine, compared to computer generated images. FFA re-imagined existing technologies to design a measuring module based on a cross-flow heat exchanger. Our system manages to provide a qualitative description of exhaust flow pattern, utilizing available interfaces. The system's physical architecture avoids exposing electrical equipment to exhaust gas directly, resulting in a cost-efficient solution.

Contents

Chapter revision

Chapter	**I1**	**E1**	**E2**	**C1**	**C2**	**C3**	**T**
Project Model	V1	V2					
Risk	V1		V2	V3		V4	
Test	V1		V2		V3	V4	
Requirements	V1	V2					V3
Iterations		V1				V2	V3
Conclusion							V1
Appendix figures and models	V1	V2		V3		V4	
Appendix Mechanical		V1	V2		V3	V4	
Appendix Software			V1	V2		V3	V4
Appendix Electrical			V1			V2	V3
System's evolution				V1		V2	
Appendix system visualization				V1	V2	V3	

Glossary

Anaconda A platform for data science, in this case the site Spyder and Python was downloaded.

ATEX ATEX directive is a directive that describe the working environment in a explosive atmosphere.

Comma-Separated Values A file format.

Computational fluid dynamics branch in fluid mechanics that uses numerical analysis to analyze fluid flow.

Coronavirus Coronavirus 2019 is an infectious disease.

Discord Discord is an online communication platform.

Eddy in fluid dynamics Swirling of a fluid.

Excel Microsoft Office Open XML Format Spreadsheet file .

Fluid Flow Analytics The name of our group.

Gas Turbine package The whole product Siemens provides.

Governing equation A governing equation describe how the unknown variables change with variables that are known. Mathematical model build of this equation describe a physical phenomenon..

Graphical user interface A subset of UI, often pertaining to icons on a screen or other graphical tools.

imc FAMOS imc FAMOS is a data analysis program for evaluation and visualisation of experimental results.

Integrated development environment An environment for programmers to write, debug and normally compile code.

Object Oriented Programming Computer programs that allows the user to define data types and data structures.

OPC Open Platform Communications.

OrCAD OrCAD is a company that deliveres electrical schematics, simulation and PCB editor.

Programmable logic controller A PLC system.

Proportional-integral-derivative controller A PID controller is a controller that stabilize a signal after three parameters; proportional, integral and derivative.

Python A programming language.

QT An IDE, pronounced cute.

SolidWorks SolidWorks is a modeling computer-aided design and engineering program.

Spyder A scientific Python IDE.

Technical Data Management Streaming .

Telegram Telegram is an online communication application.

Test kit Our system.

Thermistor Thermistor is a sensor that measure temperature, the temperature changes with the resistance in the thermistor.

Thermocouple Thermocouple is a sensor that measure temperature, the temperature is measured with change in the voltage in the thermocouple.

TIA selection tool TIA selection tool is a program by Siemens to choose PLC modules.

Unified Modelling Language A standardized modelling language for software design.

User Interface The connection between human interaction with machine.

Whorl in fluid dynamics Geometrical pattern made of spirals and concentric circles.

Zoom Zoom is an online communication application used for video conferences.

Abbreviations

3D Three Dimensional.

ADC analog digital converter.
AUP Unified Modelling Language.

CFD Computational Fluid Dynamics.
CPU central processing unit.
CSV Comma Separated Values.

FFA Fluid Flow Analytics.

GT Gas Turbine.

HSE Health, Safety and Environment.

NaN Not a Number.

PLC Programmable logic controller.
PNG Portable Network Graphic.

RUP Rational Unified Process.

SMD surface mounted device.

TDMS Technical Data Management Streaming.

UML Unified Modelling Language.

WHRU Waste Heat Recovery Unit.

Nomenclature

Symbol or operator	**Explanation**
∇	Nabla oprator: $\vec{i}\frac{\partial}{\partial x} + \vec{j}\frac{\partial}{\partial y} + \vec{k}\frac{\partial}{\partial z}$
m	Mass
$\vec{a}$	Acceleration
$\vec{F_v}$	Viscous force
L	Length scale
τ	Shear stress
ρ	Density
$\vec{v}$	Velocity
μ	Dynamic (absolute) viscosity
T	Temperature
t	Time
$\dot{m}$	Mass-flow rate
c	Specific heat
A	Area
k	Thermal conductivity
Re	Reynolds number
Nu	Nusselt number
Pr	Prandtl number
R	Thermal resistance
U	Overall heat transfer coefficient
h	Heat transfer coefficient

1. Introduction

1.1 Fluid Flow Analytics

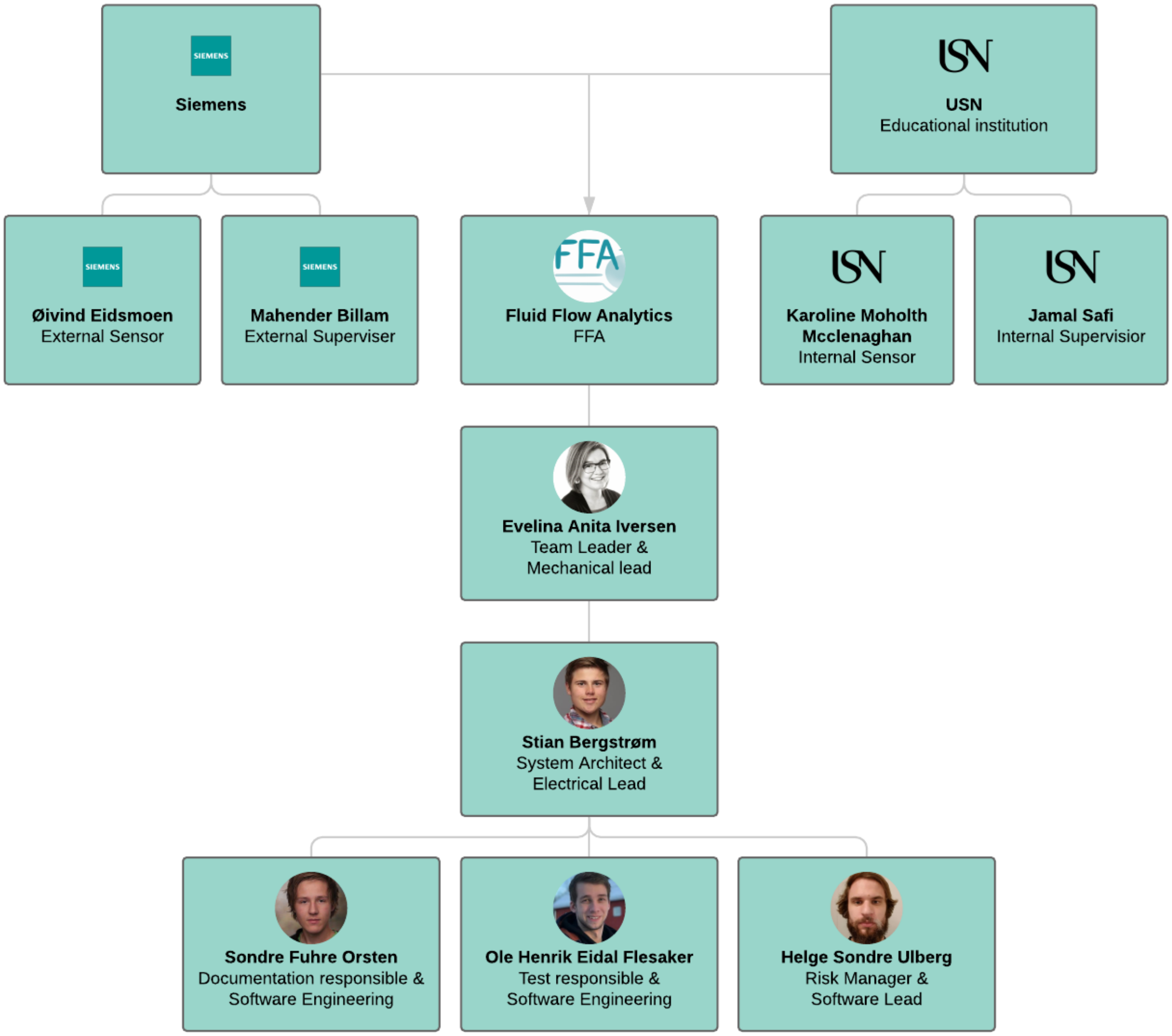


Figure 1.1: Project team chart

1.2 Siemens and an introduction to the problem

For our bachelor's thesis project FFA have received a task from Siemens Oil and Gas (client) to map gas-turbine's exhaust flow. This mission implies creating a complete "test kit" for installation on gas turbine packages on site and/or during other testing procedures.

"Siemens Oil and Gas is a global powerhouse focusing on the areas of electrification, automation and digitalization. One of the world's largest producers of energy-efficient, resource-saving technologies, Siemens is a leading supplier of systems for power generation and transmission."[1]

One of the clients' core businesses is designing and constructing gas turbine (GT) driven power generators (Figure 1.2). They design and build "packages" where the GT is connected to the driven equipment, complete with all required support systems: fueling systems, lubrication systems, noise absorbing systems, etc.

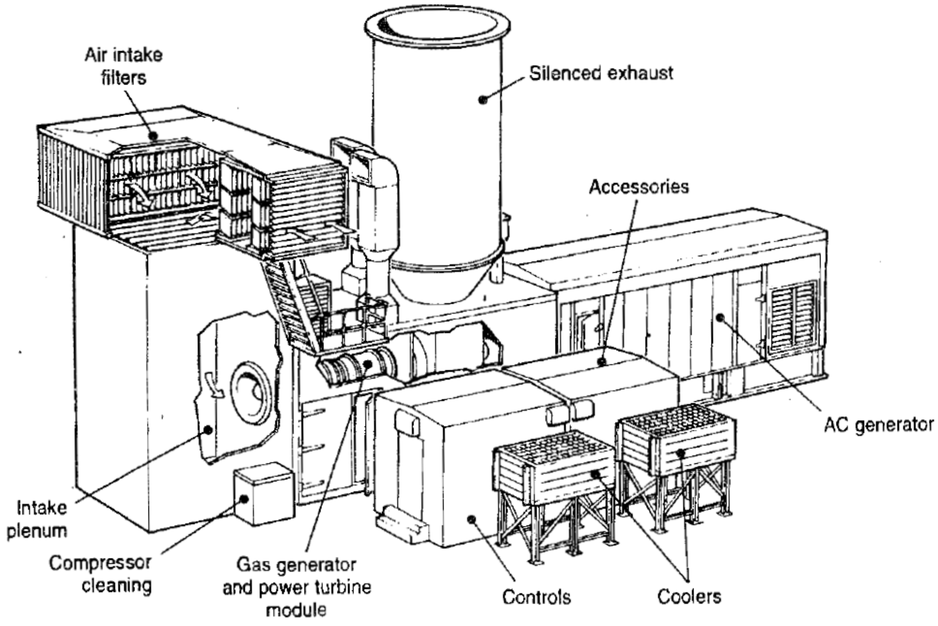


Figure 1.2: Compact generating set [2]

The turbine used by the client in our study is an industrial version of the Rolls-Royce RB211 series jet engine. Designed in the 70's originally for commercial aircraft propulsion and now with more than 750 units sold worldwide. The RB211 is today delivered in several versions with a maximum power output above 34MW. Additionally GT is known for extremely high efficiency when also measured with the respect to a power to weight ratio. Therefore GTs are commonly used for power generation in the Oil and Gas industry.[3]

Power generating package delivered by the client is a complex system, where chemical energy stored in fuel transforms into mechanical energy that generates electrical power. GT burns mixture of fuel and compressed air in combustion chamber, further hot products of combustion are forced through turbine, pushing on the blades attached to high-torque shaft that drives an electrical generator.

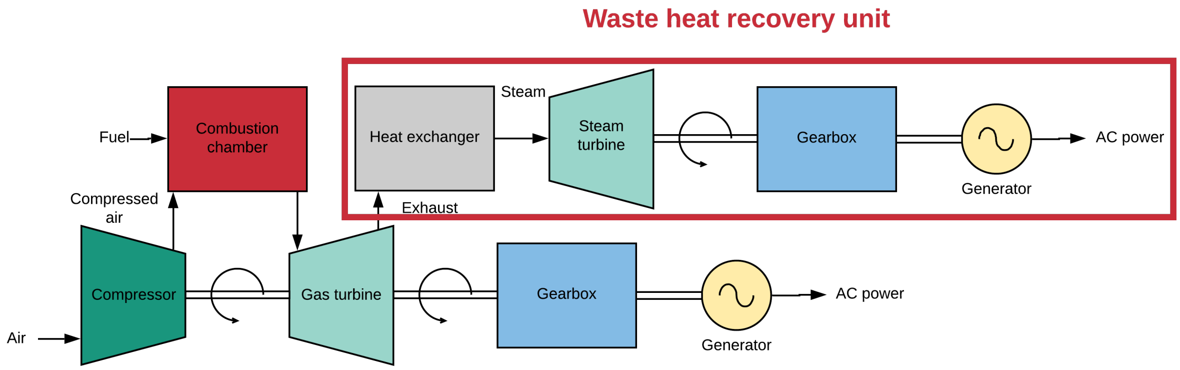


Figure 1.3: Concept of operation for gas turbine generator with waste heat recovery unit

GT generators exist in two main configurations: simple systems and combined cycle systems. Simple system consists directly of GT driving a generator, where exhaust gases flow directly out into the atmosphere through the stack. Combined cycle system implies a second-stage waste heat recovery unit (WHRU) for improved efficiency (Figure 1.3). There hot exhaust gases enter a heat exchanger that will give a rise for a working fluid that will drive a steam turbine, generating more electrical power.

GT generator package has a complex multidisciplinary design procedure, which implies aerodynamics, thermodynamics, mechanics, fluid dynamics, control systems development etc.[2] Design point studies in each discipline include detailed calculations. For the thermo-fluidic processes those are carried out using Computational fluid dynamics analysis (CFD). This technique takes into consideration physical properties of both working fluid and its environment. Mathematical models used in CFD vary according to application, providing intuitively understandable visualisation of the systems properties. An example of such a report for GT's exhaust duct can be seen on Figure 1.4.

However, CFD is a mathematical approximation of a real-world physical phenomena. Therefore any system's stakeholder may be concerned if the results obtained truthfully represent the system's behavior, hence there is a need to verify and validate CFD findings.

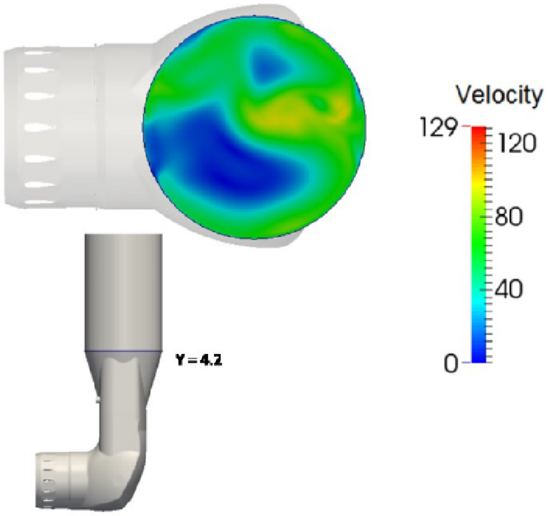


Figure 1.4: CFD analysis GT's exhaust duct. Velocity visualisation [3]

1.3 Problem description

The main study of this paper is mapping a flow profile in gas turbine exhaust. The exhaust duct is oriented vertically up and has a diameter of approximately 2 meters. Exhaust gasses form large a flow, with speed and temperature up to 90 kg/s and 550 °C respectively. Ducting downstream typically has two main arrangements:

- Silencer and stack (Figure 1.2)
- Silencer and WHRU (Figure 1.5)

From previous design studies provided by the client (Figure 1.4) it has been discovered that exhaust flow is turbulent with uneven velocity and pressure distribution over the outlet surface. However, those mathematical models are yet to be validated with respect to real-life conditions. That means that design decisions made based on performed CFD analysis are maybe not entirely satisfactory for real-world system.

In general, uneven flow can cause significant problems during operation of the GT package:

- Structural vibrations
- Cracking of the exhaust system
- Stress and possible cracking in the WHRU
- Inefficient utilisation of surface area of WHRU

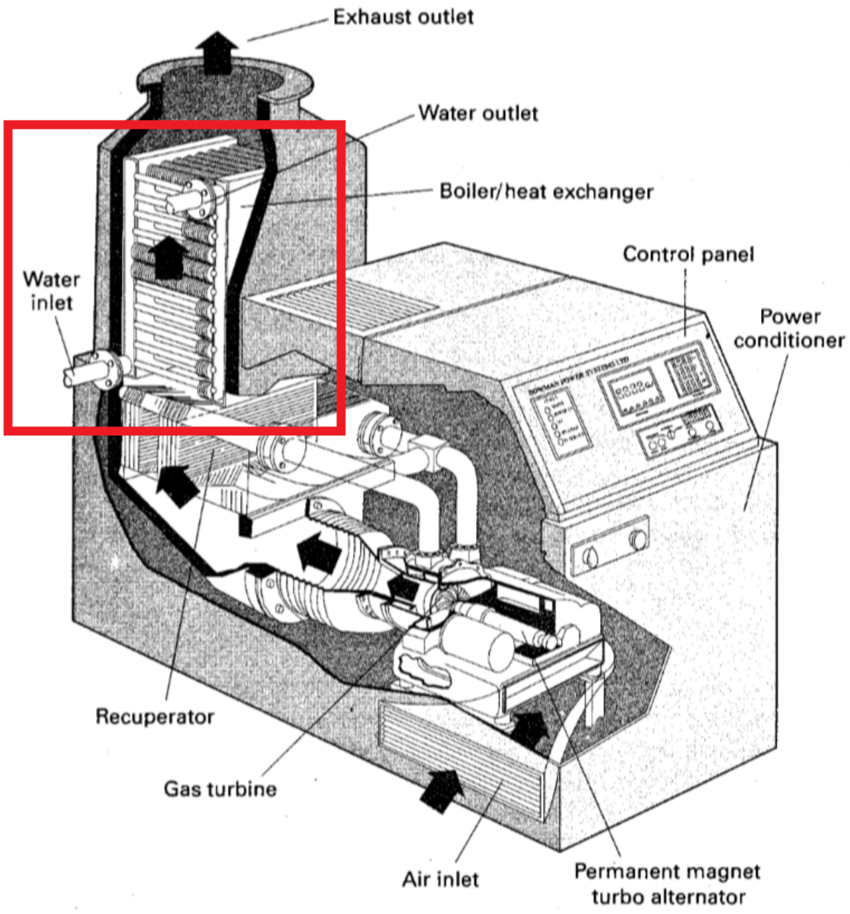


Figure 1.5: Gas turbine package with waste heat recovery unit[2]

- Thermal cracking of the exhaust system due to hot-spots formation

Measuring the actual exhaust flow in real-life conditions is believed to give an early indication on credibility of preliminary analysis preformed and show presence of any potential harmful flow. That will give design engineers reliable information to take actions for prevention of possible damage.

1.4 Our mission and knowledge value

CFD simulation is a powerful tool that can be used with confidence to predict fluid behaviour and make design decisions based on it. Essentially, one presents a mathematical model of a physical phenomena - usually a system of non-linear partial differential equations - then transforms it into a computer code and performs a simulation that yields data used in a further design. CFD simulation's credibility is evaluated through verification and validation assessment. Verification assessment is based on comparison of yielded values with exact analytical calculations. Validation on the other hand examines CFD simulation with respect to experimental results [4].

FFA assumes that verification of performed simulations is assessed by the client. This means that we believe that software used for CFD analysis of exhaust flow is credible and yields accurate results with respect to numerical calculations and initial conditions. As it has been mentioned previously, validation of the simulation is yet to be done, hence it is difficult to use CFD results to address issues rising in exhaust-stack and/or in WHRU.

FFA's mission in this project is to make an attempt in finding a way to validate CFD simulations of exhaust flow supplied by the client. As a result of a successful mission we will provide a traceable bond between theory and experiment, evaluating possible inaccuracies, making computational analysis reliable for further use in future design studies. This way client will become more certain in their product's thermo-fluidic behavior. It is expected that our system will help to make experimentally-supported engineering choices, perform reliable end-of line tests in testing facilities and run diagnostics for packages installed on site.

1.5 Solution domain and general vision

In validation of a CFD it is important to understand characteristics of the flow and the environment, where the flow is studied. In collaboration with the client it has been decided that flow profile mapping implies assessment of velocity, flow direction, pressure and pressure pulsation over the outlet surface of a GT exhaust. FFA's vision of a possible solution is a physical equipment tailored to extract experimental data in a form that is suitable for

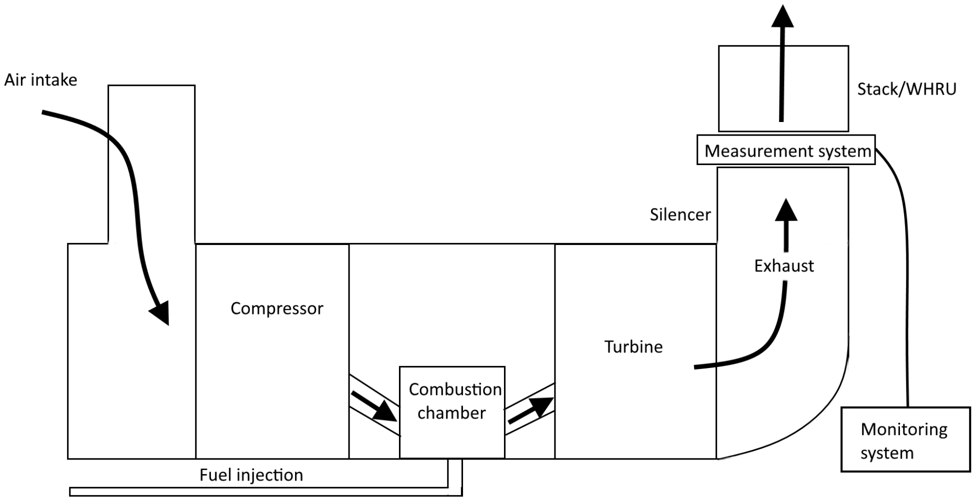


Figure 1.6: Concept of operation

comparison with existing CFD. Even though, one can assume that there is a wide range of possible solutions to the problem, the resulting system shall be designed under stiff constraints, both from operation's environment and clients wishes.

The device that we wish to design shall be in a form of an independent and portable test kit, that will include measurement instruments, monitoring and data storage unit together with all interconnections between interfaces necessary to run the test. As it is illustrated in Figure 1.6 we see our product with measurement system installed inside the exhaust duct and monitoring unit outside.

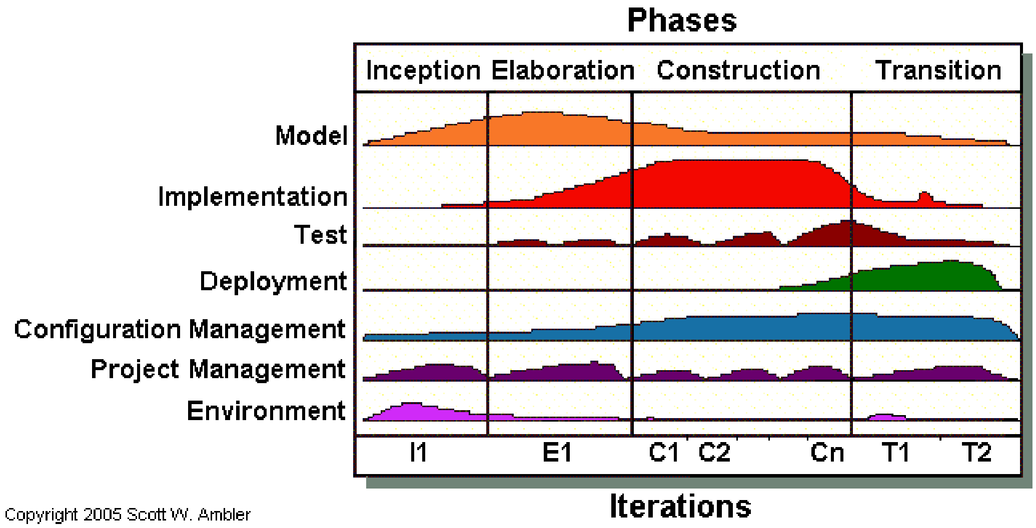
It is expected that FFA will provide an overall architecture of the complete system with user manual for installation and operation. Systems architecture additionally shall imply:

- Detailed study on methods used to measure the flow profile in the GT exhaust outlet, followed by selection of preferred methods and instrumentation with supplier data sheet and documentation.
- Evaluation of the impact of the Measurement System on the flow being measured.
- Evaluation and selection of a method of signal transfer and signal conditioning from Measurement System to Monitoring System.
- A full-scale 3D-model of the Measurement system.
- A 3D-printed model of the Measurement system.

- Evaluation and verification the required cycle time for data collection .
- Programming the Monitoring System for operation.
- Programming the Monitoring System for visual display of result.

2. Project model

An important part of a successful project is how the project modeling has been done. The project model is going to lay the groundwork on how the workflow and tasks should be done. The model that the group decided to use was AUP, which stand for agile unified process. It was chosen based on the attributes of an agile version of the unified process. The group will therefore have an opportunity to make mistakes throughout the project life cycle. The unified process is an iterative and incremental process that has been split into four different phases. During these phases there will be differing disciplines with a varying degree of focus. This can be seen in the Figure 2.1. The figure show the iterations on top and the disciplines down on the left. A more thorough description of the phases and disciplines can be found in [5]. This file is an index describing the AUP model and its different parts and aspects.
In our project we have chosen to have two elaboration phases and six construction phases.



Copyright 2005 Scott W. Ambler

Figure 2.1: Agile Unified process

With the assignment being a research and testing thesis. The group sought out a model that suited this need, the AUP works well in this regard. In the beginning of the process the model discipline is the main focus. Here the group will mainly be working on the requirements and the business modeling. In the assignment a great amount of requirements are predefined. With the AUP we can rapidly start working on translating the requirements. The agile approach, offers the group improved workflow as a multidisciplinary group. This stems from one of the project model's attributes, that rather than providing a working code or system at the end. The goal is to continually produce working code and subsystems throughout the iterations. This will make it simpler for the other disciplines, to follow the workflow and progress of the other team members.

2.0.1 Inception phase

In the inception phase the task is to evaluate the problem and start working on the project at a high level. The groups resources will mostly be spent working on the model, project management and environment disciplines. During the inception phase the main tasks are to understand the problem scope, define the requirements and begin the test plan and the risk plan.
The milestones for the inception phase are:

Milestones

- Decide what project model to use.
- Define and understand the project scope.
- Translate the predefined requirements into derived requirements.
- Identify the high level risks and create a risk plan.
- Create a test plan.
- Find out if the system is feasible.
- Have the project plan ready for the elaboration phases.
- Prepare for first presentation.

2.0.2 Elaboration phase

In AUP the main goal of the elaboration phase is to prove the architecture of the system. Architecture means that there has been made a baseline for further development. This should be high quality and satisfy the requirements. The elaboration phase builds on the work and output from the inception. The information and models that have been made in the inception phase may need to be iterated further. The test plan and the risk plan need to be further developed during the elaboration. At this point in the project the essential and critical risks needs to be addressed.

Milestones

- Detailed Risk plan.

- Detailed Test plan.
- Detailed Requirement.
- Prove the architecture of the system.
- Prepare for the construction phase.
- System concept evaluation.

2.0.3 Construction phase

During the Construction phase the main focus is the implementation of the system. The goal is to create and construct the system. In this phase need to finalize the design and start working on the programming and construction of the final system. For the different iterations of the construction phase, all the implementations need to be tested, to ensure that all the required functions and the system works. At the end of the construction phase we have the deadline for the project. In the last iteration of the construction phase we need to finalize the report and the product for the customer and the university.

Milestones

- Continuous acceptance testing.
- Test all the implemented parts continuously.
- Verify that all testing has been addressed.
- Finalize our system.
- Have all documentation ready.

2.0.4 Transition phase

The transition phase is the last phase in the AUP. During this phase the system is going through validation and preparing for production. As the system the group creates, is not being produced, the transition phase is tailored to the team. The transition phase is set after the deadline for submitting the documentation. Therefore, this phase consists of preparation for the presentation of the system.

Milestones

- EXPO.
- Product presentation in front of unbiased reviewers.

2.0.5 AUP diagram

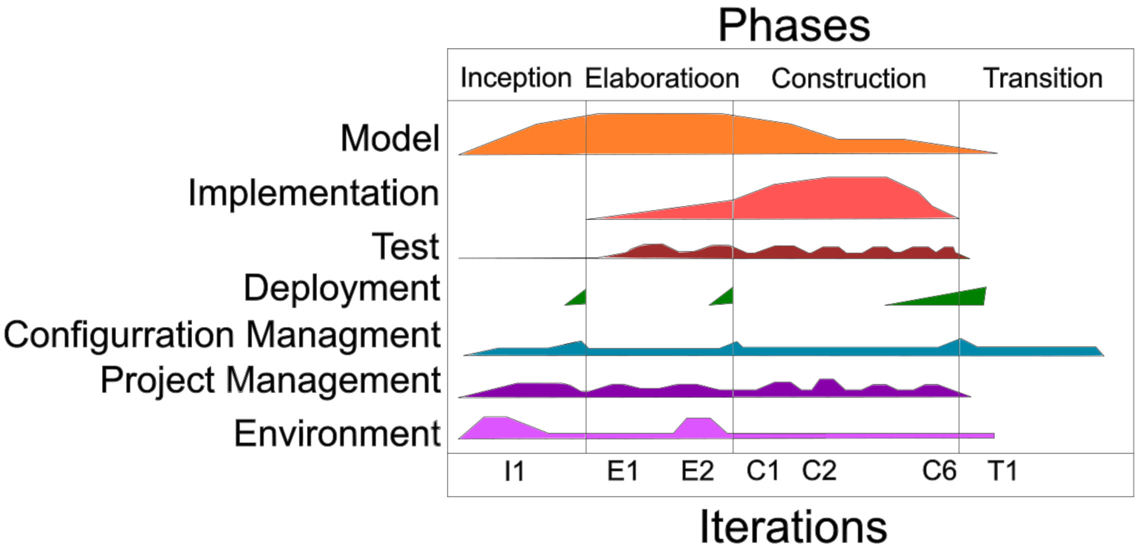


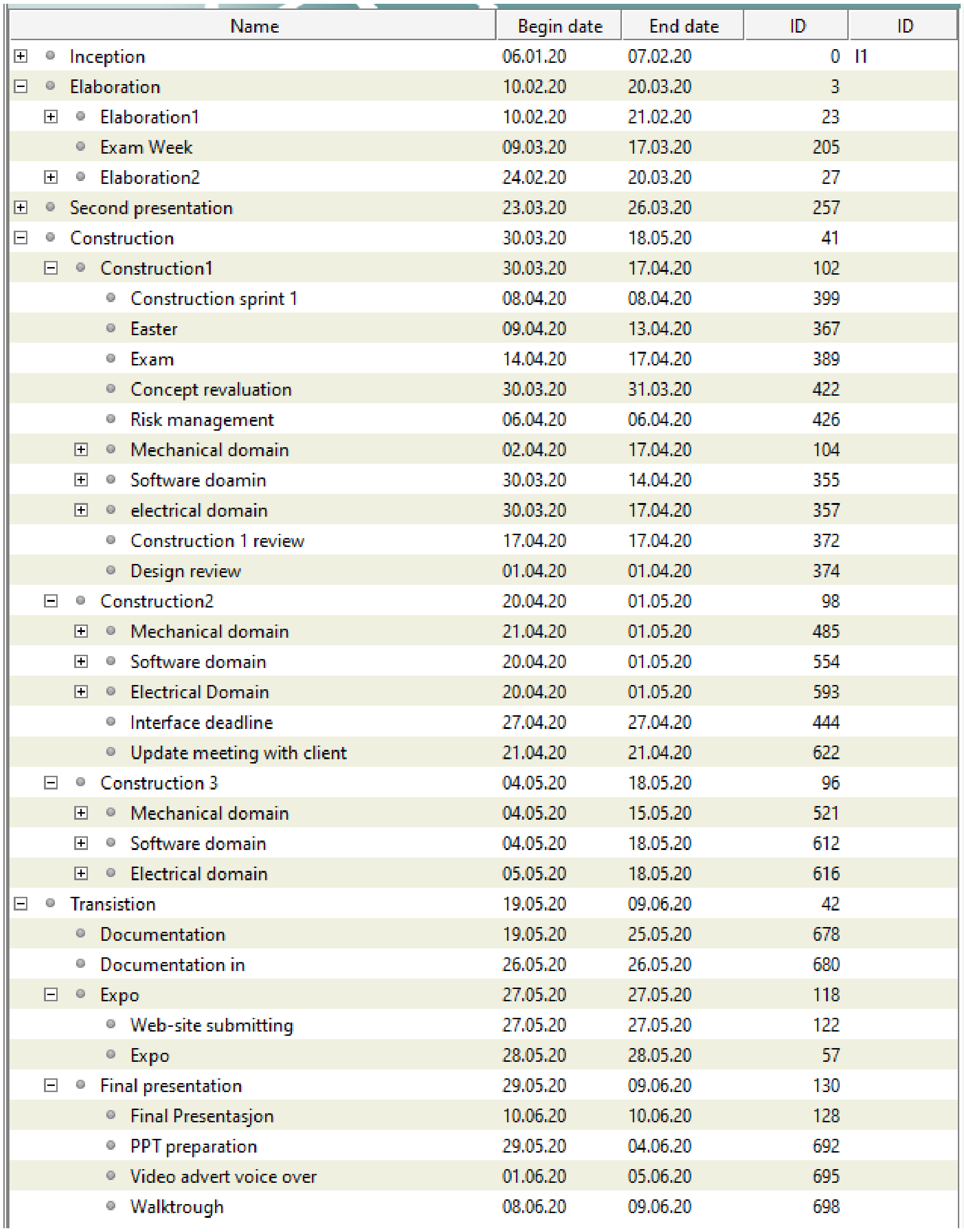
Figure 2.2: AUP model for our project

With the AUP model one do not have to follow the figure 2.1 to the point. Hence, the group modified it, as seen in figure 2.2. There has been made an estimate on how the process will differ from the original. Most notably, there will not be spent mentionable amount of time on deployment. That is because after communication with the customer, it has been decided that the system will not be physically made. The deployment describes the presentations and the delivering of documentation.

2.1 Gantt diagram

To keep the project on track, the team made a Gantt diagram. In figure 2.3 and 2.4 the Gantt diagram can be seen. The lists shows each iteration and a few of the workflows during the iteration. With the Gantt diagram the group has an overview on when different tasks are active. In the Gantt diagram there is an initial plan for when the iterations are taking place. The group has chosen to have two elaboration phases and six construction phases. When starting to plan the construction phase, there was made a change to how many iterations it would consist of, instead of having six iteration, it was merged to three iteration and an extended transition phase.
Different tools have been tried for the Gantt diagram. During the inception phase, a program called Projectlibre was used, however, during the elaboration phase, this was found to not be suitable. Therefore ganttproject, is the program that will be used going forward. In

figure 2.3 the group uses a color code. The color code describe what discipline from the AUP process the task belong to. The colors match with the colors from figure 2.2. In the group's Gantt diagram there are multiple workloads bundled together.



Name	Begin date	End date	ID	ID
Inception	06.01.20	07.02.20	0	I1
Elaboration	10.02.20	20.03.20	3	
Elaboration1	10.02.20	21.02.20	23	
Exam Week	09.03.20	17.03.20	205	
Elaboration2	24.02.20	20.03.20	27	
Second presentation	23.03.20	26.03.20	257	
Construction	30.03.20	18.05.20	41	
Construction1	30.03.20	17.04.20	102	
Construction sprint 1	08.04.20	08.04.20	399	
Easter	09.04.20	13.04.20	367	
Exam	14.04.20	17.04.20	389	
Concept revaluation	30.03.20	31.03.20	422	
Risk management	06.04.20	06.04.20	426	
Mechanical domain	02.04.20	17.04.20	104	
Software doamin	30.03.20	14.04.20	355	
electrical domain	30.03.20	17.04.20	357	
Construction 1 review	17.04.20	17.04.20	372	
Design review	01.04.20	01.04.20	374	
Construction2	20.04.20	01.05.20	98	
Mechanical domain	21.04.20	01.05.20	485	
Software domain	20.04.20	01.05.20	554	
Electrical Domain	20.04.20	01.05.20	593	
Interface deadline	27.04.20	27.04.20	444	
Update meeting with client	21.04.20	21.04.20	622	
Construction 3	04.05.20	18.05.20	96	
Mechanical domain	04.05.20	15.05.20	521	
Software domain	04.05.20	18.05.20	612	
Electrical domain	05.05.20	18.05.20	616	
Transistion	19.05.20	09.06.20	42	
Documentation	19.05.20	25.05.20	678	
Documentation in	26.05.20	26.05.20	680	
Expo	27.05.20	27.05.20	118	
Web-site submitting	27.05.20	27.05.20	122	
Expo	28.05.20	28.05.20	57	
Final presentation	29.05.20	09.06.20	130	
Final Presentasjon	10.06.20	10.06.20	128	
PPT preparation	29.05.20	04.06.20	692	
Video advert voice over	01.06.20	05.06.20	695	
Walktrough	08.06.20	09.06.20	698	

Figure 2.3: Gantt diagram list

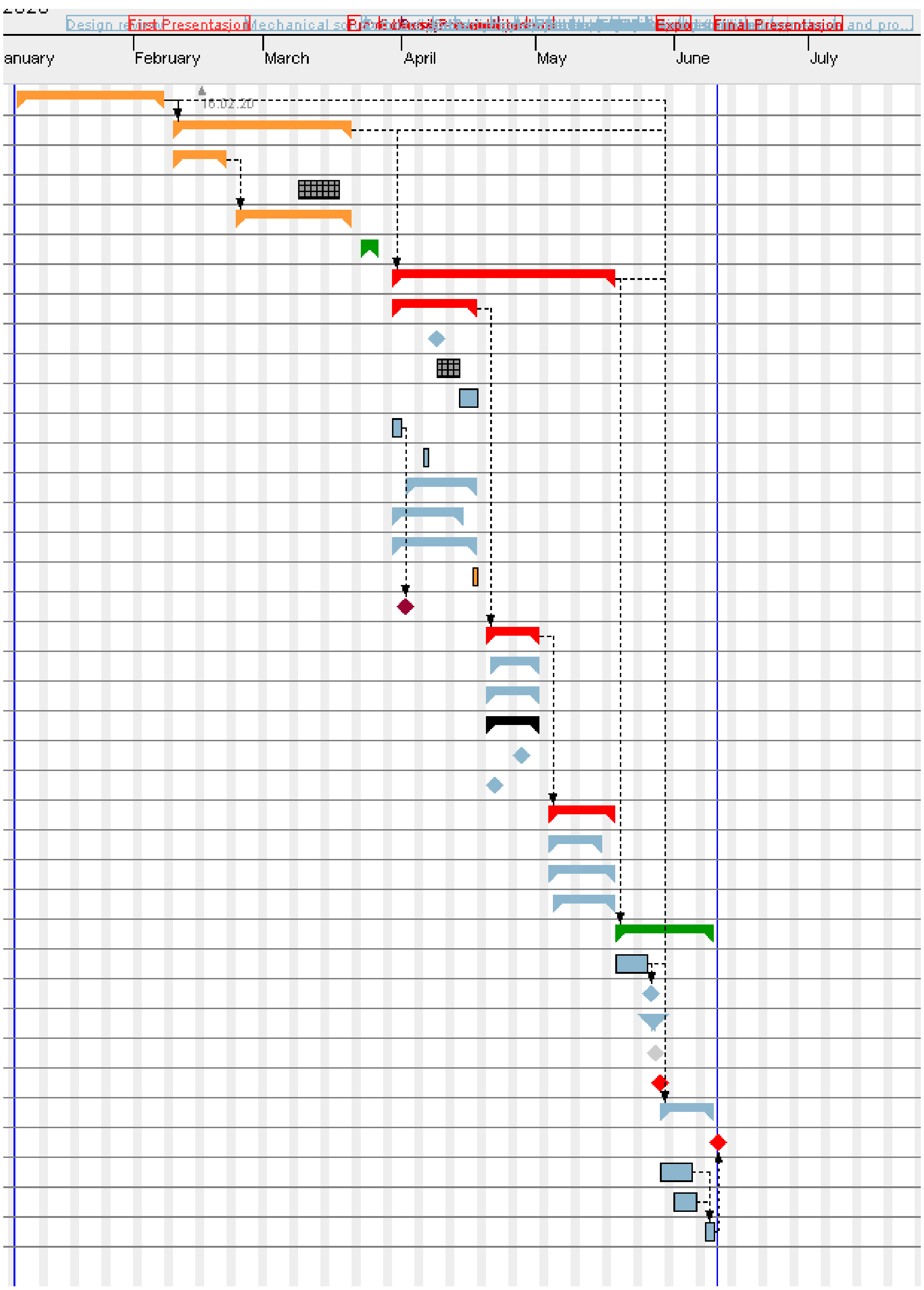


Figure 2.4: Gantt diagram

2.2 System context

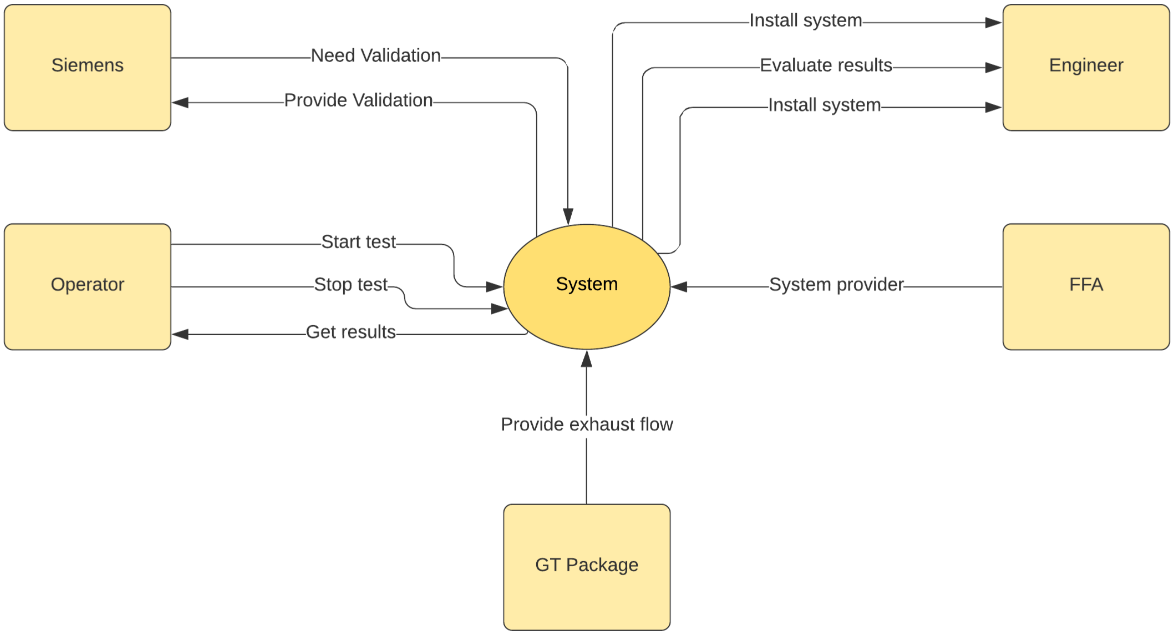


Figure 2.5: System Context Diagram

The system context diagram shows the actor‘s input and output to our system. As show in figure 2.5. Siemens need validation for their GT exhaust to ensure the calculations provided by the CFD is correct. The test kit will provide models/simulations that can be compared with the CFD analysis to validate the CFD.

The operator will be the one interacting with the test kit, starting and stopping the test. The operator will be the one seeing the results in real time. When the test is done the operator will be able to give the results to other actors. Most likely an engineer to evaluate the results. The GT package will provide exhaust flow to the test kit. The test kit will measure the flow and provide data for the operator. FFA will be the actor making the system.

The engineer will be the one installing and removing the test kit before and after testing. He will also be the one evaluating the test to ensure the data gathered from the test kit is correct. And validate with the CFD analyses.

2.3 Use Case

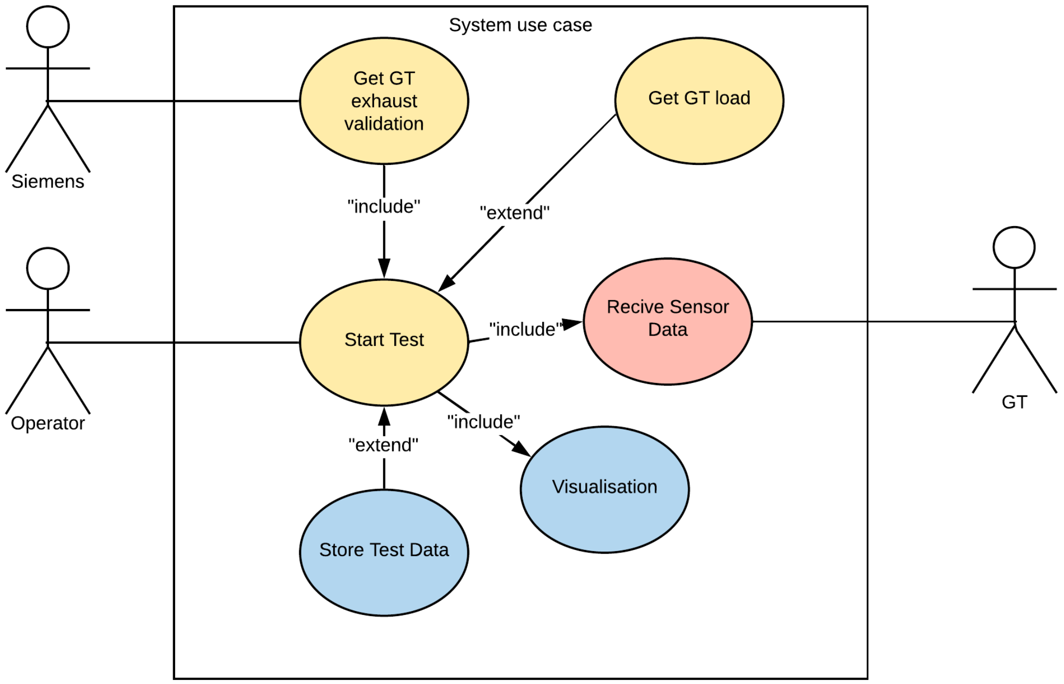


Figure 2.6: System Use case

The use case diagram shown in Figure 2.6 illustrates an overall indication for what the test kit needs, and how the test kit can verify the exhaust flow. The test kit will register data given from the turbine's exhaust and generate data based on it. That data will be then processed and visualized to give an understanding of the exhaust.

The GT is shown as an actor providing exhaust flow to the sensors. Making the sensors register the flow from the GT exhaust. The operator is shown as an actor using our test kit to run the test. Where the functions will be start/stop test, visualize and/or store test data. Then providing Siemens with results. The GT load use case will give information on the load the GT is running at. The test kit will not get the information on what load the GT is running at, therefore this will need to be an input to the test kit.

3. System's evolution

Development of a new data acquisition system for fluid flow study is a challenging multidisciplinary problem. Driving force for our decision making has been built on continuous requirements' testing and risk assessment, communication with stakeholders and specialists. Since the customer's need is to validate pre-conducted CFD analysis, resources were put to make a sufficient study on this tool (see appendices A.1.4, A.2.6 and B.2.1).

As a starting point, research team focused on understanding the assignment and studying existing commercially used technology. This research covered some techniques and underlying physics of different experimental setups, where a significant part of team's attention was dedicated to system's purchasing cost, measuring resolutions and portability (see appendices A.2.3 and A.2.4). Generally, one can divide final product versions into two categories (Figure 3.1): *multi-probe assemblies* (class V.1) and *volume measurements* (class V.2). Despite obvious differences, both classes share the same underlying concept of operation, where experimental data is extracted from the flow by means of different probing systems (see Figure F.4).

3.1 Multi-probe assembly (V.1-class)

V.1 solutions class contains more classical and broadly used data acquisition systems [6]. There is a variety of probes available on the market, that cover a wide spectrum of needs. However, in the context of this paper point-probes assemblies have been found unfeasible, caused by their inability to extract data efficiently in a sufficiently large testing volume (refer to test summaries in table A.2.5).

3.1.1 Pressure based probes (V.1.1)

V.1.1 is the first concept proposed to the customer. It has been found promising, since the system was available for purchase from *"Airflow Sciences Equipment, LLC"*[7] in the pre-assembled test-kit (see figure A.8), that fitted most of the requirements. However, even though suppliers' system collects relatively accurate data with acceptable sampling rate, it was originally designed for flows that do not change pattern in time, which is inapplicable for dynamic flow in exhaust duct. This constraint applies to all data acquisition systems that use pressure based probes [6].

At this point the software architecture was still in its infancy stage. The main focus was on understanding the problem. This was done by discussing within the group and making use

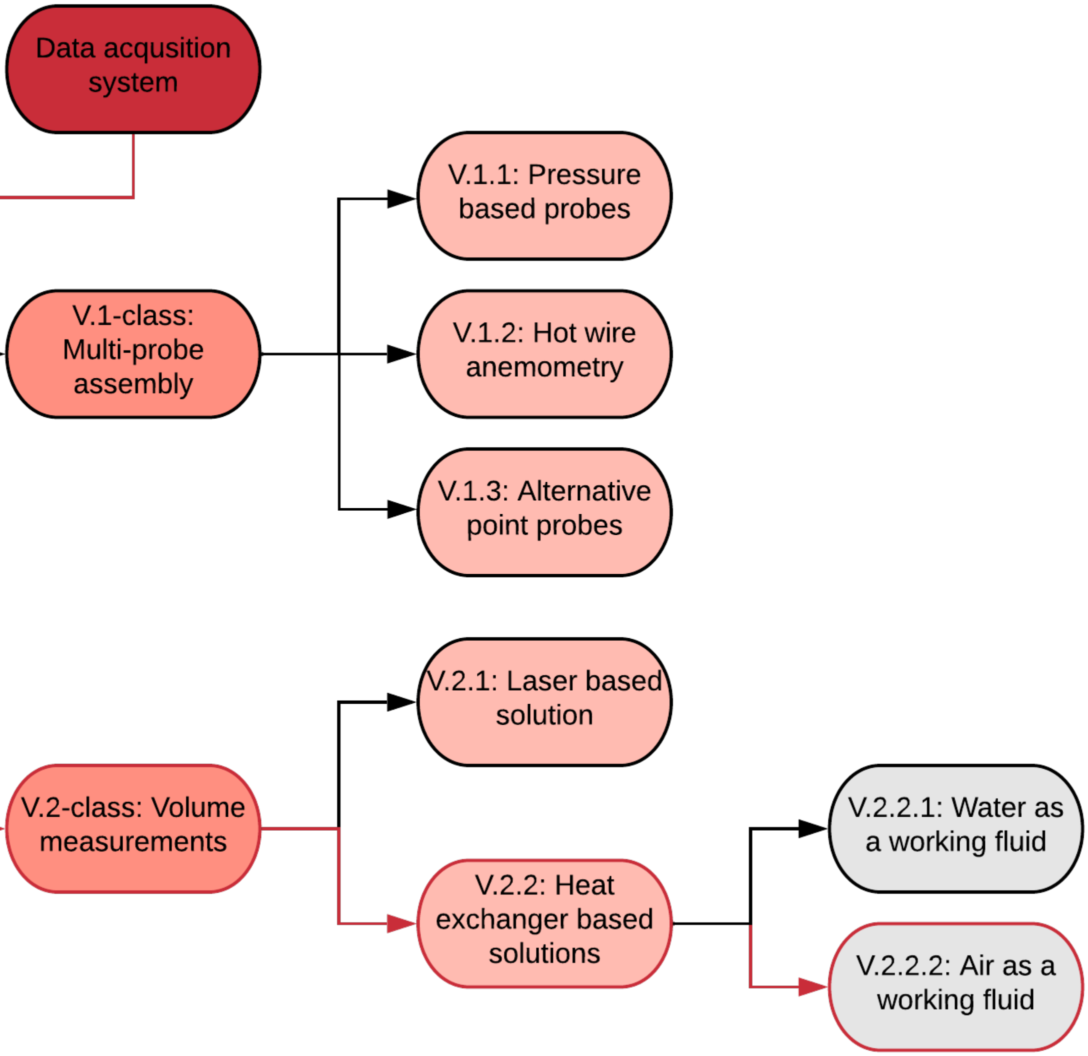


Figure 3.1: Version flow diagram with differing key elements

case diagrams and diagrams related to a projects inception phase. Use cases are important at the starting point, to create a common understanding on what the software will do and what it should do. (see F.3) for one of the earliest examples.

3.1.2 Hot wire anemometry (V.1.2)

V.1.2 version in its core is close to previous one. Hot-wire probes are accurate instruments (see figure A.4 and A.5), that are robust and measure three-dimensional speed of the fluid [6][8]. Never the less, changing type of sensor in multi-probe assembly doesn't affect the fact that it would be hard to measure the whole cross-section of the duct, using single probes. This would require purchasing a great amount of probes to generate a representable picture, which will spike a purchasing cost of the system dramatically, making it incompatible with some of the more expensive volume measurement systems.

One could argue however, that it is possible to analyse patterns provided by the customer, to narrow down measuring points to absolute minimum. Research team does agree with it, and conducted this study in a frame of V.2-systems (appendix B.4.1). Position of each sensor for multi-probe assembly has to be calculated such as it will be comparable with dimensions of the sensor itself and its measuring volume (appendix A.2.4). Achieved precision however, will be of no use, since the flow pattern would be averaged afterwards, because it is impossible to cover the whole area with point-probes without co-intrusion and disturbance of the surrounding flow.

Towards the end of this concepts time frame we were starting to model what the UI would look like,(see appendix B.1.8). By doing this we also had to concretize some of the functionality and data flow. As can bee seen by the sequence diagrams, (seen in appendix B.3.1).

3.1.3 Alternative point probes (V.1.3)

V.1.3 covers research team's attempts to re-imagine technologies used in pressure and hot-wire probe to upscale existing solutions (appendix A.3.3). However, it has been found not possible due to the physics of those data acquisition methods. This decision was also supported by thorough inspection of both techniques in their original construction.

For this concept we were waiting to see if we would keep V.1 or work on V.2 as these were in parallel. And V.1.3 was quickly discontinued in favor for the volume measurement. Therefore this concept did not provide any changes to the way we would structure our software. This is the visualization we had in mind throughout V.1-class B.11

3.2 Volume measurements (V.2-class)

V.2 solutions class is referring to measurements applied either to a specific volume in the fluid or a specific plane, depending on the construction and hardware chosen for an experiment. Compared to point-probe solution, V.2 techniques are more suitable in current context, because they give a representative picture of the flow in real time in a whole cross-plane simultaneously.

3.2.1 Laser based solution (V.2.1)

V.2.1 generalises measurement systems, that are based on photo-optic solutions. Double pulse lasers in combination with high speed cameras can provide a good quality vector filed representations (see figure A.10). However, those systems have a high purchasing cost and require special laboratory set-up. Those aspects do not combine with an environment of current problem, that has been pointed out by both the customer and an external specialist.

Shortly after the inception of volume measurement as a solution, it was improved from laser based to a solution based on heat exchanger. Therefore this concept did not change how software proceeded.

3.2.2 Heat exchanger based solutions (V.2.2)

V.2.2 covers versions of a new, commercially unavailable concept, that research team has proposed. Design process started with evaluating the system's architecture of a generic, unspecified probe and its development procedures (appendix A.3.2, Figure A.14). There team set a hypothesis (appendix A.3.3), that it would be possible to make a judgement about the exhaust gas' flow intensity by analysing the temperature change of another moving fluid, passing through the duct, encapsulated in thin pipes (see Figure 3.2). It is expected that in this heat-exchanger system the higher speed of a surrounding gas would result in a greater temperature increase in the working fluid. This way one can make at least qualitative judgement on flow's speed around each pipe: the higher the temperature, the greater surrounding speed. It is important to point out, that this works under assumption that temperature of the exhaust gas remains constant across testing volume.

Group struggled to find any academic literature on the new concept. This means that, as long as there is no available preliminary research, the system must go through all development stages: from purely scientific activities to cost-effectiveness analysis. Since the biggest technical risk associated with measurement system is that it would fail to extract expected data (risk T.8 Table 8.7), significant resources were put to plan and conduct a study that would prove the feasibility of the concept (see appendix A.3.5).

The idea of extracting access heat from one body and transferring it to another is not new, but using a heat exchanger as a sensing part of a data acquisition system is arguably an innovative solution. This method significantly simplifies mechanical architecture of a measuring subsystem: construction implies that all sensing equipment shall be moved away from the challenging exhaust gas environment. That directly correlates with purchasing cost of the sensors, since they no longer need to be as robust as more expensive options.

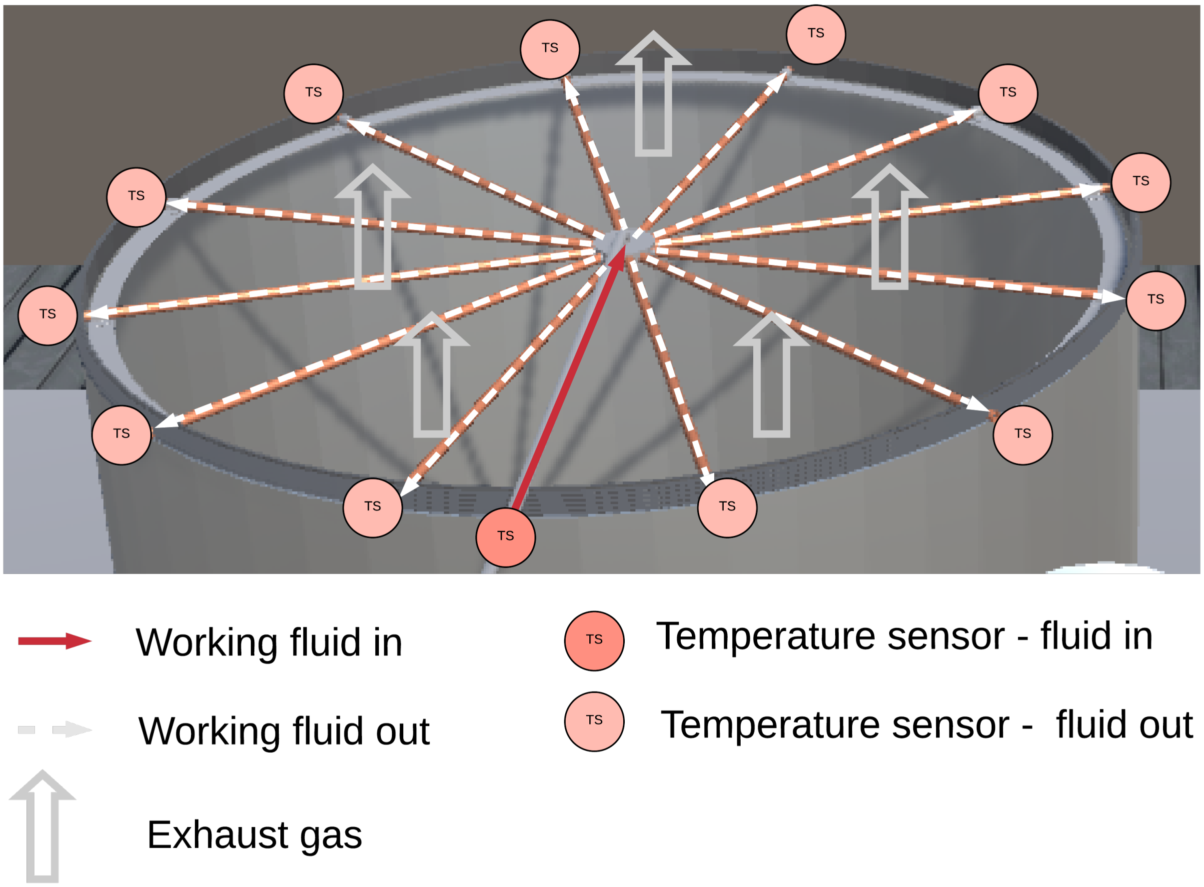


Figure 3.2: Preliminary concept of operation for heat exchanger measurement subsystem

Heat exchanger with water as a working fluid (V.2.2.1)

V.2.2.1 is an original version of a heat-exchanger data acquisition system, that was presented to the customer. Even though the underlying concept was met with curiosity and later-on was approved, water as a working fluid and support systems associated with it (see figure 3.3), were rightfully criticised by stakeholders. Since water in a liquid form would be exposed to temperatures exceeding 500°C it is likely to undergo phase transformation while traveling in the pipes. This would cause difficulties in accurate measurements of the fluids temperature, and would require excessive amount of additional hardware (further - stabilising hardware) to either hold water under high pressure to remain its liquid form, or to recycle hot water damp by means of a condensing unit (for more in depth concept of operation please refer to figure F.24 and figure F.25).

Due to the environmental constraints (see 5.2 Original stakeholder requirements document, specifically D.8) it has been decided to change a working fluid in heat exchanger from water to air. This way one removes stabilising hardware, since air remains in its gaseous state during the whole working cycle.

Towards the end of this time frame, we received some feedback, that we had not done a good enough job explaining the software and showing interfaces. As we had at this point discontinued with V.1-class and therefore changed the way we would visualize the data. This is why we created some more specific models (see figures B.21 and B.22).) as an example of how we would visualize the cross section in 2D.

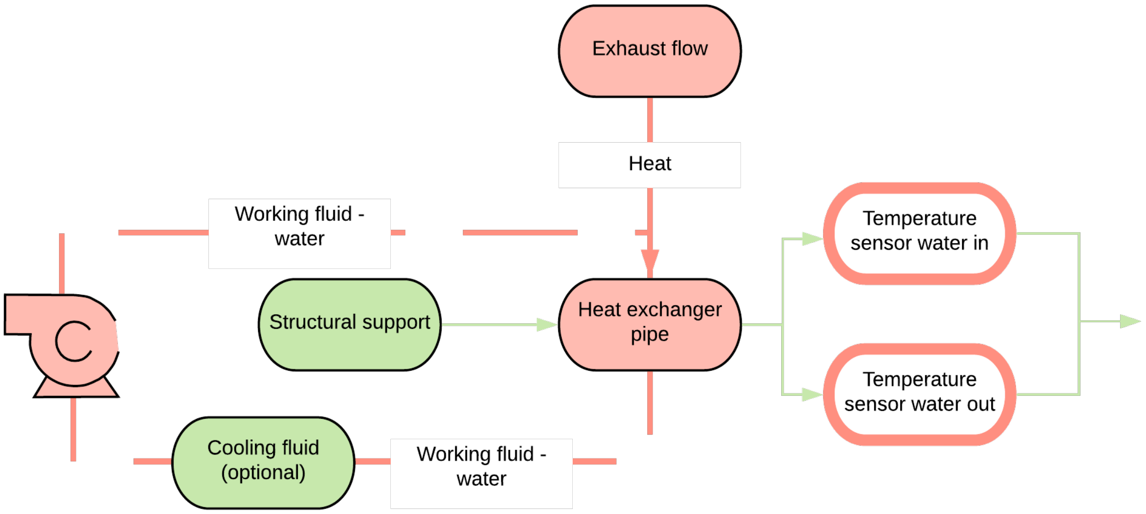


Figure 3.3: Preliminary concept of operation for heat exchanger measurement subsystem with water as a working fluid

Heat exchanger with air as a working fluid (V.2.2.2)

V.2.2.2 occurred as a natural evolutionary progression from previous concept. As it has been discussed above, a working fluid was changed to air, due to its convenience and accessibility. It has been stated by external specialist that it is possible to access an instrument air supply ($\approx 8,5bar$) directly from a gas-turbine package. Research team assumed that it would be possible to connect our system to air source, but it is important to point out that the exact interface between instrument air supply and measurement system is assumed being a "black box" until further information provided (see figure 3.4).

As this was only a change in working fluid, the software as a whole did not change from how it was in V.2.2.1.

Despite the evident difference V.2.2.1 and V.2.2.2 share the same physical phenomenon, that is at the systems core. The interaction between hot exhaust gas and cold instrumental air can be defined as air to air cross-flow heat transfer with forced convection.

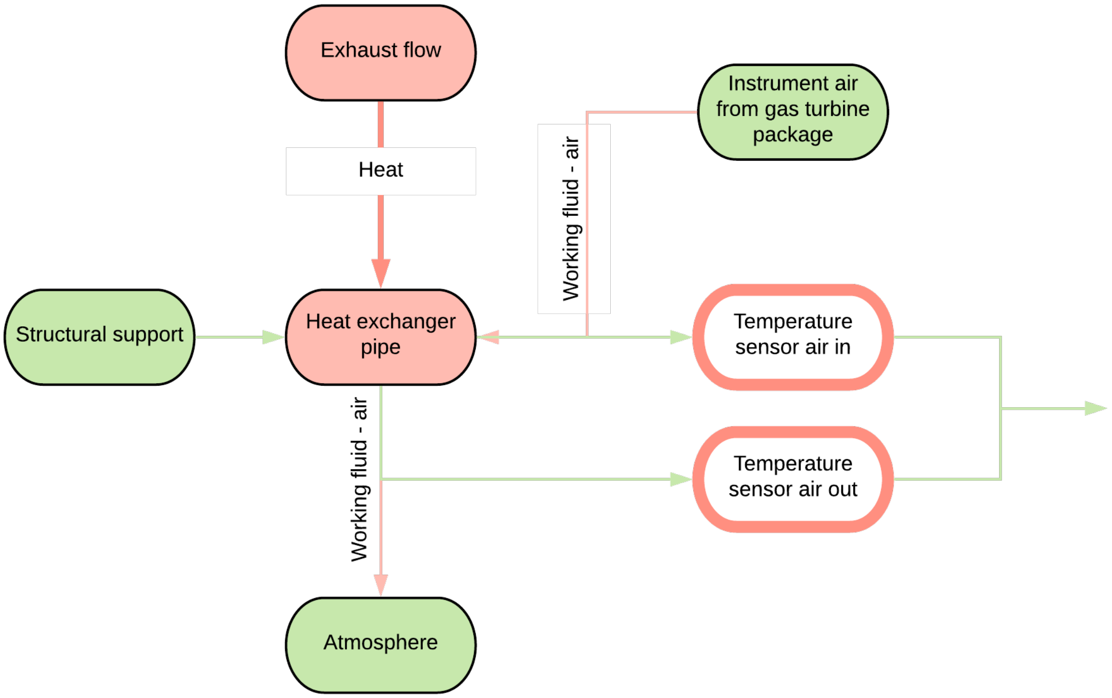


Figure 3.4: Preliminary concept of operation for heat exchanger measurement subsystem with air as a working fluid

Heat exchangers are thermodynamical devises that has been around for a long time and are considered a well researched topic of mechanical engineering, however one would have an issue trying to find peer-reviewed research on correlation between external flow's velocity and temperature gradient of the working fluid. Obtaining this correlation would be of a highest priority for this project, as it arguably has the greatest knowledge value for the customer. Finding the relationship between those parameters is a process that started with an in-depth analytical study on generic heat-exchanger system (see appendix A.3.4).

As most of mathematics associated with this paper is relatively complex, team quickly found out that in current time frame it would be impossible to run a full-scale simulation, that would describe physical processes undergoing in the measurement system. Never the less, group managed to produce a mathematical model, based on energy conservation laws for cross-flow heat exchanger. This model is yet to be resolved, however one can find a proposed numerical solution-method with references to required literature in appendix A.3.5. There has been at attempt in conducting a CFD simulation to prove the concepts feasibility and generate the desired dependency. Unfortunately, dynamic analysis simulating two separate moving fluids with heat transfer, requires a significant amount of computational power. None of the group members had enough resources to run this simulation successfully (appendix A.4.2). Therefore team turned into down-scaling the system and conducting an experimental study (see appendix A.3.5).

As a result of the experiment research team managed to obtain data that showed some confirmation of original hypothesis. It was possible to record temperature increase in the working fluid, when external gas' velocity also increased. Data sets generated were additionally used as an input to software for measurement system.

Research team was not convinced that experiment clearly proved, that it is possible to use a heat exchanger as a measuring system. It was not completely transparent, what specific process is responsible for the temperature fluctuations. Hence, it has been decided to turn to the third method of scientific analysis - mathematical calculations. One distinguished the key descriptive parameters of heat-transfer processes and performed a calculation, that singled out the dominating process in the system. For in-depth evaluation of this study, please, refer to appendix A.3.6.

It is important to point out, that the measuring system currently is in it's scientific development stage. One still has to learn how to handle, scale and control the core physical parameter (heat transfer rate) of the system, which might be a topic for further research.

3.3 Summary for software version development

The main obstacle for software has been not knowing the exact way we will receive data. The concept was to have an interface between the computer and electronics that would feed integers that software would visualize and store. There was always a backup plan to use a data set that the team generated or got provided by Siemens. However, with the circumstances having an electrical system that would generate data was seen as not feasible. Therefore the main focus for the data management was to work with a data set from excel(something that had been researched, see appendix B.2.1), resembling what the group has been shown by Siemens. When the experiment was conducted (information about it can be found in appendixA.3.5) the software team decided to use the logs from that.

3.4 Version development timeline

In the following figure 3.5, we have established the dates for each concept with the corresponding iteration. This is to give an overview of the project as a whole and a way to reference each discipline progress more clearly. Between the 17.01 and 03.02 there was a gap were we researched potential solutions. There was also some internal struggles that slowed progress down noticeably. The second gap between concepts came after feedback after the second presentation. Here we understood that we had to make small tweaks which was done over the weekend, then presented and accepted in a design review with Siemens on the following Wednesday.

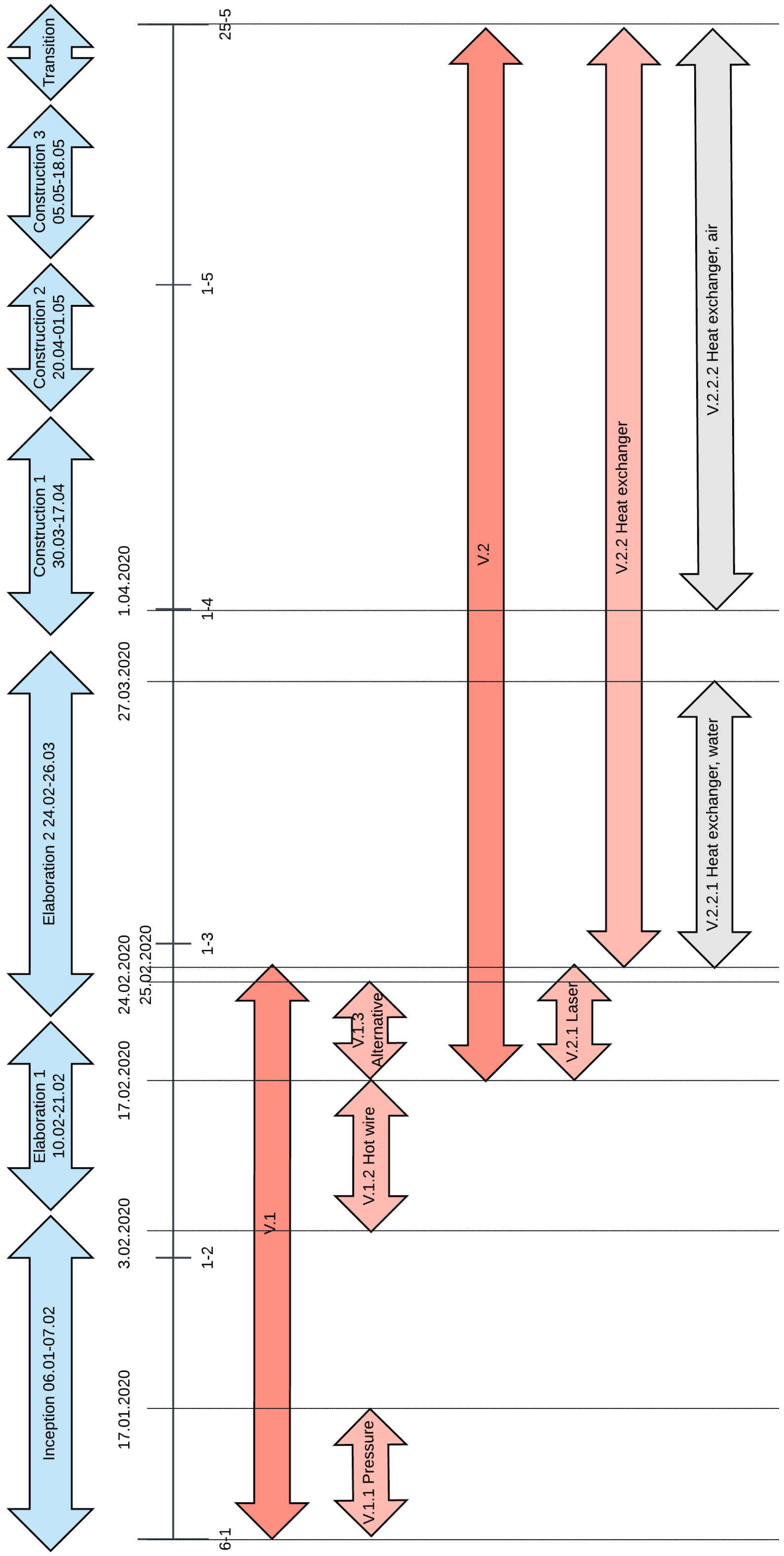


Figure 3.5: Concept development timeline

4. Iterations

In this chapter there will take a more thorough look at the different iterations. In the project model (see chapter 2) there were high level milestones, these will be subdivided and have technical milestones in the iterations.
During the process this chapter will be updated and continuously worked on. Before each iterations there will be an introduction with information on the specific phase. The information is meant to describe the iteration and describe what the team believes should be done. After the iteration there will be an evaluation. In the evaluation the team will summarize the work that was done and possible changes that were made.

4.1 Inception

Information

The inception phase contains one iteration, which will mark the start of the project. Within this time frame the group has to get familiar and make guidelines to follow during this semester. One of the important steps to make during this phase is to decide on the project model and make the framework on how to work. The group will focus on the requirements and the test and risk plan, during this iteration.

Milestones

- Decide what project model to use.
- Define and understand the project scope.
- Translate the predefined requirements into derived requirements.
- Identify the high level risks and create a risk plan.
- Create a test plan.
- Find out if the system is feasible.
- Have the project plan ready for the elaboration phases.
- Prepare for first presentation.

Evaluation

The end of our inception phase was supposed to be on Friday 31. January. During this phase there were a significant amount of decisions to make. What process model and which role each member of the group should have. For the project model the group looked at using rational/agile unified process and scrum. The group started the project using RUP, but after an internal reconstruction of the group, the role of system engineer was changed and the model was changed from RUP to AUP.

In the assignment paper there were a set of defined requirements. These requirements have been derived and made into an original requirement document, for further feedback. This is done to ensure that group has understood what the system shall deliver. There has been created a risk plan so that trace ability is enhanced. The test plan is pending while the requirements are reviewed by Siemens.

The end of the phase and the iteration was supposed to be after the first presentation on Friday 31. January. However after an evaluation meeting the group agreed that the milestones had not been sufficiently fulfilled for this phase. Hence the start of the elaboration phase was prolonged to 10. February. During this phase work was focused on the first concept that was presented to the customer (see section 3.1.1).

4.2 Elaboration 1

Information

For this iteration the goal is to get the requirements finished so that the risk and test plan can also be finalized. The requirements have to be sent to Siemens and then sorted by which discipline it relates to. The grouped requirements need to be derived further by the mechanical, electrical and software students. The requirements will serve as a foundation for the test plan. The test responsible will use them for traceability and make sure that each of the requirements can be tested. The group also need to go through all the requirements and the collective understanding of the assignment to identify the risks of the project.

The software students are going to start looking into different different IDEs and languages that are available and what will be suitable for use. The mechanical research team focused on the theoretical study of turbulence, the knowledge gathered would help to understand why the existing solutions are believed to be inapplicable. For the electrical research that implied, to start looking into what parts the system is going to have, and look into what kind of electrical system is going to be built.

Milestones

- Detailed Risk plan.
- Detailed Test plan.
- Detailed Requirements.

- Technical research.

Evaluation

For the first iteration of the elaboration phase all the milestones have not been met. That is because the group decided to spend time on feedback on the requirements. Therefore the detailed risk plan and the detailed test plan were moved to the next iteration. The requirements from the inception phase have been processed and a few of the measurement requirements have been removed.

As a part of this phase it has been planned to add a technical research workflow. For mechanical domain that implied:

- Detailed theoretical study on turbulence as a physical phenomena (appendix A.1.1-A.1.2).
- Research on technologies and techniques used to model turbulence, understanding the mechanisms, algorithms and difficulties for prediction of fluid behaviour (appendix A.1.3-A.1.4).
- Research and analysis of instrumentation techniques for conducting an experimental fluid-mechanical study (appendix A.2.2-A.2.4).
- A proposal of a technique/techniques that can be applied to case study of this paper.

It is important to point out that responsible for mechanical design has found it challenging to obtain the knowledge listed above. Most of the peer-reviewed literature on the topic is demanding even for an experienced reader and the subject itself is rather complex and yet to be fully researched. Those difficulties have caused certain drawbacks in the workflow. Mechanical-responsible have made a decision to postpone a full concept design to the next phase or until a well-elaborated decision can be made.

For software domain that implied:

- Making a software design document (appendix B.1).
- Finalize UML.
- Finish software architecture.

This is necessary to accomplish, so each member could start working independently in the language chosen. As the software team had decided to research one each, considering that this fits the agile project model.

4.3 Elaboration 2

Information

This is the last iteration of the elaboration phase. One of the main milestones for this iteration will be to prove the architecture of the system. This means making the framework to create the system.

With the choice of moving the risk and test plan to this iteration the team is going to add tests to all the requirements and find the risks for the system. The most critical risks also need to be addressed.

Milestones

- Detailed Risk plan.
- Detailed Test plan.
- Detailed Requirement.
- Prove the architecture of the system.
- Prepare for the construction phase.
- System concept evaluation.

Evaluation

As a part of systems architecture design, and solidification of the final concept proposal, in mechanical domain there have been ran a set of activities:

- Calculation of spatial and time resolution of turbulent flow in exhaust duct (added tables with data in appendix A.2.3).
- Ran test for three commercially available instrumentation techniques for fluids velocity (appendix E.1.1).
- Compared instrumentation techniques on the bases of turbulence resolution and requirements fulfilment (appendix A.2.5). Decision has been made to not implement any of those instruments.
- Gained a deeper understanding on fluids modeling activities, specifically that they work in symbiosis to create as accurate model of physical phenomenon as possible. Hence, showed how to analyse a full model accuracy in a form of flow-chart (appendix A.2.6).
- Looking into alternative solutions (appendix A.2.7).

- Design proposal, listing activities required to design and construct the system. (appendix A.3.2-A.3.3).

For software domain that implied: Each of the software members would focus on a separate programming language and IDE and research this. The main goal with the research was to create a proof of concept for the most vital parts of the software architecture. This refers to real time graph plotting by reading the data from the sensors (most likely a CSV file). Creating the four different user interfaces with all the necessary functions. Lastly generate a visual representation of the data, somewhat like a CFD, but also in 3D. This would help the team choose which IDE and language to create a software application suitable for our project. In addition it will give an indication on how long implementation of different functions will take.

- Research and proof of concept for Python see appendix B.2.1.
- Research and proof of concept for C++ see appendix B.2.2.
- Research and proof of concept for C# see appendix B.2.3.

With the recent changes due to COVID-19 our working environment has changed. The school is temporary closed and the government strongly recommends that people stay home if possible. As this has highly impacted our daily routines, it also impacted on our progression and results. The daily communication have been reduced because people work at home. As the assignment is multidisciplinary, the communication is vital for our progress. Therefore we quickly adapted and migrated most of our daily communication from text based on Telegram, to Discord for additional voice chat and Zoom for meetings. This has been a straining period, given that it happened right before an already hectic couple of weeks. As there was uncertainty regarding our exams and the second presentation being the week after the aforementioned exam. However we are adamant that we managed to take the added challenge head on and still managed to work productively.

4.4 Construction 1

Information

The plan for the first construction phase was to start working on prototyping, testing and risk analysis. However, based on feedback received after the second presentation, the group decided to alter the plan slightly, and postpone the aforementioned plan. The evaluation will explain this in greater detail.

Milestones

- Continuous acceptance testing.

- Test all the implemented parts continuously.
- Reason for picking software and programming language.
- Use Doxygen for software documentation.
- Elaborate on UML documentation.
- Elaborate visualization with models.
- Resolution for experimental image.
- Simulate prove of concept.
- Heat exchanger theoretical research.
- Create UML for system visualization in Unity.

Evaluation One can say with certainty that Construction 1 - phase has been one of, if not the most, critical phases in our project. After second presentation meeting (27.03.2020) and design review with the customer (01.04.2020) we understood weak spots in our workflow, that have been addressed immediately. Among those aspects were: lack of clarity and entirety of documentation across domains, and definition of system's interfaces. Additionally, some of the customer's wishes have been lifted: *"The purchasing cost constraint for the probes and their scientific nature ... group may [also] assume the direction of the flow being approximately uni-directional (z-axis)."* [9] These adjustments partially affect requirement D.6, since CFD-reports show flow's velocity in three dimensions, therefore group shall be focusing on 2-dimensional pattern of the flow distribution. Requirements D.22 and D.23 have been fully postponed since they refer to measuring air's velocity in the room and direction representation. Changes in D.6, D.22 and D.23 influenced requirements applied to mechanical domain: specifically, M.3, M.7, M.8 and M.9. This made current concept of heat exchanger, as a measuring system, more fitting.

Among activities performed in mechanical domain in this phase one can mention:

- CAD design and preliminary CFD analysis of heat exchanger system. CFD part was based mostly on learning how to conduct a simulation study in *SolidWorks Flow Simulation.*
- Gathering information and analysing academical literature on engineering design of heat-exchanger systems.
- Bringing documentation on current concept up to level acceptable by sensors.

Activities performed in the software domain in this phase: Following the feedback from the second presentation, the team realized that the documentation had not been performed to the expected level. Therefore the main focus was to clarify the work that had been done and improve traceability. However this had to run simultaneous to improvements on the UML and the architecture of the software. There were also done preparations for the creation of the system visualization that was to be created in Unity. These preparations were mostly about creating UML documentation of the architecture.

4.5 Construction 2

Information

While continuing to work on the system, and towards accomplishing all the milestones, in this iteration, the team is going to focus on the technical domain. With attending meetings with client, the group want to make sure that customer's wishes are the focal point, and that the group is working towards fulfilling all the requirements.

Milestones

- Read from file for testing.
- Visualisation in the main program.
- Be able to go back and forth between screens.
- Implement 2D visualization.
- Implement storing of data.
- Software library list.
- Finish the signal conditioning before the processor.
- Chose a temperature sensor and model.
- Finalize interfaces with mechanical and software.
- Chose the architecture of the processor.
- Prove of concept experiment.
- Assemble system models in Unity.

Evaluation

In the first iteration of the construction phase there was a hypothesis test. After a meeting with the client it was agreed upon that the mechanical focus was shifted. The new focus during this period was to find a correlation between temperature change and external speed. The electrical part has also changed the focus during this iteration. Electrical lead realized that there was not enough time to finish designing own micro-controller so started looking into putting together a PLC system. With the change of focus, the milestones that have been postponed, included: to choose the architecture of the processor. The software development team has worked on their tasks and milestones. The GUI is finalized(ref) and the unity program is working and being improved aesthetically(ref). Models created in SolidWorks by the mechanical engineer was also implemented in Unity. The data storing is not yet complete as the software team has decided to make some changes in consultation with changes with the electrical changes. Therefore this milestone has not been reached during this iteration. In hindsight the group realized that there were too many milestones in this iteration. Hence storing of data is being added to the next iteration.

Mechanical domain has continued work in the scientific phase of measuring system's development. The following activities have been done:

- Conducting a prove-of-concept experiment, conditioning extracted data by means of *imc FAMOS*, analysing the resulting curves (see appendix A.3.5).
- Documenting theoretical background of engineering design of heat-exchangers (see appendix A.3.4)
- Performing CFD analysis on a simplified heat-exchanger system - unsuccessfully (see appendix A.4.2).
- Creating CAD models for further application in *Unity* (see appendix A.4.1)

Activities performed in software domain:

- Created and continuously updating library lists (see software appendix, section B.4.7 and B.4.7).
- Created the GUI and how the data will be visualized (see software appendix, section ...)
- Data management changed from embedding the Python scripts in C++. It will now use the log data from the mechanical experiment and create a database (see software appendix, section ...)

4.6 Construction 3

Information

In the final construction iteration, the team is going to work to ensure that the goals for the project are reached. At the end of this phase the group is going to finalize the system to present it. After this iteration there is not going to be done more work on the concept, except for minor editing and aesthetic implementation. Each discipline is working to finalize their workloads and reach the milestones. The system architect has made some altercations to the planned time line. Resulting in the last week of the construction phase becoming a part of the transition phase. This week will now start with a concept freeze, followed by a week of documentation.

Milestones

- Ensure that all risks are addressed.
- System stability.
- Improve GUI.
- Implement graphs.
- Implement 3D visualization.
- Finalize PLC system.
- Create PLC schematic.
- Analytical analysis of the heat exchanger system.
- Prepare for transition phase.
- Improve animations and visuals in Unity.

Evaluation

The final construction phase has finished, and the concept freeze is in place, the project is now in the transition phase. One milestone for this iteration is the system stability. That is an overall goal from AUP that the system is acceptable to deploy.

The electrical domain have focused on combining modules to support two PLC platforms from Siemens. Struggling to find compatible modules for the thermocouple that supported the CPU, electrical lead found TIA selection tool from Siemens. This tool helped greatly on the task and made it possible to put together two systems. The systems are presented in a schematic made in OrCAD capture.

For the mechanical domain, work in this phase focused on building and evaluating a sufficient mathematical model, that would help to make a judgement on feasibility of current concept. This implied:

- Creating a system of partial differential equations that precisely describes a heat-exchanger behavior it two dimensions (space-time), see appendix A.3.5.
- Building and running a mathematical analysis, by utilising *Python*, to simulate the relationship between key-parameters of heat-exchanger system, see also appendix A.3.5.

- Evaluating the concept of heat exchanger as a measuring system (see appendix A.3.6).

The software domain has finalized all basic functionality to make the software operate according to the requirements.

- The excel file containing the log data has been processed and appended to an in-memory database(ref) and the C++ program uses it to plot graphs and make the CFD comparison(ref).
- The system visualization program has completed all functionality, some aesthetic improvements remain.

4.7 Transition

Information

In the transition phase the team is working on finishing the documentation and preparing to present the project. In the first week the documentation shall be finalized and delivered on 25. May. After the documentation is delivered, the focus will be on implementing the website for the digital EXPO. After the digital EXPO the team will be preparing for the third and final presentation, that will be held on campus.

Milestones

- Prepare for EXPO.
- EXPO.
- Product presentation in front of unbiased reviewers.
- Make third presentation.
- Finalize documentation.
- Deliver documentation.
- Make project poster.
- Fix defects that are detected.

Evaluation This evaluation will reflect the work done up until the first deadline on the 25. May. The group has also made preparations for the EXPO, which has a deadline on the 27. May. The original plan was having two transition iterations, however the system engineer saw it fitting to merge these into one. As no engineering work will be done, moreover some administrative tasks must be finished, in addition to minor details and improvements. In hindsight the group realizes that some work could have been avoided, had there been more structure to the documentation practices throughout the project.

5. Requirements

5.1 Solution independent requirements

In the assignment document FFA has received, there is a list of predefined instructions and expectations. The information given to project team is highly detailed, however we still attended two additional meetings with the client to gain a better understanding of the problem and create a more specified scope for our project. Information gathered from the assignment text can be separated into two main categories:

- Requested results represented in Table 5.1.
- Detailed requirements listed in Table 5.2.

Each requirement and requested result listed, has a status marker:

- Proposed: Specifically for original requirements document, indicates that it is yet to be reviewed and accepted by client.
- Approved: Indicates that requirement has been reviewed by the client and/or project team, doesn't state that it is under development or have been assessed in any way.
- Added: Used for additional requirements that have been derived, referenced or developed after acceptance of original requirements document.
- In development: Indicates that the requirement is currently used as a baseline for design, research etc.
- Test: Requirement is under testing.
- Implemented: Requirement has passed the test.
- Postponed: Requirement has been scheduled to be assess later or less prioritized at the moment.

ID	**Requested Results**
A.1	Evaluate methods and select preferred method to measure the flow profile in the exhaust, considering all specified requirements.
A.2	Design the required equipment for installation into GT exhaust flow - 3D model.
A.3	Evaluation of the probes/device impact on flow pattern.
A.4	For multi-probe assembly – 3D printed (in plastic) model.

A.5	Select equipment, preferably utilizing “standard” off the shelf proven components.
A.6	Verify the required cycle time for data collection.
A.7	Selection of instruments – supplier data sheet and documentation
A.8	Evaluate and select method of signal transfer for measurement location to display unit.
A.9	Method of signal conditioning for input to monitoring unit.
A.10	Program the monitoring unit for operation (start/ stop of signal monitoring/storage).
A.11	Program the monitoring unit for visual display of result.
A.12	An overall architecture of the complete measurement / display systems.
A.13	Describe installation and method of use.

Table 5.1: Requested Results from assignment document

ID	**Detailed Requirements**	**Status**
B.1	Measurement system shall withstand the exhaust temperature of 550 °C.	In development
B.2	The device is for temporary use only, not permanently installed, with installation / removal in less than 2 hours.	Postponed
B.3	The measurement system shall be arranged to give representative picture of the flow distribution over the entire outlet surface based on the flow pattern indicated on CFD analysis.	Implemented
B.4	The measurement device shall be designed/arranged to be temporarily installed during GT testing in assembly facility (Drammen) or at the GT package final installation.	In development
B.5	The measurement system shall be arranged for fast installation and removal.	In development
B.6	The measurement device shall be installed between the GT exhaust outlet flange and the connected to flexible bellow, but “lifting” the flexible below no more than 50 mm. (A drawing will be provided to show this).	In development

B.7	All parts of the measuring equipment, probes, any connection boxes, etc., shall take into account that the exhaust outlet flange has limited and difficult access located approx. 5 meter above the GT base-frame inside the acoustic enclosure (a drawing will be provided here).	In development
B.8	It shall be noted that no part of the equipment to be lifted may be above 20 kg.	In development
B.9	The location of the measurement devices will be inside the gas turbine enclosure and are not accessible during operation.	Implemented
B.10	The measured signals shall be transmitted to a data monitoring/display unit located on the main deck.	In development
B.11	The monitoring unit shall be in the form of a dedicated device or PC, but must be suited for outdoor location.	Implemented
B.12	The display shall be real-time and include or have output to data storage.	In development
B.13	The output shall be in the form of plots of flow pattern; velocity with direction, static and dynamic pressure, with a format similar to the CFD plots.	Postponed
B.14	Health, safety and the environment: The design must ensure that installation and operation can be carried out in a safe manner.Consideration shall be on the fact that the location is elevated, in a limited space and handling of large and /or heavy objects shall be limited.	Postponed

Table 5.2: Detailed Requirements from assignment document

5.2 Original stakeholder requirements document

It has been stated by professional systems architects that writing stakeholders requirements may be considered as a straightforward task, since it implies stating the system's needed capabilities [10]. However, it is useful to create a classification system or systems for requirements in order to lay down a solid ground for further development. One of the classifications systems applied to requirements set is separating them ”by use” so that they can be traced to appropriate modeling entities: Domains, functions or components. Classification architecture of requirements by use is such:

- Functional requirements (class FR).
  Functional requirements state what the system shall do.
- Temporal performance requirements (class TPR).
  Temporal requirements gives a value for the time there is for our system to respond to a stimulus.
- Nontemporal performance requirements (class NTPR).
  Nontemporal performance requirements gives a value to the properties of our system. These properties can be like weight, cost, size and power consumption.
- Interface requirements (class IR).
  Interface requirements describes the interfaces for our system. These requirements specify the connections and timing between the components.
- Design requirement (class DR).
  Design requirement predetermines a design choice.

ID	**Deduced From**	**Class**	**Derived Requirements**	**Status**
D.1	A.5	DR	The system should use proven equipment.	Implemented
D.2	A.10	FR	The monitoring unit shall be programmed for start/stop monitoring and storage operations.	In development
D.3	B.1	NTPR	The measurement system shall withstand the temperature of 550 deg°C.	In development
D.4	B.2	TPR	The system shall be designed for temporary use.	Postponed
D.5	B.2	TPR	The system shall be installed/removed in less than 2 hours.	Postponed
D.6	B.3	DR	The measurement shall be arranged based on the flow pattern indicated on CFD analyses.	Implemented
D.7	B.4	NTPR	The measurement device shall be portable.	In development
D.8	B.4	IR	The measurement device shall be designed for use at GT package final installation and testing facility.	In development
D.9	B.6	IR	The measurement device shall be designed to fit between the GT exhaust outlet flange and the connected flexible bellow.	In development

D.10	B.6	IR	The flexible below shall not be "lifted" more than 50mm.	In development
D.11	B.8	DR	No equipment that are to be lifted shall weight more than 20 kg.	In development
D.12	B.10, B.11	DR	The measurement shall be visualized in real time.	Implemented
D.13	B.10	IR	the data monitoring/display unit shall be connected to the measuring unit at a safe distance.	Implemented
D.14	B.11	DR	The monitoring shall be suited for outdoor location and off-shore.	Implemented
D.15	B.12	TPR	The display shall be in real-time.	Implemented
D.16	B.12	DR	The display shall include or have output to data storage.	In development
D.17	B.13	FR	The output shall be in the form of representative measurements.	Postponed
D.18	B.13	FR	The format of the plots shall be similar to the CFD plots.	Implemented
D.19	Meeting with client	NTPR	The measuring accuracy shall be representable	In development
D.20	Assignment text	NTPR	The system shall be in the form of a "test kit".	Postponed
D.21	Assignment text	FR	The system shall give a representation of the velocity.	Postponed
D.22	Assignment text	FR	The system should measure velocity.	Postponed
D.23	Assignment text	FR	The system should give a representation of the flow direction.	Postponed

Table 5.3: Original Requirements 1

5.3 Derived requirements

5.3.1 Mechanical requirements

ID	Deduced From	Class	Derived Requirements	Status
M.1	D.3, B.1	NTPR	The measurement system shall withstand the temperature of 550°C as long as the test is running.	In development
M.2	D.5, B.2, D.4	TPR	The system shall be installed/removed in less than 2 hours, without significant changes in GT-package architecture.	Postponed
M.3	D.6, B.3	DR	The measurement shall be arranged based on the flow pattern indicated on CFD analysis.	Implemented
M.4	D.7, D.11, D.20, B.4, B.8, Assignment text	NTPR	The measurement device shall be portable, with no equipment that is to be lifted weighting more than 20 kg	In development
M.5	D.8, B.4	IR	The measurement device shall be designed for use at GT package final installation and testing facility.	In development
M.6	D.9, D.10, B.6	IR	The measurement device shall be designed to fit between the GT exhaust outlet flange and the connected flexible bellow, which is not to be ”lifted” more than 50mm	In development
M.7	Meeting with client	NTPR	The measuring accuracy shall be representable	Postponed
M.8	D.22, Assignment text	FR	The system shall measure velocity of the flow in a pre-defined cross-section	Postponed
M.9	D.23, Assignment text	FR	The system should give a representation of the flow direction	Postponed

Table 5.4: Mechanical Requirements

5.3.2 Software requirement

ID	Deduced From	Class	Derived Software Requirements	Status
S.1	D.2	FR	The application shall write data received to file.	Implemented
S.2	D.2	DR	The stored file shall be in a readable format.	Implemented
S.3	D.2	FR	The application shall be able to read the sensor data.	In development
S.4	D.2	FR	The application shall have a "Start" and "Stop" function.	Implemented
S.5	D.2, D.12, D.15	DR	The application shall visualize the sensor data.	Implemented
S.6	D.2, D.12, D.15	DR	The application shall display sensor values.	Implemented
S.7	D.2, D.12, D.15	DR	The application shall visualize the data in graphs.	Implemented
S.8	D.2, D.12, D.15	DR	The application shall visualize the data in figures.	Implemented
S.9	D.15	FR	The application shall run in real-time.	Implemented
S.10	D.16	FR	The application shall place data at storage device.	In development
S.11	D.17	FR	The application shall store data according to high and low velocity	Postponed
S.12	D.18	DR	The application shall use colors equal to CFD to represent data.	Implemented
S.13	D.18	DR	The application visualization shall resemble the CFD.	Implemented

Table 5.5: Software Requirements

5.3.3 Electrical requirement

ID	Deduced From	Class	Derived Requirements	Status
E.1	A.5	DR	The system should use proven equipment.	Implemented
E.2	B.1	NTPR	The measurment instruments shall withstand the temperature of 550 deg°C.	Implemented

E.3	B.2	TPR	The system shall be designed for temporary use.	In development
E.4	B.3	DR	The measurement shall be arranged based on the flow pattern indicated on CFD analyses.	Implemented
E.5	B.4	NTPR	The measurement device shall be portable.	In development
E.6	B.6	IR	The flexible below shall not be ”lifted” more than 50mm.	In development
E.7	B.10	IR	the data monitoring/display unit shall be connected to the measuring unit at a safe distance	In development
E.8	B.11	DR	The monitoring shall be suited for outdoor location and off-shore	Implemented
E.9	B.11	DR	The electrical system outside the control room shall be explosive certified (ATEX certified)	Implemented
E.10	B.12	TPR	The display shall be in real-time.	In development
E.11	B.12	DR	The display shall include or have output to data storage.	In development
E.12	Meeting with client	NTPR	The measuring accuracy shall be representable	Implemented
E.13	Assignment text	FR	The system shall give a representation of the velocity	Postponed
E.14	Assignment text	FR	The system should measure velocity	Postponed
E.15	Assignment text	FR	The system should give a representation of the flow direction	Postponed

Table 5.6: Electrical Requirements 2

6. Requirements verification and validation

6.1 Test planing

The final stage of completing the requirements development is deriving testing sequences for the system as a whole and for system's components. This implies that the test engineer assesses if each requirement is verifiable and re-wrote it if it was either redundant, inconsistent or contained more than one objective. The main goal of testing is to ensure the project team and the client that the system delivered is of a high quality and it performs as expected. Testing procedure development can be generally described in a form of a workflow diagram an shown in figure 6.1.

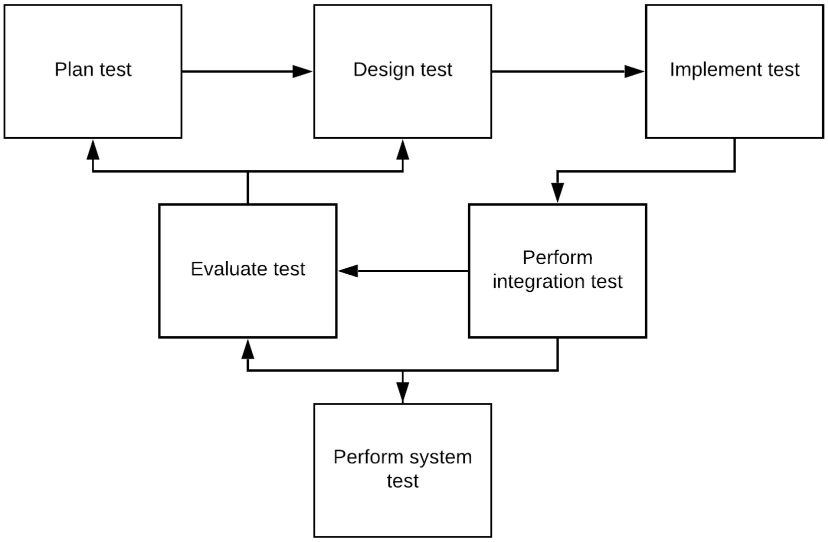


Figure 6.1: Test workflow activities [11]

Planning and designing test systems for the given iterations has three major levels [12]:

- Planning the testing process - majorly based on requirements document.
    - Review systems objectives.

  - Identify testing objectives.
  - Identify pass/fail thresholds.
- Plan the testing approaches - refer mainly to budgeting and allocating resources.
  - Define test activities.
  - Allocate activities to resources.
  - Develop testing schedules.
- Plan testing activities and specific tests - definition in greater detail.
  - Create test scenarios.
  - Identify simulation data for testing.
  - Develop test procedures.
  - Develop analysis procedures.

Test implementation involves test automation and programming required components. It is important to point out that this step is not necessary for all testing methods discussed further in this chapter. Integration testing is of the main concern for the components engineers, because at that stage systems engineers asses if a newly integrated subsystem works as expected. Systems test on the other hand takes into consideration the product as a whole. Evaluation process is focused on reflecting over results obtained under test by comparing them to pre-defined quality goal - thresholds.

Testing methods are separated into four sub-classes as it is described in Table 6.1

Method	**Description**	**Application**	**Effectiveness**
Inspection	Compare systems features to requirements	All segments of verification, validation and acceptance	Passing or failing test can be concluded bu humans
Analysis and Simulation	Use models to represent some systems aspects, for example: environment, process.	Verification testing and acceptance testing	Physical system is not available. Expense doesn't allow instrument test and inspection is not sufficient.

Instrumented	Use calibrated instruments to measure systems physical properties and/or outputs	Verification testing	Physical system is available. Need for detailed real-life information.
Demonstration	Exercise system for unbiased reviewers	Validation and acceptance testing	Expense doesn't allow instrument test on the whole system.

Table 6.1: Testing methods [12]

6.2 Method

We created a template because it's a good method to create consistent tables that includes a specific outline that is tailored for our case. This way it is more understandable for those not directly involved in the process and bystanders. We have done some changes to our old test table and also created a separate one. One table is focused on singular requirement tests and the other is for multiple requirements tests. We did this so we could have a better overview of the results for each individual requirement. As we divided our requirements into mechanical, electrical and software, we also decided to create separate tests for each discipline. For each test that is performed we will create a description of the test results and give each requirement a status on how it went respectively. This will be done through multiple iterations so it is easier to determine when each requirement of the tests are verifiable.

Test ID	ST.#
Traceability	S.#
Test Method	Inspection/Analysis/Instrumented/Demonstration
Priority	Low/Medium/High
Test Iteration	#1
Test Description	Run software
Result Description	Software did...
Status	Pass/Progress/Fail
Test Performed	Time - dd.mm.yyyy
Test Responsible	Name

Table 6.2: Singular requirement

Test ID	ST.#
Traceability	S.#, S.#, S.#
Test Method	Inspection/Analysis/Instrumented/Demonstration
Priority	Low/Medium/High
Test Iteration	#1
Test Description	Run software...
Result Description	Software did...
Requirement Pass	S.#
Requirement Progress	S.#
Requirement Fail	S.#
Test Performed	time - dd.mm.yyyy
Test Responsible	Name

Table 6.3: Multiple requirements

The recent pandemic has caused our tests to hold a lower priority because of restraints from working in physical conditions. This has caused the tests for mechanical and electrical to be down prioritized. As the work towards creating the software has not seen the same consequences, they have produced and performed planed tests. Official tests have been performed at the end of our Construction two and three phases. The test results are documented in the appendix E.

7. Project budget

7.1 Physical resources

Research team have asked the customer for physical resources, including, but not limited to: documents, possible hardware or software that they might have access to. FFA does intend to make a proof of concept, however, it will probably not be in full-scale model, and not of the correct material, as this would be quite expensive. Nevertheless, this might be possible depending on the customer's available resources.

7.2 Human workforce

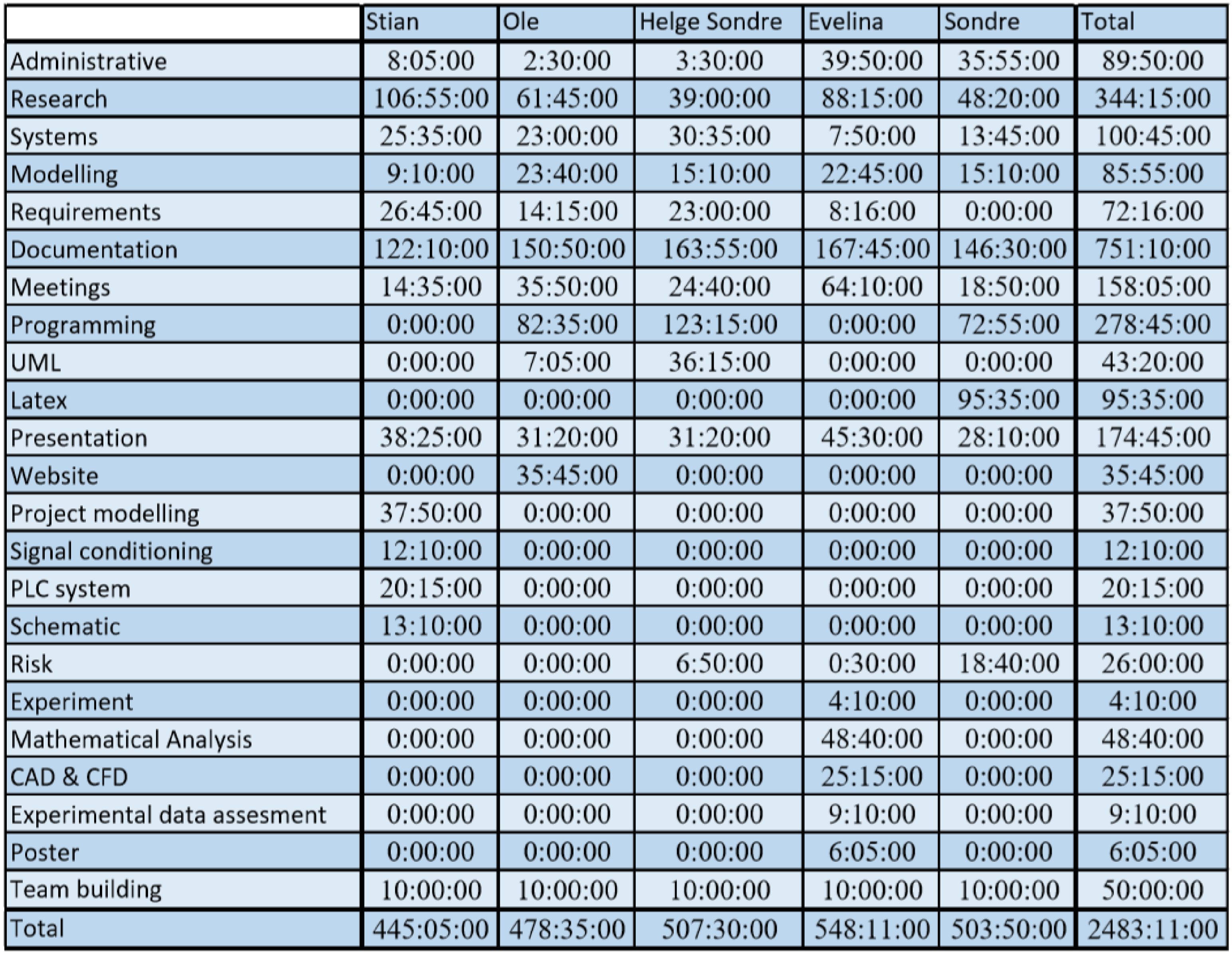
The customer will be supplying us with their time throughout the process, in the form of meetings, presentations. They also have planed a trip to Drammen, to look at their test facility. Here one counts the group as the main workforce. At the starting point of the project, it has been estimated for about 600 hours of work for each person. However, as the project continues, this may become subject to change.

7.3 Economy

As far as the economical aspect is concerned, FFA will not gain any form of external financial support. Research team have done quite a bit of research into existing products, that fit to some aspects for the test kit. This gave one an indication of approximate purchasing cost of the system. However, as it has been stated in the requested result document issued by the client, they expect to provide documentation for the full-scale system. This implies drawing a so-called bill of materials, where purchasing costs will be listed, since most of the concepts are supposed to be off-the-shelf goods.

7.4 Time workflow distribution

Over the course of the project team members have been writing personal time-sheets to keep track of their working hours. Now, at the end of the project time-frame, all time sheets have been gathered to create a combined one, to keep track of the resources utilised. One can see current hour distribution on Figure 7.1.



	Stian	Ole	Helge Sondre	Evelina	Sondre	Total
Administrative	8:05:00	2:30:00	3:30:00	39:50:00	35:55:00	89:50:00
Research	106:55:00	61:45:00	39:00:00	88:15:00	48:20:00	344:15:00
Systems	25:35:00	23:00:00	30:35:00	7:50:00	13:45:00	100:45:00
Modelling	9:10:00	23:40:00	15:10:00	22:45:00	15:10:00	85:55:00
Requirements	26:45:00	14:15:00	23:00:00	8:16:00	0:00:00	72:16:00
Documentation	122:10:00	150:50:00	163:55:00	167:45:00	146:30:00	751:10:00
Meetings	14:35:00	35:50:00	24:40:00	64:10:00	18:50:00	158:05:00
Programming	0:00:00	82:35:00	123:15:00	0:00:00	72:55:00	278:45:00
UML	0:00:00	7:05:00	36:15:00	0:00:00	0:00:00	43:20:00
Latex	0:00:00	0:00:00	0:00:00	0:00:00	95:35:00	95:35:00
Presentation	38:25:00	31:20:00	31:20:00	45:30:00	28:10:00	174:45:00
Website	0:00:00	35:45:00	0:00:00	0:00:00	0:00:00	35:45:00
Project modelling	37:50:00	0:00:00	0:00:00	0:00:00	0:00:00	37:50:00
Signal conditioning	12:10:00	0:00:00	0:00:00	0:00:00	0:00:00	12:10:00
PLC system	20:15:00	0:00:00	0:00:00	0:00:00	0:00:00	20:15:00
Schematic	13:10:00	0:00:00	0:00:00	0:00:00	0:00:00	13:10:00
Risk	0:00:00	0:00:00	6:50:00	0:30:00	18:40:00	26:00:00
Experiment	0:00:00	0:00:00	0:00:00	4:10:00	0:00:00	4:10:00
Mathematical Analysis	0:00:00	0:00:00	0:00:00	48:40:00	0:00:00	48:40:00
CAD & CFD	0:00:00	0:00:00	0:00:00	25:15:00	0:00:00	25:15:00
Experimental data assesment	0:00:00	0:00:00	0:00:00	9:10:00	0:00:00	9:10:00
Poster	0:00:00	0:00:00	0:00:00	6:05:00	0:00:00	6:05:00
Team building	10:00:00	10:00:00	10:00:00	10:00:00	10:00:00	50:00:00
Total	445:05:00	478:35:00	507:30:00	548:11:00	503:50:00	2483:11:00

Figure 7.1: Combined Time sheet

8. Risk management

8.1 The risk management process

The group's risk management is based on the ICH Q9 quality risk management process [13]. The quality risk management process, seen in figure8.1 helps provide a effective and efficient approach to a specific risk.

Starting the process:
First the process has to be initiated, this can be in form of a group meeting.
First step:
After which, entering into the assessment phase. Here a risk is identified, by e.g discussion within the group. When a risk is identified, it needs to be analyzed, so every member involved develops an understanding of it. At this point the risk score will be used. Then the risk will need to be evaluated, based on scoring and other potential factors. A decision on further actions is formed, which will be implemented in the risk control phase.
Second step:
In risk reduction the point is to mitigate the outcome, i.e. lessening the probability and/or severity of it, if possible. These measures are preemptive, for instance team building to improve moral or mitigate internal conflicts. Risk acceptance is not implying there are no consequences, but sometimes they can be ignored or handling can be postponed until suitable measures are found. However, there will likely be instances were there are no appropriate options for treatment of the risk. It is also possible that the risk has to be moved back into the assessment phase, if its not understood or otherwise assessed poorly. When the first two phases are done, the outcome will be reviewed in the last phase.
Third step:
At this point handling of the specific risk will be discussed and if needed, go back to the Risk Control phase. This can be the result of miss handling or general dissatisfaction with the outcome. Risk communication means we will need to properly orientate every involved entity. As for risk management tools, non have been decided, but table 1 in this article [14], provides ideas for tools suitable.

8.2 Risk factors

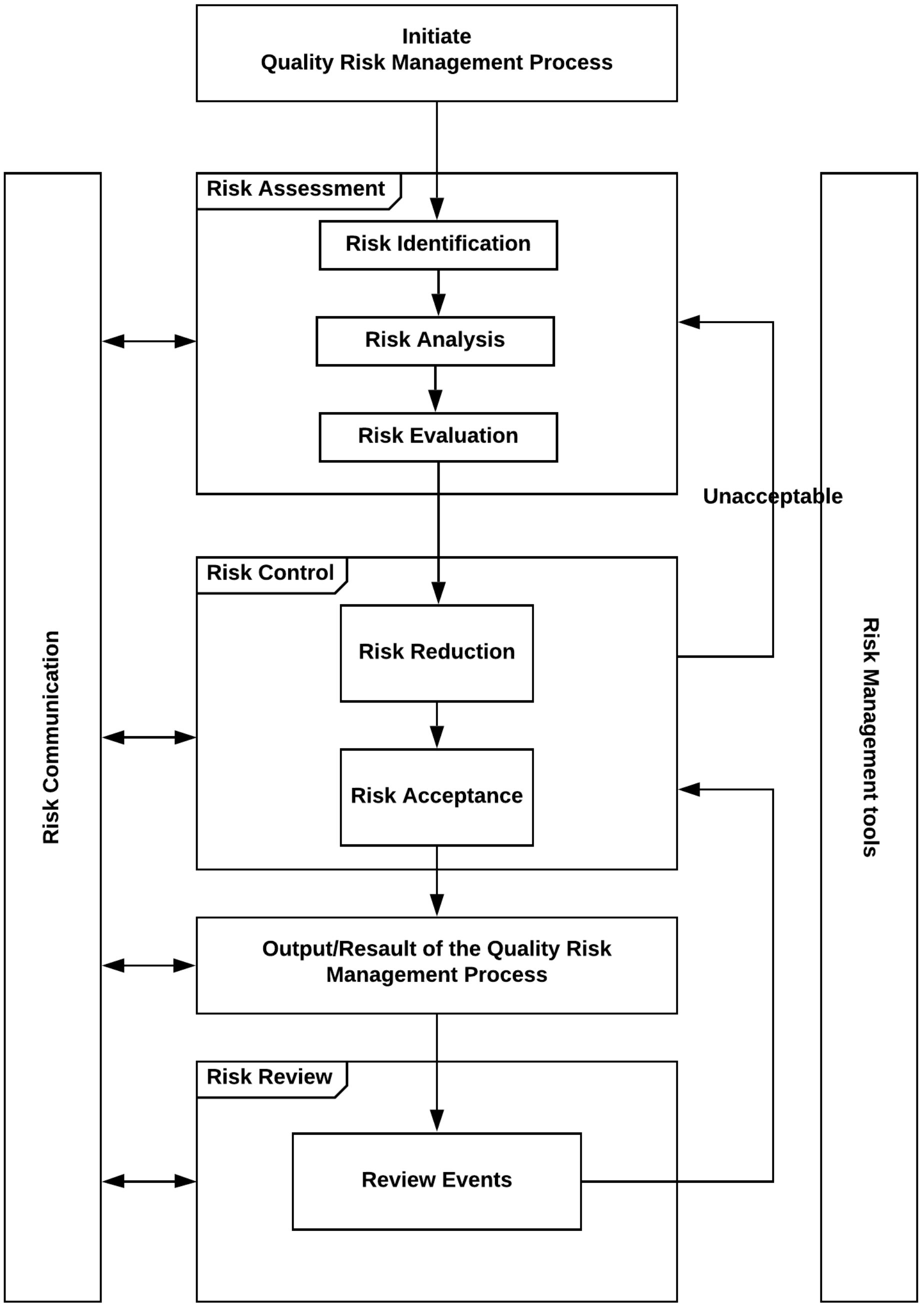
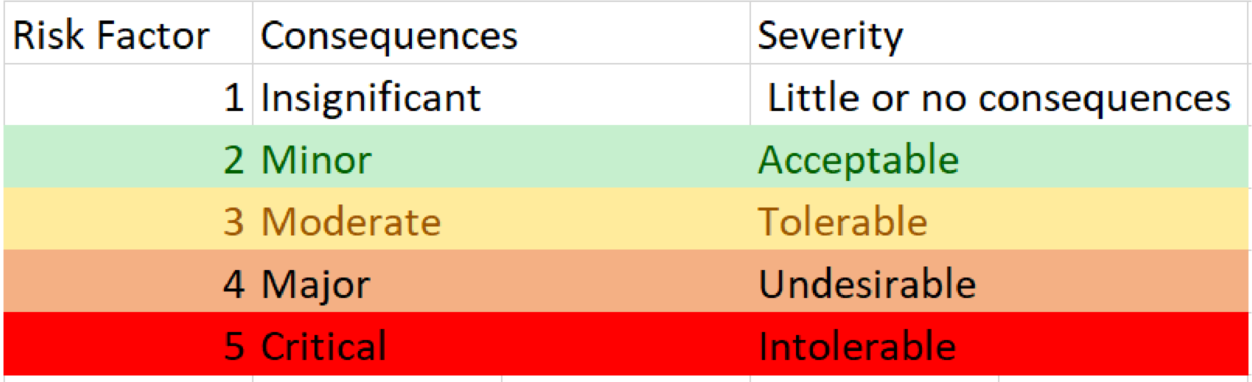
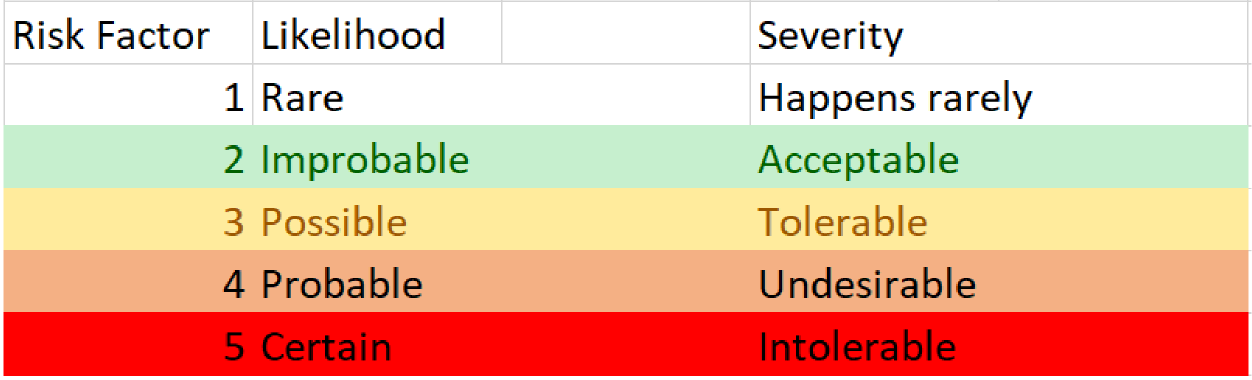


Figure 8.1: ICH Q9 model process for quality risk management[13]



Risk Factor	Consequences	Severity
1	Insignificant	Little or no consequences
2	Minor	Acceptable
3	Moderate	Tolerable
4	Major	Undesirable
5	Critical	Intolerable

Figure 8.2: Risk Factor

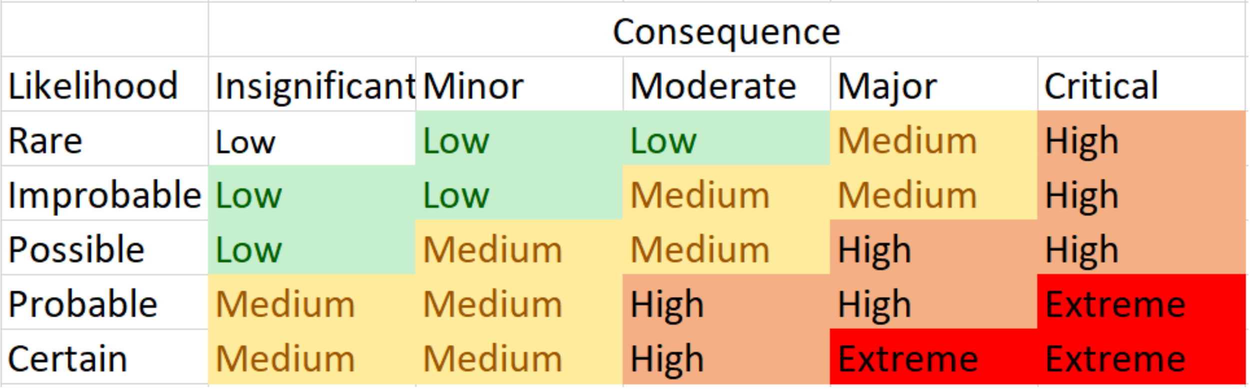


Risk Factor	Likelihood	Severity
1	Rare	Happens rarely
2	Improbable	Acceptable
3	Possible	Tolerable
4	Probable	Undesirable
5	Certain	Intolerable

Figure 8.3: Risk Factor

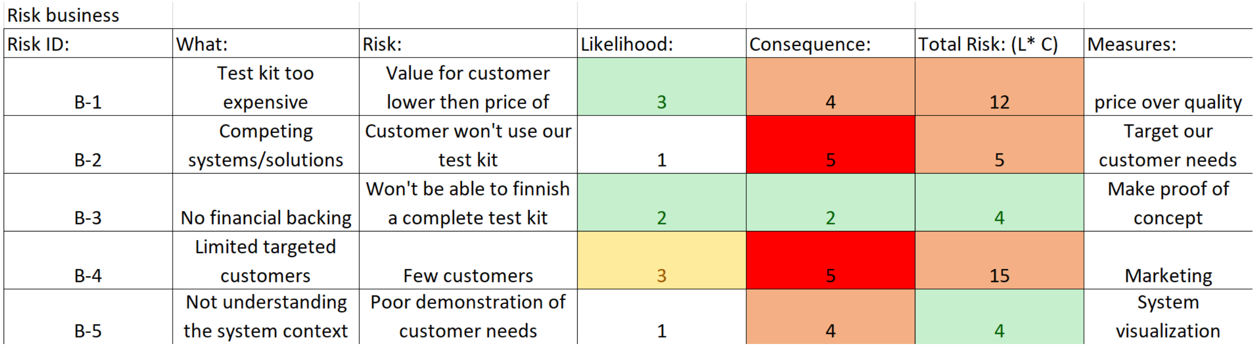
8.2.1 Description of risk factors

1. Rare: Not a problem.
2. Insignificant: not a problem.
3. Acceptable: Little or no effect, OK to proceed.
4. Tolerable: Noticeable effects, not critical.
5. Undesirable: Serious effects, can be critical.
6. Intolerable: Can result in disaster should pause progress, is critical.



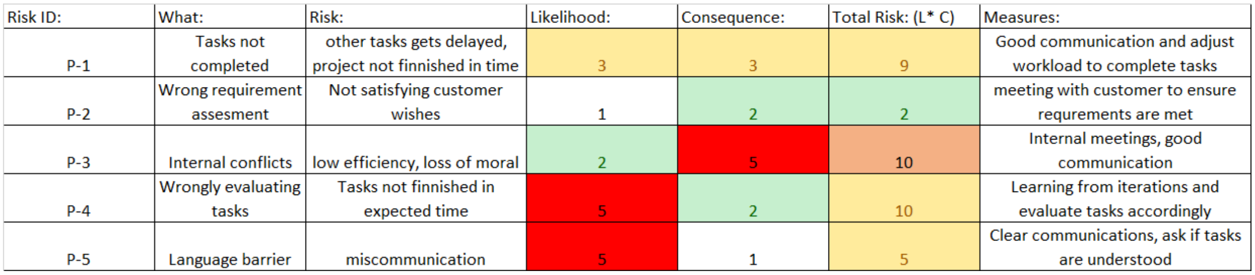
	Consequence				
Likelihood	Insignificant	Minor	Moderate	Major	Critical
Rare	Low	Low	Low	Medium	High
Improbable	Low	Low	Medium	Medium	High
Possible	Low	Medium	Medium	High	High
Probable	Medium	Medium	High	High	Extreme
Certain	Medium	Medium	High	Extreme	Extreme

Figure 8.4: Risk Matrix



Risk business						
Risk ID:	What:	Risk:	Likelihood:	Consequence:	Total Risk: (L* C)	Measures:
B-1	Test kit too expensive	Value for customer lower then price of	3	4	12	price over quality
B-2	Competing systems/solutions	Customer won't use our test kit	1	5	5	Target our customer needs
B-3	No financial backing	Won't be able to finnish a complete test kit	2	2	4	Make proof of concept
B-4	Limited targeted customers	Few customers	3	5	15	Marketing
B-5	Not understanding the system context	Poor demonstration of customer needs	1	4	4	System visualization

Figure 8.5: Business Risk



Risk ID:	What:	Risk:	Likelihood:	Consequence:	Total Risk: (L* C)	Measures:
P-1	Tasks not completed	other tasks gets delayed, project not finnished in time	3	3	9	Good communication and adjust workload to complete tasks
P-2	Wrong requirement assesment	Not satisfying customer wishes	1	2	2	meeting with customer to ensure requrements are met
P-3	Internal conflicts	low efficiency, loss of moral	2	5	10	Internal meetings, good communication
P-4	Wrongly evaluating tasks	Tasks not finnished in expected time	5	2	10	Learning from iterations and evaluate tasks accordingly
P-5	Language barrier	miscommunication	5	1	5	Clear communications, ask if tasks are understood

Figure 8.6: Project Risk

Technical							
Risk ID:	Traceability	What:	Risk:	Likelihood:	Consequence:	Total Risk: (L* C)	Measures:
T-1	E.3, E.5	Portable	Damage during moving	2	5	10	Design equipment for portability
T-2		Wrong installation	Damaging equipment or GT Package	1	5	5	Clear instructions on how to install equipment
T-3	E.4	Resolution to low or sensor misplacement	Customer not getting wanted results	3	3	9	Understanding customer need for validation
T-4	E.8, E.9	Usable offshore	Not suitable for offshore use	2	5	10	Size constraints
T-5	E.12, E.13, E.15	Measuring system not accurate enought	Not able to differentiate between low and high	1	4	4	Testing, proof of concept
T-6	S.1, S.2, S.3, S.4, S.5, S.6, S.7, S.8, S.9, S.10, S.11, S.12, S.13	Software failure	Unable to satisfy customer	5	2	10	Testing software, ensure dataflow throughout the hole
T-7	M.1	Instrument failure	Not getting sufficient data for the test, possibly damaging the exhaust duct and turbine	1	4	4	Test equipmet to withstand 550*C for at least 4 hours
T-8	M.7, M.8, M,9	Instruments do not extract required data	Data quality is poor or wrong type of data	2	4	8	Creating proof of concept for new measuring technique, running tests
T-9	M.4, M.5	Not safe to use	Doesn't follow regulations for offshore and testing facility	1	5	5	Take extra care when designing the real-life system
T-10	M.7	Instrument is too slow	Physical process goes faster then sampling rate of the measuring system	2	3	6	Creating proof of concept for new measuring technique, running tests
T-11	M.2	Long installation time	Takes more then 2 hours to mount and dismount test	1	4	4	Do not choose a higly scientific equipment
T-12	M.7	The equipment is too intrusive	Instruments disturb the flow enough to give faulty readings	2	3	6	Running CFD analysis for the measurmnet system, studying proof of concept

Figure 8.7: Technical Risk

8.3 Risk analysis

When combining the severity and the probability we get risk. This means that probability/-likelihood times by severity/consequence will give the total risk 8.4.

$$L * C = TR$$

Business risk
Business risk shown in Figure 8.5 assess the risks connected to FFA as a consulting company. Where we are able to deliver a product and if our customer will be pleased with our product or not, and the impact if we can manage to deliver wanted product or not.

Project risk
Project risk shown in Figure 8.6 shows the different risks to our project. Where the focus is what can go wrong inside the group or project, and measures to asses these problems. Group or project problems can cause latency to our process. So we need to have measures to prevent such actions. Project risk can give us an early indication and what we should focus on to avoid such problems.

Technical risk
Technical risk shown in Figure 8.7 asses risks to our solution. Whether our test kit will meet the requirements set for our product. If the requirements are not met. What consequences will this have for our final product, the importance of requirements that are not met. And how can we asses these problems and fix them.

The grading from our matrix will give us an indication on how important each risk will have. Whether this is something we need to focus on early in development or we can find new solutions later in development.

8.4 Risk consideration

Under developing of our system, we focused on mitigating the highest risks in technical risks. To ensure our development process goes as smooth as possible. To mitigate some of the risks we have been focusing on making a concept that has the least critical parts as possible. With this we have tried to take all measurements outside the GT package. To lower the chances of equipment damage during run time and for the system to be more portable. With taking the measurements outside the GT package we can lower the durability/quality the equipment as well to reduce the price of our system.

Initially we design a heat exchanger system with liquid water. This was the first step we had to get the measurements outside the GT package to mitigate the mechanical technical risks.

With this design we encountered new possible risks, due to most certain phase transformation of water during the test run. This caused our contractor's dissatisfaction. This lead to another iteration on the design, to simplify the design of our system. With removing the liquid water and replacing it with air we could mitigate a lot of the potential errors and make the system a lot smaller. All these design decisions where impact by our risk management. To lower potential failures, mitigate or avoid the risks completely.

Mechanical Technical Risk

There have been mentioned a set of risks that are associated with the mechanical domain in this project. Even though one can clearly distinguish between two separated version-classes in product development (multi-probe and volume measurements), technical risks corresponding to those concepts are arguably similar. However, risk managing procedures are different from version to version.

Since multi-probe solutions are commercially accessible, it was possible to conduct inspection tests on requirements linked to risks from *T-7* up to *T-12* (see table A.2.5). Both versions (V.1.1 and V.1.2) failed to extract required data in a representable way to construct a velocity field. This refers to the most critical risk for mechanical domain - *T-8*, where adequate measure for treating this risk was defined as: *"Creating a prove of concept for a new measuring technique, running tests"*.

Despite a direct guideline to turn into new solutions, research team made the last attempt for maintaining commercial accessibility for measuring system. First version of volume measurements version class was V.2.1 - Laser based techniques. Even though the idea seemed promising, it failed to satisfy a list of requirements associated with risk *T-11*, meaning it was impossible to use the system in the environment specified by the customer.

Thereon, after leaving existing solutions completely, the main focus for mechanical domain has been working on proving the systems feasibility and its ability to extract required data, as it was specified in the measures linked to risk *T-8.* This was achieved through an experiment and mathematical analysis (see appendices A.3.4-A.3.6).

Electrical Technical Risk

The electrical domain have worked on creating a system that are acceptable to use on site offshore. With the risk T-4 there is a risk of being unable offshore. This risk involves that the equipment need to be ATEX certified. Therefor the sensor are ATEX certified. However the control system are not ATEX certified. That can be solved with moving it outside of where an explosive atmosphere can happen. For the equipment to be portable there is need to be lightweight, Small and power easily available. The PLC system have a need to be connected to a power source to be able to function. However it is a Small data collection system that don't take up much space.

One problem that can occur is that the measurement don't have a high resolution. The chosen sensors are not the most accurate with a resolution that can be off with 1° to 2° degrees. However with a range up to 750°C there is not that big of a deviaton. Another deviaton is if one sensor measure two degrees too high and one sensor too low. The best scenario is that all the sensors have matching deviation.

Another risk with measurement is to get all the data in Real time and get the same samples. With the PLC system the analog module is designed to handle thermocouples, but with a microcontroller it may not. If the thermocouples are connected to an arduino the analog inputs are using a multiplex connected to one ADC (analog digital converter). To switch between inputs there is a need for a short delay for the multiplex to change channel.

Data Technical Risk

The risks corresponding to software have been to create a software pleasing the costumers need, and how to visualize the results in a method that can be understood. In version 1 3.1 with multi-probe assembly the main focus was the data flow in the software domain. With the focus to how we can receive information from sensors and use this information to display the information in a usable matter. To find the optimal methods to display data, the customer where shown suggestions that could be feasible and later accepted by the customer.

While the project developed and new versions where created. The visualization changed to please the customer needs. In versions with multi-probe assembly the main idea was to show graphs and a color grid with color estimated values between sensors. To show the current flow inside the GT exhaust. In versions 2 3.1 the visualization needed a slight change because of the new measurement method. This changed the visualization to make color estimations between each pipe instead of estimations between each sensor.

When the graphical interface where accepted the software development was continuously tested and compiled to ensure the software where operating as expected. Even thou there have been unexpected results from the software this have been handled continuously while creating the software.

9. Conclusion

This section covers academical findings and engineering solutions, that have been achieved within the projects time frame. Since no commercially available devices would fit the scope of the problem, it was necessary to design a completely new data acquisition system. It was also discovered, that conducting fluid mechanical experiments on the gas turbine exhaust flow, is a rather challenging task. Partially, it is caused by the nature of the fluid itself, since it is turbulent and unstable. Another issue is that measuring subsystem shall be exposed to the hostile environment of the exhaust duct for a long period of time, which put a set of tight constraints on the instruments and general mechanical architecture of the system.

Considering the hostile environment, the electrical subsystem had to be ATEX certified. This is because of an explosive atmosphere that can occur, which adds a design constraint on the electrical subsystem.

It has been challenging to find a way to compare the CFDs the group has received from the customer, to images generated by the group's software. The software team has therefore also needed to make adaptions to how visualization should be done and the way data is managed.

The product described in this paper has been evolutionary developed directly from the customer's needs with almost no previous research available. FFA had some knowledge, gathered from meetings with the customer, that previous experiments was focusing on analysing a down-scaled system, which has been proven inapplicable for current scope. The group agrees that one of the main difficulties in the projects life cycle, has been time resources, that has been reinforced by the emergence of COVID-19.

9.1 Results

It has been found that it is possible to utilize a cross-flow one-pass heat exchanger, to produce a qualitative evaluation of the flow pattern (see figure 9.1). In this concept, inlet and outlet temperature of the working fluid are used to make a judgement on the intensity of the surrounding external flow. This is believed to be possible, based on thorough theoretical research and experimental study. Even though the concept of utilising heat transfer rate as an indication of the mass-flow intensity has been proven to some extent, it is yet to be ready for implementation in a real life system. This refers to mechanical workflow being in a scientific phase of the development.

To emulate the CFD image to a sufficient level, the software team has used twelve pipes with one sensor each. The latter is located on the outside of the exhaust duct. The pipes are placed in the cross section, to map and visualize the flow (see figure B.22).

To collect the data from sensors the electrical domain has put together three electrical systems. There are two PLC systems, one basic controller and one advanced. The last electrical system is based on the Arduino mega microcontroller to connect to the thermocouples.

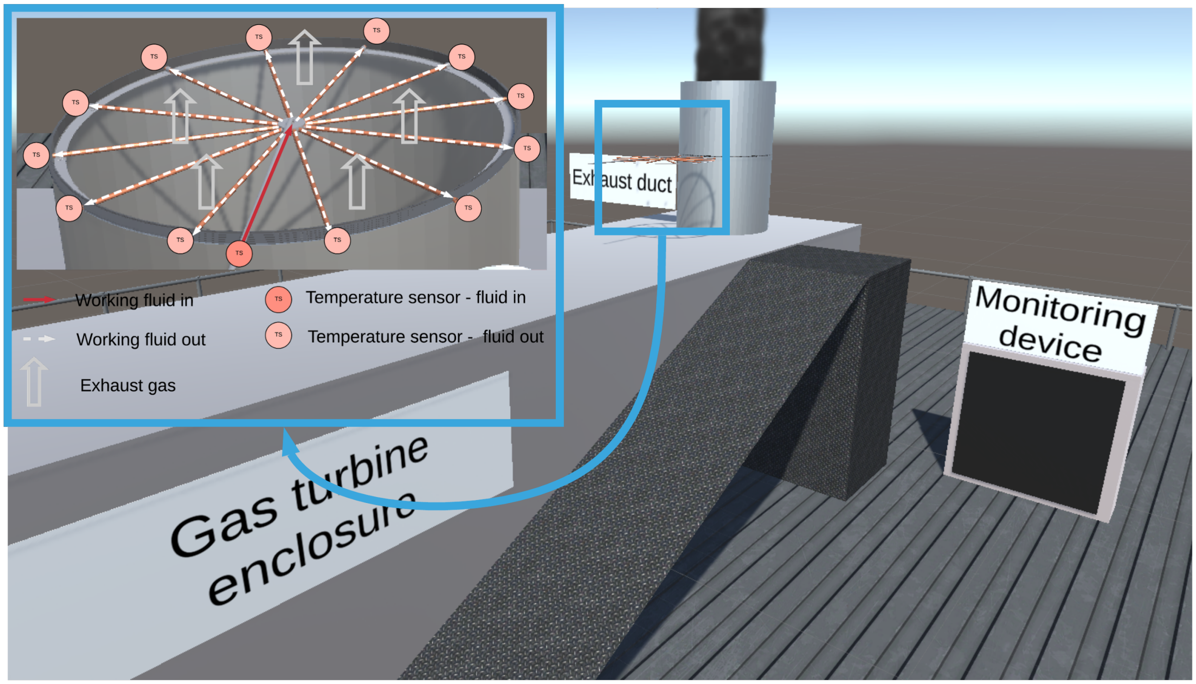


Figure 9.1: Final data acquisition system architecture

9.2 Market value

Fluid Flow Analytic's data acquisition system is arguably a cost-effective solution. It can compete with exiting commercially available products both in economical domain and the scope. It also manages to provide required experimental data, without being unnecessarily scientific. However, this means that the resolution achieved, is not greater than what is required for the customer. Nevertheless, low-resolution and qualitative description of the flow's behavior, decreases the purchasing cost of the system. In the real world this product will help the customer to make better supported engineering decisions.

9.3 Further work

Since the measuring sub-system development is currently in a scientific phase, it is advised to perform additional experiments and numerical simulations, described in appendix A.3. Numerical analysis is believed to provide an effective optimisation and tuning procedure, that can be used for choosing the working fluid and it's parameters. Those activities will finalize the scientific phase of measuring system development. This means that a possible group which supersede FFA, can then continue on engineering phase and work on structural implementation and manufacturing.

The group had an idea to use machine learning or image recognition to compare customer's CFDs to images created by the group's system. Given the time constraint on the project this was not feasible for FFA, however, this is recommended for the future of this product. The data gathered from a gas turbine test, should also be used to automatically generate a report. It is also advised to look into integration of OPC between the PLC and the data management software. 3D visualization can be implemented, based on the data gathered, making data representation more intuitive. For off shore use, the software can be made compatible with different devices e.g smart phones and tablets. General suggestion is to optimize code with threading and utilize more efficient algorithms.

For the electrical domain the further work is to have another revision on the Arduino thermocouple module and change the layout on the board to include the Arduino. For a better result there should be added an ADC to each of the input signals from the thermocouples' amplifiers. It is also advised to base the design on a microcontroller, which does not use a multiplex. The PLC system needs to be programmed for operation, using ladder logic that is normally used with PLC.

Reference list

[1] Siemens. About Siemens Global Website 2020. `https://new.siemens.com/global/en/company/about.html`. Accessed: 14.01.2020.

[2] P. Straznicky H. Saravanamuttoo H. Cohen, G. Rogers and A. Nix. *Gas Turbine Theory*. Pearson, 7th edition, 2017.

[3] Siemens. Bachelor's thesis assignment: Exhaust flow mapping, 2020. Siemens.

[4] John W. Slater. Overview of cfd verification and validation. `https://www.grc.nasa.gov/WWW/wind/valid/tutorial/overview.html`. Accessed: 21-01-2020.

[5] Scott W. Ambler. The agile unified process. `http://www.ambysoft.com/unifiedprocess/agileUP.html`. Downloaded a zip file, Current version: v1.1 May 13, 2006, Accessed: 28.01.2020.

[6] John F. Foss Cameron Tropea, Alexander L. Yarin. *Handbook of Experimental Fluid Mechanics*. Springer, 2007.

[7] Airflow Sciences Equipment. Airflow sciences equipment: Accurate flow and velocity data. `https://www.airflowsciencesequipment.com/`, 17.01.2020.

[8] Dantec Dynamics. Dantec dynamics-precision measurement systems and sensors. `https://www.dantecdynamics.com/`, Accessed 21.03.2020.

[9] Bachelor presentation follow up. Summarising document, that reflects team's, at that time (30.03.2020), current efforts, results and understanding of the project. This document also contains supervisors' inputs after second presentation.

[10] Alberto Sols. *Requirements Engineering and Management: A Systems Approach*, page 57. CreateSpace Independent Publishing Platform, 2016.

[11] Kendall Scott. *The Unified Process Explained*, pages 79, 82–85. AddisonWesley Professional, 2001.

[12] Dennis M. Buede. *The Engineering Design of Systems: Models and Methods*, pages 333, 357, 361, 364. Wiley, 2009.

[13] European Medicines Agency. ICH guideline Q9 on quality risk management. `https://www.ema.europa.eu/en/documents/scientific_guideline/international_conference_harmonisation_technical_requirements_registration_pharmaceuticals_human_use_en-3.pdf`. Accessed: 28-01-2020.

[14] E. Michael T. Raz. Use and benefits of tools for project risk management. `https://www.sciencedirect.com/science/article/pii/S0263786399000368`, Available online 27 October 2000.

[15] B. Wong. The mathematical theory of turbulence or chaos. *International Journal of Nonlinear Science*, 10:264–278, 2010.

[16] Wade W. Huebsch Alric P. Rothmayer Bruce R. Munson, Theodore H. Okiishi. *Fluid Mechanics*. John Wiley & Sons, 2013.

[17] Paul G. Tucker. *Advanced Computational Fluid and Aerodynamics*. Cambridge University Press, 2016.

[18] Engineeringtoolbox.com. Air - density, specific weight and thermal expansion coefficient at varying temperature and constant pressures. `https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html`, Accessed 21.03.2020.

[19] Engineeringtoolbox.com. Air - dynamic and kinematic viscosity. `https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html`, Accessed 21.03.2020.

[20] Arthur H Lefebvre. *Gas turbine combustion*, page 81. Taylor and Francis, 1999.

[21] Finn E. Jørgensen. *How to measure turbulence with hot-wire anemometers - a practical guide*, pages 6,8. Dantec Dynamics, 2002.

[22] Ethirajan Rathakrishnan. *Instrumentation, measurements, and experiments in fluids*, pages 203–224, 250, 295, 299–303,323. Taylor and Francis, 2007.

[23] Airflow Sciences Equipment. Map system cad animation. `https://www.youtube.com/watch?v=KroxtrnfKm0`, 22.07.2013.

[24] Department of Aerospace Engineering Dr. Hui Hu, Iowa State University. Lecture 13: Particle image velocimetry technique. AerE 311L Lecture Notes. Accessed: 28.01.2020.

[25] Kristian Debus. New accuracy study opens the door to cfd led clinical trials. `https://blogs.sw.siemens.com/simcenter/new-accuracy-study-opens-the-door-to-cfd-led-clinical-trials/`, 31.10.2017.

[26] Yunus A Cengel. *Thermodynamics. An engineering approach*. McGraw-Hill Education, 9th edition, 2019.

[27] J. P Holman. *Heat transfer*. McGraw-Hill Higher Education, 10th edition, 2010.

[28] Sen M. Goodwine B. Alotaibi, S. and K. Yang. *NUMERICAL SIMULATION OF THE THERMAL CONTROL OF HEAT EXCHANGERS. Numerical Heat Transfer, Part A: Applications*, pages 229–244. 2002.

[29] Clas Ohlson. Cocraft hh 2000 varmluftpistol. `https://www.clasohlson.com/no/Cocraft-HH-2000-varmluftpistol-/p/40-8761?gclid=Cj0KCQjw-_j1BRDkARIsAJcfmTGhrI-Q-ia6Hr80R6FipWmaM9By1x6hNA9reUASvWSRDc5vUkywKIcaAkDqEALw_wcB`, 2020.

[30] Biltema. Vindmaaler. `https://www.biltema.no/batutstyr/elektrisk-utstyr/maleinstrumenter/vindmaler-2000042811?gclid=Cj0KCQjw-_j1BRDkARIsAJcfmTHojX2Y4D5CjXyMr4TnvUKmey-V8GaYOfCitcvUSqSPvOBpcobgAMgaAl_7EALw_wcB`, 2020.

[31] KROHNE GmbH. Krohne norway. `https://no.krohne.com/nb/`, 2020.

[32] National Instruments. Multifunction i/o device - national instruments. `https://www.ni.com/en-no/shop/select/multifunction-io-device?modelId=124910`, 2020.

[33] RS Components. Rs pro type k thermocouple 0.3mm diameter, $-50°c - +400°c$. `https://no.rs-online.com/web/p/thermocouples/6212170/?sra=pstk`, 2020.

[34] RS Components. Mean well, mdr din rail panel mount power supply, 24v dc output voltage, 1a output current. `https://no.rs-online.com/web/p/products/0428433/`, 2020.

[35] imc Test & Measurement GmbH. imc famos - data analysis framework. `https://www.imc-tm.com/products/measurement-software/imc-famos/B`, 2020.

[36] Sandip Mazumber. *Numerical methods for partial deirrential equations*. Academic Press, 2016.

[37] Engineeringtoolbox.com. Air - prandtl number. `https://www.engineeringtoolbox.com/air-prandtl-number-viscosity-heat-capacity-thermal-conductivity-d_2009.html`, Accessed 02.05.2020.

[38] Tiobe index. `https://www.tiobe.com/tiobe-index/`. Accessed: 23.03.2020.

[39] Open source tool for python. `https://pandas.pydata.org/`. Accessed: 21.02.2020.

[40] Lorena A. Barba. Cfd python: 12 steps to navier-stokes. `https://lorenabarba.com/blog/cfd-python-12-steps-to-navier-stokes/`, 22.07.2013. This post describes the first practical module of Prof. Barba's Computational Fluid Dynamics class, as taught between 2010 and 2013 at Boston University.

[41] Gui package for python. `https://wiki.python.org/moin/TkInter`. Documentation for Tkinter, accessed: 23.03.2020.

[42] WeaverDev. Unity graphs. `https://assetstore.unity.com/packages/tools/gui/debuggui-graph-139275`. Downloaded: 18.03.2020.

[43] Docs.python.org. `https://docs.python.org/3.7/library/sqlite3.html`. Sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.7.7 documentation.

[44] Doc.qt.io. `https://doc.qt.io/qt-5/qtimer.html`. QTimer Class — Qt Core 5.14.2.

[45] Doc.qt.io. `https://doc.qt.io/qt-5/qapplication.html`. QApplication Class — Qt Widgets 5.14.2.

[46] Doc.qt.io. `https://doc.qt.io/qt-5/qtwidgets-index.html`. Qt Widgets 5.14.2.

[47] Doc.qt.io. `https://doc.qt.io/qt-5/qtcharts-index.html`. Qt Charts 5.14.2.

[48] Doc.qt.io. `https://doc.qt.io/qt-5/qchartview.html`. QChartView Class — Qt Charts 5.14.2.

[49] Doc.qt.io. `https://doc.qt.io/qt-5/qpieseries.html`. QPieSeries Class — Qt Charts 5.14.2.

[50] Doc.qt.io. `https://doc.qt.io/qt-5/qpieslice.html`. QPieSlice Class — Qt Charts 5.14.2.

[51] Emanuel Eichhammer. Qt plotting widget qcustomplot - introduction. `https://www.qcustomplot.com/index.php/introduction`.

[52] Pandas.pydata.org. `https://pandas.pydata.org/docs/`. Pandas 1.0.3 documentation.

[53] Docs.python.org. `https://docs.python.org/3.7/library/tk.html`. Graphical User Interfaces with Tk — Python 3.7.7 documentation.

[54] Pillow.readthedocs.io. `https://pillow.readthedocs.io/en/stable/releasenotes/7.0.0.html`. 7.0.0 — Pillow (PIL Fork) 7.1.2 documentation.

[55] Docs.python.org. `https://docs.python.org/3.7/library/random.html`. Random — Generate pseudo-random numbers — Python 3.7.7 documentation.

[56] Docs.python.org. `https://docs.python.org/3.7/library/itertools.html`. Itertools — Functions creating iterators for efficient looping — Python 3.7.7 documentation.

[57] Docs.python.org. `https://docs.python.org/3.7/library/csv.html`. CSV File Reading and Writing — Python 3.7.7 documentation.

[58] Docs.python.org. `https://docs.python.org/3.7/library/time.html`. Time access and conversions — Python 3.7.7 documentation.

[59] Numpy.org. `https://numpy.org/doc/1.18/`. Overview — NumPy v1.18 Manual.

[60] Matplotlib.org. `https://matplotlib.org/3.1.3/contents.html`. Overview — Matplotlib 3.1.3 documentation.

[61] Nobiax / Yughues. Yughues free metal materials. `https://assetstore.unity.com/packages/2d/textures-materials/metals/yughues-free-metal-materials-12949`. Downloaded: 25.04.2020.

[62] William Bolton. *Mechatronics Electronic control systems in mechanical and electrical engineering*. Pearson, sixth edition edition, 2015.

[63] Data sheet ad8494/ad8495/ad8496/ad8497. `https://www.analog.com/media/en/technical-documentation/data-sheets/AD8494_8495_8496_8497.pdf`. Accessed: 20.05.2020.

[64] Two ways to measure temperature using thermocouples feature simplicity, accuracy, and flexibility. `https://www.analog.com/en/analog-dialogue/articles/measuring-temp-using-thermocouples.html`. Accessed: 23.05.2020.

[65] Rnd 205-00034 - rekkeklemme for kretskort. `https://www.elfadistrelec.no/no/rekkeklemme-for-kretskort-32-3mm-22-12awg-5mm-pitch-poler-rnd-connect-rnd-205-00034 p/30042860`. Accessed: 23.05.2020.

[66] Arduino mega 2560. `https://store.arduino.cc/arduino-mega-2560-rev3`. Accessed: 23.05.2020.

[67] Atex and explosive atmospheres. `https://www.hse.gov.uk/fireandexplosion/atex.htm#background`. Accessed: 20.05.2020.

[68] Explosive atmospheres - classification of hazardous areas (zoning) and selection of equipment. `https://www.hse.gov.uk/fireandexplosion/zoning.pdf`. Accessed: 20.05.2020.

[69] Tia selection tool. `https://new.siemens.com/global/en/products/automation/topic-areas/tia/tia-selection-tool.html`. Accessed: 14.05.2020.

[70] Simatic s7-1500 homepage siemens. `https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html`. Accessed: 23.05.2020.

Appendices

A. Appendix - mechanical

A.1 Introduction to turbulent flow phenomena

This appendix gives the general information on the physical phenomena of turbulence in fluid's flows. Mathematical apparatus engaged in studying the turbulence is complex and requires knowledge in some modern, yet to be fully researched, theories. In this paper governing equations are presented amongst their rather simplified explanations, focusing on their practical meaning and not on formal proofs. If more in-depth information is desired, refer to *"Handbook of Experimental Fluid Mechanics"*[6].

A.1.1 Motivation

When providing validation of a CFD analysis one implicitly assumes that the verification of a numerical solution has been done previously, meaning the mathematical/algorithmic framework on which the simulation is running, are accurately solved. However this implies that the person performing the validation understood the underlying physics of the process being simulated and the techniques used. This knowledge will be defining in finding and choosing flow parameters and respective instrumentation that could give a better representation of a flow.

A.1.2 General understanding of a physical phenomenon

Some may say that it is difficult to obtain a strict definition of the term "turbulent flow" or "turbulence". Main reason for this inconsistency is an absence of a fully developed mathematical description of this specific fluid behavior. One can than go on and provide a qualitative explanation of phenomenon or define it by comparing it to a contrasting term - laminar flow. Laminar fluid motion is characterised by smooth flow, where moving particles are confined to distinct parallel layers without any disruption and/or mixing between them. Turbulence on the other hand is defined by chaotic changes on flow pattern: velocity, pressure, etc.

One can say with confidence that laminar flow motion is well studied, theories and mathematical models on this topic are covered in most of the university courses on fluid mechanics. Unfortunately laminarity in a real-life systems is an anomaly and not considered standard. Turbulent flow can be observed in nature: swirling clouds in atmosphere, smoke rising from a campfire. It can also be easily seen around moving bodies: ships, planes and cars (see Figure A.1). One of the biggest problems with describing turbulence is that a path taken by a particle in these conditions is continuous, but not differentiable: it causes eddies and whorls. However, despite flow's visual irregularity, there is a self-similar (some say fractal-like) pattern of the motion. For turbulence is representative:

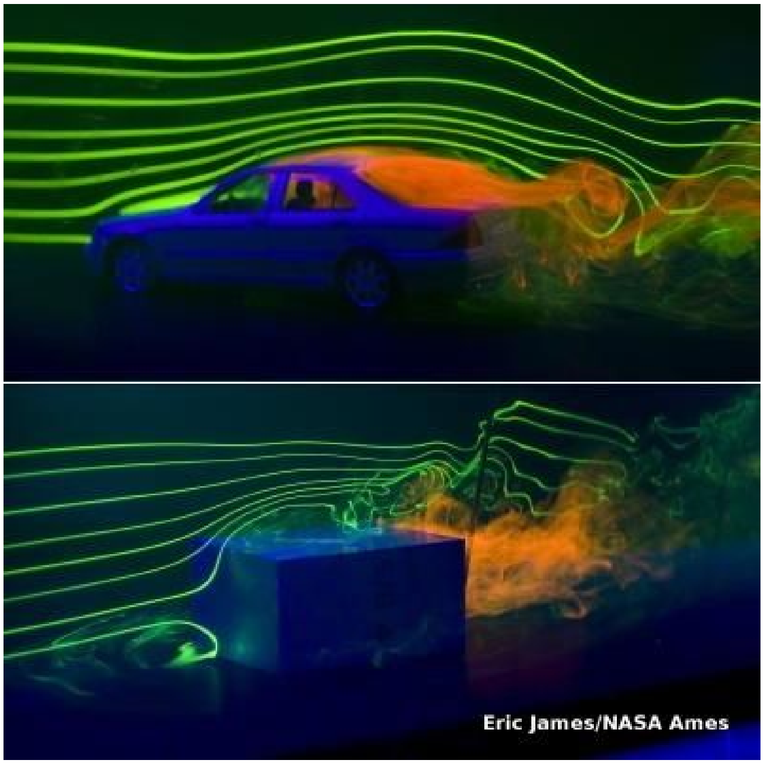


Figure A.1: Representation of transition between laminar and turbulent flow by Eric James - NASA

- High sensitivity to initial and boundary conditions: caused by, among other reasons, high degrees of freedom for each particle.
- Unpredictability, but not randomness: continuous non-linear feedback (see Section A.1.3) in the system causes difficulties in obtaining analytical prediction. However, since equations describing the phenomena are derived from fundamental conservation principles, one can not say that the process itself is random.
- Wide range of scales needed to study the flow: presence of eddies within eddies, that interact with each other.
- Diffusion: mixing together different parts of the fluid, diffusing also energy and momentum.
- Fully 3D nature: caused by large range of scales, shapes and strengths of of eddies, depending on the initial and boundary conditions.

- Cross-dependant parameters: sensitivity to initial conditions causes correlated changes in fluid-mechanical parameters.

A.1.3 Turbulence mathematical modeling

The motion of the fluid can be described by equations obtained by Georges Stokes in 1845. He derived them by applying Newton's second law to a single "fluid particle" taking into consideration fluid's pressure, viscosity and density (A.2). Those equations are based on universal laws of physics and they are able to model any fluid on the earth.

$$\nabla \cdot \vec{v} = 0 \tag{A.1}$$

$$\rho \frac{\partial \vec{v}}{\partial t} = -\nabla p + \mu \nabla^2 \vec{v} + \rho \vec{F} \tag{A.2}$$

The first one is conservation of mass or incomprehensibility equation (in vector form). This means that for the fluid following (A.1) it is true that it maybe changes shape, while moving, but no matter is added or subtracted from it.

The second one (A.2) is a Newton's second law in vector form, in this context it is usually referred to as "momentum equation". The expression on a left hand-side is an acceleration written as a time derivative of the velocity ($\frac{\partial \vec{v}}{\partial t}$) multiplied by fluid's density. Right hand side represents all the forces acting on the fluid:

- Internal - "$-\nabla p$" is a force contribution from a change in pressure. Difference between pressure levels causes fluid to move (from high to low) along that pressure gradient, which is creating a force.
- Internal - "$\mu \nabla^2 \vec{v}$" is viscous friction between layers of fluid.
- External - "F" combines all forces that are acting on a fluid externally, usually it is, for example, gravity.

In three dimensions those equations represent a system of second order, non-linear, tightly-coupled partial differential equations. In this context term "tightly-coupled" means presence of cross dependant variables. Mathematically speaking this means that change in any parameter of the system propagates and effects the original parameter that changed. The non-linearity of those equations refers to the fact that this system's behavior can not be predicted with absolute certainty[15]. These complexities prevent possibility of an analytical approach towards the problem, however some believe that analytical solution exists and even offering *"the 1000000 USD prize [...] for a proof of the existence and uniqueness of solutions to the Navier-Stokes equations"*[6].

Since the accurate solution to equations modeling turbulent flow is yet to be found, it is impossible to theoretically predict it's behavior. However those equations are used by engineers in design making decisions by simplifying them: Reynolds-averaging on a grid covering an area/volume of interest or time-averaging. Unfortunately, averaged governing equations contain not only variables of interest, but also products of those variables, caused by original system of equations being non-linear. Thus, it is impossible to obtain governing equations including only averaged quantities. Hence, they need a further mathematical treatment.

There are some solutions proposed by mathematicians to provide a so called ”closure” (making the number of equations the same number as number of unknowns) by introducing eddies viscosity, mixing length or turbulence kinetic energy. Those methods are topics of study in statistical thermodynamics and mathematical physics, which are out of the scope of this chapter. However, it is worth noticing another approach rooted in a modern branch of science, that might provide an insight on the nature of turbulence. Chaos theory covers *”behavior of nonlinear dynamical systems and their response to initial and boundary conditions”*[16].The main idea of ”chaotic approach” is to provide properties of turbulent flow directly from governing equations without any previous mathematical treatment. Unfortunately, this promising idea is still under development, but if one is interested in current results obtained by researches, please refer to, for example, *”The mathematical theory of turbulence or chaos”*[15]

Reynolds number

A fundamental parameter describing transition from laminar to turbulent flow is dimensionless Reynolds number. It is defined as ratio between internal and viscous forces in the moving fluid:

$$Re = \left| \frac{ma}{F_v} \right| = \frac{mv^2}{L} \cdot \frac{1}{\tau L^2} = \frac{\rho L^4 v^2}{L^3 \mu v} = \frac{\rho v L}{\mu} \tag{A.3}$$

The Reynolds number gives general information on fluid's behavior, it is calculated for specified flow geometry - boundary conditions, and other particular flow parameters at given time - initial conditions. When Re is below a certain critical value viscous forces are large enough to overpower internal forces and smooth down possible instabilities in the flow. Whenever Re is equal or above critical value the flow starts to show the features of fully developed turbulence.[16] To quantify the relation to Reynolds number's size one can imagine some common fluid dynamical processes. For example, a stone dropped in a cup of honey or another thick syrup will create a fluid motion with $Re \approx 10^{-3}$, for the blood flow in our bodies $Re \approx 10^2$, a car driving on a highway $Re \approx 10^6$ and for the submarine rising from the sea $Re \approx 10^8$. A general rule of thumb in this case is that $Re > 1000+$ shows a developed turbulent nature of the flow.

To assist with making theoretical study on predicting a fluid behavior, it is convenient to write Navier-Stokes equation of momentum (A.2) [6] in a dimensionless form. Non-dimensionalization helps to compare the contribution of acceleration, pressure change and viscosity to a fluid motion. For simplicity, one can also assume that there are no external forces acting on a fluid. To remove dimensions from the equation parameters one can refer to scales in A.4. It is worth noticing that length (in a form of an operator ∇), velocity and time are scaled naturally. However, pressure has to be non-dimensionalized for a specific case of a fluid motion. Since in this paper the high speed flow is studied, we choose a scaling factor that depends on velocity:

$$\vec{v} = \tilde{v}v; \quad p = \tilde{p}\rho v^2; \quad t = \tilde{t}\frac{L}{v}; \quad \nabla = \frac{\tilde{\nabla}}{L} \tag{A.4}$$

Scaling down velocity, pressure, time and length:

$$\frac{\rho v^2}{L}\frac{\partial \tilde{v}}{\partial \tilde{t}} = -\frac{\rho v^2}{L}\tilde{\nabla}\tilde{p} + \frac{\mu v}{L^2}\tilde{\nabla}^2\tilde{v} \tag{A.5}$$

Dividing both sides of the equation A.5 by $\frac{\rho v^2}{L}$:

$$\frac{\partial \tilde{v}}{\partial \tilde{t}} = -\tilde{\nabla}\tilde{p} + \frac{\mu v L}{L^2 \rho v^2}\tilde{\nabla}^2\tilde{v} \tag{A.6}$$

$$\frac{\partial \tilde{v}}{\partial \tilde{t}} = -\tilde{\nabla}\tilde{p} + \frac{\mu}{L\rho v}\tilde{\nabla}^2\tilde{v} \tag{A.7}$$

Substituting $Re = \frac{\rho v L}{\mu}$:

$$\frac{\partial \tilde{v}}{\partial \tilde{t}} = -\tilde{\nabla}\tilde{p} + \frac{1}{Re}\tilde{\nabla}^2\tilde{v} \tag{A.8}$$

Where in A.8 dimension free contributions are: $\frac{\partial \tilde{v}}{\partial \tilde{t}}$ for acceleration, $-\tilde{\nabla}\tilde{p}$ for pressure fluctuations and $\frac{1}{Re}\tilde{\nabla}^2\tilde{v}$ for internal friction forces. For turbulent flow the Reynolds number is per definition high, which means that viscose forces for that fluid motion can be assumed negligible compared to pressure and acceleration effects.

$$\frac{\partial \tilde{v}}{\partial \tilde{t}} = -\tilde{\nabla}\tilde{p} \tag{A.9}$$

Resulting equation A.9 doesn't imply any external forces, but still it is difficult to solve, since it is non-linear due to the acceleration part. However, even though one doesn't have the solution, equation can be used to elaborate on the parameters that play the key role in fluid's motion. It can be seen then that the governing process in particle movement for this case is pressure gradient.

A.1.4 Fluid flow analysis methods

In general there are three analysis' methods of a fluid behaviour: analytical, computational and experimental (Figure A.2). Analytical method usually yields exact precise solutions, but works only on a few relatively simple cases, since governing equations of fluid motion are not easily approachable for majority of real-life problems. Another way of obtaining high-accuracy data is conducting an experimental study. This method unfortunately has its own drawbacks, since it is usually expensive and requires special facilities. Instrumentation techniques used in those studies must be carefully designed and calibrated, which implies that one has to already know what kind of phenomena is being tested. That means that this method fails to predict the outcome of the experiment if wrong instruments have been used. Computational analysis can be as accurate as experimental methods, due to proven mathematics and computational power behind it. At the same time CFD is most often cheaper and more flexible then conducting an experiment.

At their core CFD and analytical methods use, among others, Navier-Stokes equations to describe real-life phenomena. This on a bigger scale means that CFD software can be compared to a sophisticated graphing calculator, where computed results obtained are dependant on an operator, who conducts the simulation. CFD is not a tool that can be used without sufficient preliminary knowledge on the subject of fluid- and thermodynamics. Operator has to perform a quality control procedures, checking if the simulation resembles the real world phenomenon.

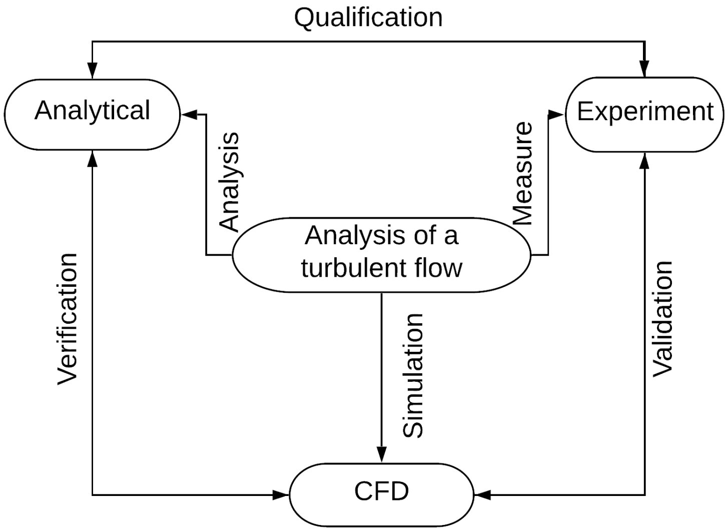


Figure A.2: Fluid dynamics' methods of analysis

For mathematical analysis one applies governing equations for the whole testing volume or mass of fluid. Brief explanation of those methods is presented in section A.1.3. To summarise the workflow one can follow:

1. Write the down the governing equations.
2. Derive boundary conditions from the environment of the phenomenon.
3. Choose parameters of interest for assessment.
4. Apply appropriate mathematical treatment. Usually averaging or statistical treatment.
5. Solve and evaluate.

For computational simulations, in contrast to analytical approach, equations are applied to a single, very small finite volume of fluid. Body in the study then is subdivided into small particles or cells, that are usually (under condition that no stress applied) are in a shape of cubes with side-lengths $\delta x, \delta y$ and δz respectively (Figure A.3). Each single cell interact with the neighbouring ones by the laws of physics following predefined equations. The accumulation of those cells is also called *mesh*. Software solves a set of partial differential equations for each of those particles, combines the resulting values and plots the data either in form of graphs, tables or field plots (Figure 1.4). This way of handling a fluid mechanical problem resembles to some extend averaging technique that has been discussed previously in mathematical modeling (section A.1.3).

The force contributions to a single cell are generalised in a form of: normal stresses, shear stresses and body forces. Normal stresses represent direct pressure. Shear stresses are formed usually by friction, and body forces are built up of gravity, magnetic or other external forces. One can the write down the equation of momentum (A.2) in terms of cell's stresses:

$$\rho \delta x \delta y \delta z \frac{\partial v}{\partial t} = (\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z}) \delta x \delta y \delta z + \rho F_x \delta x \delta y \delta z \tag{A.10}$$

Where boundary conditions are represented by the geometry interacting with the fluid. It is important to notice that in A.10 only one axis is considered, full cell definition is achieved through assessing all tree axis. The major differences from A.2 is that viscous forces and pressure gradient contributions are combined into common sum $\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z}$.

Conducting a computer simulated fluid dynamic study usually requires thorough preliminary analysis of the problem. Operator has to build scientifically reasoned expectations before running the software to be able to judge if resulting solution is valid. This can be pointed out as a major weakness of a CFD, since there is no clear automatically generated indication if the simulation's outcome is faulty. The workflow for this method can be generalised in a number of steps; for more detailed description of the process please refer to *"Advanced Computational Fluid and Aerodynamics"* [17]:

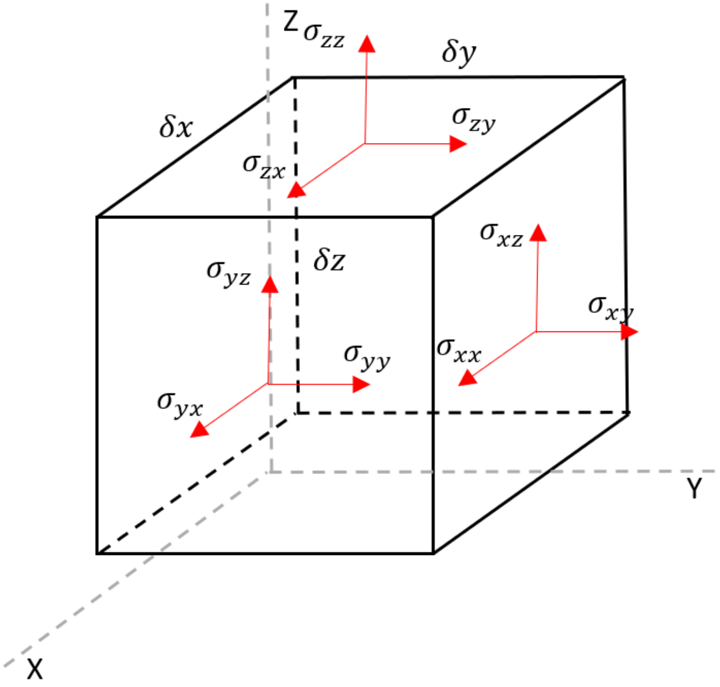


Figure A.3: Finite element of fluid with stresses

1. Geometry preparation: building the geometry or importing model in supported format; removing unnecessary small details, that would otherwise put additional load on the processor.
2. Simulation - has to be run several times to obtain best results
    - Meshing
    - Set up physics environment
    - Choose a solver (mathematical model)
    - Run the simulation
    - Post processing
3. Evaluation

CFD, mathematical analysis and experiment have their own strengths and weaknesses. Some may say that they are not independent techniques used to make engineering decisions, they rather coexist in a symbiotic relationship: one can also be used as either qualification, validation and verification tool for another (see Figure A.2). In the context of current paper authors assume that mathematical modeling has been done correctly with following verification of CFD reports. The task of the research group then is to design an experiment that will work together with existing solutions to create a clearer picture of a fluid-dynamic process.

A.2 Turbulent flow study: experimental approach

This appendix covers basic instrumentation techniques used in experimental fluid mechanics nowadays. The main focus is the equipment and techniques suitable for studying a turbulent flow. Instruments and their use will be compared on the basis if their general applicability to exhaust gas mapping problem addressed in this paper, their cost-effectiveness measure and other clients wishes.

A.2.1 Motivation

Even though CFD simulation is recognised as precise and powerful analysis tool in fluid and aerodynamics, nevertheless it fails to give an independent description of exhaust gas flow. It has been stated that authors' mission is to find a way to conduct a sufficiently accurate experimental study that is capable of providing validation to simulation. The initial idea would be to study existing solutions on the market and try to apply them directly. However, if no such solution exist, the goal will be to use well-researched and proven technologies in a new context.

A.2.2 Difficulties in conducting the experimental study

An experimental fluid mechanics technically is an attempt to isolate a real-life fluid and measure its mechanical and thermodynamical properties. Turbulent flow has been proven to be a challenging fluid-state to study. In a context of measuring the gas-turbine exhaust gas flow additional challenges occur. CFD analysis that is to be validated, has been performed by the client for the variable loads on the engine. This means that it is desired to have a relatively continuous range of data for each characteristic point on the cross-section that corresponds to points in the simulation. Most of the data acquisition systems available on the market (see section A.2.3) are for steady state fully developed fluid motions. Steady state here means that mechanical parameters for each point in the volume of fluid do not change with time. In practise this implies that one can use a single sensor to gather data on a bigger volume by moving it in space. This approach is not suitable for gas turbine exhaust mapping as sensor must be in the same place during the whole test run.

Another difficulty associated with space is that data acquisition system might intrude with original movement of the fluid in the test volume. That can cause collection of the wrong data, since boundary conditions of the original phenomenon are changed, hence intrusiveness of the sensors must be assessed with care.

It has been mentioned above that turbulence is a fully three dimensional physical occurrence that rapidly changes in time. This means that testing system must be capable to detect changes in the fluid both for the smaller and the bigger scales. Those time and space characteristic lengths must be either derived analytically for a given fluid, or extracted from CFD analysis.

Three dimensional nature also causes difficulties in measurement volume definition. For this paper it has been stated that data shall be collected in a specific cross-section of exhaust duct, which significantly simplifies the task.

A.2.3 Characteristic lengths and spatial resolution calculation

Numerical predictions for time and length scales is a crucial preliminary stage for choosing instruments for data acquisition system. Those quantities are direct functions of a physical state of the fluid: density, viscosity and linear speed of fluid's motion. It is important to notice that those parameters are not constant, but change with the temperature of the environment. Since in this paper authors study motion of gas-turbine exhaust gas, they choose to asses those quantities in a range of temperatures from $300°C$ to $600°C$ with $100°C$ increments (see Table A.1). Viscosity and density are collected directly from engineering data spreadsheets. They also define velocity through averaged mass-flow ($90kg/s$). The linear velocity in this case is a highly approximate number and can be used only to determine the precision range of the instruments.

$T[°C]$	$\rho[\frac{kg}{m^3}]$	$\mu[10^{-6}\frac{kg \cdot s}{m}]$	$v[\frac{m}{s}]$
300	0,62	47,54	46,47
400	0,52	63,82	54,72
500	0,46	77,72	62,76
600	0,40	94,62	70,89

Table A.1: Physical properties [18],[19]

Calculations for spatial (space) resolutions base on Reynolds number assessment. The minimum characteristic length is derived from A.3 as $d_{min} = \frac{Re \cdot \mu}{v\rho}$ with $Re = 1$. For this Reynolds number value turbulent kinetic energy is beginning to be destructed by viscous forces[17]. Maximum spatial length is calculated using Kolmogorov's theory of internal turbulence. There, based on kinetic energy dissipation cascade, he proved that maximum and minimum length scales being proportional with coefficient $Re^{\frac{3}{4}}$[6]. Since at the time of writing this chapter authors didn't receive a complete CFD-report from the client, they assume general value for Reynolds number for turbo-machinery exhaust flow. It is decided to be $Re \approx 10^4$[20] until better quality data is obtained.

$T[°C]$	$d_{min}[10^{-6}m]$	$d_{max}[10^{-3}m]$	$t_{min}[10^{-8}s]$	$t_{max}[10^{-4}s]$	$f[kHz]$
300	1,7	9,3	3,6	2,0	5,0
400	2,2	12,5	4,1	2,3	4,4
500	2,7	15,2	4,3	2,4	4,1
600	3,3	18,6	4,7	2,6	3,8

Table A.2: Measuring resolutions for velocity components instrumentation

After one derives spatial resolutions, it becomes relatively easy to calculate time scale: one simply divides obtained lengths by the linear speed of fluid. Since most of the documentation on instruments lists detectable frequency as their sampling rate characteristic, it is useful to represent time range in $[Hz]$. The minimum time scale for turbulent fluctuations is much smaller then time period it takes for flow to develop. Therefore, for most of the engineering calculations and experiments the small-scale unsteady processes are out of interest.[17] Hence, the descriptive frequency in Table A.2 is a function of a macro-scale parameters.

A.2.4 Measurable parameters of interest and existing instrumentation techniques

To decide what kind of flow variables are relevant to measure for obtaining a fully defined fluid flow's experimental model, one can refer to fundamental law of substance and Navier-Stokes equations. There in fundamental law pressure can be derived as a function of density and temperature. Navier-Stokes then provides connection between pressure, viscosity, density and velocity. Since authors' current focus is turbulent flow, they cannot rely completely on analytical approach by measuring only some of those parameters. Therefore, to collect sufficiently accurate data it would be more effective to engage measurements for all variables of interest. However, since the topic of this paper is to validate a CFD generated velocity profile, authors will focus on data acquisition systems for this parameter. From Table A.3 one can see a range of commercially available instrumentation techniques for measuring velocity components.

	Velocity components
Techniques	1. Thermal anemometry 2. Particle image velocimetry 3. Pressure based

Table A.3: Measuring techniques for relevant flow variables [6]

When choosing a technique that will be used for measuring parameters describing fluid in motion, it is important to address a number of aspects, which define one or another technique[6]:

- Space scale of the measuring system or its ”spatial resolution” - the range of detectable lengths.
- Time scale - how often the system is capable to extract data from the fluid.
- The characteristic size of the system measured.
- The duration of an experiment.
- How intrusive the system is.

Thermal anemometry

Thermal anemometry or hot-wire anemometry is a technique that is used to measure instantaneous directional velocity and temperature. This method relies on changes of electrical resistivity in a small heated wire that is exposed to the moving fluid. When the electric current is passed through the wire, the heat transfer varies with the flow rate. Changes in resistance are picked up in form of the signals, which are then conditioned and monitored by specially designed electronic circuits (see Figure A.5). For measuring velocity in three dimensions a multi-wire probe is used (see Figure A.4). Each wire responds to the component that is perpendicular to it. Filaments are usually very small, about $4 \cdot 10^{-3} m$ from one supporting needle to another, this causes detectable length of the sensor being $\approx 10^{-5} m$. The sampling frequency of this method is comparable to average fluctuations speed in a real fluid (see Table A.4 and A.2) However, even though hot-wire probe is a high precision tool, it is not necessary suited for conducting an experimental study on bigger volumes. The limitations occur due to wires finite length and finite thermal inertia. To take a quality

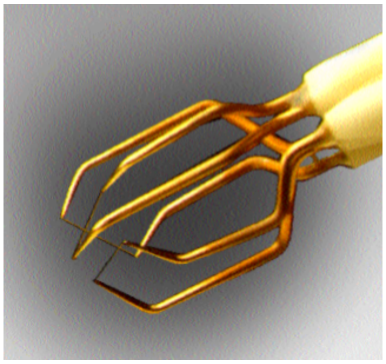


Figure A.4: Gold plated hot-wire probe. Manufactured by Dantec Dynamics. [21]

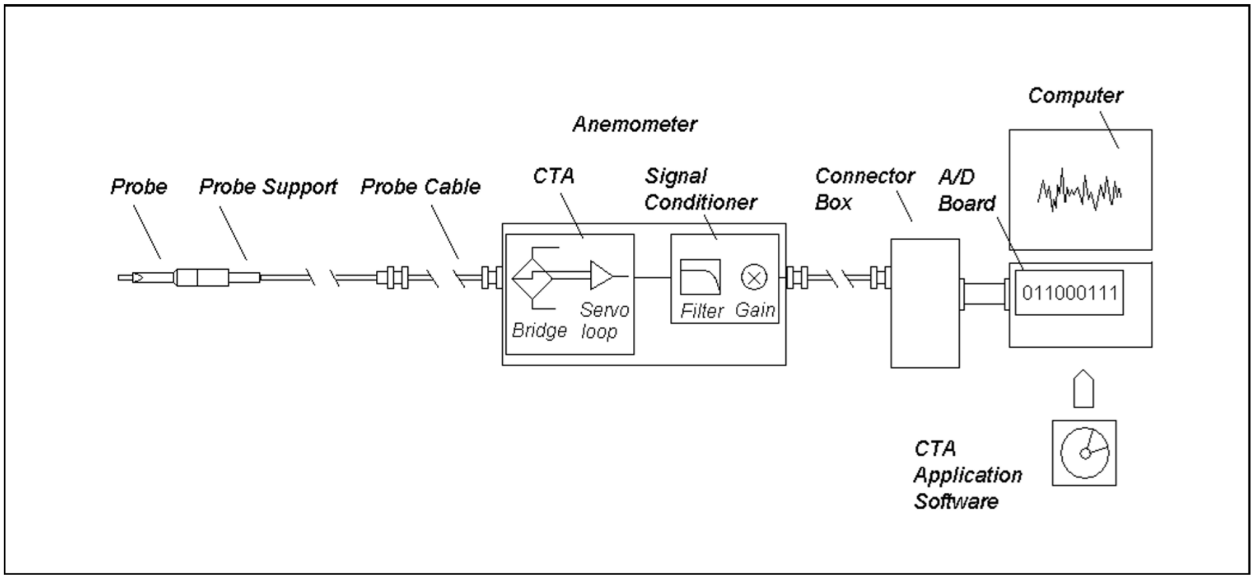


Figure A.5: Typical measuring chain with hot-wire probe by Dantec Dynamics. [21]

measurements in a system with a bigger characteristic size would require a multi-probe assembly, which increases investigation costs dramatically. One heavy duty three-wire probe, that can withstand temperature up to $700°C$ costs 34000 NOK [8] with calibration expenses, mechanical support and cables not included. The limitation of using a multi-probe solution is that probes will need supports for being placed in the positions that are further from the walls. The diameter of a support pole is 45 mm, which is expected to intrude significantly with moving fluid and possibly cause collecting of inaccurate data.

Pressure-based velocity measurement

Fluids velocity can be derived from total and static pressures using Bernoulli equation, which allready puts some constraints on this instrumentation technique, because equation A.11 is valid along a single streamline in steady flows where incompressibility equation A.1 holds.

$$\frac{dp}{\rho} + vdv = 0 \tag{A.11}$$

To extract static and total pressures from the stream a well-known device called Pitot tube is used (Figure A.6). It has two holes that are exposed to the fluid. The front whole is positioned towards the stream and is measuring stagnation or total pressure, the hole on the side is measuring static pressure. Those values are then plotted in equation A.11 and speed is calculated.

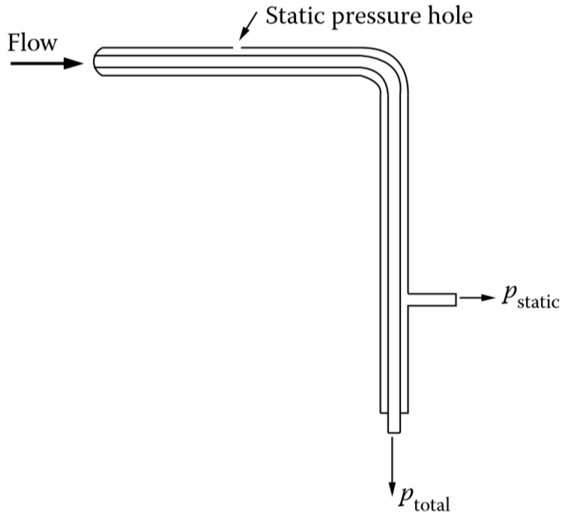


Figure A.6: Pitot probe schematic [22]

To derive flows velocity in the room a larger three-dimensional multihole probes are used (Figure A.7). For all pressure-based equipment determining the velocity major issues are:

- Intrusiveness due to the larger size of the probe.
- Difficulties in the positioning of the probe, where angle of the incline constraint is usually violated in highly turbulent flows.
- A single streamline assumption almost never holds for real-life conditions.
- Measurements must be averaged over longer time, hence there are difficulties in obtaining instantaneous velocities with frequency demanded.

However, carefully designed measuring devises can minimize the effect of those issues [6]. The fabrication of pressure-based acquisition systems is relatively inexpensive and it can be also supplied in a from of portable pre-calibrated test-kit that can be used by a single operator (Figure A.8).

Concept of operation for 3D pressure probe is somewhat close to hot-wire sensor. That implies similar drawbacks in covering larger measurement volumes. Spatial resolution (Table A.4) of a single probe is simply not enough to cover the area of exhaust duct, therefore one has to use a multi-probe assembly (Figure A.9).

For gas-turbine exhaust gas study probe or probes must withstand high temperature environment for at least 4 hours. Suitable equipment can be provided by *"Airflow Sciences Equipment, LLC"* in a form of a spherical 3D-probe with a water-cooling system. Purchas-

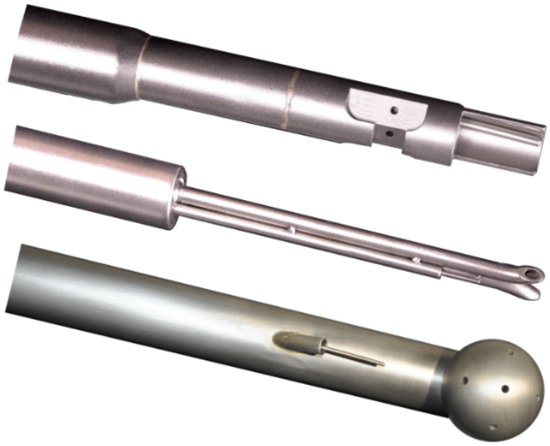


Figure A.7: Heavy duty 3D probes. Manufactured by Airflow Sciences Equipment, LLC. *Photo taken from 3DDASTM Data Acquisition system and Accessories Price List – 2020*

Figure A.8: Full system for measuring flow velocity. Manufactured by Airflow Sciences Equipment, LLC. *Photo taken from 3DDASTM Data Acquisition system and Accessories Price List – 2020*

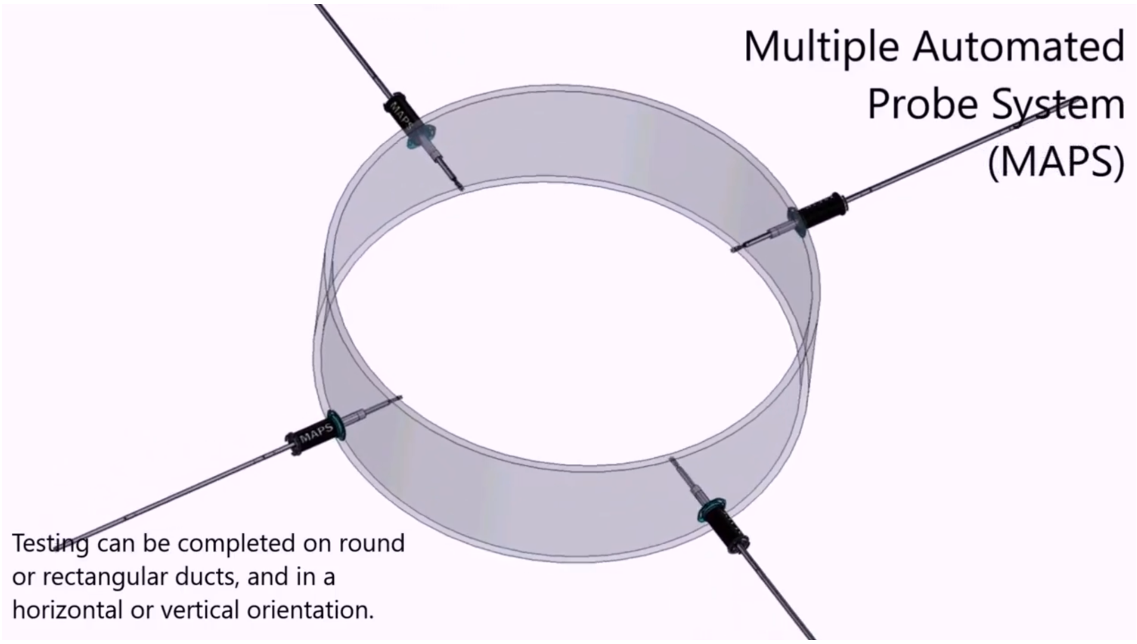


Figure A.9: Multiple Automated Probe System. Manufactured by Airflow Sciences Equipment, LLC. [23]

ing cost of one sensor will be 90300 NOK with additional costs for umbilical cord for signal transferring (17000 NOK) and calibration expenses (15200 NOK per sensor). It is also important to point out that it is impossible to use their full test-kit with PC and power supplies in offshore environment, since there is no documentation available that it is qualified for this application.

Particle image velocimetry

Particle-based instrumentation techniques are based on tracing seed-particles placed in the flow using sophisticated optical systems. This method works under assumption that particles will move with the same velocity as surrounding fluid and are present in sufficient number to provide desired resolutions. Seed-particles must also have good light-scattering features, being at the same time light enough and neutrally buoyant to minimise effect of gravitational forces. Silica particles (2 μm in diameter) is a frequent choice for measurements in exhaust gases in combustion processes [6][24].

Simplified concept of operation with all hardware required can be seen on Figure A.10. Laser and optics there are used to illuminate seed particles to track fluids motion. One usually uses high power double pulse laser to capture movement with high spatial and time resolution. High-speed camera is connected to the laser through a synchronizer to obtain simultaneous illumination and image capture. Visual frames are then transferred to computer for storage and processing. Specially designed software with machine learning algorithms correlates particles' position in two consequent images, computes displacement that occurred during the time interval between frames and then calculates velocity of a particle (Figure A.11). This method can be used to conduct an experimental study for combustion processes. With the help of high-power lasers and diffusion optics it is possible to cover sufficiently large testing areas, making method's wide spatial resolution a very unique feature compared to other solutions (see also minimum detectable length in Table A.4). With application of high-speed cameras it is possible to acquire data with frequency comparable to fluctuations-rate of a real-life fluid. At the same time particle image velocimetry is almost non intrusive, since seed-particles are chosen to make as little impact on the flow as possible, and no other physical objects are inserted in the measuring volume.

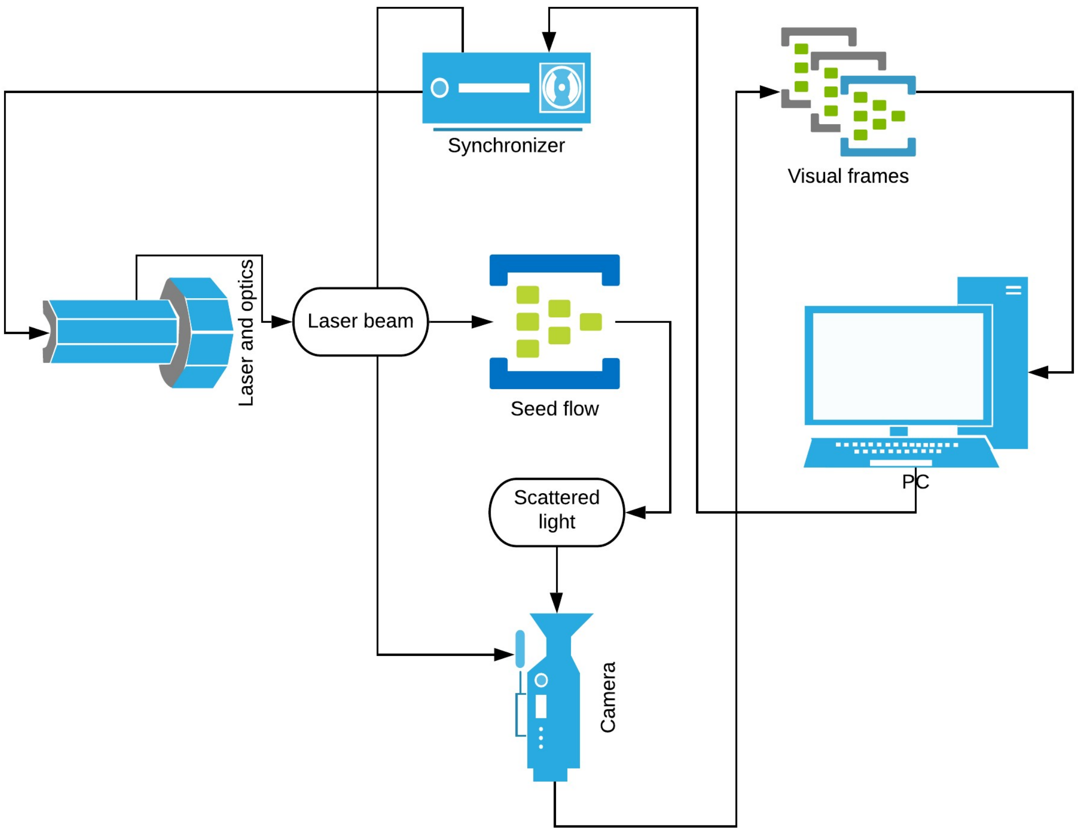
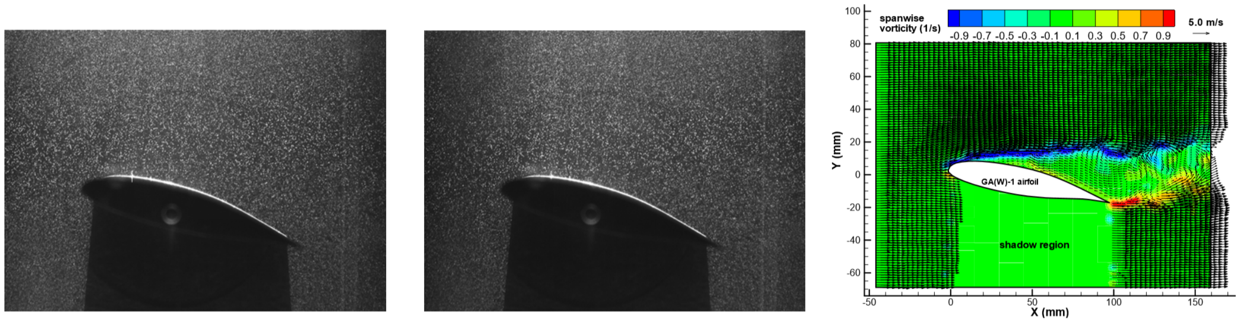


Figure A.10: Particle image velocimetry concept of operation

The main drawbacks of this instrumentation technique are the high purchasing cost and its general immobility. Particle-based solutions require special devises and test-rigs that can



A. $t=t_0$

B. $t=t_0+10\ \mu s$

C. Derived Velocity field

Figure A.11: Particle image velocimetry example [24]

take days to assemble: particle feeding constructions in case of gas-turbine exhaust duct will imply structural changes in current testing facility layout. Additionally one would have to purchase special shielding systems made of see-through durable materials to protect costly optical equipment. Even though this method doesn't necessary need frequent calibrations, funds required to purchase of, for example, a high-power laser would cover the budget for complete 3DDASTM system (over 500000 NOK depending on supplier and model).

A.2.5 Instrumentation techniques comparison and evaluation

Commercially available data acquisition systems come with with different detectable lengths and with variety of time resolutions. Time and space scales for thermal anemometry, particle image velocimetry and pressure-based techniques parameters are summarised in Table A.4.

Parameter	Formula	Thermal anemometry	Pressure based	Particle image velocimetry
Minimum detectable length $[m]$	d_{min}	10^{-5}	$3 \cdot 10^{-3}$	10^{-4}
Minimum detectable time $[s]$	t_{min}	10^{-4}	10^{-3}	10^{-4}
Spatial resolution $[m^{-1}]$	$SR = \frac{1}{d_{min}}$	10^5	$3 \cdot 10^2$	10^4

Time resolution $[Hz]$	$TR = \frac{1}{t_{min}}$	10^4	10^3	10^4
Real detectable frequency $[kHz]$	$f_{max} = \frac{1}{2t_{min}}$	5	$0,5$	5

Table A.4: Measuring resolutions for velocity components instrumentation

Comparing real detectable frequencies for each method with averaged fluctuation frequency of turbulent flow from Table A.2 ($\approx 4,2kHz$) one can easily see that pressure based technique with $f \approx 0,5kHz$ will not provide desired time resolution for the experiment.

Test ID	Description	Thermal anemometry	Pressure based	Particle image velocimetry
MT1, MT2, MT3	Withstand $550°C$ for 4 hours	Passed	Passed	Passed
MT4, MT5, MT6	Is portable and can be assembled in short time (2 hours)	Passed	Passed	Failed
MT7, MT8, MT9	Is capable of extracting data that is comparable with CFD reports	In progress	In progress	In progress
MT10, MT11, MT12	Can be used in testing facilities and in a final product destinations	Passed	Passed	Failed

MT13, MT14, MT15	Can be used to produce plots that are representable of velocity field	Failed	Failed	Passed
MT16, MT17, MT18	Can be fitted in the opening under flexible. Can reach all points in section.	Passed	Passed	In progress
MT19, MT20, MT21	Measures three components of velocity	Passed	Passed	Passed

Table A.5: The acceptance testing for the scope: commercially available velocity measuring techniques tested against mechanical requirements (refer to appendix E.1.1 for detailed test results)

Thermal anemometry, with hot-wire probe as its sensing tool, is capable to pick up changes in velocity with sufficient spatial and time resolution on the smaller scale. However, it has been discussed previously that this method will require a multi-probe assembly which is both expensive and highly intrusive. To solve this issue one must run a statistical analysis based on CFD reports to find the required sensors quantity and their geometrical positioning. One has to run additional CFD analysis and analytical calculations for new boundary conditions (together with a multi-probe system) to find if it causes critical disturbance in the flow, causing hot-wire probe picking up the faulty data. If errors will be comparable to ±5% of probes resolution, then using this system will defeat the purpose of high precision expensive hardware.

At the first glance particle image velocimetry is a method that would be the solution for the scope of this project. There are number of research papers proving that this technique is applicable for experimental studies on high speed, high temperature gasses in variety of test-volumes' sizes. Unfortunately, it is currently impossible to implement particle image velocimetry neither in the clients testing facility nor at the gas-turbine power generating packages final destinations.

All three commercial methods have been tested against mechanical domain requirements (see Table A.2.5). It can be seen that all techniques failed at least one of the tests, making them at current stage being inapplicable for the scope. Theoretically it is possible to design a multi-probe system using thermal anemometry, addressing MT13-test failure, but it has been decided together with the client that not any of those techniques shall be implemented.

A.2.6 Combining experimental solution and simulation

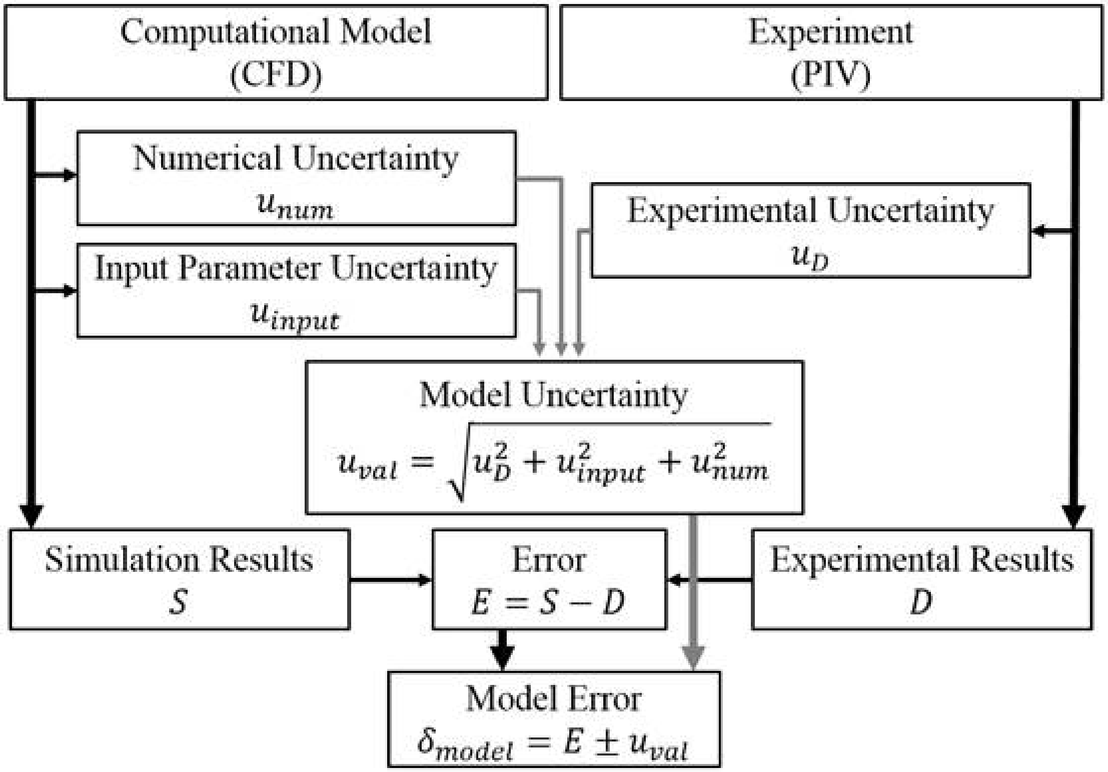


Figure A.12: Sources of error in fluid dynamic study[25]

It is known fact that not any research method, besides maybe mathematical analysis, give an absolute answer, therefore it is important to discuss errors and uncertainties connected to the model description. Some may even say that not any model is valid until uncertainty is defined.

As any physical phenomenon turbulence has three general modeling methods:

- Describing analytically, usually using partial differential equations.
- Simulation - solving equations with numerical methods and other mathematical methods, using computers.
- Conducting experimental study.

It has been mentioned in A.1.4 that three methods above work together in piecing up a model of fluid in motion. Where one of the techniques fails to add to the knowledge about fluid's behavior, another one rises up to the task. Additionally they are connected through their general architecture: CFD is based on numerical solutions produced by mathematical analysis and experimental research often refers to CFD-reports. Hence it is logical to combine error associated with different techniques and use it as a general indicator of the quality of the model.

Figure A.12 represent a flow-chart diagram with sources of errors in fluid dynamic study. The process summarised there will be used further in this paper when discussing results obtained for exhaust flow mapping.

A.2.7 Validation of CFD

Finding the way to validate the CFD-based fluids model is the main purpose of this research paper. The goal is to study the input parameters, geometry, meshing algorithms and mathematical solver used by the client in the most recent available report. Based on this information research team will build a testing system replicating the CFD set-up to extract a similar data for velocity from the real fluid.

It expected that designing the experiment may take several iterations on the entire project, which is not possible to implement in residual time for current research team. However, another solution can be proposed. It is possible to measure a couple of values for another thermodynamical parameter, preferably scalar one, in the cross section of interest. Then one plots extracted data against one that has been simulated. This way one can make a judgement on general validity of the CFD algorithm.

It has been discovered that previously such a point study in exhaust duct has been done in clients testing facility. The measured parameter has been temperature. After contacting test-responsible in Drammen research team found out that it is difficult to conduct this study immediately, because that would require purchasing additional equipment that has not been budgeted with management. However, even though this solution can not be run in current project iteration, one can see it with certainty as an alternative type of validation technique.

A.3 Mechanical design

This appendix describes a design procedure of the mechanical part of data acquisition system. It starts with outlining the generic design workflow, then goes on to choosing a suitable technique for extracting data from the moving fluid.

A.3.1 Motivation

At present time there is a variety of commercially available equipment on the market. Point-probes in the form of pressure-based and hot-wire sensors are inapplicable to current scope of the problem. Particle image velocimetry has been a promising solution, but it has been found impossible to implement in the clients testing facilities. This gives a rise to the need of designing new solution that would use proven scientific methods of data extraction.

A.3.2 Design process

Starting point of mechanical design will be to study existing techniques implementing for the probes in industrial scale data acquisition systems. They either use electro-chemical (hot-wire), elastic (pressure transducers) or photo-electrical (particle image velocimetry) properties of the material that is exposed to moving fluid. Changes in those parameters are collected usually in a form of electrical signals that afterwards are treated in micro controllers (see Figure A.13).

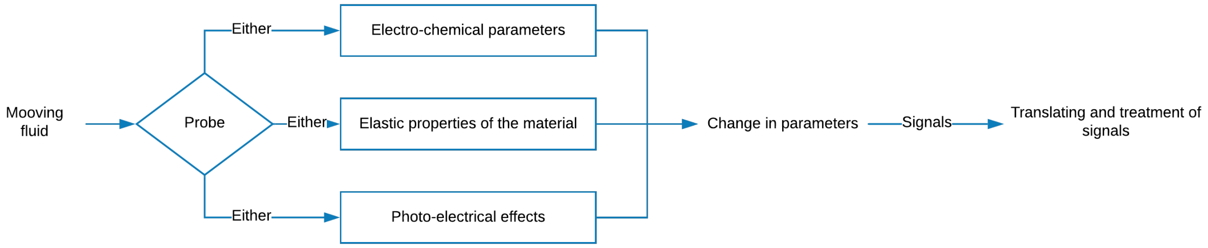


Figure A.13: Concept of operation generic probes

Based on existing solutions one can generalise a design process of the test set up as:

- Choose what kind of the material/mechanical/chemical/electrical/photo property will be utilised in the probe
- Conduct preliminary study - mathematical modeling on how this parameter changes in time when the probe is exposed to the moving fluid

- Find how change in fluids velocity affect this property - find mathematical relationship between this parameter and fluids velocity
- Learn how to control changes in this property
- Learn how to record changes in this property
- Conduct the study on probes spatial and time resolution: how fast the property can change, identify the volume of interaction with fluid. Is averaging possible?
- Preliminary analysis for structural implementation:
  - How to insert the probe into the moving fluid.
  - How to keep in stationary.
  - For how long can it withstand the environment.
  - How intrusive it is.
  - How to retract the probe.
  - List down additional support systems: power supplies, cables, pumps, cooling systems etc.
  - Assess storing, rigging, packing and transporting of the probe and support systems.
- Manufacturing processes of the data acquisition system.
- Cost-effectiveness analysis.
- Cross analysis with existing solutions.

A.3.3 Preliminary design proposal

Discussing redesign of currently available probes

When designing a new probe for three dimensional velocity measurements it is useful firstly to see if current techniques can be implemented if some changes are applied. It has been mentioned that off-the-shelf instruments are not suitable for the current scope, therefore one can discuss if the issues can be somehow resolved, keeping the technology in the probe the same.

Pressure based methods utilise deflection in a thin membrane: changes in pressure generate force that affects membranes shape. To achieve required precision of data extraction it must be small, so measured deflections would be comparable with membranes size. Hence, using higher quality transducers might increase sampling rate of the probe, addressing high frequency nature of the turbulent flow. However, another major issue with this technique has been "Bernoulli" - constraint (see Appendix A.2.4). Since Bernoulli's equation is the mathematical model that transforms pressure into velocity, it is impossible to eliminate it, making technique inapplicable for turbulent flow experiments.

Hot-wire techniques rely on changes in electrical resistivity of the material. The main issue there has been its inability to measure speeds on larger volumes, due to the size of the probe. One solution to this problem could be to choose a sensing wire that is significantly thicker or longer, to make contact area between moving fluid and filament larger. However, longer wire will easily brake in harsh environment. On the other hand, making it thicker will result in uneven heating of the filament, so common mathematical models used to compute velocity components will fail.

Particle tracing techniques rely on expensive optical equipment. Major difficulty there was method's immobility and price. Unfortunately, in this case cutting costs would result in purchasing the system that is incapable of measuring required parameters and even being destroyed before the test is finished. As mobility and purchasing cost was of a greater issue for the client, research group decided to not continue development based on this method.

Hypothesis: heat conduction as the property of interest

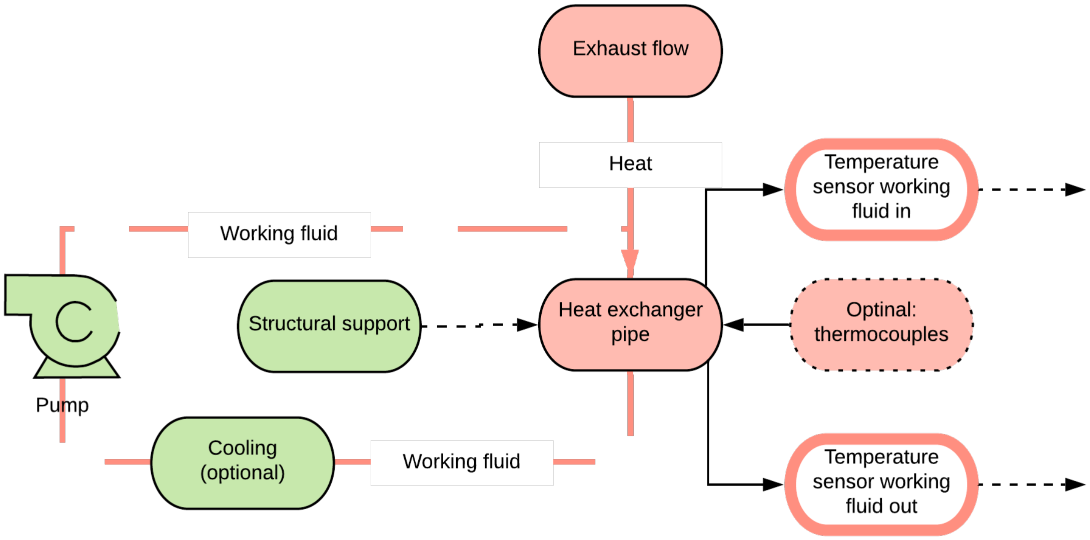


Figure A.14: Mechanical concept of operation for sensing heat exchanger prototype

After studying existing techniques and consulting with external experts, an idea came up, that sensing media that experiences exhaust gas not necessary has to be stationary. Like electrons stream in hot-wire, one can use other moving particles that are contained in some volume to make a judgement about the surroundings. Those particles might be exposed through a certain interface to the moving fluid, which then will cause changes in the material's properties they are a part of.

Based on the fact that exhaust gas carries a significant amount of heat, and heat is directly connected to kinetic energy of particles, research group sets up a hypothesis:

"It is possible to use material's (fluids) heat conducting properties to detect movement of surroundings".

To test this hypothesis one can model a heat-exchanger-like construction system with a fully-defined working liquid, which is circulating in the pipe of known dimensions and material, that is inserted in the hot air flow.

Research team would hope to prove or disprove, that if knowing physical parameters, mass-speed and temperature gradient of the working fluid; geometry and material of the pipe, it is possible to derive average speed of the surrounding exhaust gas.

In real-life testing this system would have a general concept of operation illustrated in Figure A.14. This solution will require at least one additional device to circulate the working fluid. In our case this will be a pump, that must be chosen with care, as fluid can be extremely hot and under high pressure. Additionally design engineer working on this concept must conduct a thorough thermodynamical analysis on the working fluid, since it is unwanted that it will boil or evaporate, which could be solved by introducing a cooler.

It is obvious that this system will likely have drawback by not being as compact as some existing solutions on the market, however this could be resolved if the system could be connected to the boiler in the Siemens' testing facilities. Additionally heat exchanger's tubes can be manufactured based on CFD analysis contour shapes, which can possibly give sought after results.

An example of possible implementation can be seen in Figure A.15.

Working fluid problem

It has been mentioned above that phase-transformation of the working fluid is a big concern for the design. Originally, research team proposed to use liquid water as a sensing media. However, it has been pointed out that introducing support systems that would either maintain water in its liquid state or recycle it after boiling and vaporisation would result in unnecessary complication of the system. Additionally, one can not rely that boiler-interfaces are identical both in testing facility and in the final destination of the GT-package. Hence, it has been decided to use air as a working fluid, since it is not expected to change its state from gaseous during the test-run. Therefore, all further analysis shall refer to instrumental

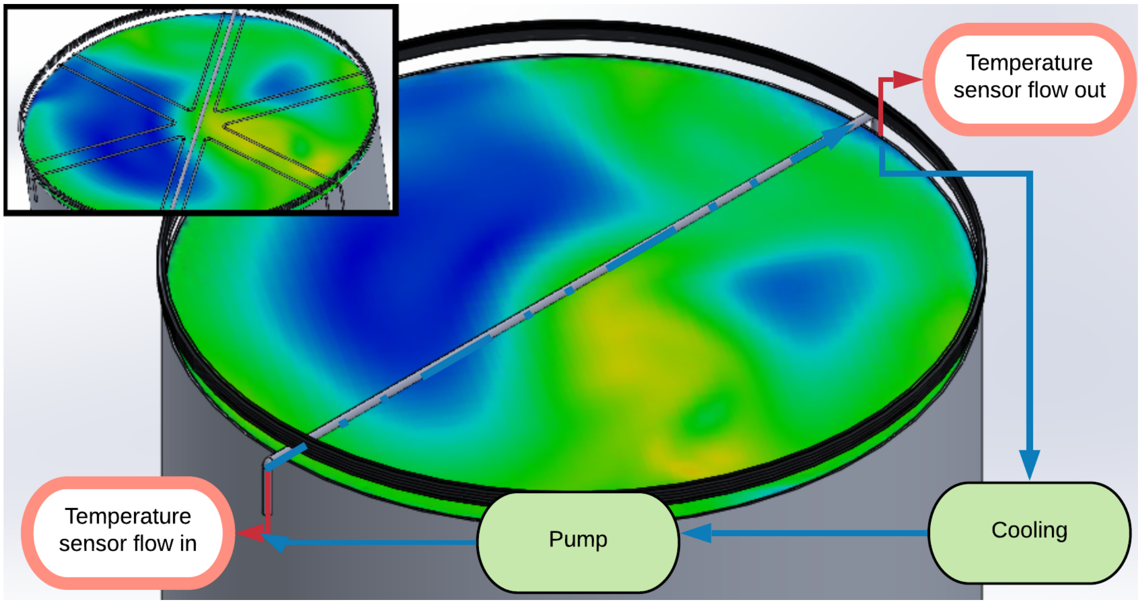


Figure A.15: Mechanical concept of operation for sensing heat exchanger approximate visualisation

air as a sensing media. Interface between measuring system and instrumental air is unknown in the current state of the project. Until further information is introduced, this connection is assumed to be a black-box (see subsection 3.2.2 in System's evolution).

A.3.4 Heat exchanger background theory

Heat is undoubtedly something that everybody experiences on daily basis. For example, a cup of freshly brewed coffee gets colder overtime. This happens because warm liquid looses some of it's *heat* content into cooler atmosphere.

By definition, *"heat is the form of energy that is transferred between two systems (or a system and its surroundings) by virtue of a temperature difference"*[26].

To predict the outcome of phenomena, where temperature difference is present, one refers to thermodynamics and it's analysis techniques. The behavior of any thermodynamical system first and for most obeys the conservation of energy principle (the first law of thermodynamics) and Kelvin-Planck statement (the second law of thermodynamics)[27]. Those laws set up a framework for heat balancing in the system: the energy flowing from one body to another is neither created or destroyed, and not any real working fluid is capable of transferring all of its heat to another body. Recognition of energy interactions happens at the systems boundary, therefore it is of a biggest priority to model the process before engaging into any analysis.

Part of the modeling process requires understanding the mechanisms of energy interactions. They are as follows (see figure A.16):

- Heat transfer - change of internal energy of the system through varying energy of the molecules/atoms.
  - Conduction: temperature gradient in the solid body causes heat transfer form high to low energy area
  - Convection: heat transfer between moving fluid and solid body, specifically refers to interface between solid and fluid.
    - Natural: fluid movement happens naturally by the means of gravitational forces, as the density change due to the temperature gradient, lighter fluid moves away from heat source.
    - Forced: fluid is forced to move by an external device
  - Radiation: heat transfer by electromagnetic radiation
- Work transfer - the system that is doing any work will lose some of its energy; work that has been done on the system will increase its internal energy.
- Mass flow - as mass is one of the primary carrier of energy in general, its entering or exiting influences the systems total energy.

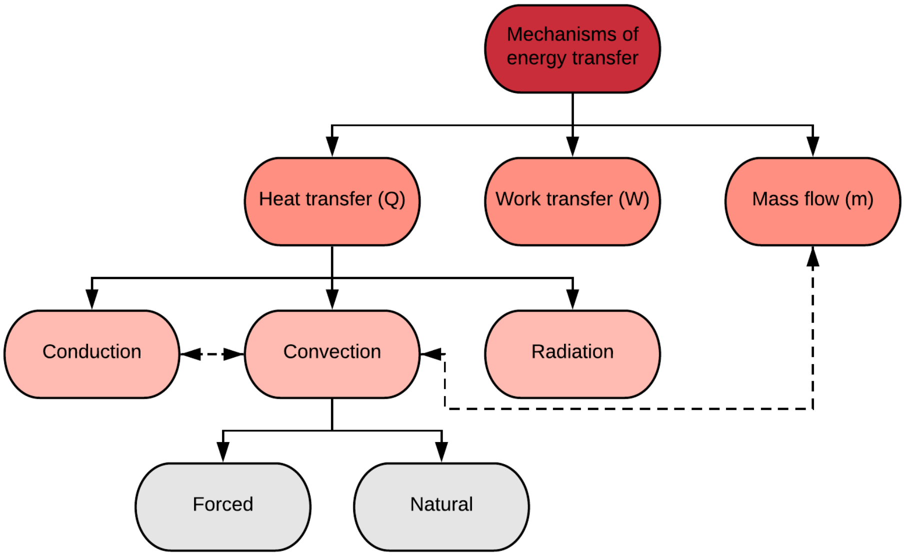


Figure A.16: Energy transfer in thermodynamical system classification

In a context of heat exchanger the mechanisms that one shall focus on are: heat transfer and mass flow. This refers directly to the definition of the device itself: *"heat exchangers are devices where two moving fluid streams exchange heat without mixing"*[26]. On the basis of its architecture, an exchanger in this paper is categorised as *shell and tube*[27]. There, *shell* is the part of the exhaust duct where the piping system - *tube* - is inserted. Interacting fluids in this heat exchanger are air in the pipes and exhaust gasses surrounding them, this makes the device an *air to air - type.* Since fluids' flows are directed perpendicular to each other with no redirecting, additional geometrical classification may be applied, making the device *one pass, cross-flow.*

After consulting specialists and analysing commercial heat-exchanger systems, one can assume that copper is a metal that is most commonly used for constructing heat exchanger pipes. Hence, for further analysis it has been chosen to use 8x0,8 mm copper pipe, as it is broadly available from the majority of hardware suppliers.

The first stage of the heat-exchanger modeling is to define the direction of the heat transfer. At the starting point, before media interaction, temperature of the exhaust gasses is much higher then temperature of copper pipe and air, which are being assumed in thermal equilibrium with much colder testing facility. Therefore, since all thermodynamical systems obey Kelvin-Planck statement, heat shall flow from the media with the highest energy to the lowest one. As each material has a thermal inertia (heat transfer does not happen immediately), shortly after interaction start, copper pipe will become hotter then air, defining heat direction as it can be seen on figure A.17.

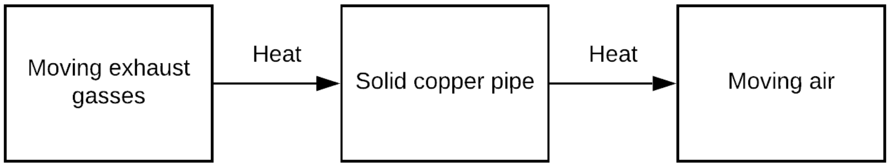


Figure A.17: Heat transfer direction

After heat transfer direction has been established one shall specify the mechanisms that govern the physical process. If one imagines an arbitrary cross-section of the heat-exchanger pipe (figure A.18) of a small length Δx and assumes that there is no heat gradient present in circumferential direction (temperature gradients exist only in x-direction)[28], than one can organise main mechanisms involved. Each moving fluid - boundary set corresponds to convection heat transfer: exhaust gas with outer surface of copper pipe and air with inner surface of copper pipe. Presence of a solid material, that separates areas of different temperatures gives a rise to conduction process.

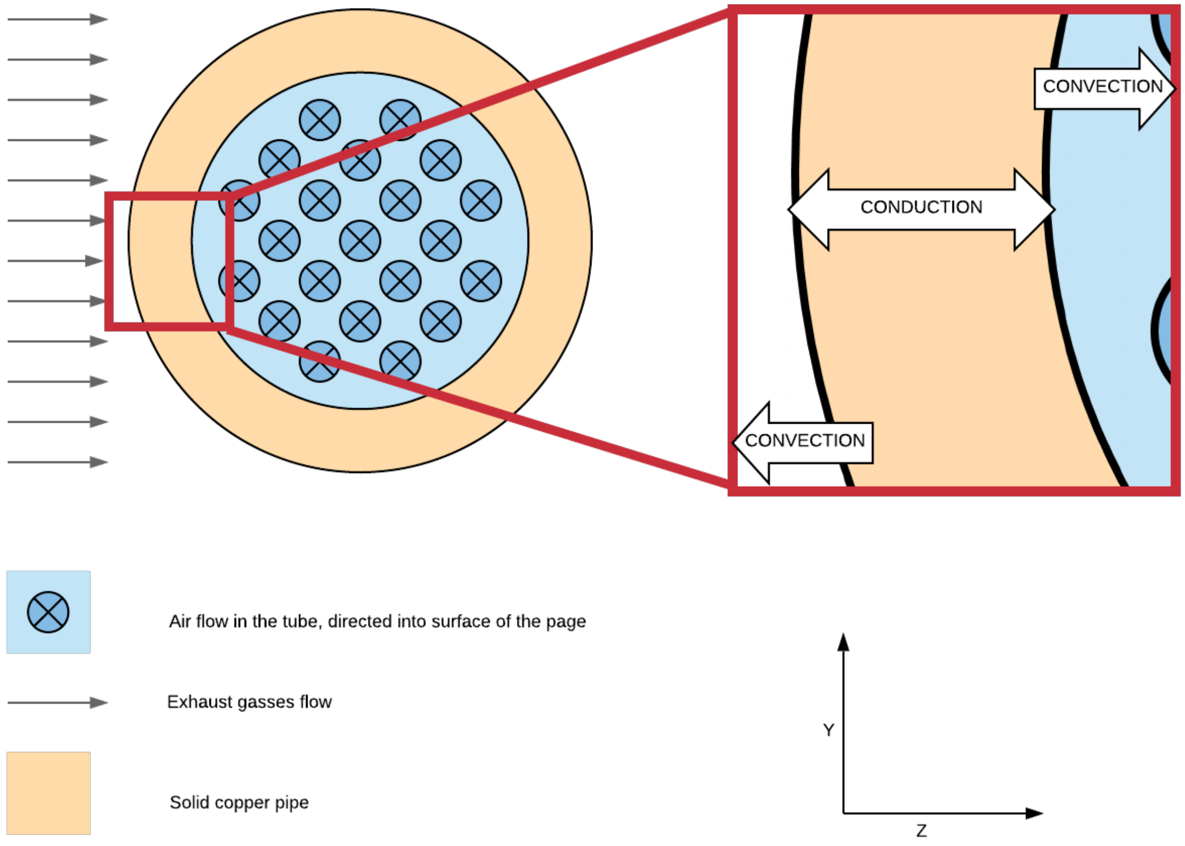


Figure A.18: Heat transfer modeling in the pipe cross-section

Since all thermodynamical processes also obey the energy conservation law, one can break-down each mechanism to specific ”building blocks” of heat transfer (see figure A.19). There, around each small Δx piece of pipe, moving hot exhaust gas’ mass carries thermal energy and transfers it via convection to solid copper. Thereon heat is conducted through the metal’s body. Temperature change in solid material is influenced by: convection rate with exhaust gas, heating or cooling from neighbouring Δx pieces of pipe and convection rate with internal air. The last one is determined by heat conduction in the air and it’s mass flow.

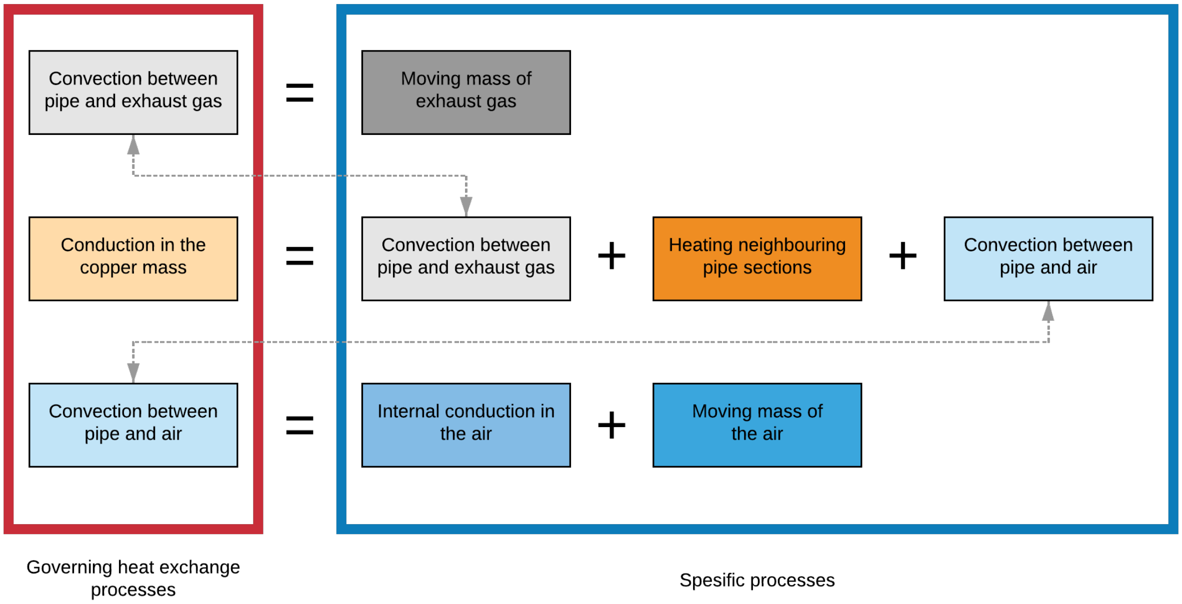


Figure A.19: Building blocks of thermodynamical processes based on energy conservation

Since all processes influence each other and exhaust gas’ mass flow is one of the governing parameters, one can say with certainty that temperature gradient of air in the copper pipe is a function of the speed of external gas flow. To distinguish the relation between exhaust flow-rate and temperature increase in the air, one can turn to either of three methods: experiment, mathematical analysis or CFD (see appendix A.2.6). It is important to notice that one must conduct at least two of those studies to give a high quality investigation. This way, the first study shall act as a starting point, providing prediction to the system’s behaviour. The second study shall than evaluate validity of behavioural model and give clarification of the aspects, that previous study failed to provide.

Research team made a decision to firstly conduct an experiment, because it would show an immediate quantifiable indication if varying of external fluid's speed (with all other parameters unchanged) would affect internal fluid's output temperature. Additionally one will also find out if this change is measurable by a commercial off-the-shelf instrument.

A.3.5 Concept analysis

In this section one will find documentation on experimental study and numerical simulation conducted to analyse validity of a concept proposed in appendix A.3.3. It has been of the main priority to find out if a change of external flow in heat exchanger would affect the internal flow temperature gradient, and if this difference would be measurable. At the same time one did not unconditionally assume that the external flow was the main cause of temperature fluctuations, if those would occur. There has been done a numerical simulation that studied heat exchange processes from the perspective of a heat resistance. This study showed what mechanism had the biggest affect on the air's exist temperature.

Experimental proof of concept setup

Conducting an experimental study to prove a concept validity has been one of the critical points in this research paper. It is known that for processes involving dynamical fluids, one often turns to physical experiments, since numerical solutions are usually challenging to perform (see appendix A.1.3). Additionally, depending on the problem in the study, sometimes it is impossible to achieve precise replication of all aspects of the real-life phenomena. Therefore, engineers downscale the environment and its parameters to make experiment cheaper and more manageable. As this paper focuses on gas-turbine exhaust gasses, research team found it impossible to build a full-scale test for experimental research. Therefore, has been decided to utilise resources that were available at that moment.

The main goal for this investigation was to build a cross-flow air-to-air heat exchanger that resembles an "exhaust duct and copper pipe" - system in miniature. Architecture of experimental set-up is illustrated in figure A.20. There one can see that hot external gasses were replaced by heated air supplied by a heat-gun. This device is equipped with controls for flow temperature and flow-speed [29]. However, since it was not calibrated by a trained specialist and there was no documentation on it's performance, it was decided not to use values on the control-panel in precision calculations and mathematical models building. Cold air supply into heat exchanger's copper pipe has been fed from an $8bar$ instrumental air supply, which is relatively close to full-scale system. Air velocity for both heat gun and cold air (see values on figure A.20) was measured by an unprofessional uncalibrated anemometer, founded by the research team itself [30]. Instrumental air velocity was used in numerical analysis as an input *range* of values, therefore keeping the analytical integrity of the mathematical model.

Laboratory equipment used in the study is listed in table A.6. Those devices were provided

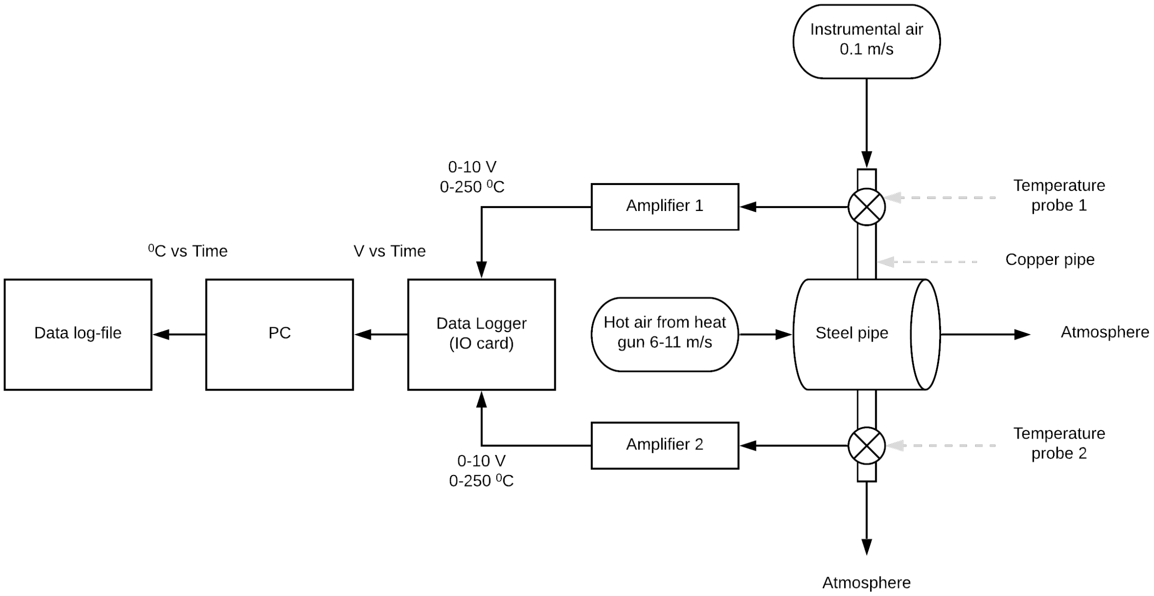


Figure A.20: Experiment set up architecture and data flow

Device	Type	Supplier	Specifications	Price	Quant.
Operational amplifier	PR4114	KROHNE [31]	Modification of operational parameters in system 4000	1398 NOK	2
Data log I/O device	NI USB-6210	NI [32]	Multi-function I/O device	12520 NOK	1
Temp. sensor	K-ThermoC	RS [33]	$-50°C - +400°C$	96,10 NOK	2
Power supply	24V	RS [34]	1A	241,02 NOK	2

Table A.6: Hardware utilised in the experiment and wired by external specialists at Kongsberg Automotive AS (further - KA). They also performed the experiment with one of the research team members available via web-camera solution. This measures had to be implemented due to the global pandemic situation, since physical presence at the laboratories has been restricted. KA engineers provided raw unconditioned data-log in a form of TDMS-file, which was further analysed by FFA research team and used as an input data for software testing.

Important aspects of mechanical construction, that were controlled with care have been: overall structural integrity - air tightness of the pipe-system and mounting of thermo-couples (figure A.21). Leakages were prevented by using gasket sealer and thermo-couples were positioned such that sensing part was exposed only to moving air without physical contact to copper pipe.

Increasing the flow intensity has been performed by physically changing position of the trigger-key on the heat-gun. In data-set this transition was marked by a pulse that was set there manually. This solution may caused significant impact of human factor, making further analysis possibly inaccurate.

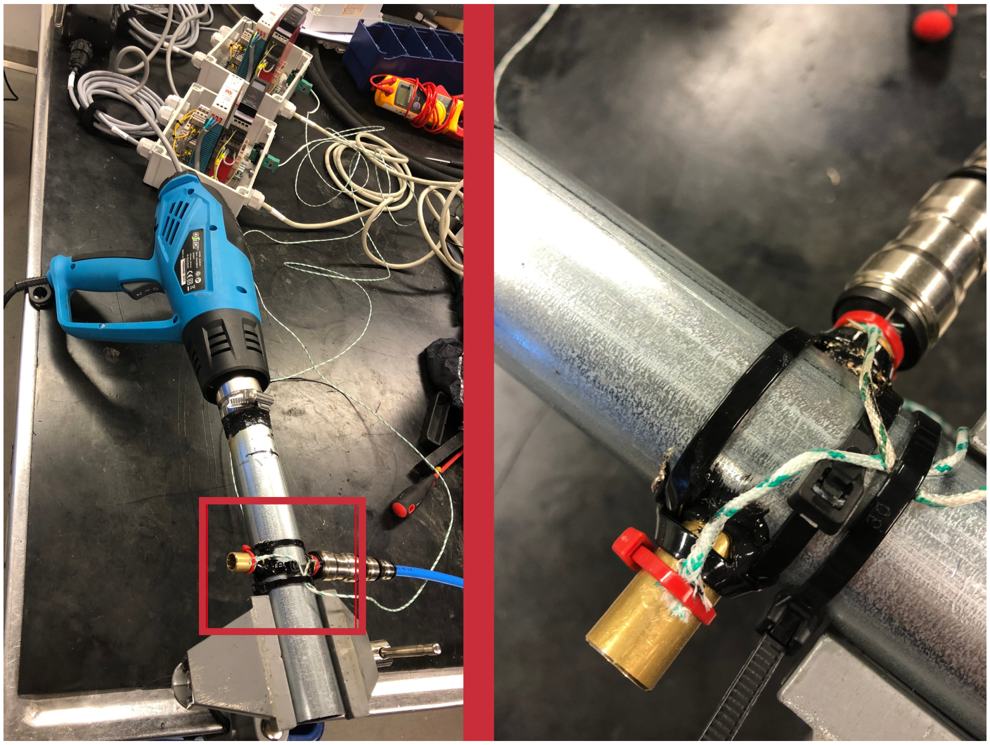


Figure A.21: Experimental set up

After all laboratory hardware has been rigged and wired, one may proceed to the experiment, that was performed as follows:

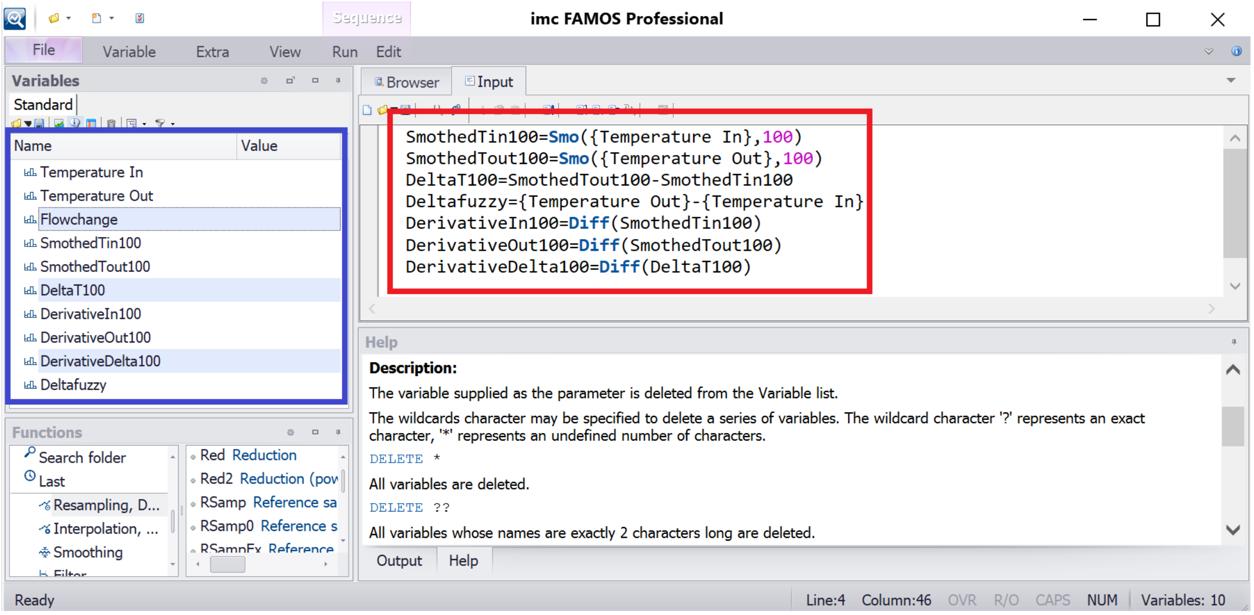
1. Resetting thermo-couples. Room temperature is assigned a value of $0°C$.
2. Start logging
3. Turn on cold instrumental air ($\approx 0.1m/s$) and heat gun at the lower speed setting ($\approx 6m/s$)
4. Record air temperature at the inlet and the outlet of the copper pipe
5. When outlet temperature stabilises, change the settings at the heat gun to the higher speed ($\approx 11m/s$) and mark this as a pulse in the log-system.
6. Wait until outlet temperature stabilises
7. Turn off the heat gun, mark it as a pulse
8. Log data until outlet temperature fall significantly
9. Stop logging
10. Export log-file

This procedure has been performed for two temperature settings on a heat-gun and two cold air velocities ($\approx 0.1m/s$ and $\approx 0.2m/s$), resulting in four sets of data, that software team could use for testing of their code.

One could rightfully describe the experiment above as unfulfilling and limiting. Possible adjustments then would be: introducing additional sensors for measuring in and out temperature of hot air, measuring flow speeds continuously during the experiment and building the system that would provide a precise temperature and flow control from a heat-gun. Unfortunately, those improvements were not possible to implement immediately due to time and resource constraint on KA side. Research team strongly suggest for possible followers to conduct similar experiment with all suggestions above applied.

Experimental data analysis

To condition and analyse experimental data, research team used specially designed software *imc FAMOS 7.5 Professional* [35]. This program was found immensely useful, since it could take as an input log-files directly, without any additional conditioning. It represents experimental data on two-dimensional plot with time on $x - axis$ and voltages-to-degrees on $y - axis$. Software was also capable of performing a variety of mathematical operations on the whole data set simultaneously. Those operations were called out in the from of direct commands in the input window (see framed in red on figure A.22). Among all functions used the most convenient was *Smo()*, which removes distortions and noise, that is common to record when conducting an experiment with electronic devices.



```
SmothedTin100=Smo({Temperature In},100)
SmothedTout100=Smo({Temperature Out},100)
DeltaT100=SmothedTout100-SmothedTin100
Deltafuzzy={Temperature Out}-{Temperature In}
DerivativeIn100=Diff(SmothedTin100)
DerivativeOut100=Diff(SmothedTout100)
DerivativeDelta100=Diff(DeltaT100)
```

Description:

The variable supplied as the parameter is deleted from the Variable list.

The wildcards character may be specified to delete a series of variables. The wildcard character '?' represents an exact character, '*' represents an undefined number of characters.

`DELETE *`

All variables are deleted.

`DELETE ??`

All variables whose names are exactly 2 characters long are deleted.

Figure A.22: imc FAMOS main environment

If data is plotted directly from a log-file, it is arguably difficult to see any trends in the process development, even if time markers were carefully placed in the positions of flow change (see figure A.23; mind that this is a close up view, for compelete plot please refer to F.26). Hence performing mathematical operations, like finding rate of change (first derivative), on any set would result in low quality visualisation.

First step for data analysis, after loading up respective files (see framed in blue on figure A.22), was removing noise from inlet and outlet temperature signals. Then after uploading new conditioned data in other variables (*SmothedTin100, SmothedTout100*), one calculates temperature difference between those two points in the copper pipe (*DeltaT100*), and finally computes the first derivative for temperature change. Differentiation of a function is a useful tool that can help to indicate major trends in function's behaviour. If derivative's value is equal to zero, there is no change in values at that point, if the derivative is a horizontal line, then function either increases or decreases linearly. If derivative plot has irregular shape, one has to address relative positioning of neighbouring points on the plot. If one point is higher then another, then rate of change in the second point was greater then in the first one.

Data analysed in this section is specifically for $\approx 100^{\circ}C, 6m/s - 11m/s$ hot air and $\approx 21^{\circ}C, 0.1m/s$ instrumental air. The main object of experimental study was to see if immediate change in external flow would result in changes in temperature difference in the internal flow. One also was interested into time period it would take to register this change if it would occur.

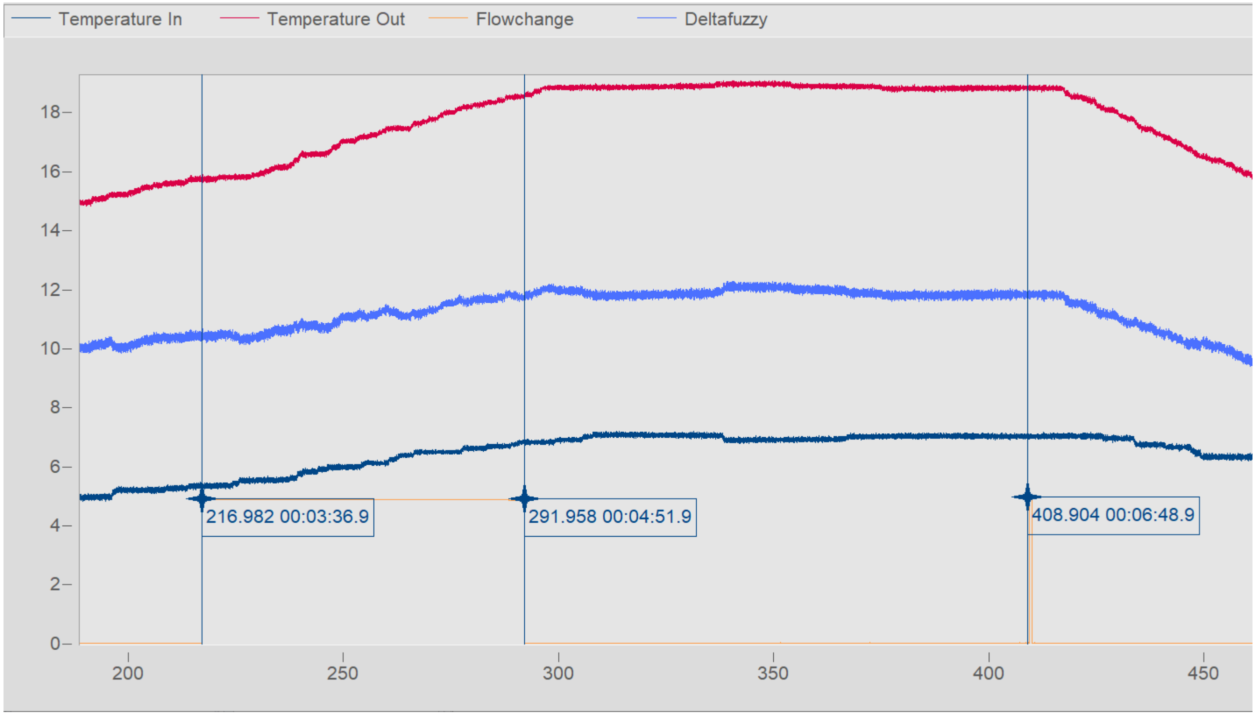


Figure A.23: Raw unconditioned data for inlet and outlet copper pipe temperature, and temperature difference plotted in imc FAMOS environment

On figure A.24 (close up view, for compelete plot please refer to F.27) one can see the illustration of temperature change with rate around the time, when external flow has been increased (03:37.0 - left marker). It is clear that at this point temperature was linearly increasing. Then, utilising graphing tools in the imc FAMOS one builds horizontal reference lines to allocate where increase-rate has changed. Mind that voltage markers on the plot are not scaled for derivative graph, since they refer to original vertical axis.

Referring to intersection of horizontal line indicating the increase-rate and its time-stamp (03:42.8 - right marker), one can see that it took $5,7sec$ for the system to react to the flow change. This to some extend proves that internal fluid is sensitive to external flow-rate. Additionally, the time that required the system to show a measurable change lays in the boundaries required by a customer ($\approx 5sec$). However, one cannot say with certainty that this change was influenced solely by air fluctuations from the heat-gun. To support those concerns one might point out that temperature was not brought to compelete equilibrium (derivative was not close to zero), but continued increasing. Even though test engineers tried to adjust flow change to a constant temperature, they clearly couldn't achieve it, mostly because those adjustments were not automated, but performed by a human. This means that it is impossible to separate thermal effects of the low-speed flow and high-speed flow on instrumental air, as those effects most certainly overlapped. Overlapping may be also caused by medias' thermal inertia, meaning that heat emitted from copper pipe could be belated. Originally the idea was to analyse four sets of data and build a correlation function between hot-flow speed and temperature change in instrumental air. But, since the experiment performed could neither provide a continuous data on hot air speed, or distinguish between speeds' thermal effects, it has been decided to perform an analytical study on a heat-exchanger. There the main goal would be to find out which heat transfer mechanism influenced temperature fluctuations the most.

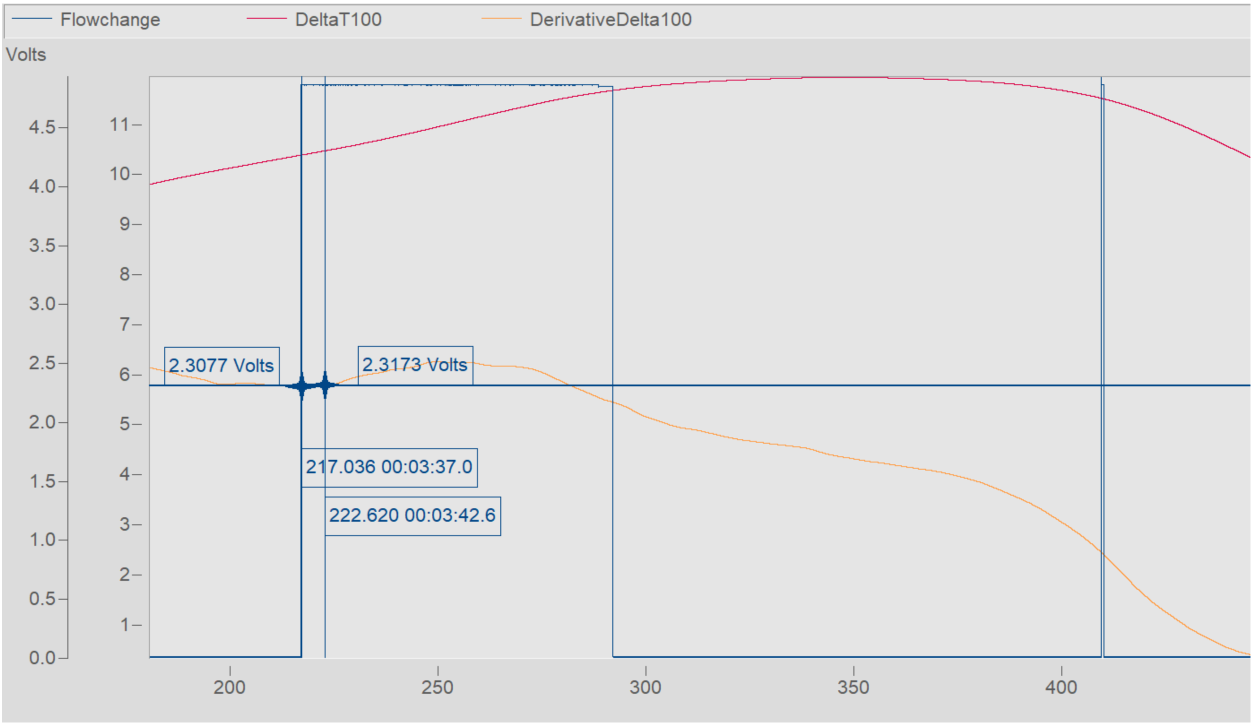


Figure A.24: Conditioned temperature increase visualisation with time stamps

Analytical and computer simulated analysis

Experiment performed by test engineers clearly provided a quantifiable indication to some thermodynamical process. But it is yet to be decided if this process was: external speed fluctuations, inert conduction in the cooper, or a mass-flow influenced heat transfer in instrumental air. Arguably, the best way to answer this question would be to describe the heat exchanger system with a set of equations, based on energy conservation illustrated in figure A.19. As it has been discussed previously, it was assumed that for an air-tube-air system there is no temperature gradient in the circumferential direction. This means that one could apply energy conservation law to a single small section Δx and sum it up both in space and time domain afterwards (see figure A.25). Space domain there refers to positioning of the Δx - piece on the pipe's length and time domain corresponds to exact moment it time since the heat interaction started.

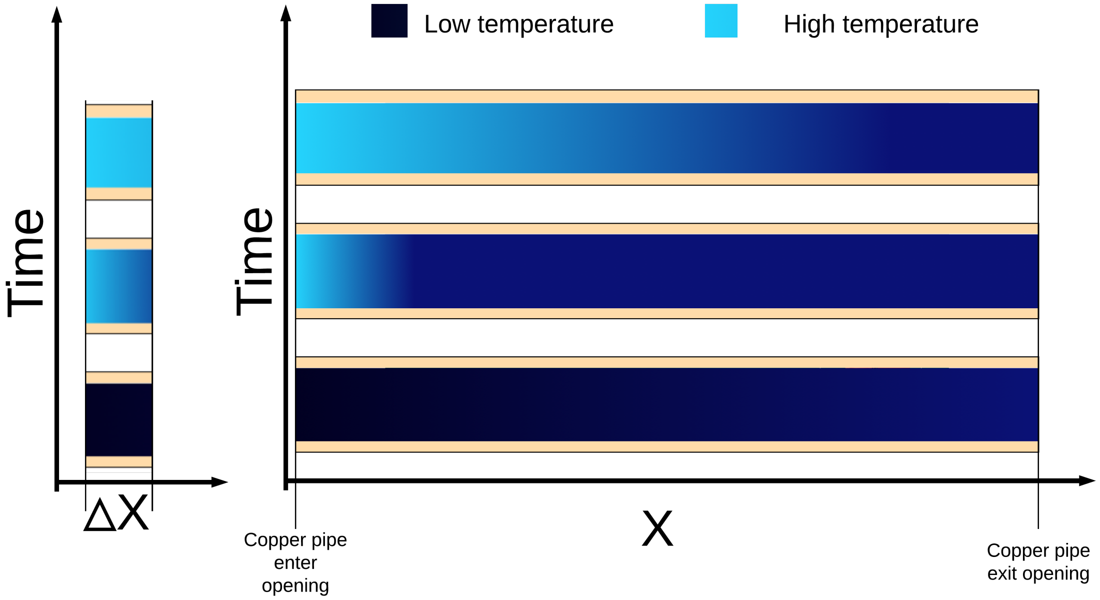


Figure A.25: Time-space visualisation for temperature distribution of air in the copper pipe

Each single block in figure A.19 can be replaced by a respective formula [27][28]. Mathematical expressions there depend on the mechanism of heat transfer and can be divided in four distinguished groups:

- **Heat transfer by moving mass** blocks can be replaced with $\dot{m}c\frac{\partial T(x,t)}{\partial x}$ for a fluid moving in x-direction and with $\dot{m}c\Delta T(x,t)$ otherwise. There $\dot{m}$ is a mass-flow rate, c is a specific heat of the fluid (can be found in engineering data sheets), $\frac{\partial T(x,t)}{\partial x}$ is a partial derivative of temperature distribution function in the fluid with respect to coordinate and $\Delta T(x,t)$ is temperature difference between boundaries in y-direction.
- **Conduction across a solid material** blocks are replaced by $\rho c A\frac{\partial T(x,t)}{\partial t}$. There ρ is the material's density, c is a specific heat of the fluid (both are from engineering data sheets), A is a cross-sectional area of a solid body in yz plane and $\frac{\partial T(x,t)}{\partial t}$ is a partial derivative of temperature distribution function in the solid body with respect to time.

- **Conduction from one solid to another (for neighbouring pieces of pipe)** is replaced by $kA\frac{\partial^2 T(x,t)}{\partial x^2}$, there k is a material's thermal conductivity (found in engineering data sheets), A is the area of contact between two neighbouring pieces and $\frac{\partial^2 T(x,t)}{\partial x^2}$ is a second partial derivative of temperature distribution function in the solid with respect to coordinate.

- **Convection heat transfer between a moving fluid and solid body** blocks are replaced by $hA|T_{solid}(x,t) - T_{fluid}(x,t)|$. There A is a contact area between solid and fluid, $|T_{solid}(x,t) - T_{fluid}(x,t)|$ is a temperature difference between pipe's wall and fluid's boundary layer, and h is a heat-transfer coefficient. It is a relatively complex parameter, which is calculated as $h = Nu\frac{k}{d}$, where k is fluid's thermal conductivity, d is either inner or outer diameter of a pipe and Nu (Nusselt number) is a dimensionless number, that represents a relation between conductive and convective heat transfer in fluid-solid system. Nu is usually difficult to obtain for irregular shapes and flow patterns, but for more common models, one can find in the literature [27], either as the number's averaged value, or as a mathematical formula that can estimate it for a specific process. Usually Nu is a function of two other dimensionless parameters: Re (see appendix A.1.3) and Pr (Prandtl number). Pr gives an indication on how heat is dissipated in the fluid. It relates thermal and molecular diffusion, and can be found in engineering spreadsheets.

After all blocks in energy conservation diagram have been replaced by respective formulas, one will get a system of three coupled elliptic partial differential equations (equations A.12-A.14). The subscripts are assigned as follows: o refers to outer hot air parameters, p is assigned to values associated with copper pipe, i is for inner instrumental air, in and out refer to either inner or outer geometrical measures of the pipe. There are also two superscripts in place: t stands for top and b for bottom. Those are connected to temperature of a hot air before and after it interacted with the copper pipe. One also notices that cross-sectional and contact areas has been replaced by a quantity of a lower order. It has been done by dividing each of those area-expressions by a common member - L. This also caused a slight variation from originally stated formula for energy by mass transfer for outer air: $\frac{\dot{m}_o}{L}c_o(T_o^t(x,t) - T_o^b(x,t))$. Since originally it did not have length of the pipe in the expression, it was placed in the denominator, making mass flow-rate a scaled value.

$$2\pi r_{out} h_o\left(\frac{T_o^t(x,t) + T_o^b(x,t)}{2} - T_p(x,t)\right) = \frac{\dot{m}_o}{L}c_o(T_o^t(x,t) - T_o^b(x,t)) \tag{A.12}$$

$$\rho_p c_p \pi (r_{out}^2 - r_{in}^2) \frac{\partial T_p(x,t)}{\partial t} = 2\pi r_{out} h_o \left(\frac{T_o^t(x,t) + T_o^b(x,t)}{2} - T_p(x,t) \right) + k_p \pi (r_{out}^2 - r_{in}^2) \frac{\partial^2 T(x,t)}{\partial x^2} - 2\pi r_{in} h_i (T_p(x,t) - T_i(x,t)) \tag{A.13}$$

$$2\pi r_{in} h_i (T_p(x,t) - T_i(x,t)) = \rho_i c_i \pi r_{in}^2 \frac{\partial T_i(x,t)}{\partial t} + \dot{m}_i c_i \frac{\partial T_i(x,t)}{\partial x} \tag{A.14}$$

As for any system of partial differential equations one would need to define a set of expressions for initial and boundary conditions. In this case those equations would be the initial temperature distributions and end temperatures for pipe - inner air system. Those conditions must be formulated and assigned with extreme care, since they will influence the outcome significantly. One way of obtain those functions is to conduct an experimental study, and construct them from real-life data. Unfortunately, this was not possible to obtain neither for down-scaled venison or for real-life exhaust duct. However, research team advises possible followers to solve those equations after obtaining precise boundary and initial conditions functions. As a result of this study one will get a two-dimensional functions for temperature distributions in pipe $T_p(x,t)$, inner air $T_i(x,t)$ and existing temperature for hot air $T_o^b(x,t)$. Team suggests to utilise a finite difference method [28] to compute those time-dependant functions numerically. If one would like to study procedure of constructing the code with detailed theoretical background on the method, please refer to *”Numerical Methods for Partial Differential Equations”*[36].

One could say that energy conservation - based solutions would give a great overview on a complex dynamic process and can be considered superior to other methods of analytical analysis, because level of precision for those techniques is limited only by accuracy of initial conditions, stability of a numerical algorithm and computational power available. However, there possibly is a simpler method to determine dominating heat-exchange mechanism. It bases on inspection of thermal conductivity and heat transfer coefficients of thermodynamical processes. Here it will be useful to introduce two new engineering concepts, that will be of assistance in further analysis.

Thermal resistance is a thermodynamic property of an object (solid body or fluid), that quantifies it’s opposition to the heat flowing through it’s volume. It depends on geometrical, mechanical and thermal parameters of that object. One can find respective values for thermal resistance in engineering data-sheets or can compute it utilising mathematical formulas, that are different for distinct heat transfer mechanisms. For convection processes please refer to equation A.15 and for conduction in a thin-walled pipe to A.16 [27].

$$R_{conv} = \frac{1}{hA} \tag{A.15}$$

$$R_{cond} = \frac{ln(r_{out}/r_{in})}{2\pi kL} \tag{A.16}$$

To grasp a practical meaning of a new property, one can refer to an analogy with electricity. There temperature is imagined as an electrical potential or voltage, and therefore thermal resistances correspond to ohmic resistances. Illustration of this analogy applied to real-life exhaust-to-air system can be seen on figure A.26. There heat flows from a point with higher potential T_{in} (incoming hot gas) through series of resistances to a point with the lowest potential T_{out} (exiting warmed up air in the pipe). A process that would cause a biggest "potential drop" across respective resistance is the dominating one, hence it would be responsible for bigger total changes in total temperature. Ideally, T_{out} will be utilised as an indication of velocity fluctuations in exhaust gas, meaning that the greatest temperature drop would happen at the exhaust-pipe junction.

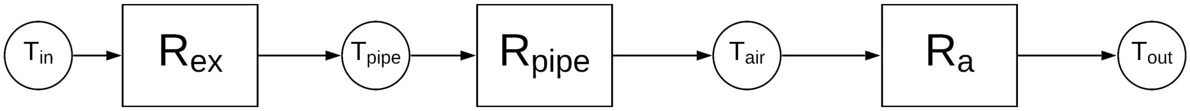


Figure A.26: Thermal resistance chain for exhaust gas-to-air heat exchanger system

A second concept that will be used to compare heat transfer mechanisms is **overall heat-transfer coefficient** (equation A.17). It is used for calculating energy flow-rate through the whole system. It also can be utilised as a value that one compares other thermal parameters to (like thermal conductivity k or heat transfer coefficient h). If one of the k or h values is close to U, then respective process can be considered dominant.

$$U_{overall} = \frac{1}{A\sum_{j=1}^{N} R_j} \tag{A.17}$$

To analyse and compare thermal resistances one can employ mathematical simulation written in Python. It is a programming language with a relatively simple syntax, that is often used in scientific calculations. Code listed down below performs a simple simulation, that provides relative visualisation of parameters of interest. Since R for convection process is dependant on fluid's velocity it has been decided to evaluate $R_{exhaustgas}$ on a range of speeds. Minimum and maximum values there ($\approx 1m/s{-}169m/s$) were extracted from CFD reports provided by a customer. Since there is no information on initial temperature distribution in the copper pipe or instrumental air, all thermodynamic parameters, including Prandtl number [37], were found in respective engineering data sheets [18][19] with respect to mean temperature between exhaust gas ($550°C$) and estimated wall - room temperature ($21°C$).

Code A.1: Python based numerical simulation for analysing thermal resistance of processes involved in air-to-air heat exchanger

```python
import numpy as np #library for mathematical operations
import matplotlib.pyplot as plt #library for plotting graphs

VoMin = 1.0 #minimum exhaust flow speed
VoMax = 169.0 #maximum exhaust flow speed

N=100 #number of sampling points
ROavg=0.0 #average thermal resistance of exhaust gas

RoO=0.5 #average density of exhaust gas
KinVisc=0.00007772 #average viscosity of exhaust gas
kO=0.5 #average thermal conductivity of exhaust gas
PrO=0.717 #Prandtl number exhaust gas

RoI=0.7 #average density of air
KinViscI=0.000035 #average viscosity of air
kI=0.3 #average thermal conductivity of air
PrI=0.698 #Prandtl number air
VI=0.1 #Feeding velocity of air

kTube=375.0 #Thermal conductivity of copper

dO=0.008 #Copper tube outer diameter
dI=dO-0.0008 #Copper tube outer diameter
L=1.0 #Length of copper tube
AO=np.pi*dO*L #Outer contact area
AI=np.pi*dI*L #Inner contact area

#Arrays for storing data
Vspace = np.linspace(VoMin, VoMax, N)
```

```
ROspace = np.zeros((N), dtype=float)
ROavgspace = np.zeros((N), dtype=float)
RTubespace = np.zeros((N), dtype=float)
RIspace = np.zeros((N), dtype=float)
UTotspace = np.zeros((N), dtype=float)

#Thermal resistance of copper tube
RTube=(np.log(dO/dI))/(2*np.pi*kTube*L)
ReI=(VI*dI)/KinViscI #Average thermal resistance of air
NuI=0.023*(ReI**0.8)*(PrI**0.4) #Nusselt number air
hI=NuI*(kI/dI) #Convection heat-transfer coefficient for air
RI=1/(hI*AI) #Average thermal resistance of air

#loop for exhaust gas thermal resistance calculation
for i in range(0,N,1):
    Re=(Vspace[i]*dO)/KinVisc
    Nu=(0.35+0.56*(Re**0.52))*(PrO**0.3)
    hO=Nu*(kO/dO)
    ROspace[i]=1/(hO*AO)
    ROavg=ROavg+ROspace[i]
    RTubespace[i]=RTube
    RIspace[i]=RI

ROavg=ROavg/N #average thermal resistance for exhaust gas

for i in range(0,N,1):
    UTotspace[i]=1/(np.pi*dI*L*(ROavgspace[i]+RTubespace[i]+RIspace[i]))
    ROavgspace[i]=ROavg

print("Average␣RO",ROavgspace[N-1])
print("Rtube",RTubespace[N-1])
print("Rinner",RIspace[N-1])

#overall heat transfer coeffisient
Overall=1/(np.pi*dI*L*(ROavgspace[N-1]+RTubespace[N-1]+RIspace[N-1]))
print("Overall",Overall)
print("hi",hI)

#graphing commands
plt.figure(1)
```

```
ax=plt.subplot(211)
ROpl=plt.plot(Vspace,ROspace,c='red',label='RO')
UTotpl=plt.plot(Vspace,UTotspace,linestyle=':',c='red',label='Utot')
RTubepl=plt.plot(Vspace,RTubespace,c='blue',label='RTube')
RIl=plt.plot(Vspace,RIspace,c='green',label='RI')
ax.legend()

ay=plt.subplot(212)
ROpllog=plt.plot(Vspace,ROspace,c='red',label='RO_log')
ay.set_xscale('log')
ay.set_yscale('log')
ay.legend()
plt.show()
```

The first result obtained in the simulation addressed the concern regarding thermal inertia in the copper pipe. When plotting $R_{exhaustgas}$ and R_{pipe} as a function of speed one can clearly see (figure A.27 - top) that convection process from exhaust gas is superior to conducting through solid copper. Thermal resistance of the pipe is lower on the whole range of gas' velocities. This means, firstly, that in the context of current system, copper is a good heat transferring media. It works just as a thin membrane between two interacting fluids and does not cause great disturbances in heat exchange.

Another important finding provided by the simulation can be seen in the bottom plot on figure A.27: thermal resistance of external convection process decreases with exponential rate. This means that the greater is exhaust flow, the lesser will become it's contribution to potential drop in the temperature. This can give a background for distinguishing precision of thermo-couples required to pick up the slightest temperature changes occurring at higher velocities.

The third finding produced by the simulation was a significant dominance of convection process between copper pipe and instrumental air (green graph in figure A.28). This immense influence is supported by comparing overall (averaged) and convection heat transfer coefficients (equation A.18).

$$U_{overall} = \frac{1}{A\sum_{j=1}^{3} R_j} = 9,295W/m^{2\circ}C \approx h_i = 9,325W/m^{2\circ}C \tag{A.18}$$

This indicates that overall heat transfer is controlled by internal convection process. Meaning that exhaust gas - copper pipe system could be theoretically replaced by a constant heat flux of $550°C$, removing effect of external speed fluctuations completely. In practice, this

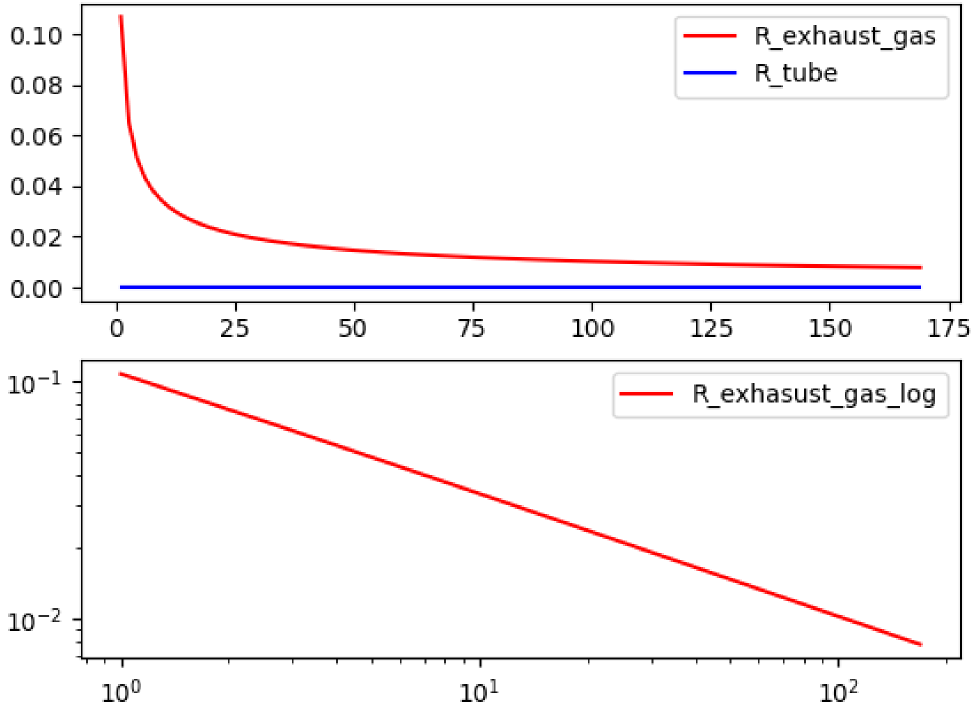


Figure A.27: Top: thermal resistance of heat transfer in exhaust gas - pipe system vs thermal resistance of conduction through copper pipe. Bottom: log-log plot of thermal resistance of exhaust gas - pipe system

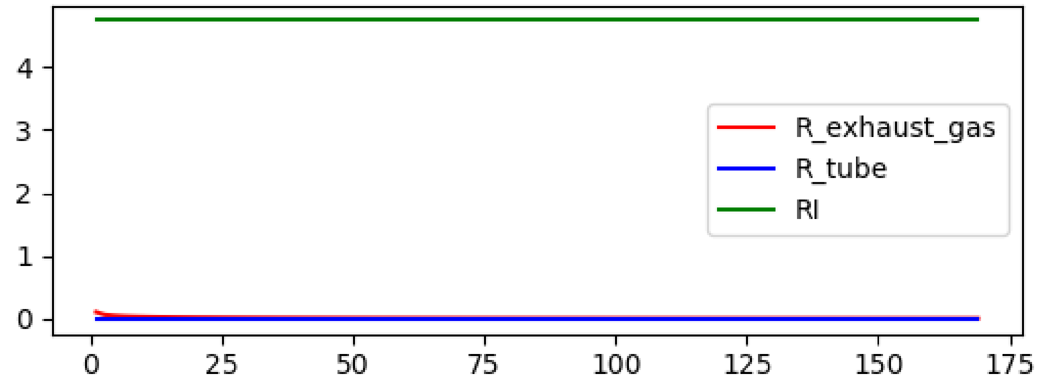


Figure A.28: Thermal resistance of heat transfer in exhaust gas - pipe system vs thermal resistance of conduction through copper pipe vs thermal resistance of heat transfer in pipe - air system at real instrumental air velocity.

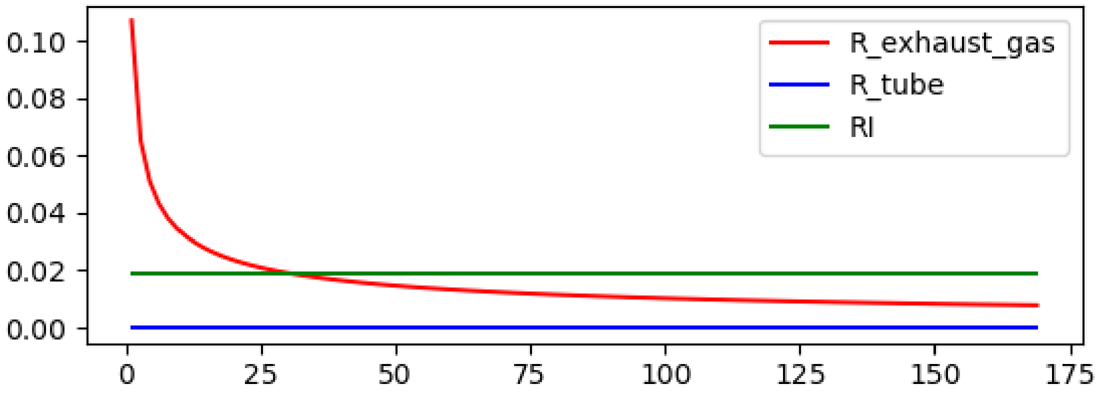


Figure A.29: Thermal resistance of heat transfer in exhaust gas - pipe system vs thermal resistance of conduction through copper pipe vs thermal resistance of heat transfer in pipe - air system at altered instrumental air velocity.

will imply that at higher exhaust's velocity, air temperature at the pipe's exit will not be influenced by external velocity fluctuations.

However, one could alter descriptive parameters of internal air such that flow regime would be not as contrasting with respect to external hot gas. This would result in significant improvement of thermal resistance distribution. As an example, simulation refers to the model, where instrumental air's velocity was increased from $0.1m/s$ to $100m/s$ (see figure A.29). Among other parameters that could be adjusted are: initial temperature distribution or an internal fluid itself.

This finding also explains the reasoning behind that test engineers managed to record the influence of external flow fluctuations through instrumental air exit temperature assessment. In the experiment set up the difference between flow states was not as substantial as for real-life system. There, the thermal resistance distribution most likely would be similar to the leftmost part of figure A.29, where external flow is dominant due to lower velocities.

A.3.6 Results' evaluation

Appendices A.3.4 and A.3.5 are referring to a scientific stage of measuring system development (for the whole development procedure, please, refer to appendix A.3.3). The main goal here was to prove that heat conduction rate, and associated with its temperature fluctuations, can be used as an indicator of external mass-flow behavior in heat-exchanger system. In other words, the research focused on finding if fluctuations of hot gas' velocities influenced exiting temperature of cold inner air. Meaning, that by measuring this temperature one could make a judgement on external flow's behaviour.

The analysis has been conducted via experiment and analytical simulation. Experiment has shown some positive correlation between the parameters of interest, proving, to some extent, that the greater outer velocity results in greater temperature increase. However, since it was difficult to distinguish the dominating processes in the experiment, it has been decided to conduct an experimental study to bring more clarity on the topic. This decision has been supported by previous research in fluid mechanics (see appendix A.1.3), where one can learn that most of the parameters describing moving fluid are connected, or *tightly coupled*.

The analytical simulation has shown that heat transfer rate indeed is dependant on flows' velocities. Mass-flow rate influences dimensionless numbers Re and Nu, that play a key-role in predicting fluids' behavior. However, it has been discovered, that increase in fluid's velocity causes respective decrease in output temperature sensitivity to flow fluctuations. This does not necessary bring the concept of heat-exchanger as measuring system to rest, but gives a ground for further, more specific, research. Team suggests to study the system's behavior with different working fluids at different feeding speeds.

Another knowledge value added by the simulation is the general understanding of a WHRU installed in the exhaust duct. Since WHRU is a type of a heat exchanger, research team proposes to engage in a closer assessment of this device's performance. Based on CFD reports provided by the customer, one can assume even distribution of pressure and temperature in the cross section of interest. Therefore one has to distinguish another parameter that could be causing uneven performance and failure of WHRU. FFA does not affirm, but suggests, that it is of a first priority to obtain a fulfilling documentation on thermodynamical performance of the device that is already exposed to exhaust duct. This way one can narrow down the parameters that can possibly cause decrease in efficiency or failure. Research team cannot state with absolute certainty that velocity distribution is irrelevant factor, since it has been discovered that for the flow ranging from $1m/s$ to $\approx 10m/s$ (see figure A.28), hot-flow does affect an overall heat transfer coefficient, that influences general heat transfer rate. Never the less, since the study has been conducted for a very specific type of heat-exchanger, one cannot apply the same knowledge to a real-life device, meaning that thermal resistance curves there could be dramatically different from the ones obtained by the research team. Therefore, one can not claim, that low speeds would have the same effect on real-life WHRU.

A.4 Miscellaneous

This appendix covers graphical modeling performed by mechanical responsible in *SolidWorks 2018*. It is important to notice that models' architecture and features were not designed for blue-prints production, but for either exporting in binary format for *Unity* applications, or for CFD analysis.

A.4.1 3D modeling for Unity

Since it has been decided to utilise gaming engine *Unity* for visualisation and presentation of the system, one had to generate the geometry for environment in the game. Figure A.30 shows the last version of measuring system.

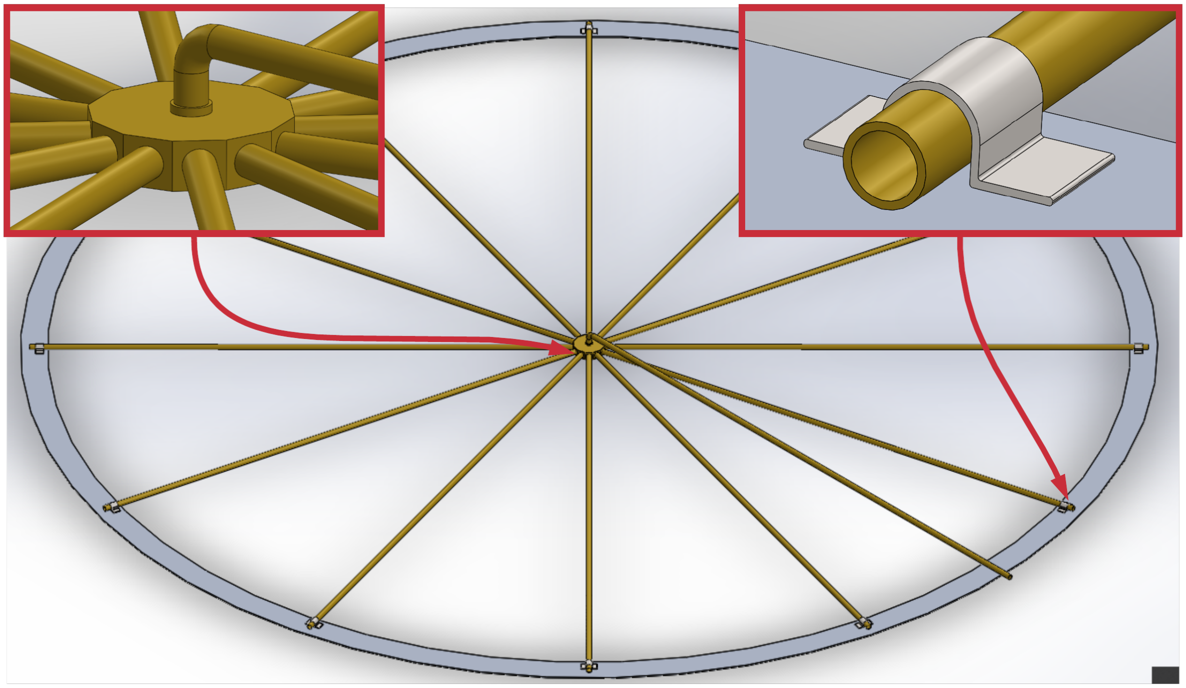


Figure A.30: Measuring system visualisation produced in SolidWorks. Close up on copper junction and fastening parts.

A simplified geometry of the exhaust duct with the feature-tree can be seen on figure A.31. Notice that all geometry so far has been produced up to real-life scale.

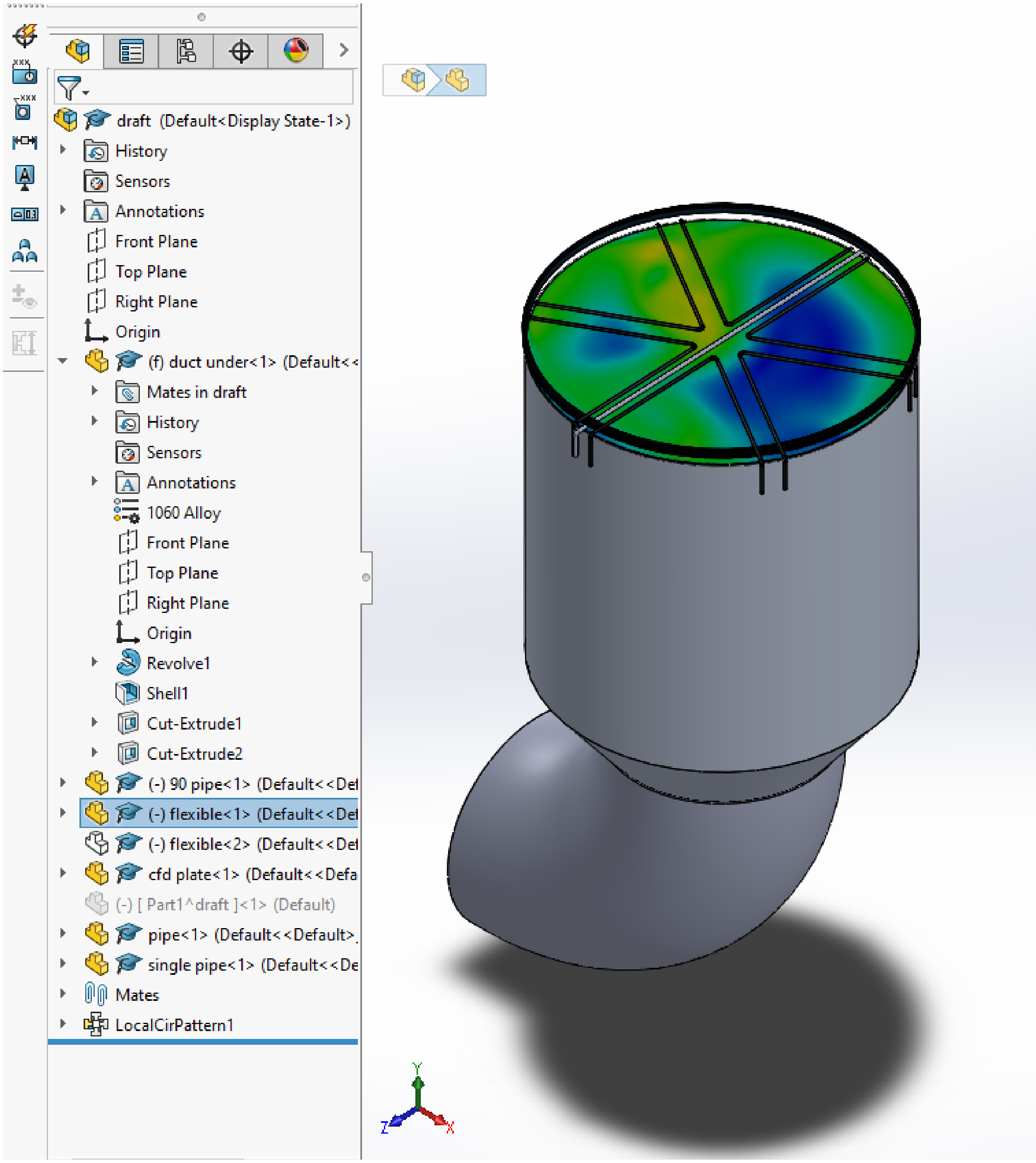


Figure A.31: Preliminary design assembly with feature tree

A.4.2 CFD analysis attempt

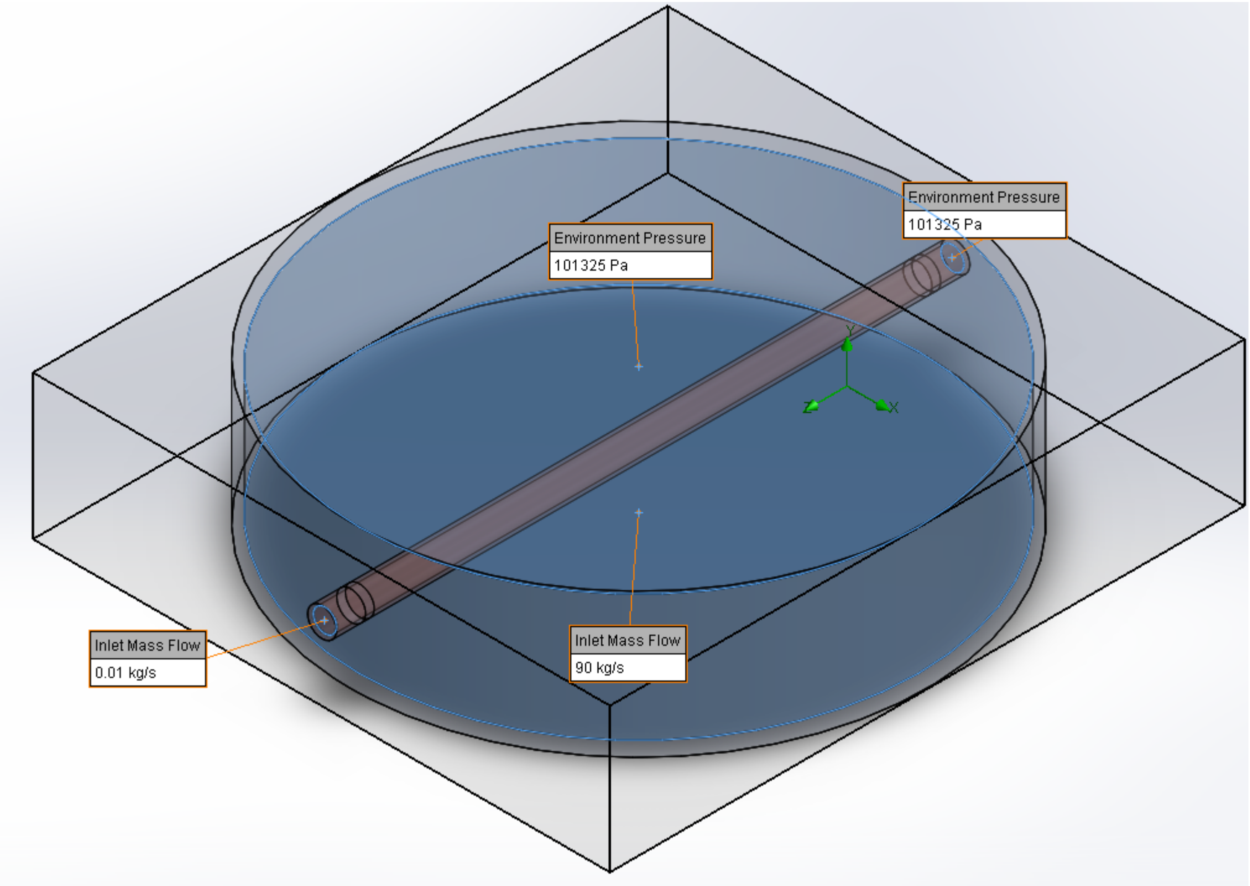
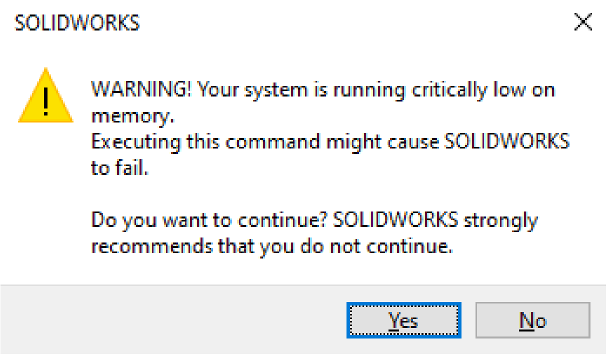


Figure A.32: SolidWorks Flow Simulation. Measuring volume and assigned boundary conditions for prove of concept study

As a part of a prove of concept workflow, there has been made an attempt to perform a CFD analysis of cross-flow air-to-air heat exchanger. Generally, one followed the procedure summarised in appendix A.1.4. On figure A.32 one can see a simplified geometry in measuring volume, with applied boundary conditions onto two separated fluids' domains. Mesh for this study has been auto-generated, with no additional refinement, to not force any unwanted load on the processor. For computation, the standard solver has been utilised; one however applied features as: fully developed high-speed turbulent flow, conduction through solid material (copper pipe - colour coded as brown on the figure of reference) and adiabatic outer-wall. Even though PC used for running the simulation managed to successfully mesh the measuring volume and solid bodies, it has struggled to run the simulation itself. The operator experienced a set of warnings (see figure A.33) before forcefully running a CFD analysis. This resulted in either unrealistic solutions that would not converge, or software failure.

Since at the time research team was working on prove of concept, there was no other computational resources available, FFA decided to turn to other methods of analysis. However, working on with *SolidWorks Flow Simulation* had helped mechanical responsible to better understand the CFD reports, provided by the customer. It also gave a better overview on original problem domain, since research team's main goal was to validate a computer generated model of exhaust gas' flow.



WARNING! Your system is running critically low on memory.
Executing this command might cause SOLIDWORKS to fail.

Do you want to continue? SOLIDWORKS strongly recommends that you do not continue.

Figure A.33: SolidWorks warning showing lack of computational power

B. Appendix - software

B.1 Software design document

B.1.1 Goals

- The software needs to write input data from sensors to file (likely a CSV file) for readability.
- The data needs to be visualized in form of graphs in real time, figures representing actual values.
- The software should be easy to use for the user.
- Upon test completion the application will automatically create a readable report.

B.1.2 Milestones

This will be continuously updated throughout the project.
Elaboration 1

1. Use case diagram.
2. Sequence diagram.
3. Class diagram.
4. functions.
5. GUI models.

B.1.3 Research: Software and programming language

Elaboration 2

1. C++ and visual studio or C++ and QT.
2. C# and unity.
3. Spyder and python.

B.1.4 Outcome of research

Construction 1

1. Reason for picking software and programming language.
2. Use Doxygen for software documentation.
3. Elaborate on UML documentation.
4. Elaborate visualization with models.
5. Resolution (Resulution on pipes)

B.1.5 Create demo software with minor functionality

Construction 2

1. Read from file for testing.
2. Visualize GUI.
3. Be able to go back and forth between screens.
4. Implement 2D visualization.
5. Implement storing of data.
6. Library list.

B.1.6 Implementing functions and testing

Construction 3

1. improve GUI
2. Implement graphs.
3. Implement 3D visualization.
4. (Communication between hardware and software. Implement math if needed)

B.1.7 Functional Description

This application will be able to gather data from a set of sensors and further visualise this data in form of graphs and models. At startup the user will input various input data such as: Test name, time, date, username, if you want to input the GT loads manually or not, if you are offshore, if you want to store all data and how often you would like to store data. When all these inputs have been decided you can start the test and the application will send you to the second screen. The second screen will display the sensor values numerically as well as in a graph. There are also a function which allows you to store data for a specific amount of time, this may be useful depending on your input on the initial screen, this function may be hidden depending on input. From here you can navigate into the 2D or 3D visualisation. The 2D and 3D visualisation screens only function is to visualise, they also have a back button that will bring you to the second screen. All three screens that are running during test also has a Stop Test function, this is in case something unexpected happens and you have to shut it down.

As the input phase of the test has human interaction it is prone to get errors. The input windows will require you to type your inputs in correct format, if this is not upheld you will get error messages and the test will not be able to start. We are prone to encounter more errors with this software, therefore we hope to encounter as many errors as necessary during testing.

B.1.8 User Interface

Initial screen

Initially we discussed the various functions our software will require, and what kind of input it needs. We then started designing the first iteration of our software. Underneath you can see the initial screen of the software B.1. This contains a set of inserts that are required:

- Name of test.
- Time and date.
- Test Person.

It also contains some check-boxes, and some input fields may appear if a box is checked, these are illustrated with grey. If you check yes for manual loads you will get new input fields for load amount and how long the load will be running.

- Manual load.
- Offshore.
- Store all data.
- Sample rate.

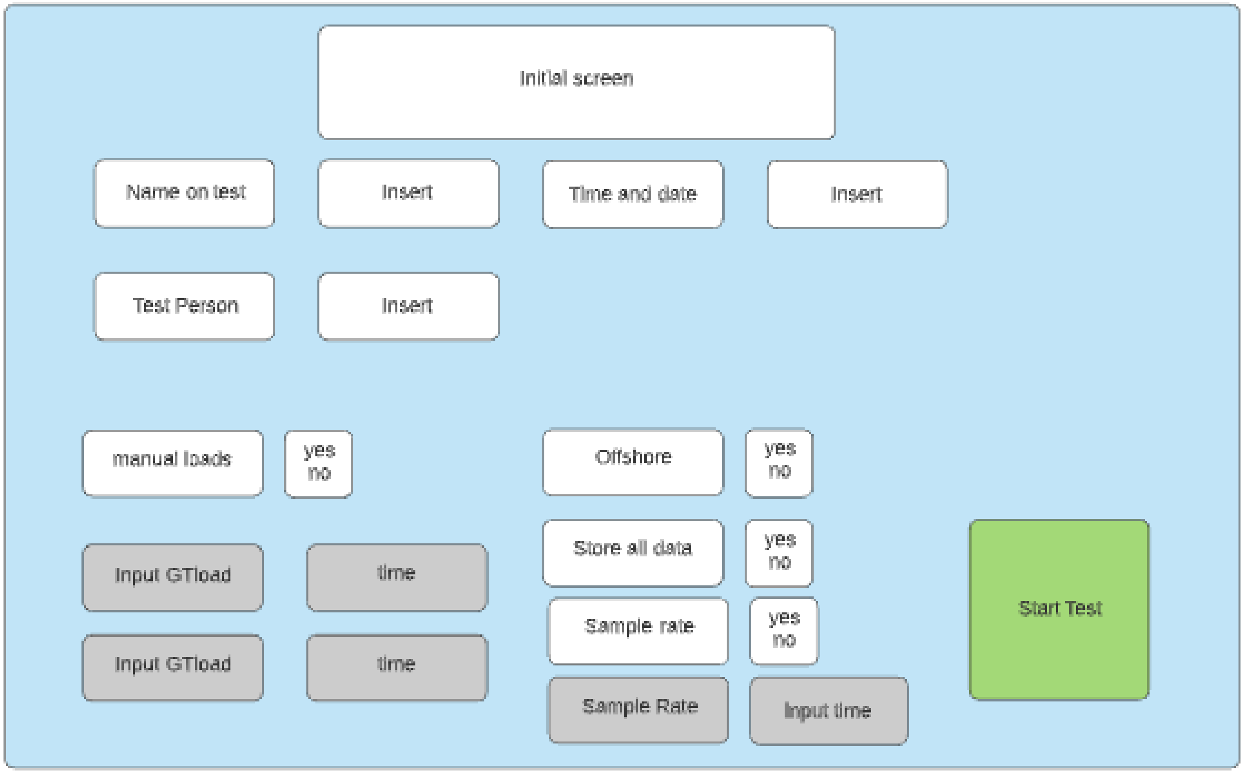


Figure B.1: Initial interface made in lucidchart

The second screen
Will be the default "Running test" screen B.2, here we can see each sensors value displayed individually as well as in a graph. You can also see a button "Start storing data" with a input field besides where you input how long you want to store. There are also a button that will bring you to the 2D Visualisation B.4 and the 3D Visualisation B.3 as well as a "Stop Test" button.

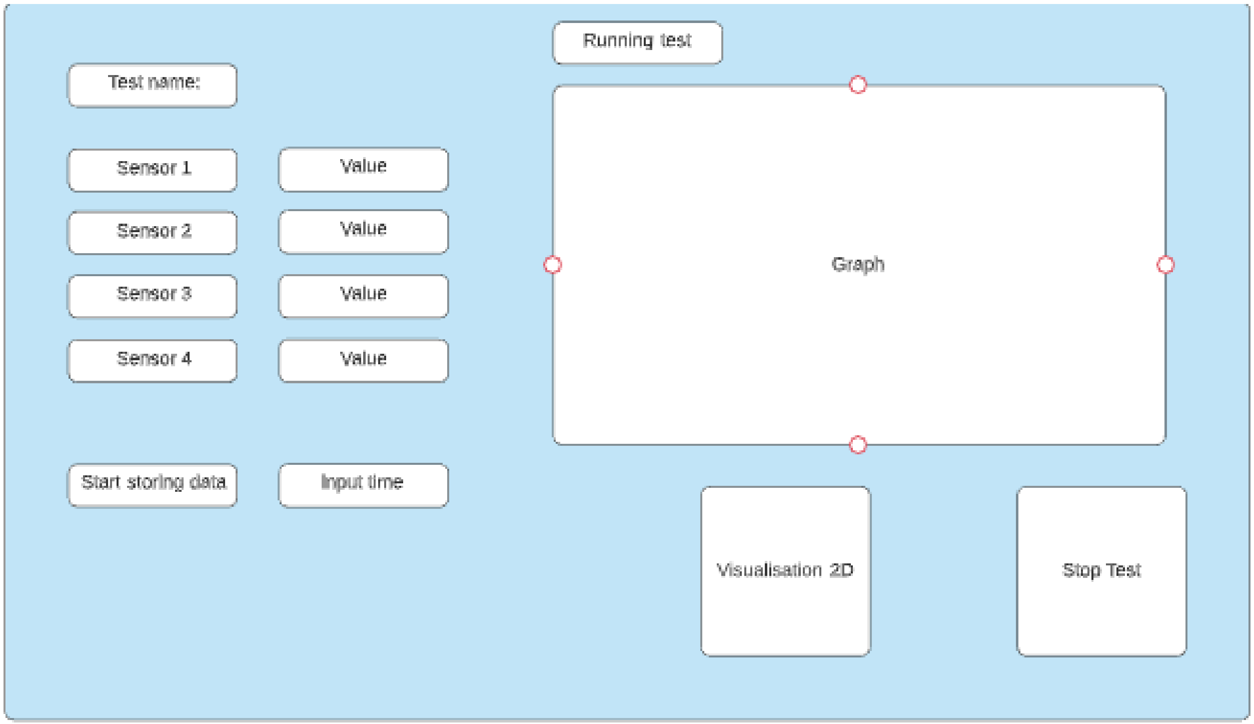


Figure B.2: Running test interface made in lucidchart

The third/fourth screen
Depending on you pressing the 3D or 2D Visualisation button you will enter a visualisation. For now the visualisation is undecided. You will also have a "Back" button to bring you to the "Running Test" screen and a "Stop Test" button.

Figure B.3: Interface for visualisation 3D made in lucidchart

Figure B.4: Interface for visualisation 2D made in lucidchart

B.2 Research

B.2.1 Python

Elaboration 1
The reasoning for choosing Python boiled down it being highly recommended from engineers we know that work with software. As well as it being very dynamic with a lot of supporting libraries and other extensions. It is also popular programming language, ranked as number three according to [38]. It also focuses on a simpler or more minimalist syntax, making it easier to progress in. The process started with downloading Spyder (Python 3.7) from Anaconda, Spyder being the IDE and Python the language. Looking up tutorials and information on useful libraries.

Elaboration 2
One of the first that was read more about was Pandas, an open source data analysis/manipulation tool [39]. After spending some time on Pandas, reading and writing to file was manageable. With this we made a simple script that would make a CSV file and write in

"random" values at a set interval, emulating sensor data. Then another one that would read these values and plot these in a graph as new values were created. After this was done, the focus was on creating a 3D visualization of the data. What we had in mind was a CFD, but in 3D. As this was researched, we found an online class provided by [40]. Here I went through the twelve steps, however the underlying math and physics was too complex to learn or understand within our time frame. The math and physics used also differed somewhat from what we were going to have use. So after discussing with mechanical responsible, we decided that it would be more efficient they focus on theoretical preliminary research and supply us with generated numerical algorithms along side with their practical explanations. Another solution will be to simply keep it as a black box when making the software. However after going through all of the steps and the videos for each. One of the figures that was made figureF.30 was of interest, even though this is for flow were there is a cavity (hole), and it being 2D. This is highly resembles what we envision for our visualization both in 2D and 3D. So while this will not be used, it is added to show the possibility with Python. After this conclusion I started working on the UI, with the library Tkinter [41]. When working on the UI the main focus was to learn how to use the Tkinter library. It was not about adding a lot of functionality as this would have been unnecessarily time consuming. It was also important to verify that it had what was needed to create the four UI screens we envisioned (Appendix B.1.3). In the figure F.31 the big blue canvas in the middle is supposed to contain either the graph seen in figure F.28 or another visual representation. *For more information on the scripts, see attachment four.*

B.2.2 C++ and Qt

We considered a few possibilities on how to create our final software project. Where we wanted to do some research in each of them. With the language C++ we have not worked on making any GUI. Therefor the main focus in C++ research was how to make a GUI. In the beginning of the research we used visual studio and found that we could use visual studio "forms". This made it possible to drag and drop buttons and text-boxes to a GUI. In early research this looked promising but we found it hard to connect simple functions to buttons or text-boxes. While we tried to figure out how visual studio forms worked, we started to notice another software called "Qt creator". This software used C++ as language and had a more user friendly way to create GUI software. So we started to try and make simple GUI software in Qt. Qt has integrated many different tools to help software developers, that visual studio where missing. While continuing with Qt creator we wanted to make a simple GUI software with very limited functions. To see how efficient it wold be to create our final software with C++ and Qt creator.

Current results from qt and C++:

Initial screen

In Figure B.5 you can see the result of the initial screen, where the user can input: Name on test, Time and date, Test person. If the user wants to input manual loads the GT will run at. As a checkbox. If it will be a test for offshore yes/no, storing all of the data yes/no. If the user wants to store some of the data. There will be a need to input how often the data shall be store in time interval. After all the necessary information is input the test can start. As of now the only function added to the initial screen is the possibility to write and check the boxes, and the "Start Test" button will lead to the next screen.

Time and date 01.01.2000 00:00

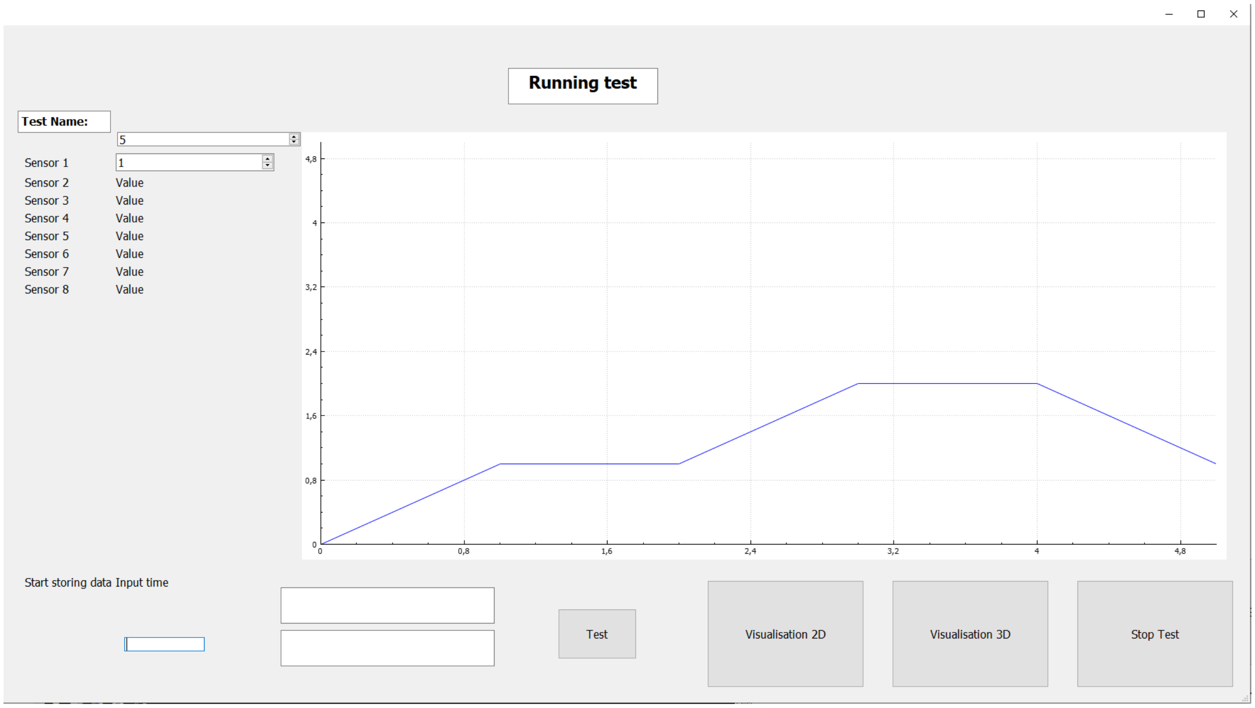
Figure B.5: Interface for initial Screen made in C++ with QT

Test started

In Figure B.6 you will see the values of each sensor as shown, with a graph representing the values live to each sensor. This is a simple representation of the data from the sensors. Easy to understand easy to use. With the use of graphs you can often more easily see changes to the sensors. As if one or more sensors have some noticeable changes. The functionalities added to this point in B.6 is the two scroll buttons on the top left with the test button. These scroll buttons are used to add values to the graph, and implemented with the "Test" button. This was done to see how we could implement values to the graph. The "visualise 2D" will take you to visualise 2D screen(B.4). The "Stop Test" button will take you to the initial screen(B.5 again.

Visualisation2D

In Figure B.4 you can see the intended idea on how the 2D can be represented. With a model in the middle of the screen showing where the sensors and equipment is placed in the



Running test

Test Name: 5

Sensor 1 1
Sensor 2 Value
Sensor 3 Value
Sensor 4 Value
Sensor 5 Value
Sensor 6 Value
Sensor 7 Value
Sensor 8 Value

Start storing data Input time

Figure B.6: Interface for test Started made in C++ with QT

GT exhaust outlet, And the values from the corresponding sensors. The intentions to make this in C++ and Qt was to see if I could add models or figures and use them. The functions available at this point is the "Test" button to change the variables to corresponding sensors to see how to implement this. And the buttons have the same functions as described in previous paragraph.

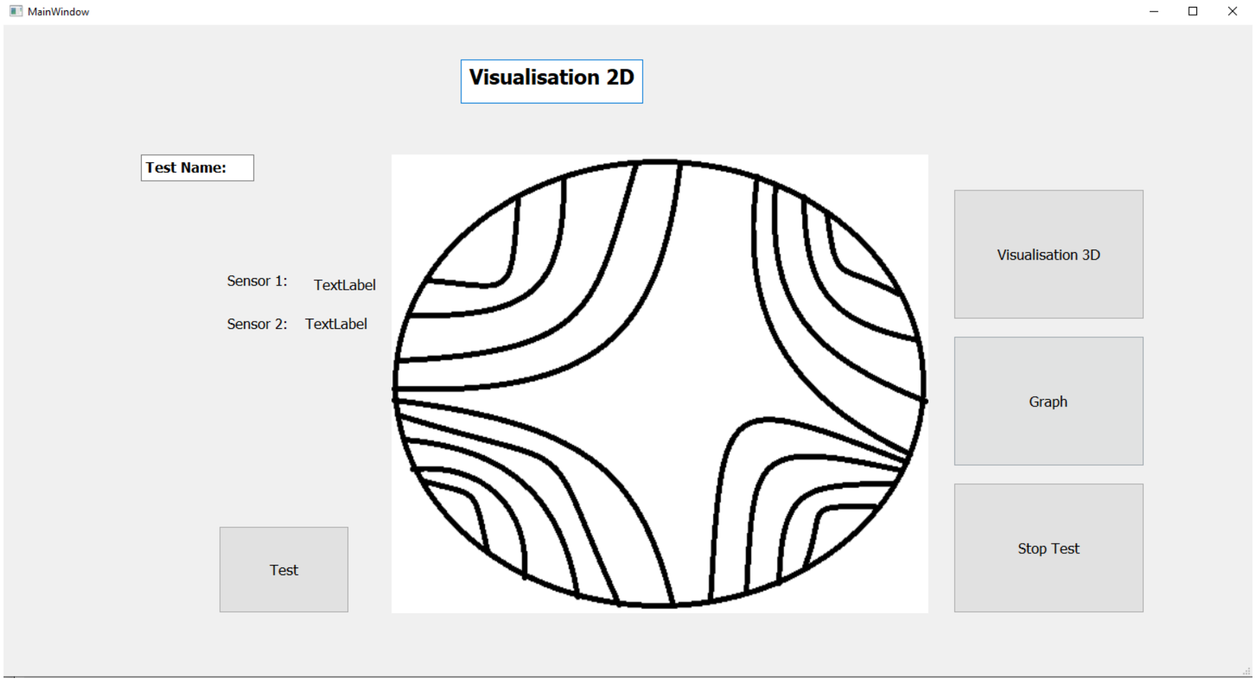


Figure B.7: Interface for visualisation 2D made in C++ with QT

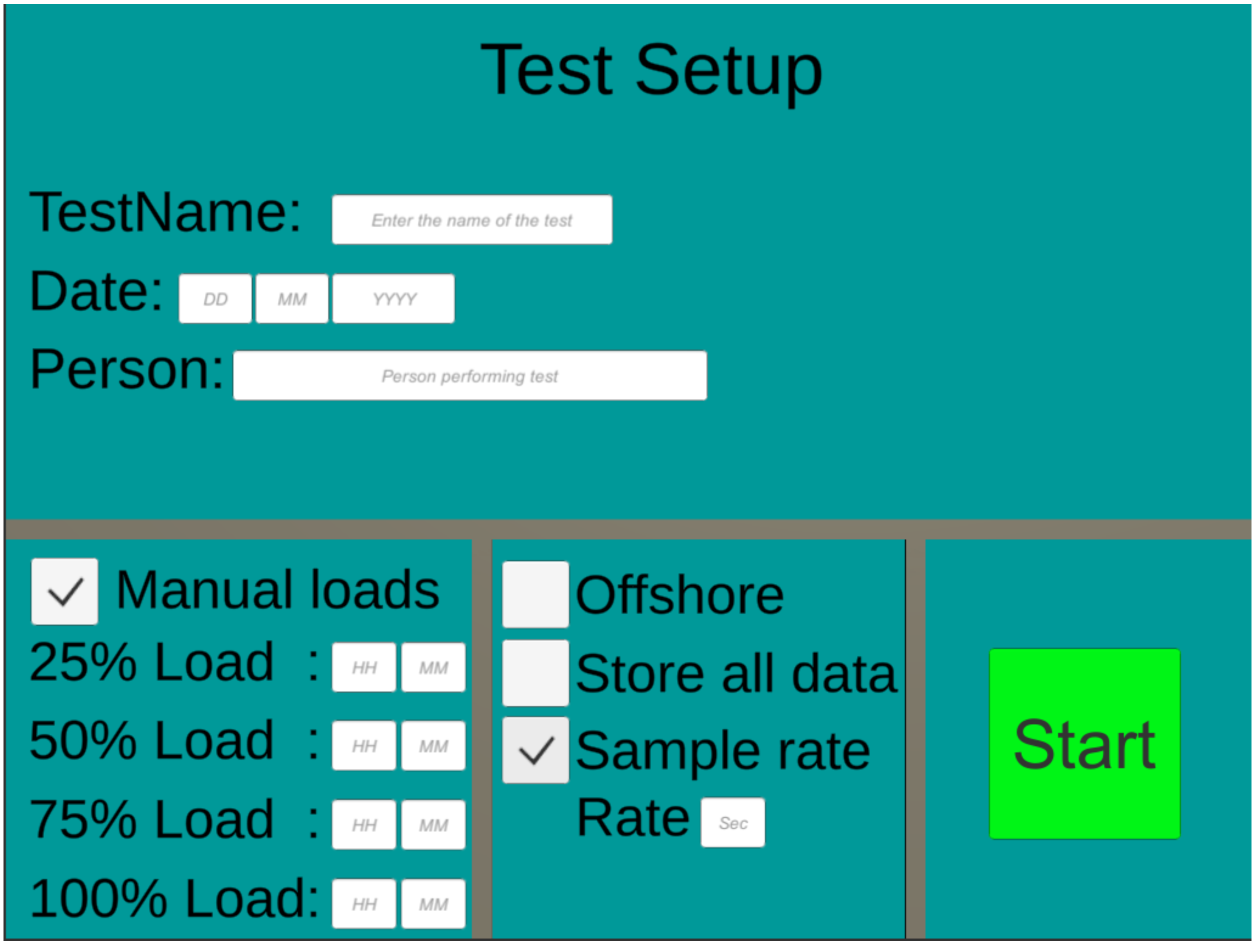
Result
The main focus for this research and simple testing was to see how things could be implemented by using Qt and C++. To see if we could create and implement everything we want for our software system. This has also given us an idea of how long different task will take and what we can do in Qt. With this in mind we can more easily choose software and language according to our needs and customer needs.

B.2.3 C# and Unity

One of the options we have considered was to use the game engine unity. We thought of this because we from software have been involved with this program earlier, this had given us some indication to its many uses. After researching more into Unity we have discovered that the ideas we have had will be possible to perform. Unity uses the programming language

C# which is a object oriented language. Unity is a great way to build a program, as it has a user friendly layout and a simple way of creating. Creating text bubbles, buttons and insert fields are as easy as drag and drop, but they will still need some code to make them perform any action. This way coding in C# is more intuitive as it is easy to observe changes directly. While researching Unity and C# we have produced a demo of our program, although the program still lacks functionality we have gotten a good indication of what will be needed to develop the program further.

Here is displayed the current results:



Test Setup

TestName: Enter the name of the test

Date: DD MM YYYY

Person: Person performing test

☑ Manual loads

25% Load : HH MM

50% Load : HH MM

75% Load : HH MM

100% Load: HH MM

☐ Offshore

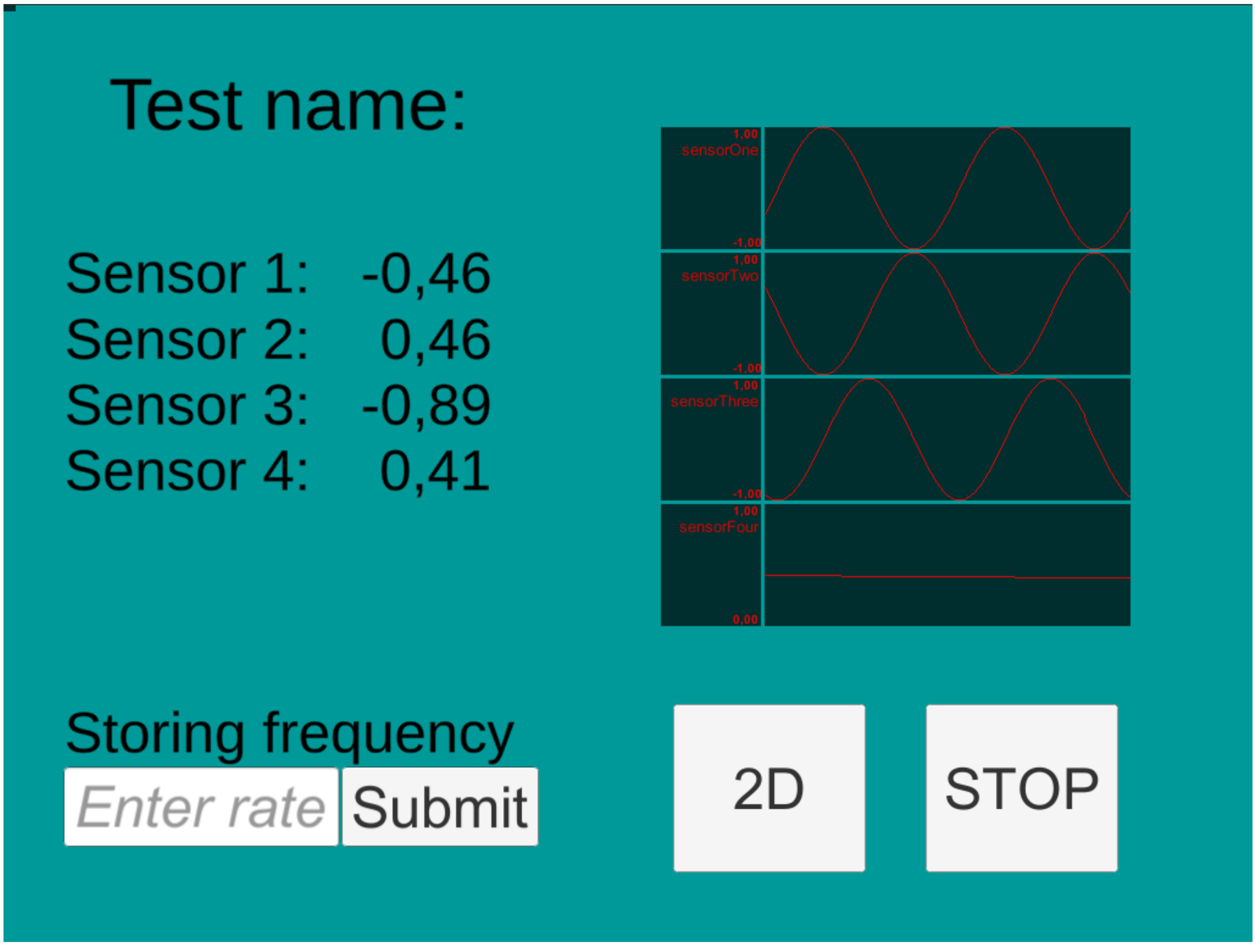
☐ Store all data

☑ Sample rate

Rate Sec

Figure B.8: Interface for start Screen made in C#

Here in the setup screen the user will write the test name, date and his/her name. There are also check boxes where you will check for manual loads, offshore, store all data and sample rate. Checking manual loads will spawn the 4 loads underneath where you will enter how long the test will run on each load, the percentages presented here are placeholder and will be changed. The offshore checkbox will give feedback to the documentation and is a way to trigger additional functions. If the store all data checkbox is ticked the program will store at the highest frequency possible, and the option underneath is if you want another frequency. When pushing the start button the test will start and the document created, the document will contain the information given in the setup screen and further data collected during the test.



Test name:

Sensor 1: -0,46
Sensor 2: 0,46
Sensor 3: -0,89
Sensor 4: 0,41

Storing frequency

Figure B.9: Interface for graph and values interface made in C#

This screen will display graphs and sensor values. These graphs comes from a library downloaded from the Unity Asset Store [42], the code for this is not our own but modified to fit our own program. From here you can modify the rate of data sampling as well as enter the

2D visualisation and stop the test. The reason we want to fit a sampling rate modifier here is because we want to be able to modify our gathering based on observation. If something unexpected happens we can sample more frequent and if its all normal and quiet we can lower it.

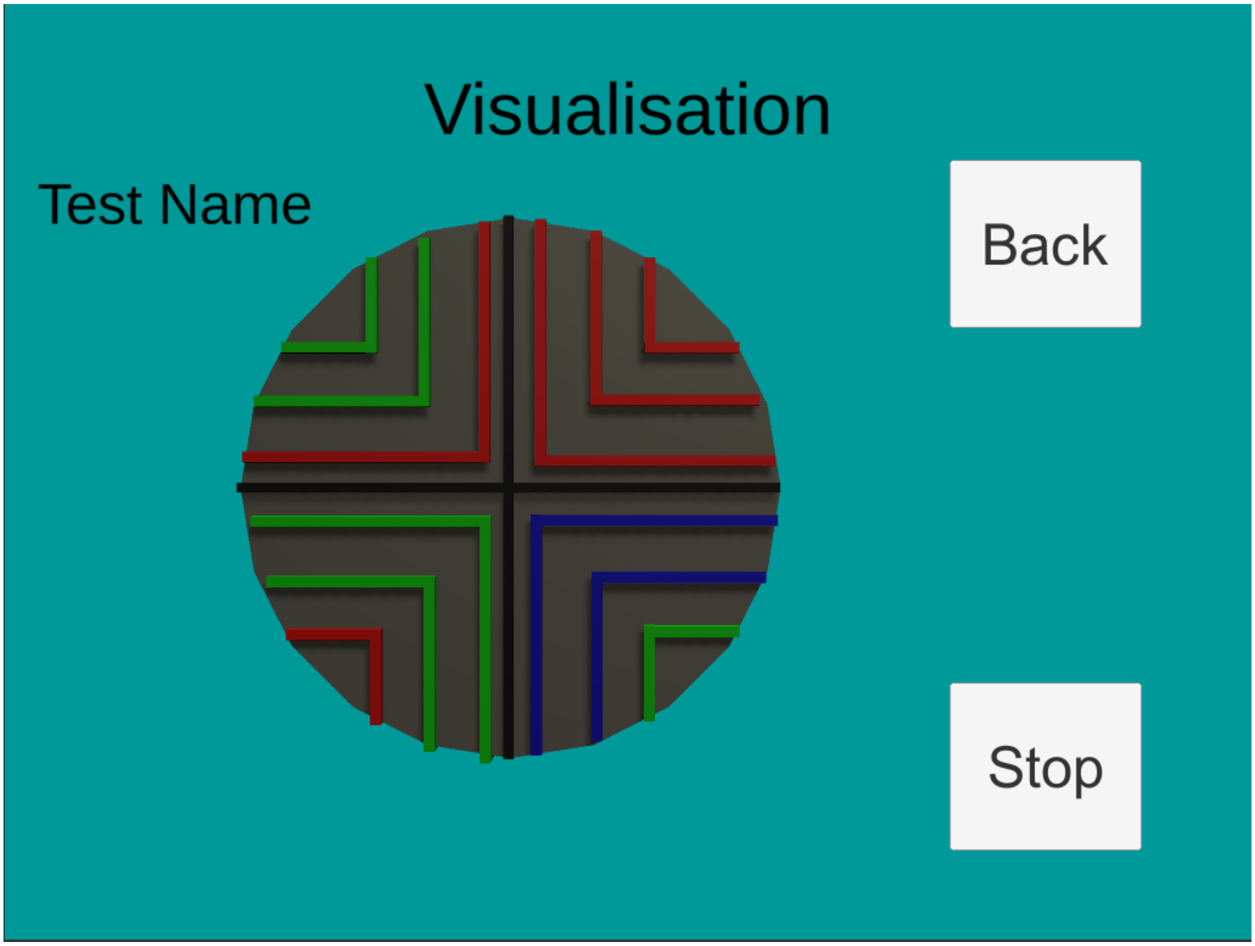


Figure B.10: Interface for 2D visualisation made in C#

This is a iteration of how our data can be illustrated in 2D. The idea is that fluids will circulate the illustrated pipes, there will be sensors at both enter and exit where we will get temperature before and after. Based on the temperature before and after we will give the pipes colors based on the outcome. Value to color chart have yet to be made, this is nothing more than a placeholder.

B.2.4 Visualization

With our solution we need to create a image with a lower resolution than the CFD software. The CFD consists of a large amount of points that gives a details image, but we cannot get as many points of measures equals to the CFD. Therefor will our visualization contain a grid with sensor containers and empty grid containers. A container can be colored according to its velocity.

With this we can use the sensor values and give them a fitting color according to its velocity, and can estimate the color of empty containers according to its neighbor as shown in the Figure B.11. Depending on how many sensors will be used the grid will be smaller or larger. Depending on the amount of sensors the resolution of our software will increase or decrease.

To clarify the white spots will be colored accordingly. This is just a concept drawing of how the visualization can be handled.

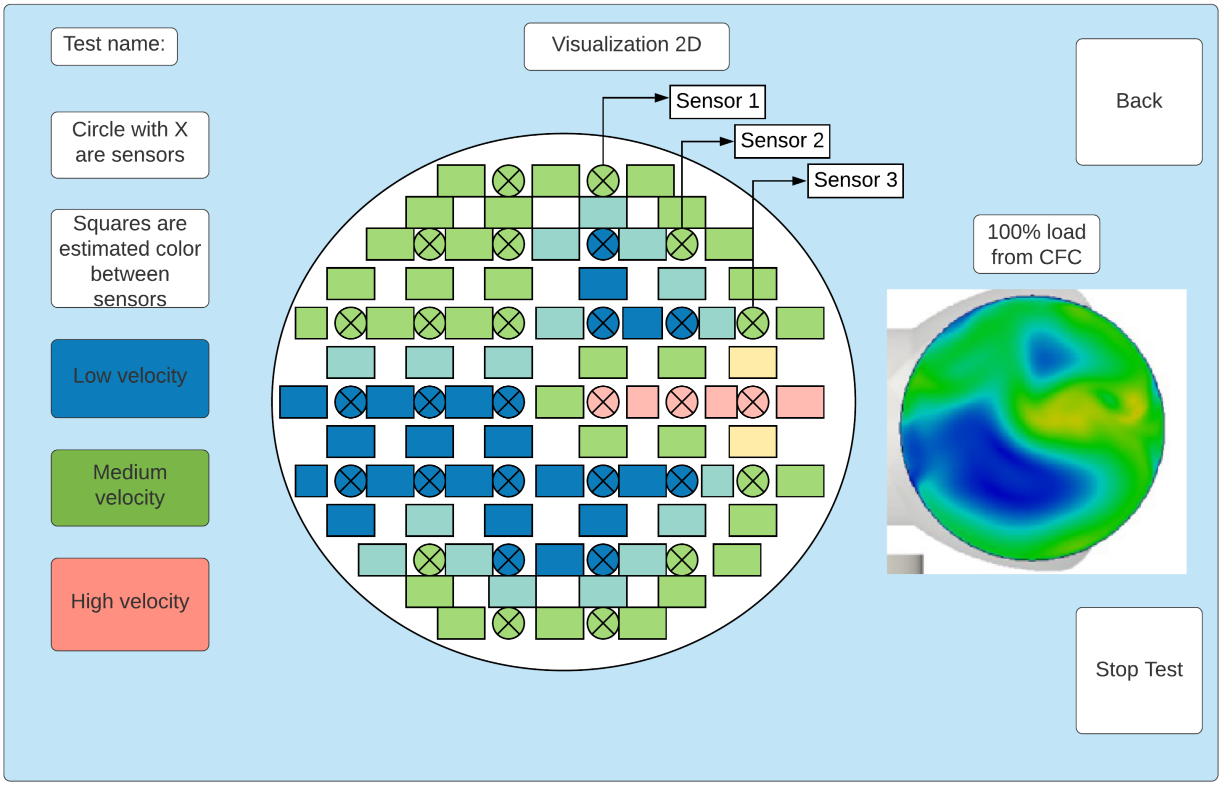


Figure B.11: 2D visualization

B.3 Software architecture

B.3.1 Sequence diagram

First iteration

To initially understand how the system would operate, sequence diagrams were created. This gave us an idea of what functions was needed, and how the data would be handled in the system.

The first iteration of the sequence diagram shown in figure B.12 shows the operator operating our system. Where he starts to fill inn inputs according to the test. So the system will use this information when it stores the data. When the operator has inserted the information needed and the test is ready to start, he will start the systems test.

While the software is running, it will display the sensor data from the sensors and visualization of the data will be shown to the operator. While the operator is running the test he can choose to either store the data or not store the data. The software will be running as long the operator wants it to run.

When the operator stops the test. The program will stop requesting anymore data from the sensors and stop the test.

This is our initial idea of how the software will be used. According to our requirements and use-case diagram.

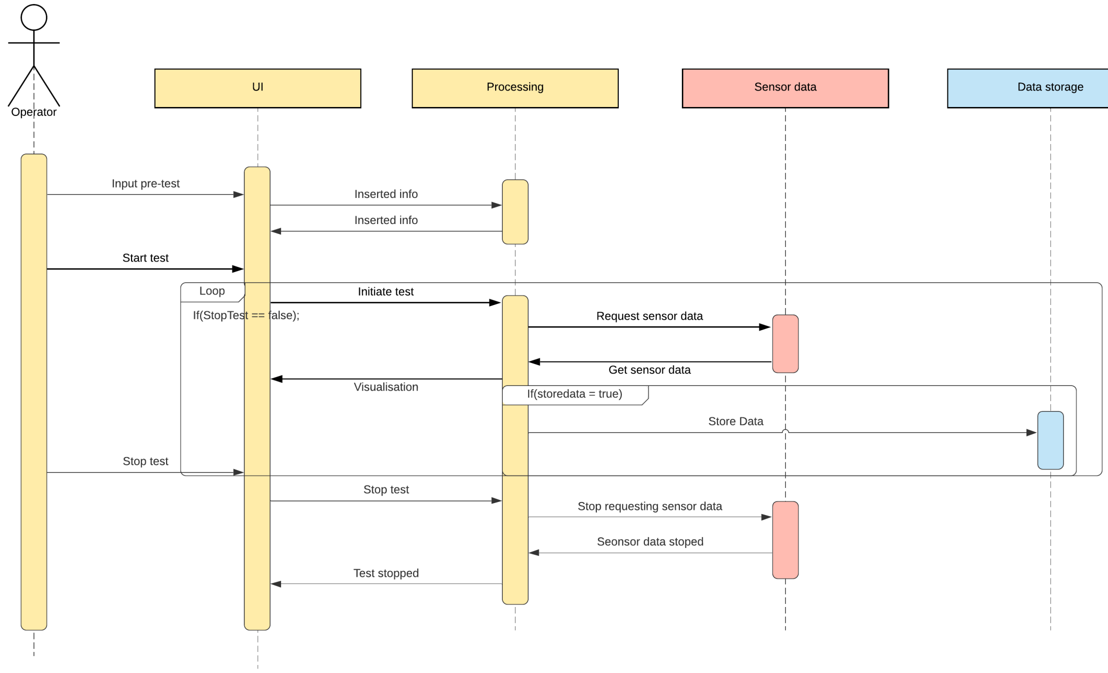


Figure B.12: Sequence Diagram first iteration

Second iteration

After going into a greater depth the sequence diagrams were reiterated. There was created 3 new sequence diagrams for the program resulting in the old diagram being out dated. The results are more detailed sequence diagrams, displaying how we envision our code and structure. The diagrams are the following For the starting phase B.13, the data storing phase B.14 and the visualization phase B.15.

The starting phase sequence diagram will handle the requirements S.1, S.2, S.4 and S.10. The process of this sequence is as described First the operator starts the software, this brings him to the initial screen where the test performer will input the test name, date of test, and their name. When this is entered the inputs will be checked for faults. If the inputs are correct the input will be written to a CSV file within the DataStorage. If the input is false the user will be directed back to the initial screen with an error message.

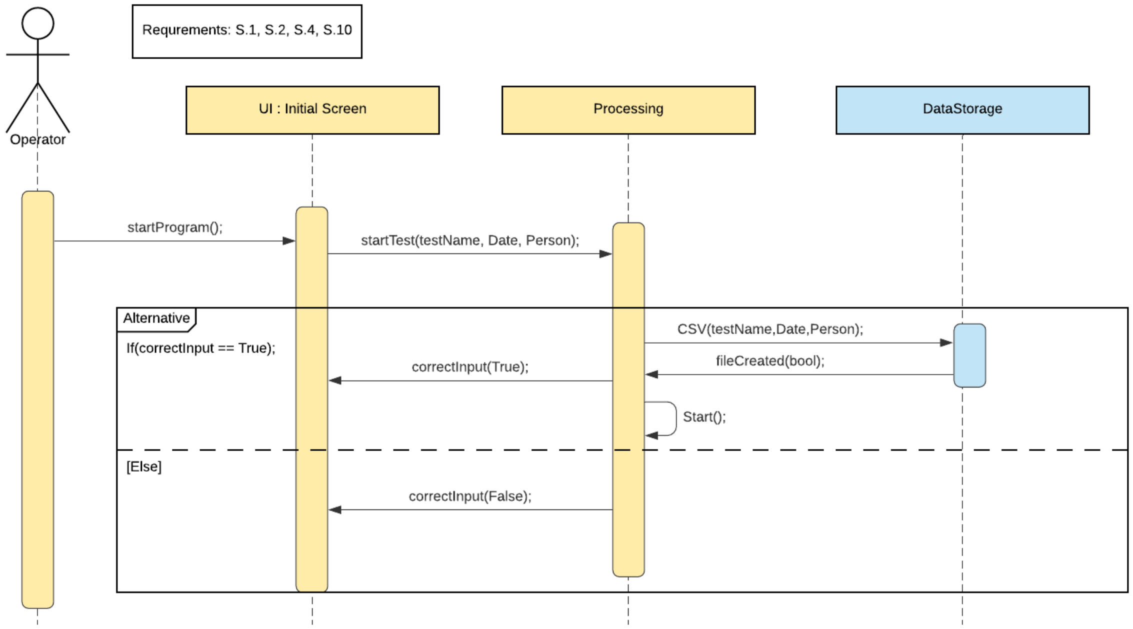


Figure B.13: Sequence Diagram startProgram

The data storing phase sequence diagram will handle the requirements S.1, S.2, S.3, S.9, S.10 and S.11. This sequence diagram illustrates how the program will request, receive and handle the data from the external hardware. The way this will be performed is that the program will run a loop requesting data connected to the respective sensor tags, this returning the value attached to the sensor tag. This will be relayed to the data storage in the same way.

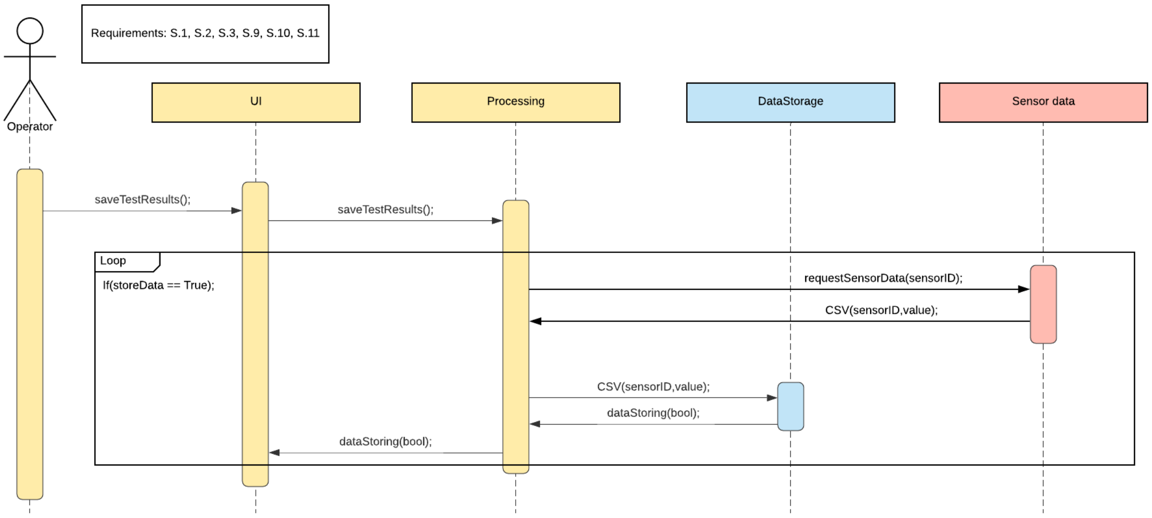


Figure B.14: Sequence Diagram saveResults

The visualization phase sequence diagram will handle the requirements S.3, S.5, S.6, S.7, S.8, S.9, S.12 and S.13. This sequence diagram illustrates how the readings will be visualized. This sequence starts at the same time the test is started. The sensor data will be requested and received with the same method as mentioned above. After the sensor data is received the sensor tag and value is given to the plots function, this function will convert the value of each sensor tag into a RGB value. The coloring will be done by comparing the RGB value of each sensor to their neighbor and the space between will be colored based on the average.

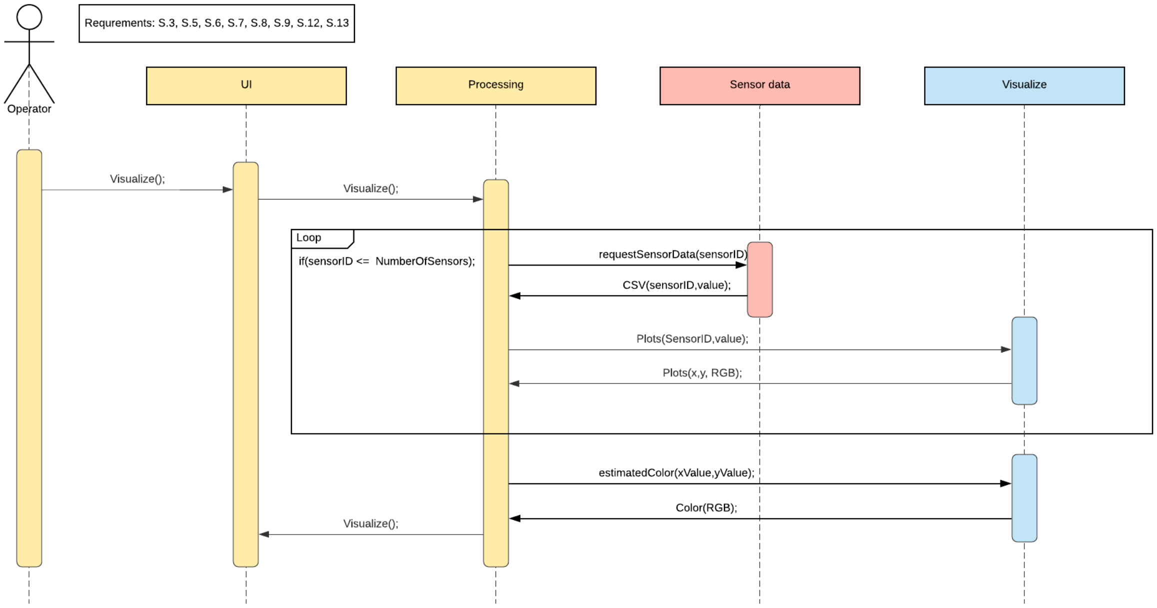


Figure B.15: Sequence Diagram VisualizeResults

B.3.2 Activity Diagrams

To get an even better illustration of how the sequences will be performed activity diagrams were created. This is to show the association each activity holds to the others. Starting with B.16 which is the activity diagram belonging to the sequence diagram B.13. This activity diagram illustrates that after the operator enters his input he can start the test. When the operator enters his input the program will check the values of the input, if this is incorrect there will be displayed an error message. If the input is correct the test will start.

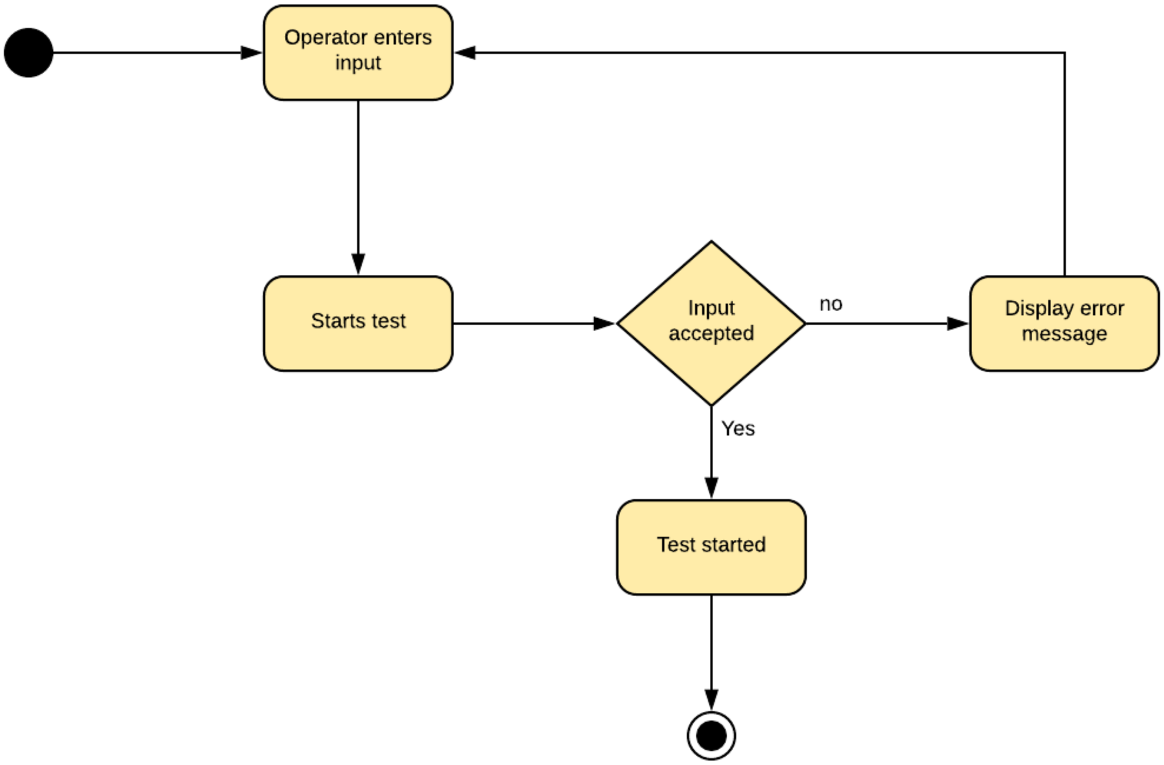


Figure B.16: Activity diagram

The next activity diagram is B.17 which belongs to the sequence diagram B.14. This illustrates that if the operator enables to save test results, he will enter a loop. This loop will request sensor data, then store the data until the test is ended. If the operator does not enable to save test results, the program will not enter the loop.

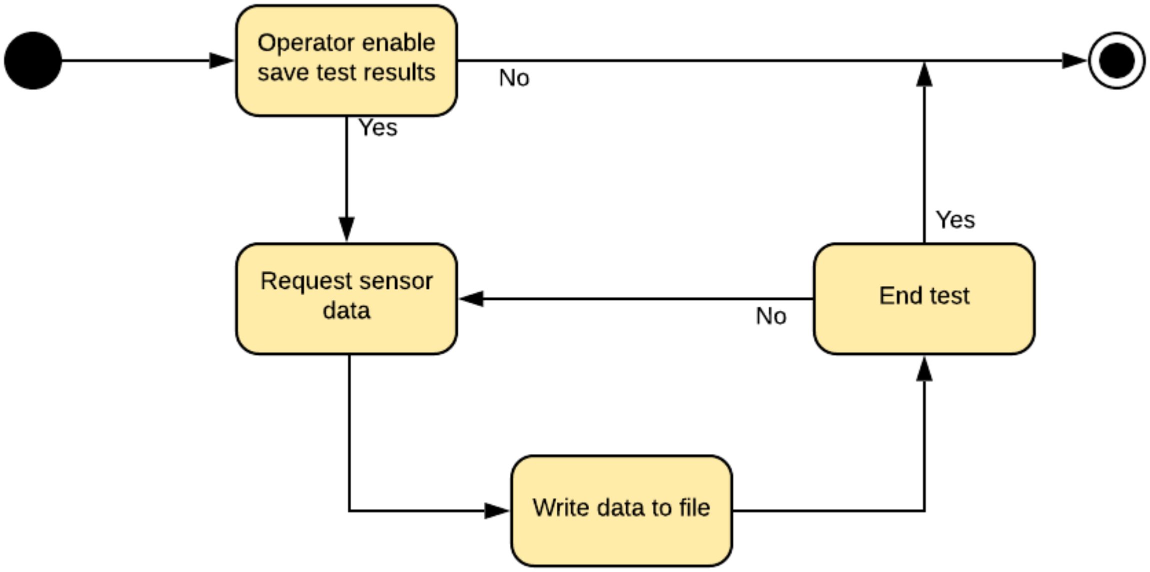


Figure B.17: Activity diagram 2

The last activity diagram is B.18 which belongs to the sequence diagram B.15. This activity diagram is for the visualization part of the software. This illustrates that when the test is started the visualization will begin. It will check if "SensorID" is the same as "Sensors", if it is not the same it will check for the data for the "SensorID". When it gets the data for "SensorID" it will receive the plots for the image and then color it. When this is done the value of "SensorID" will increase by 1, and when it is the same as "Sensors" it will be displayed. If the test is stopped during this time the program will be stopped, if not it will repeat itself. B.15

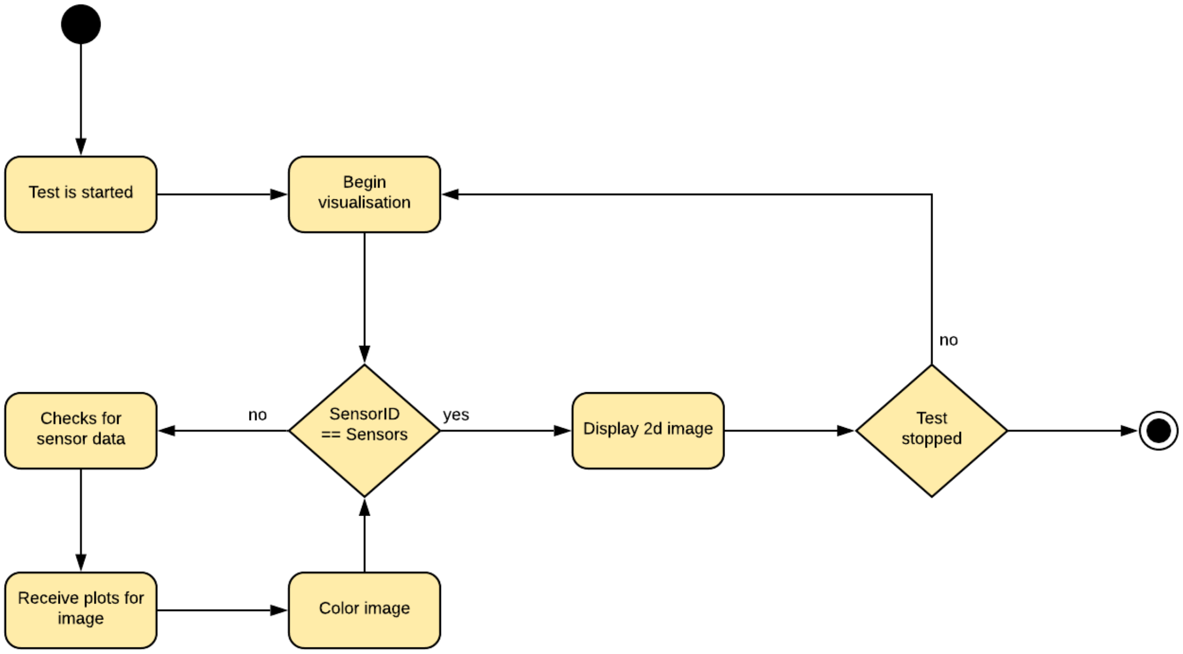


Figure B.18: Activity diagram 3

B.3.3 Class Diagrams

First iteration

To further help us get an understanding of the software, there has been made a class diagram as shown in figure B.19. This came out to be a very simplified version of the classes. To give an initial understanding of the system. This far two classes has been modeled, they are "SensorData" and "DataStorage". The "SensorData" class will have the attributes of the sensor number and the sensors value, and "DataStorage" class will have the attributes of the information for the test and the sensor data. At this point the IDE and programming language have not been decided. Depending on the IDE and language we choose, might have an impact on how we will handle our classes and functions.

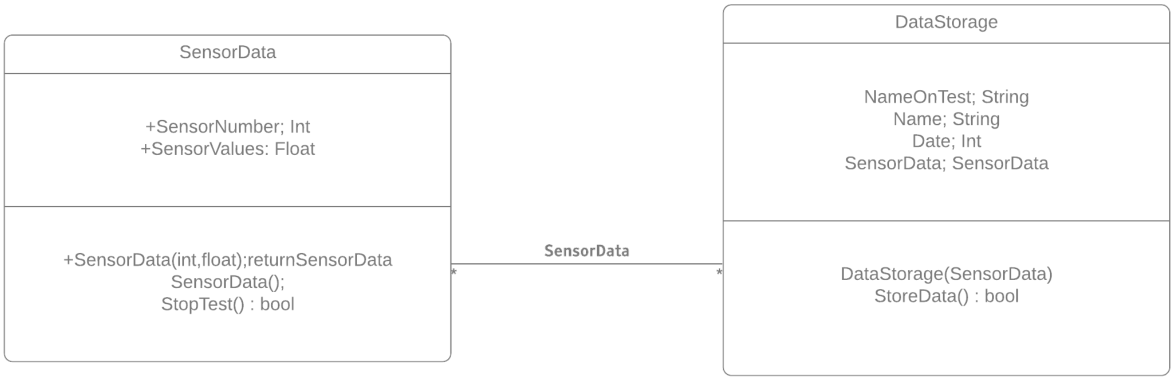


Figure B.19: Class Diagram first iteration

Second iteration

As it was time for the second iteration there was made changes to the class diagram. A new class diagram have been created and there has been changes to the previous classes. They will act as an indication of how the coding is approached when the creation of the software is initiated. The classes consists of "DataStorage", "SensorData" and "Visualization". Initially for the "DataStorage" class there has been created some functions; "CSVFile(string,int,string)" that will take the variables "testName", "Date" and "Person" that are entered by the user on initiation to create the CSV file. Next is "CSV(int,int)" that takes the variables "sensorID" and "value" to continuously send the data value along with its parent.

Our "SensorData" class will be the class that reads from the CSV file. It has a function called CSVinfo(sensorID, value), this will read from the CSV file the number of each sensor and return the value of said sensor. As implied in the figure B.20, the "SensorData" class distributes data to both "DataStorage" class and "Visualize" class without any return.

The "visualize" class has tree different functions. "plots(int,int)" takes the sensors value as "yValue" and time as "xValue". The "estimatedColor(int,int)" takes the sensor value from two different sensors and estimate the colors between two sensors. "Color(int)" takes the value for each sensor and gives it a color "RGB; int". The "plots();" function updates and initialize the colors every time its called.

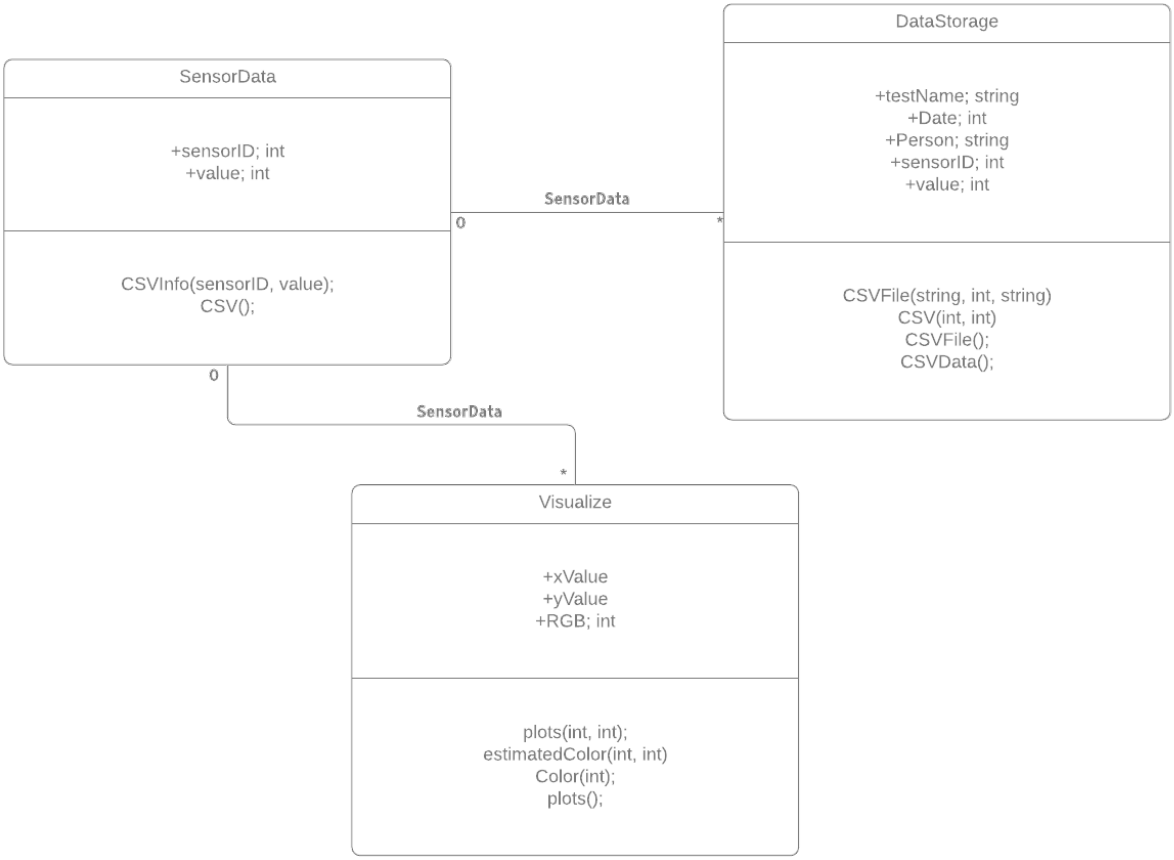


Figure B.20: Class diagram second iteration

B.4 Software solution

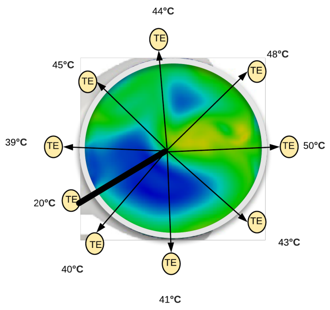
For complete documentation for software look into attachments. Source code is provided in attachment 1. Doxygen documentation is provided in attachments 2.

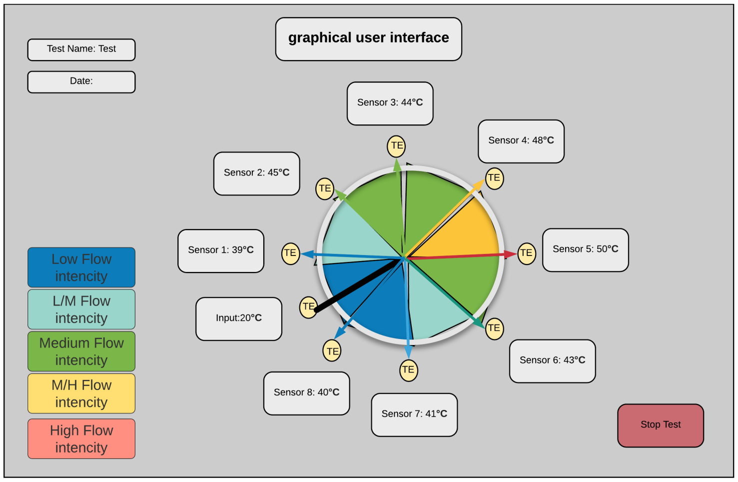
B.4.1 Resolution

The systems resolution will be a correlation with the amount of pipes and sensors we will use in our system. Even though this is an important aspect of our solution we have decided too not use a theoretical approach to this complex problem. Instead we have focused on solving this problem with a more of a testing approach. With equal distance between each tube and straight tubes. To ensure the airflow inside each tube travels the same distance before reaching the sensors.

We have looked at the CFD analysis given from Siemens and tested with the amount of pipes needed to cover the most important aspects of the CFD analysis. The method we used was to implement the CFD analysis to lucid chart, then we drew lines at critical point. We tried first with eight tubes as shown in Figure B.21. With the use of eight tubes we can see that the resolution can be to low and critical points can be missed. With increasing the resolution with the use of twelve tubes and sensors we can see an improvement for detecting critical spots B.22.

With the results from our testing we have decided to go for a solution with twelve tubes and sensors. With the use of twelve tubes we hope to find all necessary spots in the GT exhaust flow. Even though the testing only occurred at 70% and 100% loads, with the even distribution method on the pipes we hope to detect all critical spots at lower loads.





Test Name: Test

Date:

graphical user interface

Sensor 1: 39°C

Sensor 2: 45°C

Sensor 3: 44°C

Sensor 4: 48°C

Sensor 5: 50°C

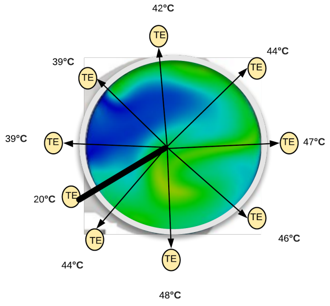
Sensor 6: 43°C

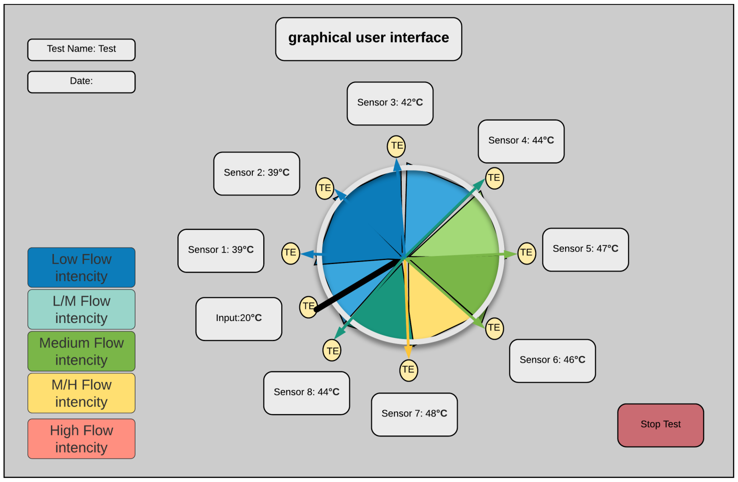
Sensor 7: 41°C

Sensor 8: 40°C

Input:20°C

- Low Flow intencity
- L/M Flow intencity
- Medium Flow intencity
- M/H Flow intencity
- High Flow intencity





Test Name: Test

Date:

graphical user interface

Sensor 1: 39°C

Sensor 2: 39°C

Sensor 3: 42°C

Sensor 4: 44°C

Sensor 5: 47°C

Sensor 6: 46°C

Sensor 7: 48°C

Sensor 8: 44°C

Input:20°C

- Low Flow intencity
- L/M Flow intencity
- Medium Flow intencity
- M/H Flow intencity
- High Flow intencity

Figure B.21: 8 Tubes test

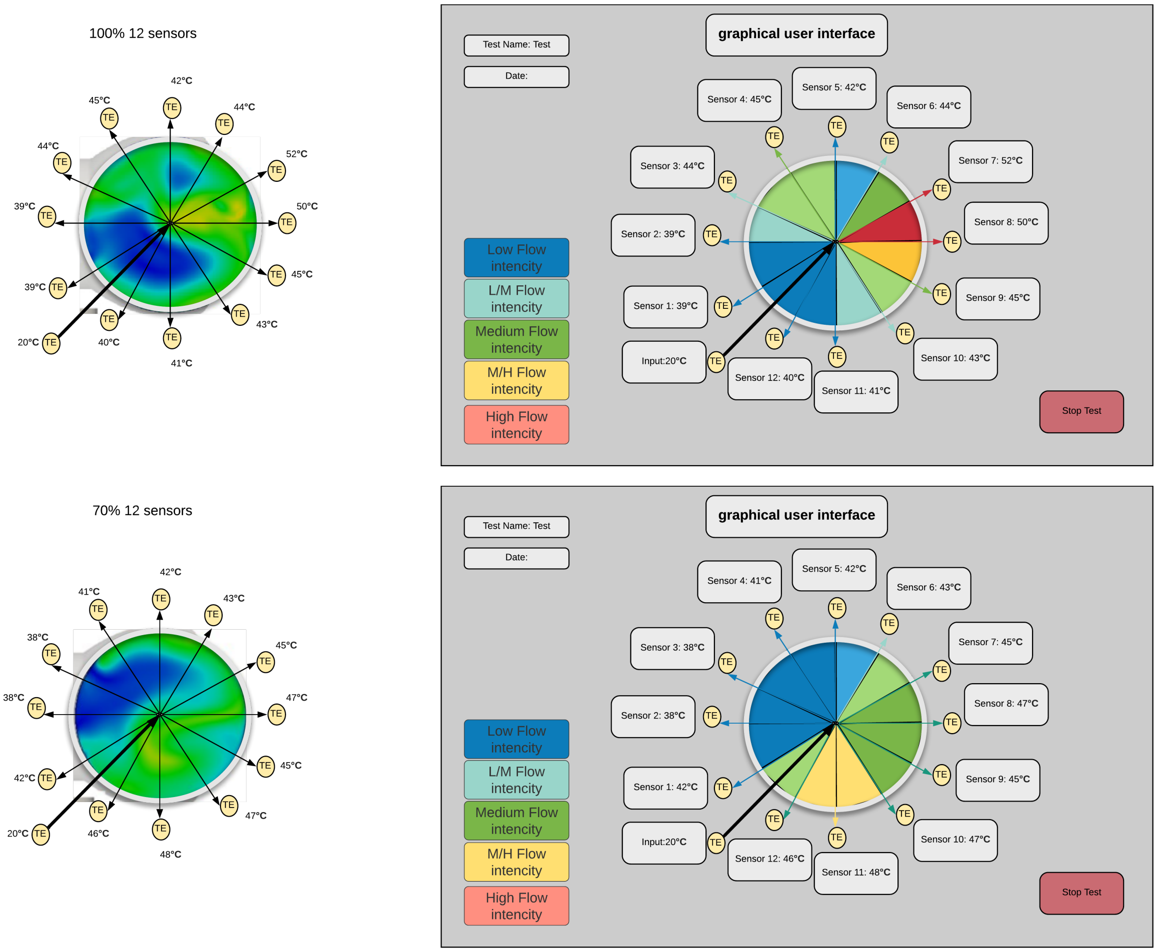


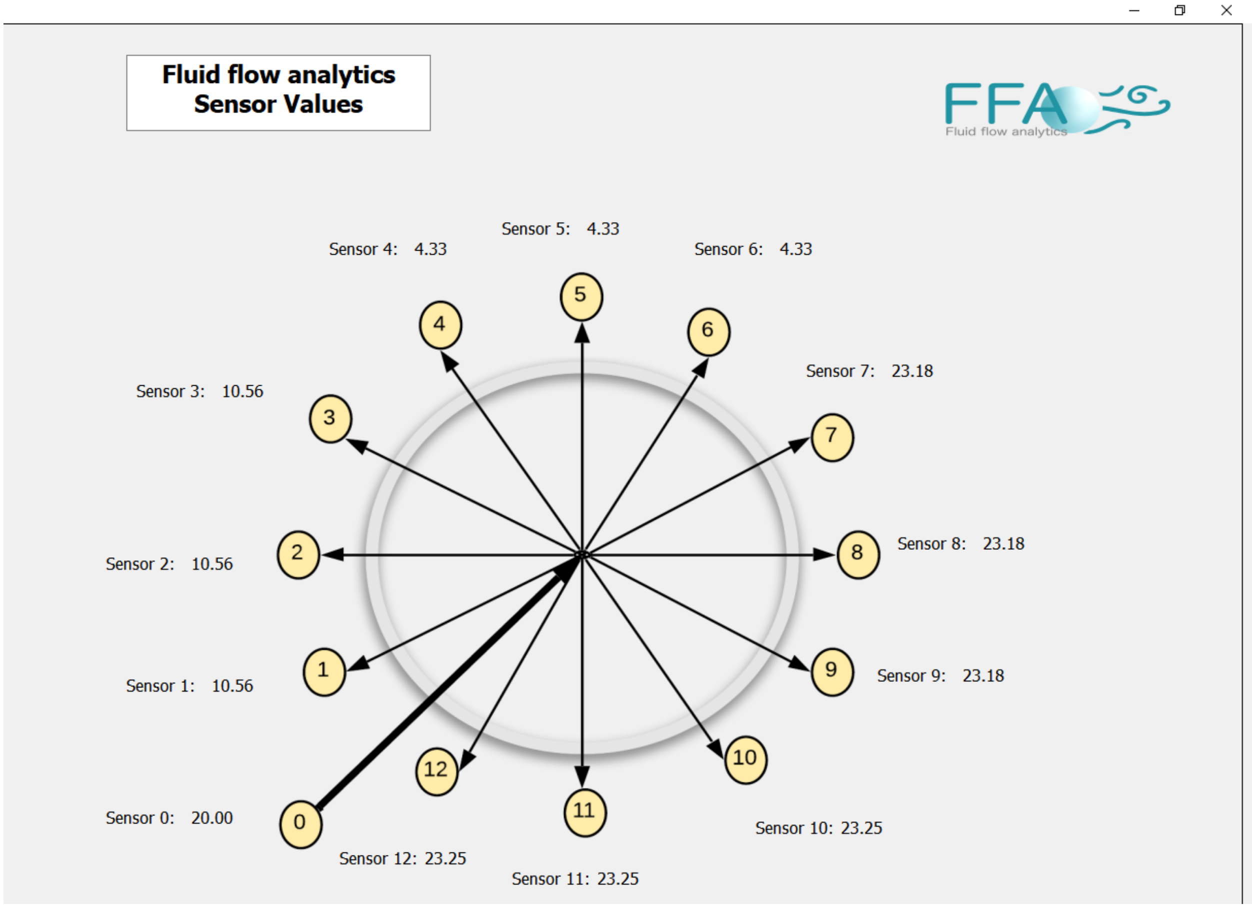
Figure B.22: 12 Tubes test

B.4.2 First iteration of the solution

In the early stages of implementation of our software system, software domain started with simple GUI objects and tried to create functions and connect them to the GUI objects. The first implementation software domain where working towards where to get values shown to the GUI. The first problem that occurred was that variables that where made would not connect to our GUI elements. This made the progressing in early development slower then expected. Instead it has been decided exercise another approach. That implied an attempt to follow the software architecture modeling procedures, instead of engaging in software design without solidified strategy.

The software modeling helped to see what functions and classes that where needed. Therefor the software domain initiated with starting to creating the first class instead. The "Sensor-Data" class. With the class created, the software domain where able to link parameters with the GUI, and this gave the Software domain a spark in the development. With how to interact and use the Qt creator.

With parts of the first class created and text objects in QT forms in Qt creator, where able to visualize data in to the GUI and show where it was expected and wanted. Shown in Figure B.23. Notice this is the early stages of the development, but the software domain could now start to further develop the software.



Fluid flow analytics
Sensor Values

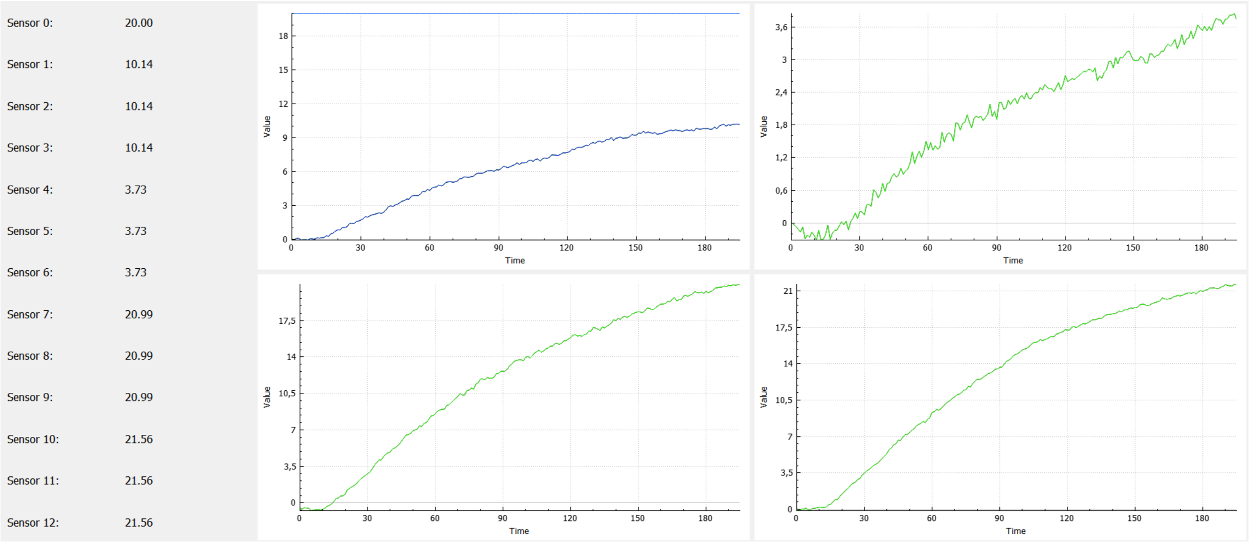
Sensor 5: 4.33
Sensor 4: 4.33
Sensor 6: 4.33
Sensor 7: 23.18
Sensor 3: 10.56
Sensor 8: 23.18
Sensor 2: 10.56
Sensor 9: 23.18
Sensor 1: 10.56
Sensor 0: 20.00
Sensor 10: 23.25
Sensor 12: 23.25
Sensor 11: 23.25

Figure B.23: Sensor Values visualization

There where created 13 objects of the "SensorData" class to each have the information from one sensor. To then display its value to the GUI. The displaying images of the Figure B.23

where created to show the exact values the sensors measured. Therefor can the operator see all the sensor values at the same time. The idea is a simple view that everyone that operates the system would understand. The visualization shows where each sensor is placed and the value corresponding to each sensors.

The second implementation we worked towards was the graph implementation. Graphs have an unique way of displaying values over timer. To visualize how the values of each sensor have changed over timer. With the knowledge of placements of each sensor and the combination of graphs. It becomes easier for the operator to see where the flow changes over time inside the GT exhaust.



Sensor 0:	20.00
Sensor 1:	10.14
Sensor 2:	10.14
Sensor 3:	10.14
Sensor 4:	3.73
Sensor 5:	3.73
Sensor 6:	3.73
Sensor 7:	20.99
Sensor 8:	20.99
Sensor 9:	20.99
Sensor 10:	21.56
Sensor 11:	21.56
Sensor 12:	21.56

Figure B.24: Graphs visualization

In the implementation process of the graphs, we used external third party library called "QCustomPlot". "QCustomPlot" have finished graphs and designs that we can link to our GUI. Shown in figure B.24. Because we have 13 sensors where one sensor checks the input value to our system and 12 sensors that checks the output value of our system. This led to our design of making 4 graphs where we implemented 3-4 sensors in each visualized graph. To better see the values changed to each sensor. The current value of each sensor and sensor number is visualized next to the graphs.

The functionality of the "QCustomPlot" draws each graph from an vector. Each sensor have their own vector in the software and prints the vector values to the graph and the sizes of the graphs are expanding according to the values of the plots. This means that the graphs starts small and visualized accurately and over time the graphs expands to fit more plots to visualize. To limit misreadings we have added the sensor values to the left of the graphs. To show current value of each sensor.

B.4.3 Second iteration of the solution

The second evolution of the software, the software team started implementing a 2D visualization with color mapping. The idea of the color mapping is to show the operator where we expect the flow to operate inside the GT exhaust, and visualize it with colors. The problem is that sensors only measure a small area of the flow, and get an average of the measured area. The area that won't be measured will be estimated, and the method of estimation is with an algorithm. The algorithm uses the values from two sensors and find the average between these senors. With the average value the software can estimate the flow between two sensors. This is illustrated in Figrue B.25. The colors are chosen depending on estimated flow. High values are shown as red, low values are shown as blue and middle values are shown as green.

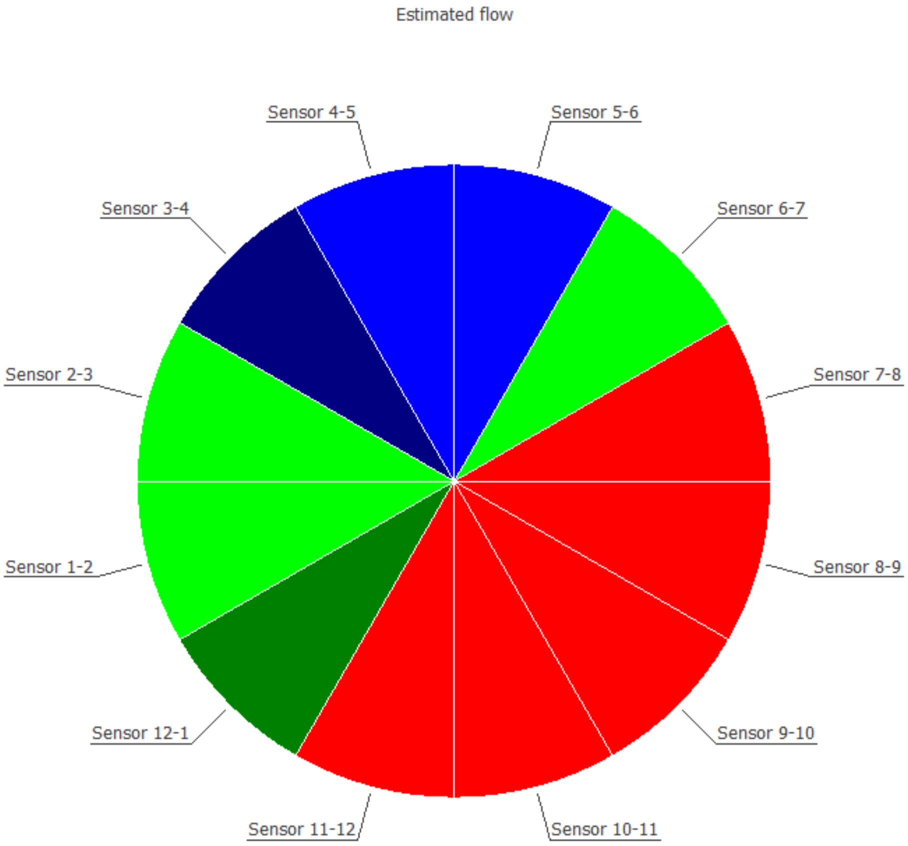


Figure B.25: Flow Estimation visualization

The implementation method used to create this object where by using "QtCharts". With "QtCharts" we where able to create a piechart. Designing and creating objects in Qt where complicated and very time consuming, hens the reason to use the "QtCharts" to design the visualization instead. The similarities between the original design and the pie chart made this a viable option. The pie chart where divided into equal slices and use the algorithm to estimate the color to change the color for each slice accordingly.

While trying to implement this into the software some problems occurred. The main problem was to connect it to our GUI. The software team tried to connect it in many different ways. The first implementation method that where tried was to make a class that handled everything with the pie chart object and connect this to a widget in the GUI. The widget was set to inherit everything from the class, but where still not able to connect it and visualize it to the GUI. The second attempt where to try the same method used for visualizing the graphs, but this also failed. To visualize a ”qtCharts” object you need to make an graphic widgets with the right properties being that the graphics widgets needs to inherit from ”QtCharts and QtSlices”.

B.4.4 Data management

As the concept has changed throughout the project (explained in system's evolution 3)the team's mechanical student conducted an experiment (A.3.5) that served as a proof of concept. By doing this, there will be real life values instead of values the team generated, which was used in the Python plot for the second presentation. From this experiment, there was created a vast amount of sensor data which was presented in an excel format and not in CVS as was anticipated and prepared for. However after some initial research the file was readable into a data frame, unfortunately there were errors when doing this. This stemmed from how the excel file was made from TDMS. Therefore the file needed to be manipulated so that all the data could be accessed in the Python program. The manipulation consisted of restructuring columns, removing special letters, characters and changing/removing rows with ”Nan” values. After the restructuring was done, it was decided that the log file would be separated into four csv files. These would contain the sensor ID, the change in temperature and a timestamp with when it was sampled. This would be used to created an in-memory database with sqlite3 [43]. By having a database containing the sensor values, there was no longer a need to make changes in code or in the file itself. *A simple script to connect and append to the data base can be found in attachment four.*

B.4.5 Storing

The method used to insert the sensor values have been through a database. When implementing databases connection the software team decided to create a demo software. The demo software where created to ensure the communication between software and the database where executed in the right manner. A visual example for how the communication where tested can be shown in F.29. Where we created a text window for each sensor from the database. The windows display all gathered information for each sensor. When the SQL queries for gathering information where executed correctly. The software team created SQL queries for storing the information. Both was tested and created in the demo software. Therefor the natural solution to store information was to choose database storing of the values. The software creates a table using the information inserted at the start page, and uses the inserted location as name. The table created holds 13 column, one for each sensor and one extra column for the test responsible name. The button is placed at the start screen. The reason is that when the test is stopped you return to the start screen and then can store all the sensor data from the test.

B.4.6 User manual

When the software is opened you have the possibility to enter the operators name and the location for where the test will be executed. These inputs can be inserted both before and after the test is finished. When the test is ready to start the operator can press the "Start Test" button. Figure B.26 shows the start page of the program. When the start button is pressed you initiate the test and connect to a database where the values will be called from. The operator will be guided to the "Sensor Values" page in the program.

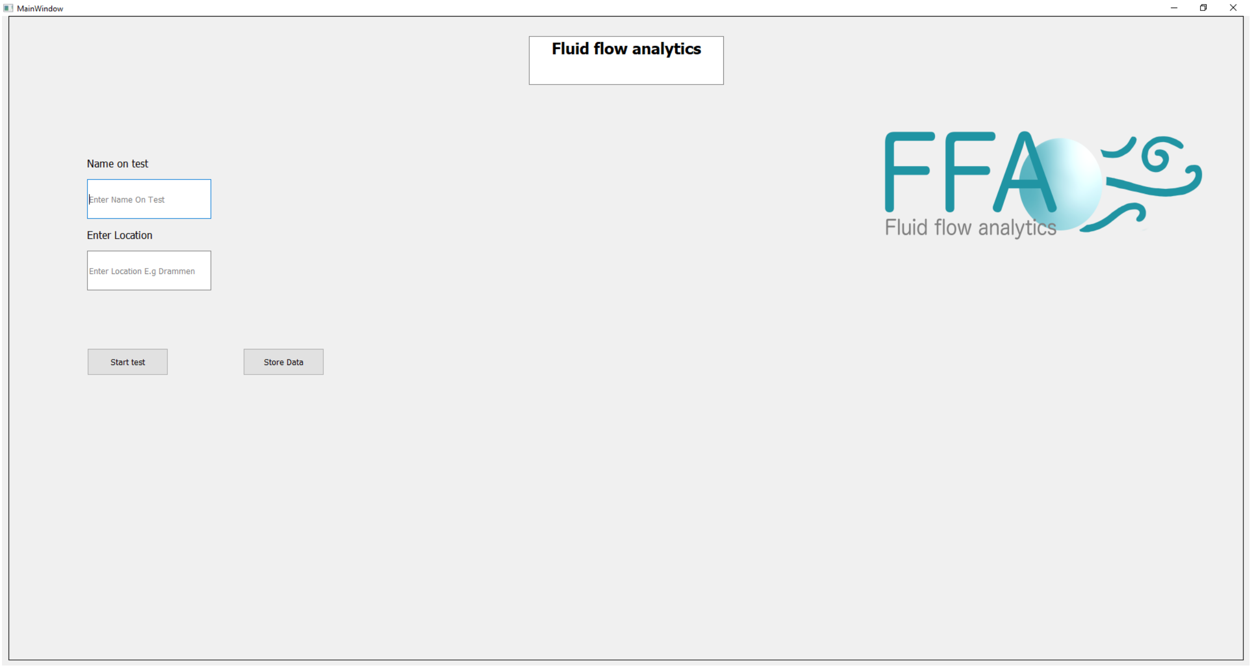


Figure B.26: Start Page

The "Sensor Values" page shows the operator the placement of each sensor and the values corresponding to each sensor. AS shown in figureB.27. The refresh rate for values are set to 1 time each second. The buttons at bottom right guides the operator to the other visualizations methods. The "Stop Test" button returns you to the start screen.

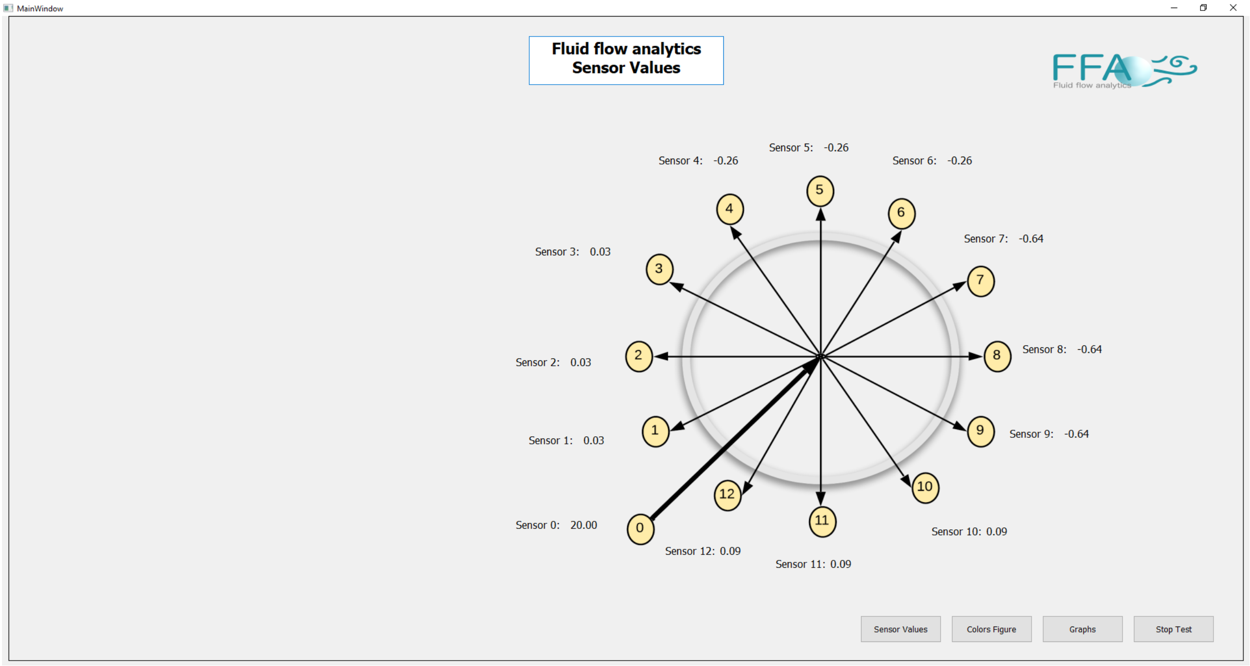


Figure B.27: Sensor Placement And Value

When the "Color Figure" button is pressed it guides the operator to the color figure page. As shown in Figure B.28. To the left the operator can find the information each color represent and the visualization to the left shows the estimated flow between each sensor.

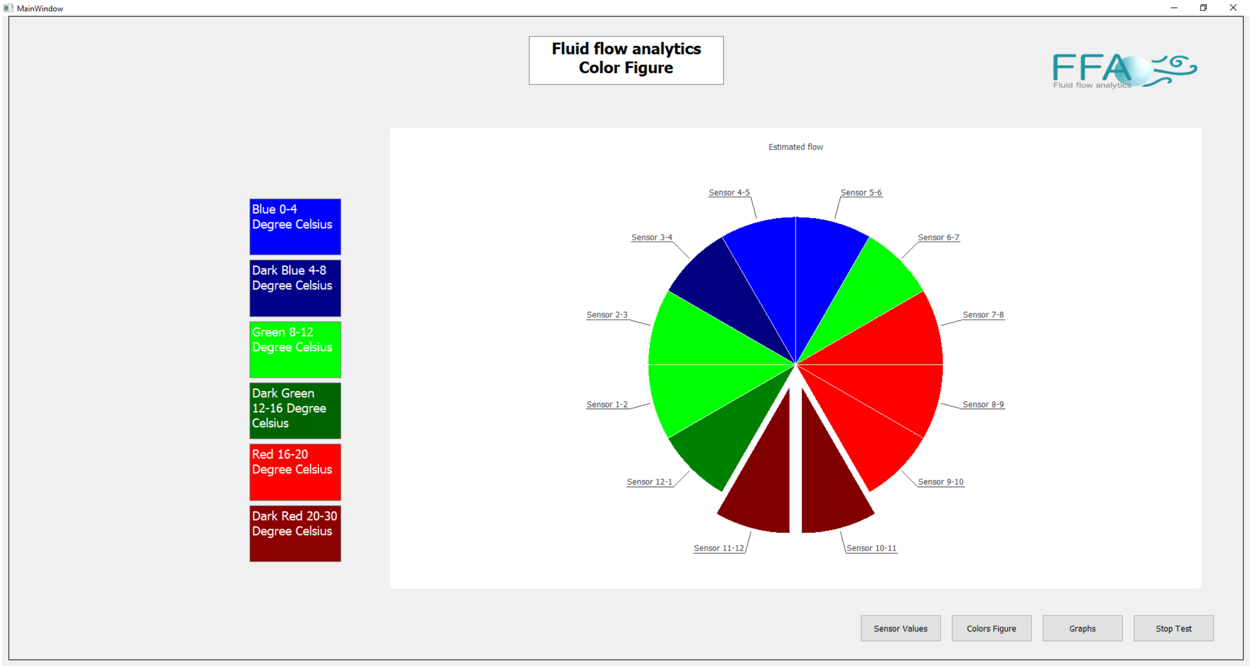
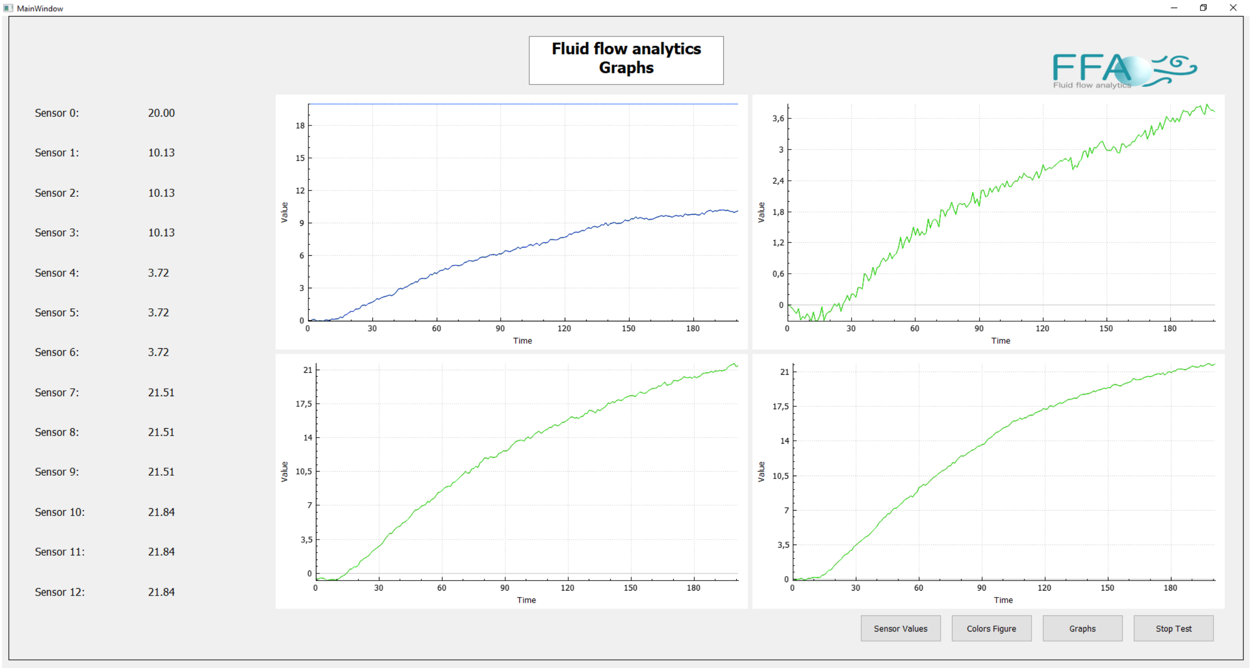


Figure B.28: Flow Estimation

When the ”Graphs” button is pressed it guides the operator to the Graphs page. As shown in Figure B.29. To the left it shows sensor values to each corresponding sensor. The graphs to the middle and right side shows all the values a sensor have measured over the test duration.



Sensor 0:	20.00
Sensor 1:	10.13
Sensor 2:	10.13
Sensor 3:	10.13
Sensor 4:	3.72
Sensor 5:	3.72
Sensor 6:	3.72
Sensor 7:	21.51
Sensor 8:	21.51
Sensor 9:	21.51
Sensor 10:	21.84
Sensor 11:	21.84
Sensor 12:	21.84

Figure B.29: Graphs

When the test is finished or stopped, and the operator wishes to store test results. Press the "Stop Test" button to return to the start page. At the start page press the "Store Data" button. The stored data will be saved in a database with a table name equal to the entered location and the "Name on test" input field creates a column with the inserted information. All sensors have automatically columns created for each sensor.

B.4.7 Library list

For C++

QTimer is a class within Qt. We need to use Qtimer to enable functions to loop within a period of time. In our case we use Qtimer to run our software in realtime, using it to simulate the frequency we would get from sensors. Because we are not able to connect our system to any sensors. For more documentation on Qtimer [44]

Qapplication is a class within the Qt creator. This helps connecting the code to our GUI in Qt creator. Connecting button, text boxes and other drag and drop items in the Qt creator. for more documentation on Qapplication [45]

QtWidgets is the premade elements Qt lets you use to create simple elements for your GUI.

You can draw objects yourself or implement finished models into your GUI with the use of the QtWidgets. For more documentation on QtWidgets [46].

QtCharts is a premade component with animations to help create charts within your GUI in QT creator. This have been used to create a circle model for our visualization in our GUI. For more documentation on QtCharts [47].

QchartView is an class that connects that has been used to connect a QGraphicsScene to an object to visualize the circle created in QtCharts. For more documentation [48].

QPieSeries is a premade class that can display data in a pie chart, to reduce the time on drawing or creating the visualization for our GT exhaust. The QPieSeries have been used, because the way we ended up designing the system ended up a lot like a pie chart. for more documentation [49].

QpieSlice lets us define how each of the slices made in the QPieSeries will act. We have used this to make a color mapping for our GT exhaust to display the estimated sensor values. For more documentation [50].

QCustomPlot is a library created to display graphs in Qt. This is a third party library that have predefined designs to graphs and have added functionalities to help create the graphs for your projects. For more documentation [51].

For Python

Pandas (documentation for version used found here [52]) is used us to handle the connection between files stored on the pc to dataframes/two dimensional arrays in our program. It is also used to manipulate it and handle the read/write functionality.

Sqlite3 (documentation for version used found here [43]) is a library integrated in python, which is used to managed the database operations and sql. We have also downloaded and used the Sqllite3 GUI, so that we did not have to run everything to cmd. And to also have a program were we see the database and can make changes and commits through, rather than only commands. In order to use it, we had to add sqlite3 to path and point it to the directory were it was downloaded (shown here).

Tkinter (documentation for version used found here [53]) is used in Python to created GUI, this was tinkered with in the research phase. But was subsequently discontinued after we decided to use the QTlibrary in C++. We did this because it was less time consuming.

PIL (documentation for version used found here [54]) in this case Pillow which is a fork from the image library PIL for Python. This means that it has branched out form the library using its source code and has been built upon by different programmers/contributors. Was used in the research phase, when finding ways to save pictures made from plots or manipulate these images. It was also used when looking at the CFD and was used in the first attempt at using meshing to find the most suitable way to place the tubes in our final concept. This was however discontinued as a time saving measure, for a different method.

Random (documentation for version used found here [55]) is used here to generate pseudo-random (referanse her) numbers. We chose to set the range for the numbers as this was not meant to pass a randomness test (kanskje referanse her) for a simulation, but rather just show how our graph plotting would handle real time sensor data. Which we emulated by writing the random numbers to a file, then reading and plotting them.

Itertools (documentation for version used found here [56]) is a module to iterate or count, the count() takes a starting point in this case start time is set to zero and a step which indicates how much it should iterate with. Here we used one as to be used as a timer for seconds. This was also only used in the research part, with the loop where that emulated time passing as sensor values were made.

CSV (documentation for version used found here [57]) is a Python module for handling read/write with csv files. This was used early in the research phase, before finding the pandas library, which has more attributes and was there preferred over the csv lib.

Time (documentation for version used found here [58]) is a module for time, date, calendar etc. for us it was used to with the time.sleep() function. Which emulates a fixed sampling rate for sensor time stamps. In our case each new value was added after 1 second.

Numpy (documentation for version used found here [59]) is a library used for scientific computing as well as a way to handle arrays and higher than one dimensional arrays. As Python does not have a built in way to handle arrays, but rather makes lists. For us it is used for arrays in one and two dimensions.

Matplotlib (documentation for version used found here [60]) is a 2D plotting library for Python, we have used to for primarily for graph plotting and this was in the research phase.

B.4.8 Future work to finalize software

The software is missing a few parts to be finalized and ready to use. One element that is missing is the communication between the electrical and software domain. The solution to this problem is to use ”Open Platform Communications” to connect signals from PLC into the software. The PLC sends the signals to ether a PC or a stand alone device with ”Open Platform Communications”. The PC or device can convert the signal input from the PLC and give output in forms such as excel or database. Other options are also available, but with the design of the software, the recommendations is to use a database based connection. Shown in Figure B.30.

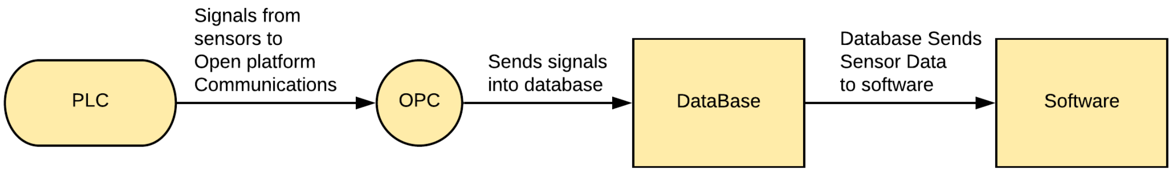


Figure B.30: PLC to Software

Depending on the chosen method for connecting the PLC to the software, some elements will need new iterations. If the database option is the selected option. File path for the database will needed to be updated in the source code. Another element that need iteration and testing is the color estimations. The measuring results from a test with flow equal to the environment inside the GT exhaust flow has not been proven. Therefor the colors represented in the current software is not calibrated for such a test.

C. Appendix - system visualization

C.1 System Visualization

C.1.1 Software architecture

The research team decided to create a visualization of the entire system so that it could be used as a commercial for our product. The visualization will be created in the game engine Unity. This decision was made as one of the software engineers are familiar with this program from earlier research during the project. To prepare the creation of this the visualization responsible have constructed a use case diagram, a sequence diagram, a couple of class diagrams and a couple of activity diagrams that will be displayed and explained here. Doxygen documentation of the code for the system visualization is provided in attachment 3.

Starting with the use case diagram fig:C.1, this is quite straight forward. The operator of the program will be playing the program, he will then be able to assemble the sensor. After the sensor is assembled the operator can start the test, if the test is started the results will appear.

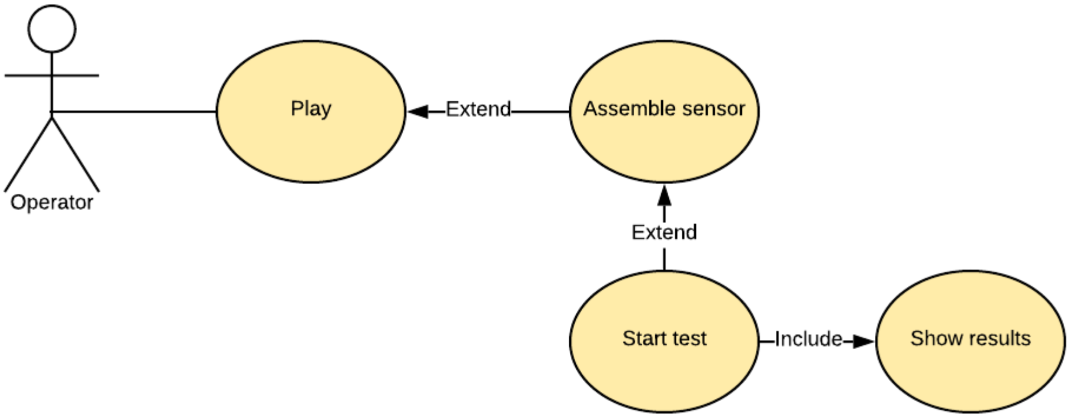


Figure C.1: Use case diagram

The sequence diagram fig:C.2 is also quite straight forward. The operator starts the game, he presses the keys and if the key has an action it will be performed, otherwise it will not.

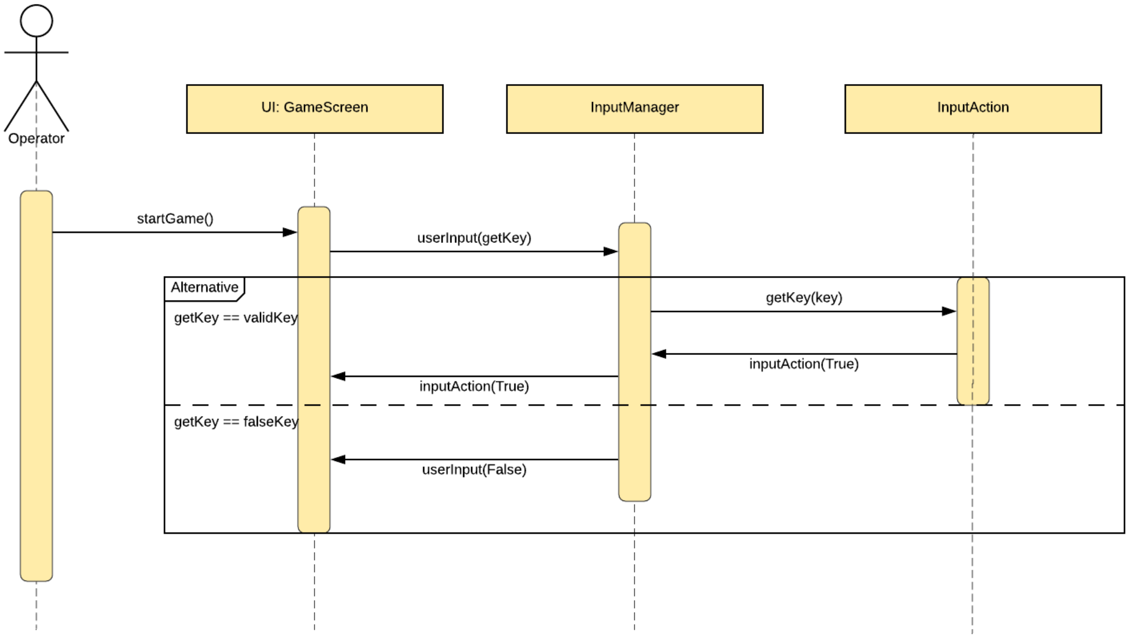


Figure C.2: Sequence diagram

The class diagrams fig:C.3 represents the classes. These are not all of the datatypes or functions in use but rather a representation. The additional datatypes and functions will be described further in the documentation of the code.

The ”GameFunc” class consists of a public string which consist of what button the operator is pressing. A public Vector3 ”moveDirection” which is a datatype that contains 3 kinds of data in x direction, y direction and z direction. A GameObject type called ”figure”, this is the figure representing the operator. It also has a function ”inputAction(getKey)”, this handles what happens based on what the operator presses.

The ”playVideo” class is a class made for playing a short video of our software when the test is started. This contains a boolean called ”playing”, the boolean is true when the video is playing and false while paused. The value of the boolean is controlled by the ”inputAction()” within the ”GameFunc” class.

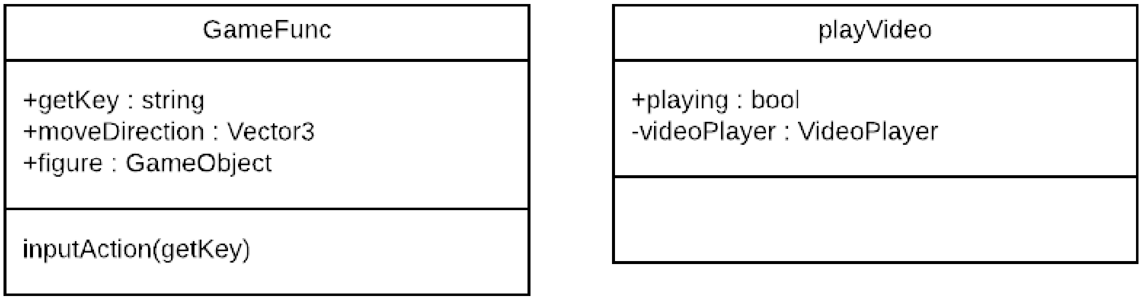


Figure C.3: Class diagrams

Starting with the smaller of the activity diagrams fig:C.4. This is an overview of the users actions to the program. The game is started, the program gets user input, if the key is invalid it will return for new input, if the key is valid it will perform the key's action. If the key's action is to quit the game it will, if not it will return to get new input after the action is performed.

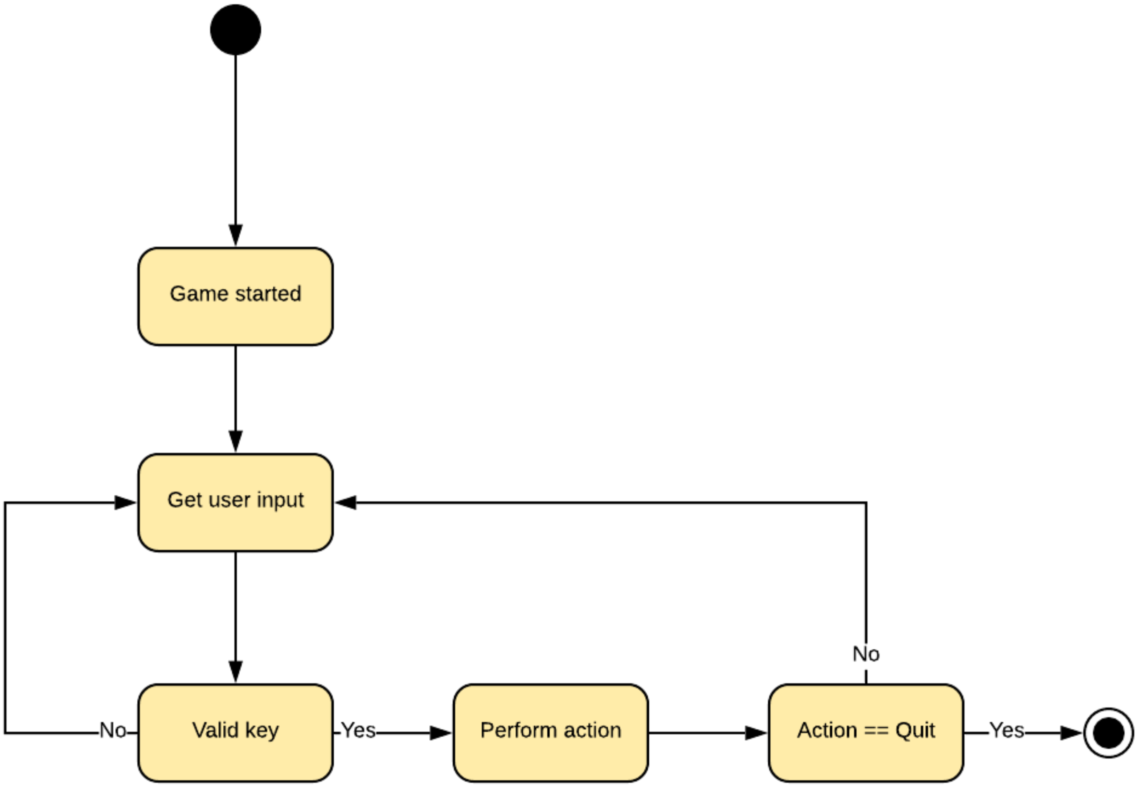


Figure C.4: Activity diagram

The bigger activity diagram ref:C.5 is describing the action following the user input. Starting by looking to the right on the diagram there is displayed that if the action is "Assemble" it will check if the figure is inside its designated assembly area. If this condition is true then the sensor will be assembled and the action will stop. If the condition is false the action will be stopped. If the action is "Start test" it will check if the figure is inside the designated test area. If it is not the action will be stopped, but if true it will continue to check if the Sensor has been assembled. If the sensor is assembled the testing display will become active and run the video. If it is false the action will be stopped.

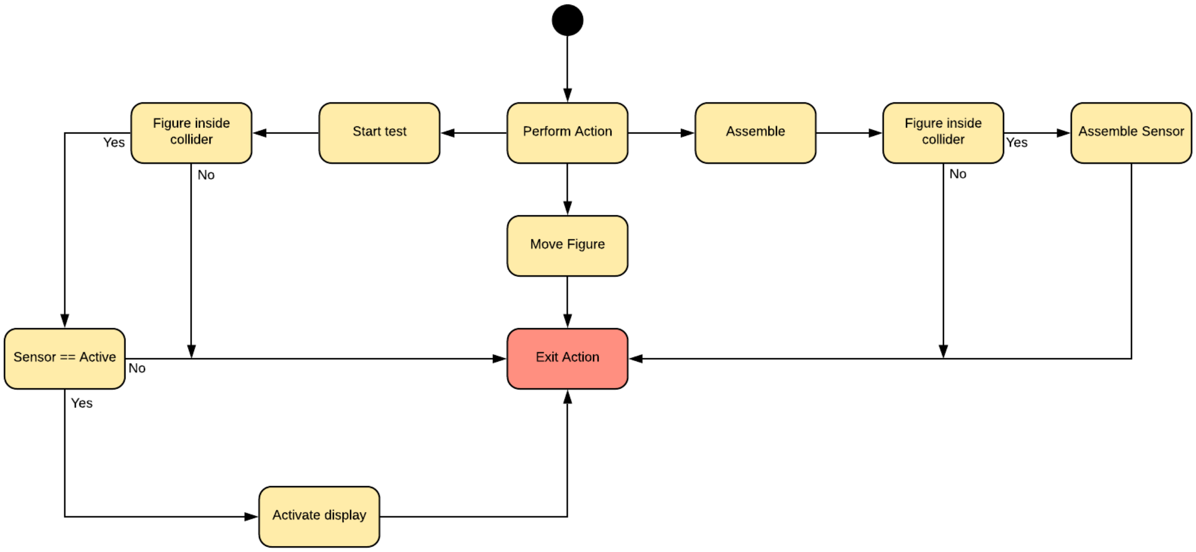


Figure C.5: Activity diagram 2

C.1.2 First iteration

During the first iteration the top priority have been to work on getting the preparations in order. The preparations consisting of the use case diagram, sequence diagram, class diagrams and activity diagrams. After the UML was completed, the Unity project could be built and the visualization work commence. During this iteration the physically correct models have not yet been created. To compensate for this boxes and other default models from Unity have been used as placeholders. To start the project, boxes, planes, and cylinders were used to represent different models, as visualized in this picture fig:C.6. This picture displays the results of the first iteration. Here there has been placed standard models to resemble the gas turbine enclosure, exhaust pipe, a ramp and a figure for the operator to move. There has also been written some code to make the figure movable.

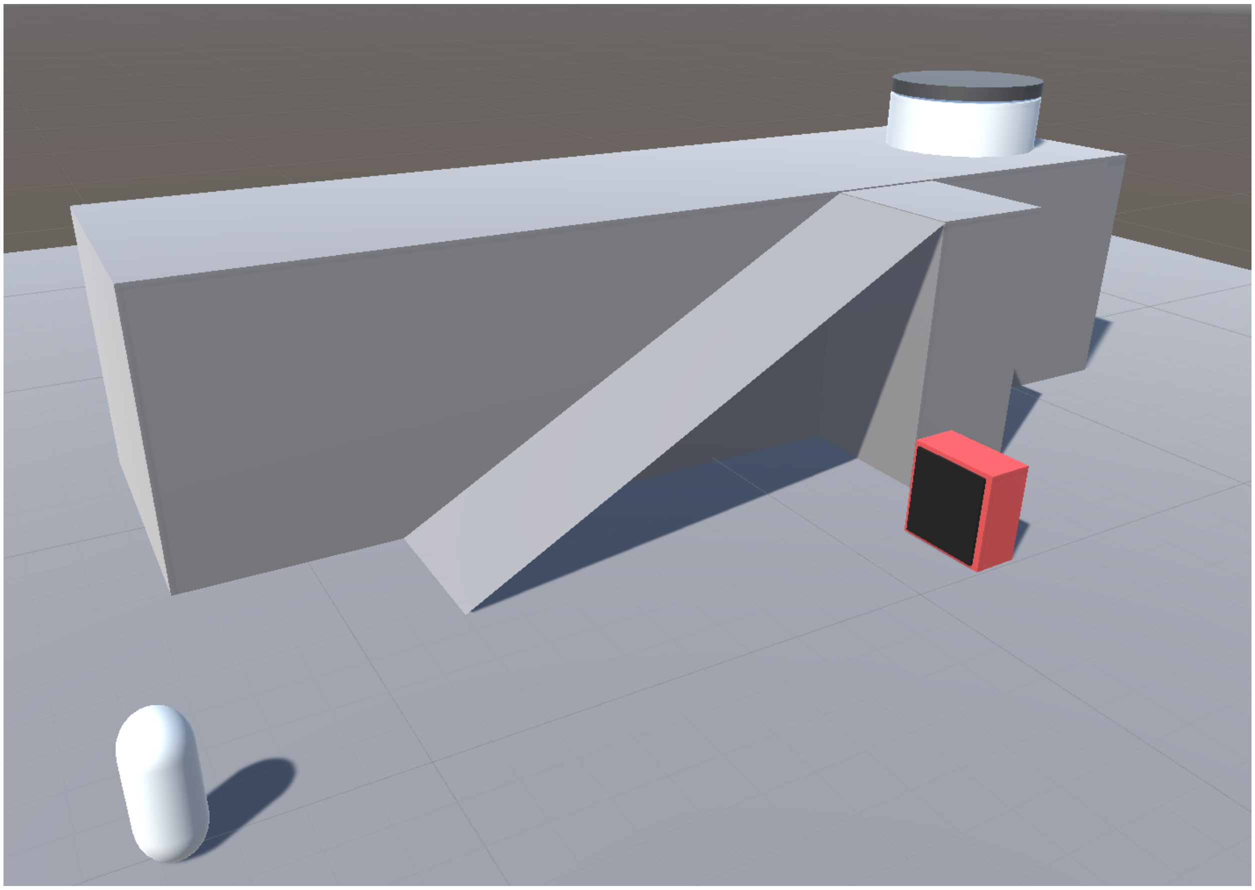


Figure C.6: Visualization overview

C.1.3 Second iteration

When it was time for the second iteration the mechanical lead gave some support for the visualization responsible. The mechanical lead gave their support through creating models in solidworks A.4.1. These models were exhaust pipe, flexible, support, pipes and connector. These models have been used to visualize how the product would look and how it would be installed. To visualize this the visualization responsible have researched how to make animations for the models, giving the program an illusion of how they would be connected to the system. Additionally a smaller object have been created beside the gas turbine. This object is representing the computer of the system, here it will be displayed a video of the end result of the software.

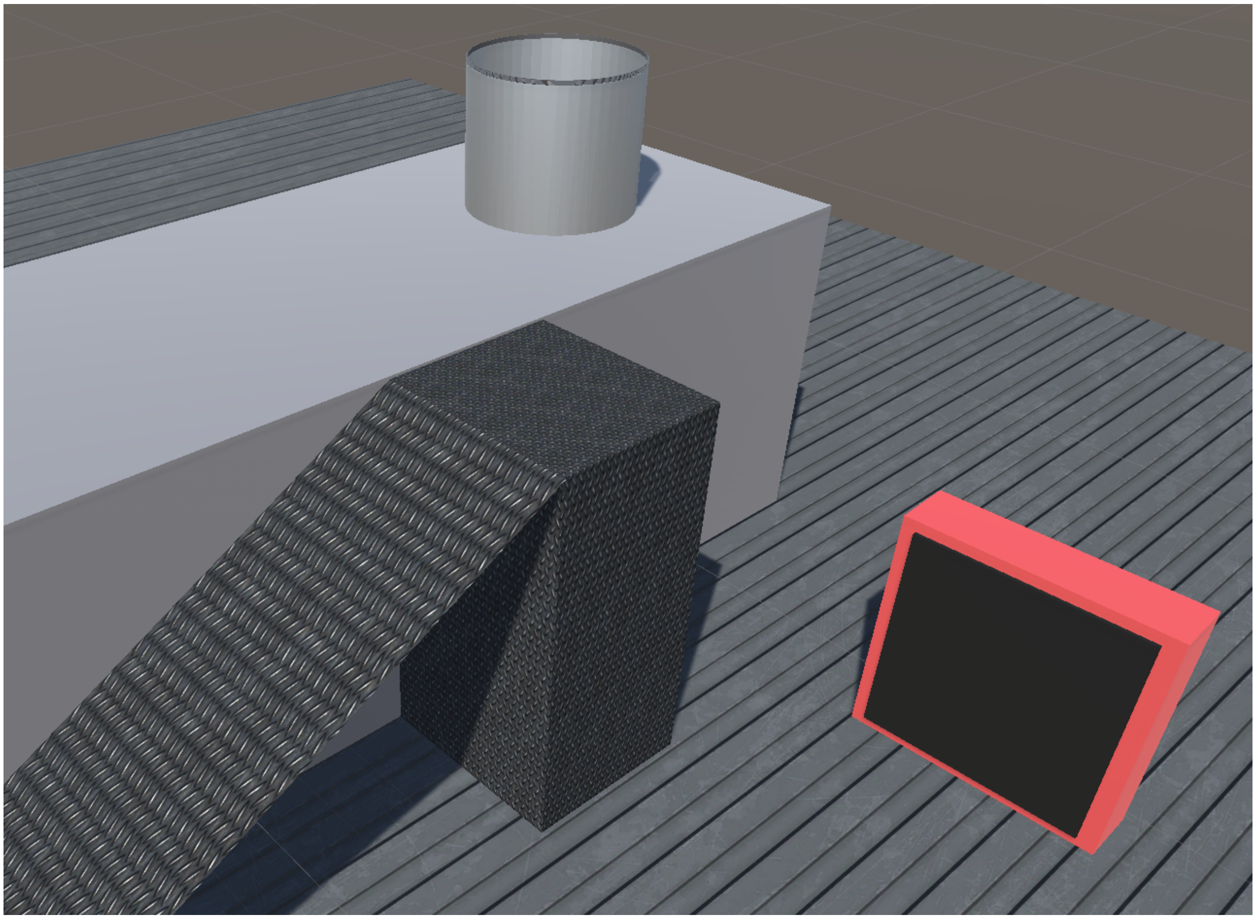


Figure C.7: Visualization overview

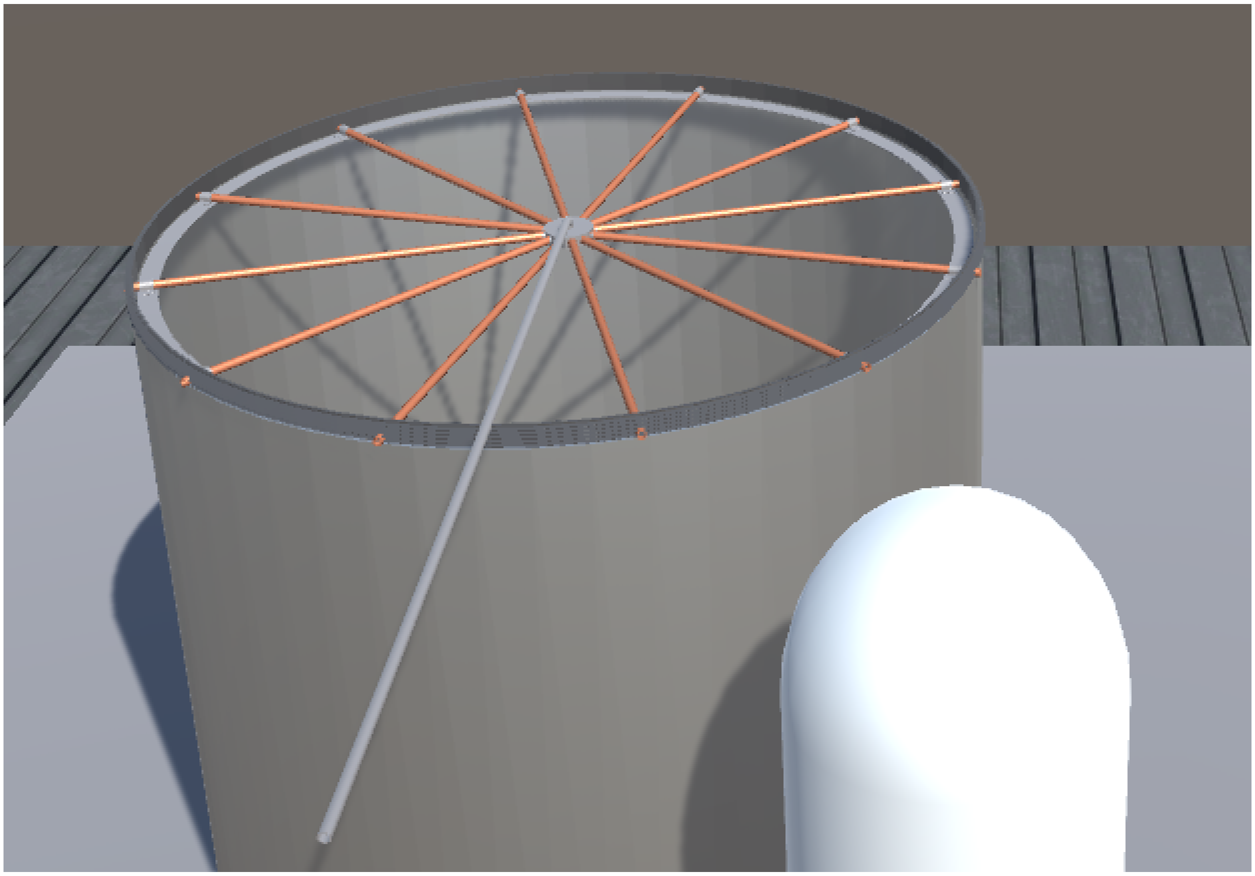


Figure C.8: Exhaust pipe with measurement system

For the materials applied to the pipes, platform floor and the ramp, the visualization responsible searched the Unity asset store. This is where a pack of materials was located and downloaded for free[61].

C.1.4 Third iteration

Early in the third iteration the visualization responsible realised that the way the animations were made, made it hard to understand for someone outside the project. Early in this iteration measures were taken to improve the comprehensibility of the visualization. Most of the animations had to be redone and this proved to absorb a lot of time. Additionally, the mechanical lead created another model in solidworks, namely the silencer for the GT exhaust. The silencer is the part of the exhaust pipe belonging above the measurement system. The biggest change is the way the system is visualized, instead of moving manually with the figure, animations have been created for the camera. This proved way more effective as it provided easy navigation, as a figure would have a hard time inspecting inside the exhaust. More items have been created, these are the thermocouples, cabling and a junction box. Another feature that has been added is exhaust particles, this is a stream of particles spawning in the bottom of the exhaust pipe.

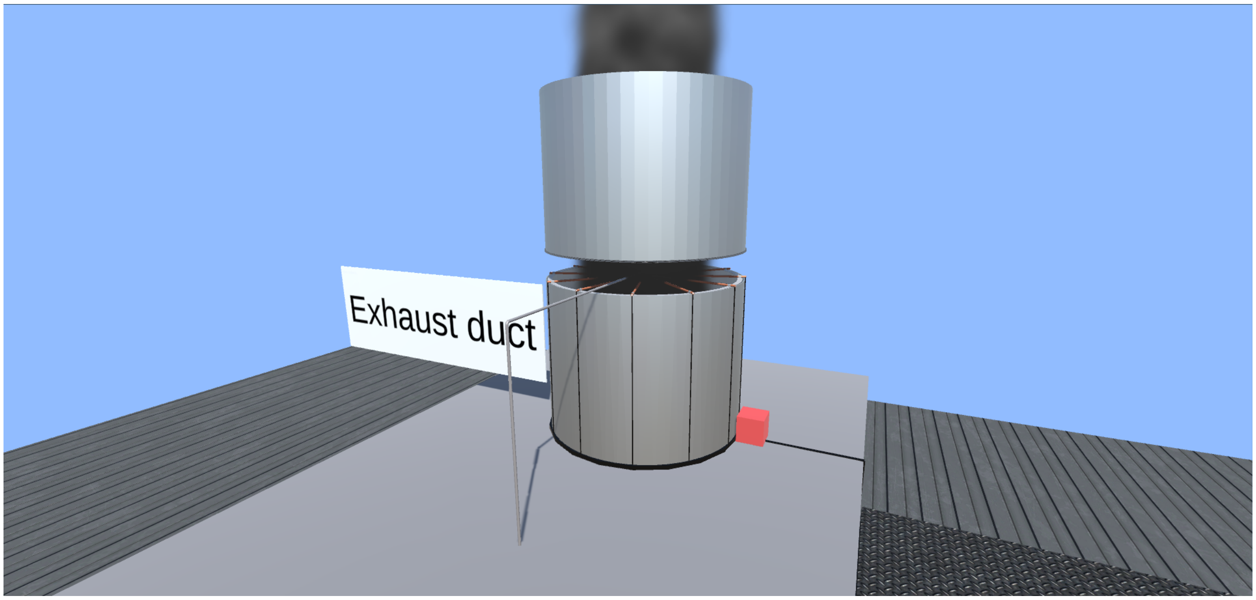


Figure C.9: Open exhaust pipe with all components

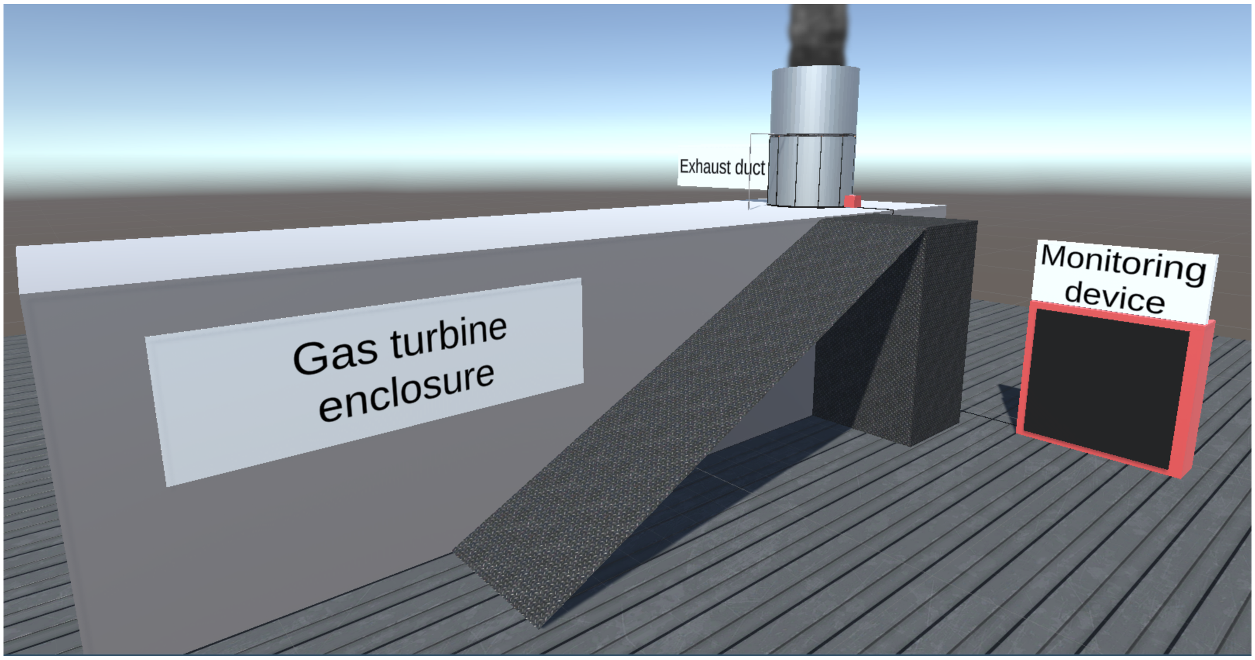


Figure C.10: Full system overview

The system visualization is yet to be completed as there are tasks remaining that would require us to complete the rest of the project. However this will be completed before the third presentation.

D. Appendix - electrical

D.1 Introduction

In this Appendix we are looking at the electrical system for the project. The electrical system is focusing on collecting data from temperature sensors for our concept. With this the systems presented in this appendix is two PLC systems and one microcontroller. The goal for the electrical domain is to present different systems that will monitor our sensors, and be for use for the client.

D.2 Preliminary Concept

The electrical system for our project, will have a lot of data inputs and send it out in real time for visualizing. With our electrical requirements the most important one is to be able to give a presentable measurement of the mass flow. To be able to do that, there is a need to finalize a system concept. During the brainstorming the group found a few systems that could work. The biggest constraints on the system is that all the sensors that are put inside of the exhausts outlet, need to withstand high temperature. This make it problematic because the sensors then need to be cooled down. With the need to cool down sensors, standard sensors used for our measurements are not fitting. For our concept the group have gone for one that don't need sensors on the inside of the gas turbine. In our concept there is going to be pipes with working fluid on the inside of the exhaust outlet, and measuring of the temperature of this liquid. The liquid is supposed to be measured with the input temperature and output temperature from the exhaust outline.

Team's electrical system are going to be made out from a microprocessor instead of using a PLC system. One shall then look at how much it will cost to produce this system versus how much a PLC system cost.

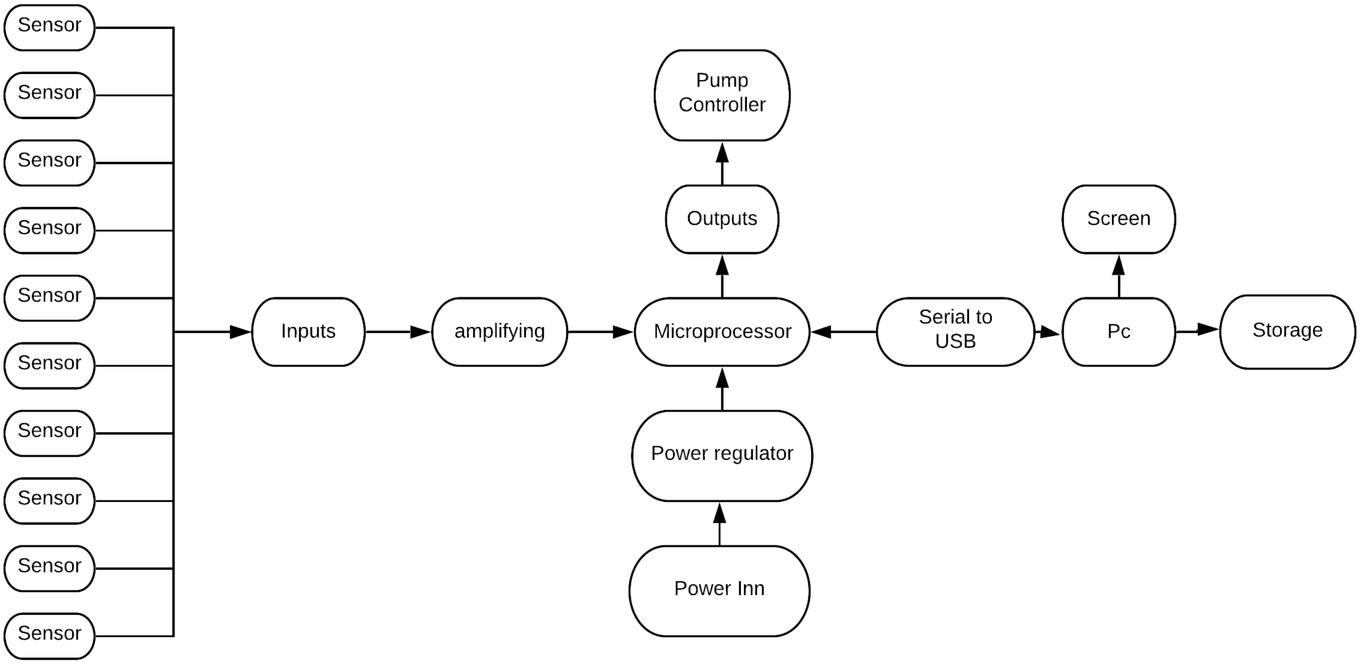


Figure D.1: Electrical concept block diagram

In figure D.1 there is a example on the main parts of the electrical system. The main part that connects everything is the microprocessor. For the processor it needs to have enough ports for our sensors and need to be able to process allot of sensor data in real time. Another constraint that our system have is it may need to be ATEX certified. This certification describe equipment used in a explosive atmosphere and add a design constraint when selecting components.

In the diagram, the electrical system have a significant amount of sensors. These sensors are temperature sensors. They can be thermistors or thermocouples, depending on what kind of fluid temperature they are going to measure. Thermistors and thermocuoples have both strengths and weakness. The choice will be taken at a late time when the electrical domain have gotten further understanding on the temperature measures that is needed.

These signals are then being processed. If one is using a thermocouple the signal needs to be amplified because the voltage change from the thermocouple is very low. The signal from the thermocouple goes through an amplifier, like the differential amplifier, and then are sent to the microprocessor. In the processor the signal needs to be interpret as a temperature. With these signals going in to a mathematical model that gives the representative mass flow from. In this point of time the electrical domain are going to look at the mathematical model as a black box. The mass flow measurement is then sent through our serial to USB, to our computer for visualizing.

The microcontroller is also controlling the fluid pump. For the control of this pump there can be used a PID controller. A PID controller(proportional-integral-derivative controller) is a controller that with three parameters will control and stabilize the pump. The response, overshoot and how long time the pump stabilize depend on the three parameters to each part of the PID. With using a controller the pump can be controlled to run at a chosen speed and pump as much liquid as it is needed.

To power the electrical system there is either going to be a battery solution, or connection to the power that are available on site. This need to be decided, and need to be addressed with the client, if there is possible to connect to a power source on site.

D.2.1 Proof of concept system

An early idea was to make a proof of concept model. This was imagined to be able to ensure that the concept is working. The model will be a simple one, where we test one pipe and see if team can get the correct readings. For the test system one can make a miniature model that can fit around a hair dryer. The hair dryer will then blow hot air at the pipe. The liquid in the pipe will then hopefully be heated up and the thermistors can measure the temperature change. The thermistors can be connected to the arduino. The arduino will translate the signal to temperature and control the pump.

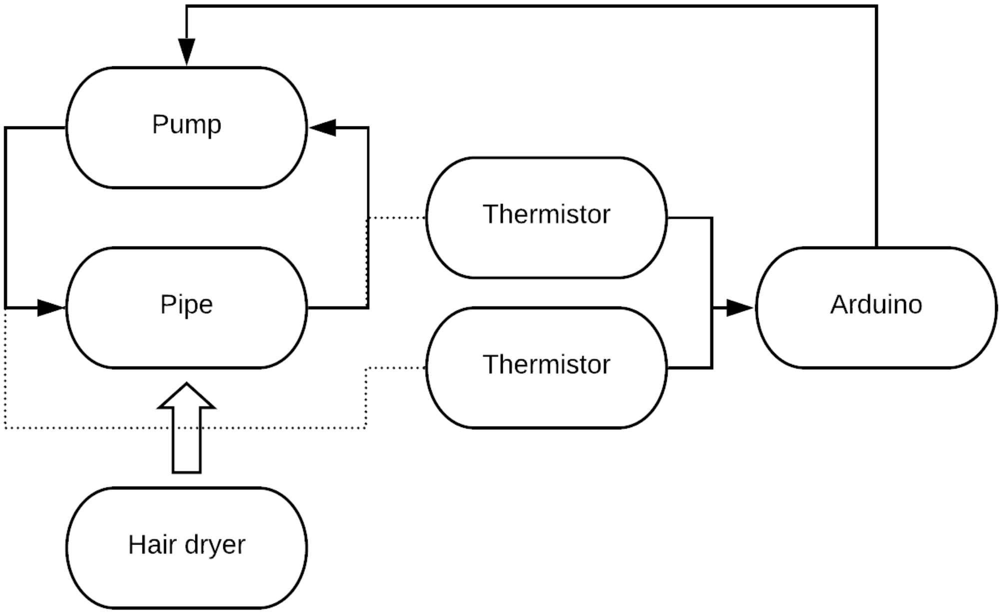


Figure D.2: Electrical proof of concept

The final proof of concept experiment can be found in appendix A.3.5.

D.3 Final concept

In the final concept of the electrical domain the goal is to collect data from temperature sensors. There is thirteen temperature sensors needed in this concept, and the data need to be processed and sent to software for visualizing. The thirteen sensors have one sensor to measure the temperature into the system, and twelve sensors measuring different output temperatures.

D.3.1 Sensors

For our system we are going to need to measure temperature. There exist multiple types of sensors that we can use for this purpose. Commonly used sensors for this purpose are thermocouple, thermistors and RTD (resistance temperature detectors). In our case the electrical domain are working with thermocouples.

With using thermocouples there is a need for signal conditioning. The signal that comes from the thermocouple are a low voltage. It needs to be amplified so that an ADC can distinguish between the different levels. Another problem with the thermocouple is that it needs to be cold junction compensated. That is because the thermocouple work with creating a potential difference across the junction [62]. If the reference junction temperature are unknown, the temperature measured are the difference in the temperature from reference junction to measure point.

AD8495 The chosen amplification for the circuit is to use the AD8495 thermocouple amplifier with cold junction compensation. The AD8495 amplifies the output signal to give 5mV/°C with using a instrumentation amplifier. With that the gain of the amplifier is 122,4. More information can be found in the datasheet in [63].

The circuit in figure D.3 consist of a thermocouple as the input connected to low pass filters. The function of the filters is to remove unwanted noise signals. With the resistor R1 as 1MΩ the thermocouple is grounded.

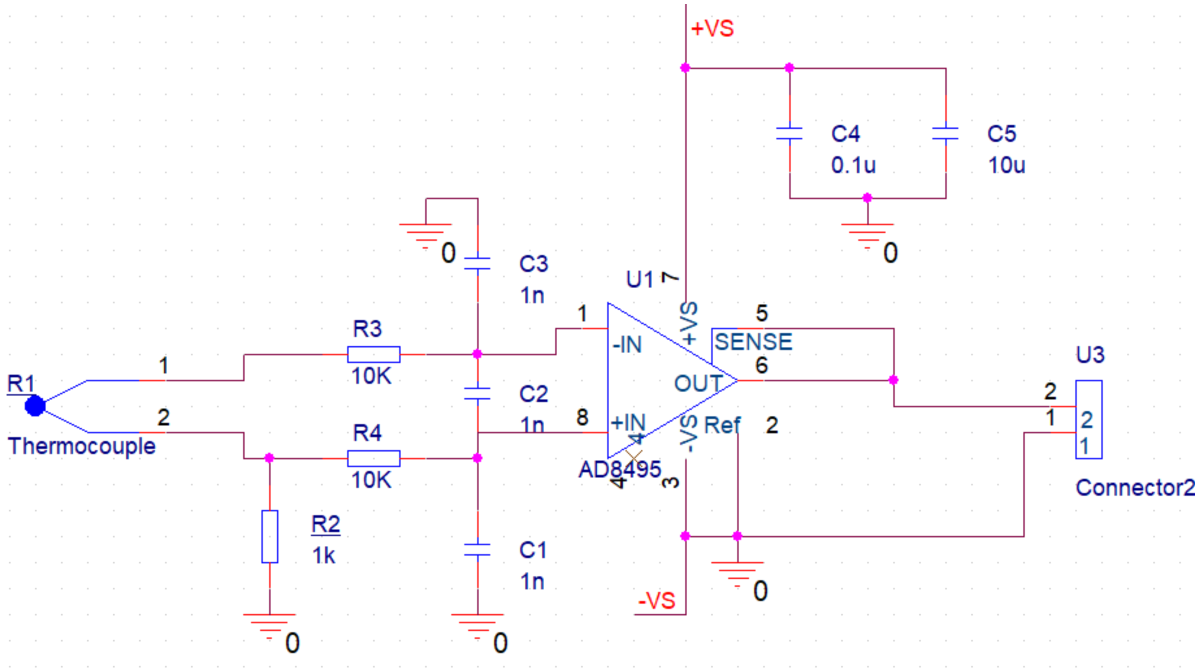


Figure D.3: AD8495 thermocouple amplifier with cold junction compensation

The low pass filters values are calculated with the formula D.1. This formula describes the filter frequency given from the resistor and the capacitor. C3 and C1 have the same value and are represented as C_C in the equation. C2 are represented as C_D and R3 and R4 are R.

- C3, C1 = C_C

- C2 = C_D
- R3, R4 = R

With the filter frequency, the goal is to not interfere with the thermocouple signal, but remove unwanted noise signals. To chose this frequency the equations D.1 and D.2 are used. In This circuit the electrical domain is looking at the filters that analog devices made in[64].

$$FilterFrequency_{DIFF} = \frac{1}{2\pi R(2C_D + C_C)} \tag{D.1}$$

$$FilterFrequency_{CM} = \frac{1}{2\pi R C_C} \tag{D.2}$$

Output signal: the output signal is described in equation D.3 as the temperature of the measurement times 5mV. That gives a 5mV change of each degrees of temperature. The output are then put into an analog to digital converter for a processor to convert it to temperature.

$$V_{OUT} = (T_{MJ} \times 5mV/^{\circ}C) + V_{REF} \tag{D.3}$$

From the microcontroller equation D.3 can be used to get the temperature. The temperature is given by equation D.4. With this equation the microcontroller have the temperature and can send it for visualizing.

$$T_{MJ} = \frac{V_{OUT}}{5mV/^{\circ}C)} + V_{REF} \tag{D.4}$$

In figure D.4 there is a design for a simple PCB board for the AD8495 chip. The design is build up from the circuit in figure D.3. For the footprints on the PCB standard through hole components are used for the resistors and capacitors. If it occurs that the filtering is not set to the right frequency, the components can be changed to test another filter frequency. With AD8495 being a smd (surface mount device) component the footprint was created after the defined size in the datasheet [63]. The thermocouple, power and output are connected with a terminal block as seen in [65]

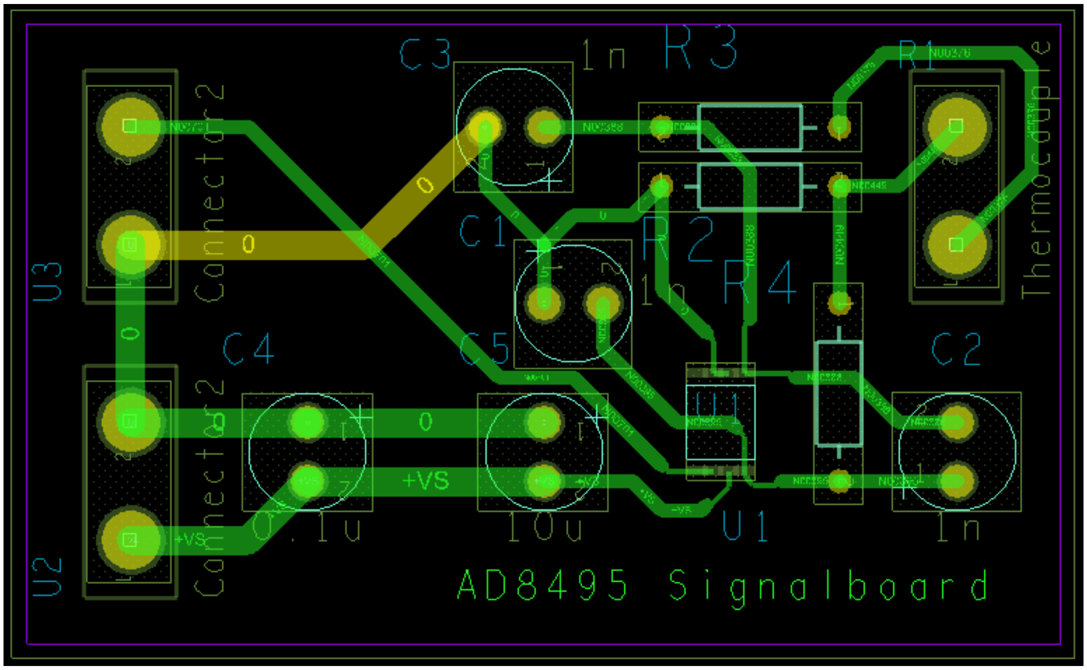


Figure D.4: AD8495 PCB layout

D.4 Arduino module

With the signal conditioning circuit made the next stage is to connect it to a microcontroller. In this chapter the electrical domain is looking into creating an Arduino shield. An Arduino shield is circuit that are put on top of an Arduino. The Arduino that are going to be used is the Arduino Mega 2560. [66] The reason for using the Arduino Mega is that it have 16 analog inputs, but it only have one ADC. With the ADC there is connected a MUX(multiplexer). That is one risk with this system, there is only possible to read one input at the time. Therefore, there is need for a short delay between each reading. The ADC have a resolution of 10 bits. With this resolution there is 1024 levels that can represent 1024 degrees with one degree per level.

From Arduino's site there is Eagle files available. The eagle files include schematic and PCB layout for the Arduino mega and can be converted to files supported by Orcad. In Orcad there is a setting to translate eagle files that import the files and convert them with all necessary files. From there the layout for the Arduino with footprints are available and can be used to map out the connectors that the shield board needs to connect too. In figure D.5.

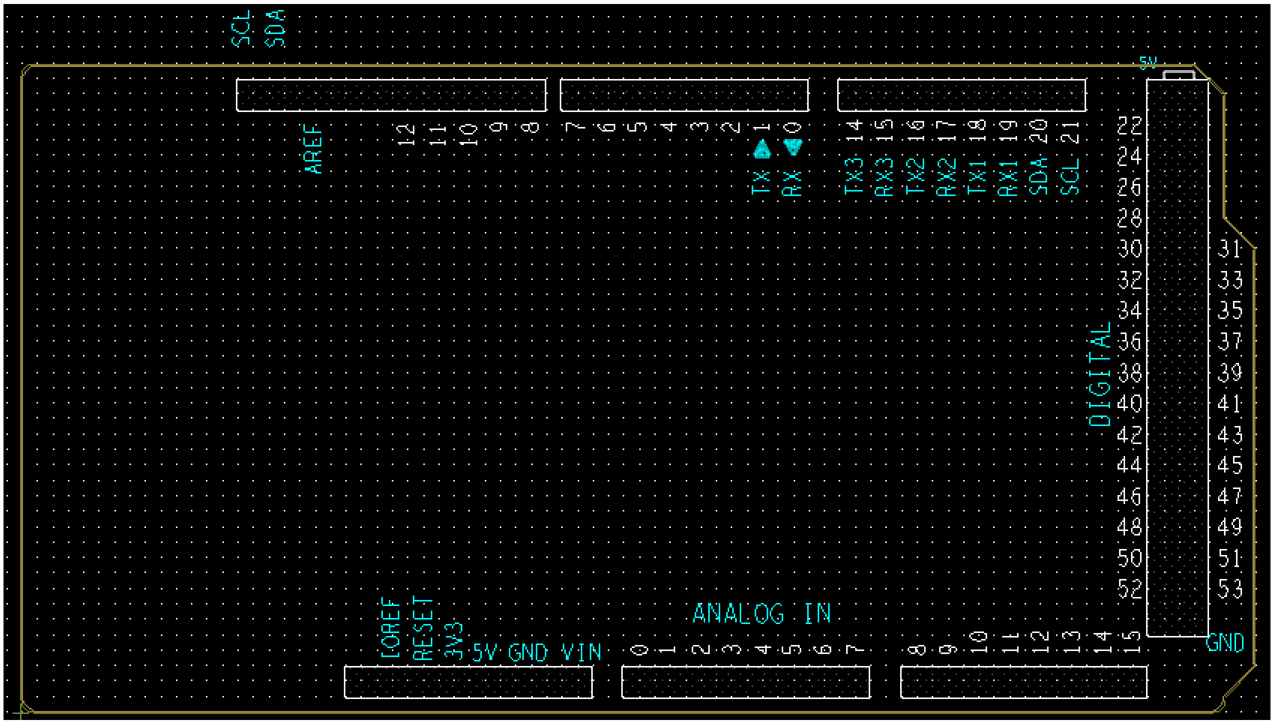


Figure D.5: Arduino Mega 2560 Pin outline

The outline of the pins is visible, but all the components and traces from the Arduino are removed. With that there is less likely to be mistakes when placing components that needs to be connected to the Arduino. For the footprints on the design there is used SMD (Surface mounted devices) components. The resistors and capacitors use 0805 footprint. and the tantalum capacitors uses the size 7343.

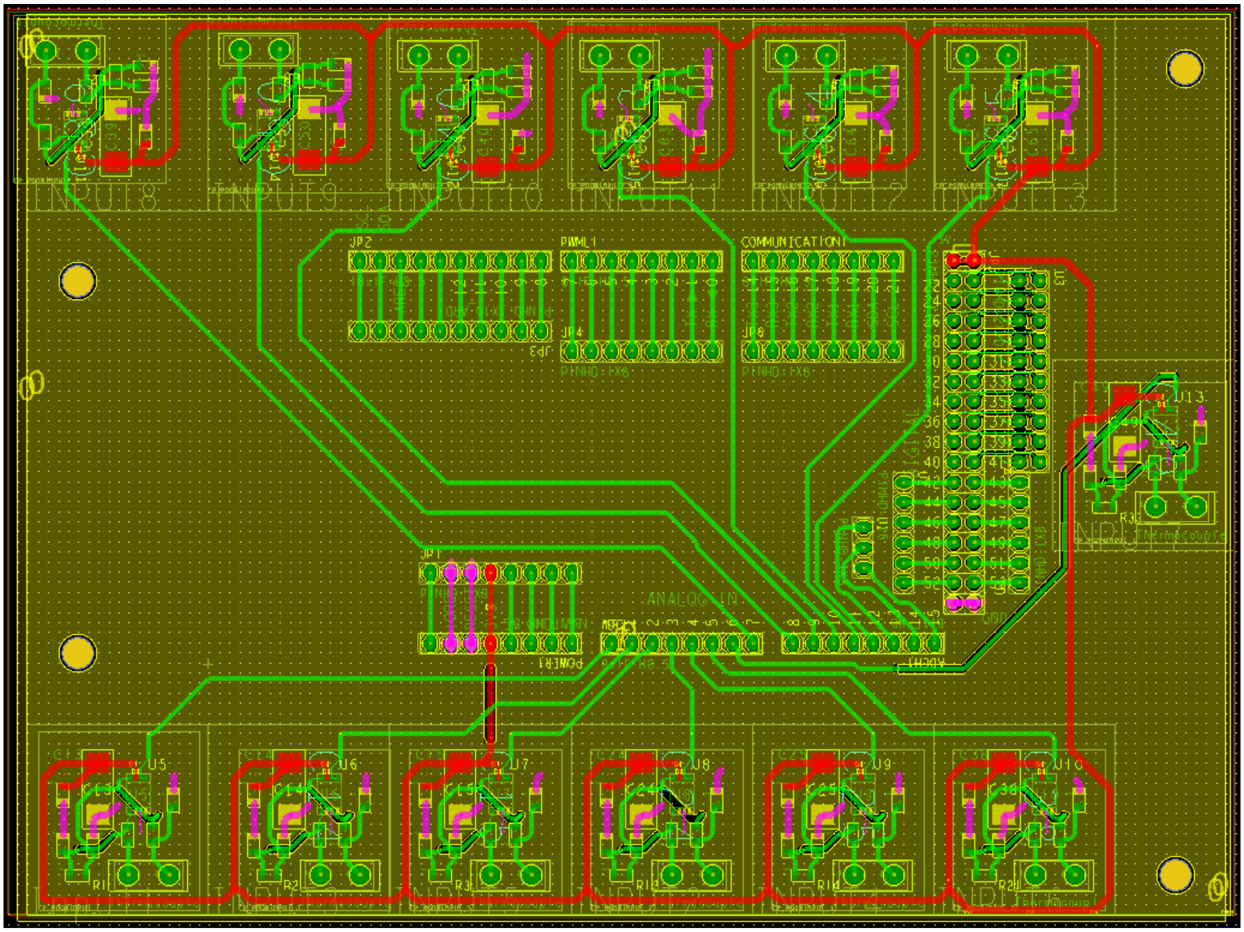


Figure D.6: Arduino Mega Thermocouple shield PCB layout

In figure D.6 the PCB layout for the circuit D.4.1. To not remove any functionality from the Arduino, the pins that are unused, are extended to new pins on the top of the PCB. The rest of the PCB are split under each of the thermocouple input circuits. In figure D.7 the PCB is shown in 3D view with models of the components.

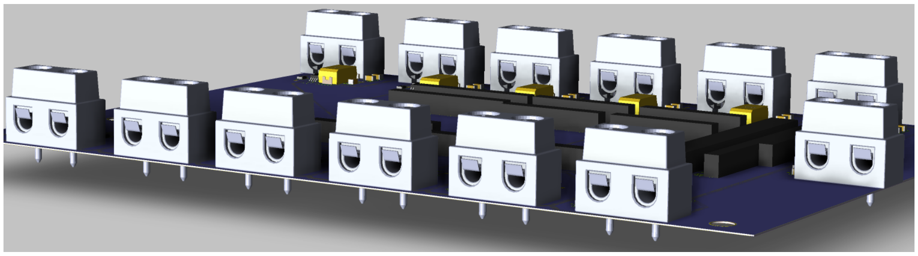
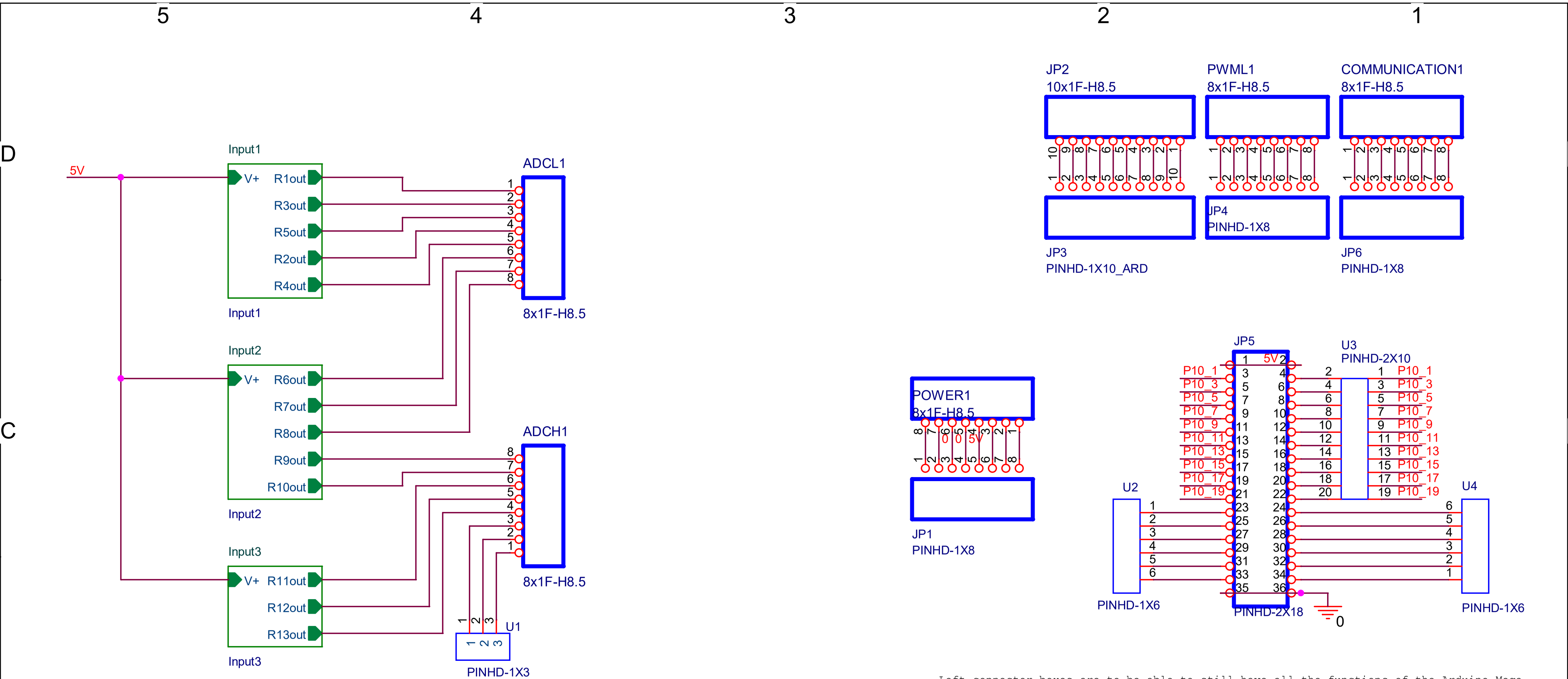


Figure D.7: Arduino Mega Thermocouple shield 3D view

D.4.1 Schematic



```
Left connector boxes are to be able to still have all the functions of the Arduino Mega.
They are meant to be extenders for the pins.

ADCH and ADCL are the analog inputs from the arduino that are connected to the input stages.
```

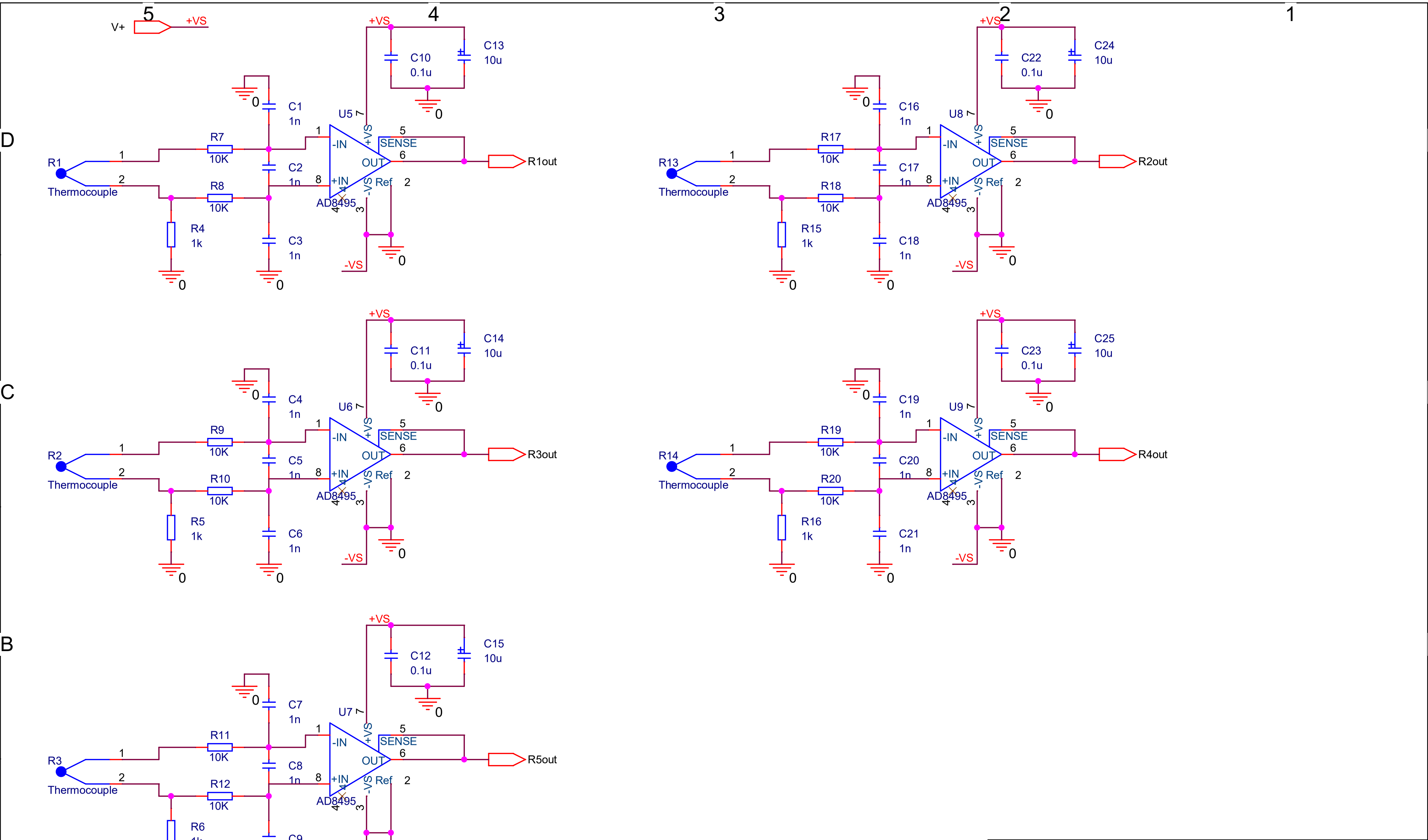
University of South-Eastern Norway

Stian Bergstrom
Postbox 235
3603 Kongsberg
Norway

Title
Arduino Thermocuple shield

Size A4
Document Number
Arduino connections
Rev ?

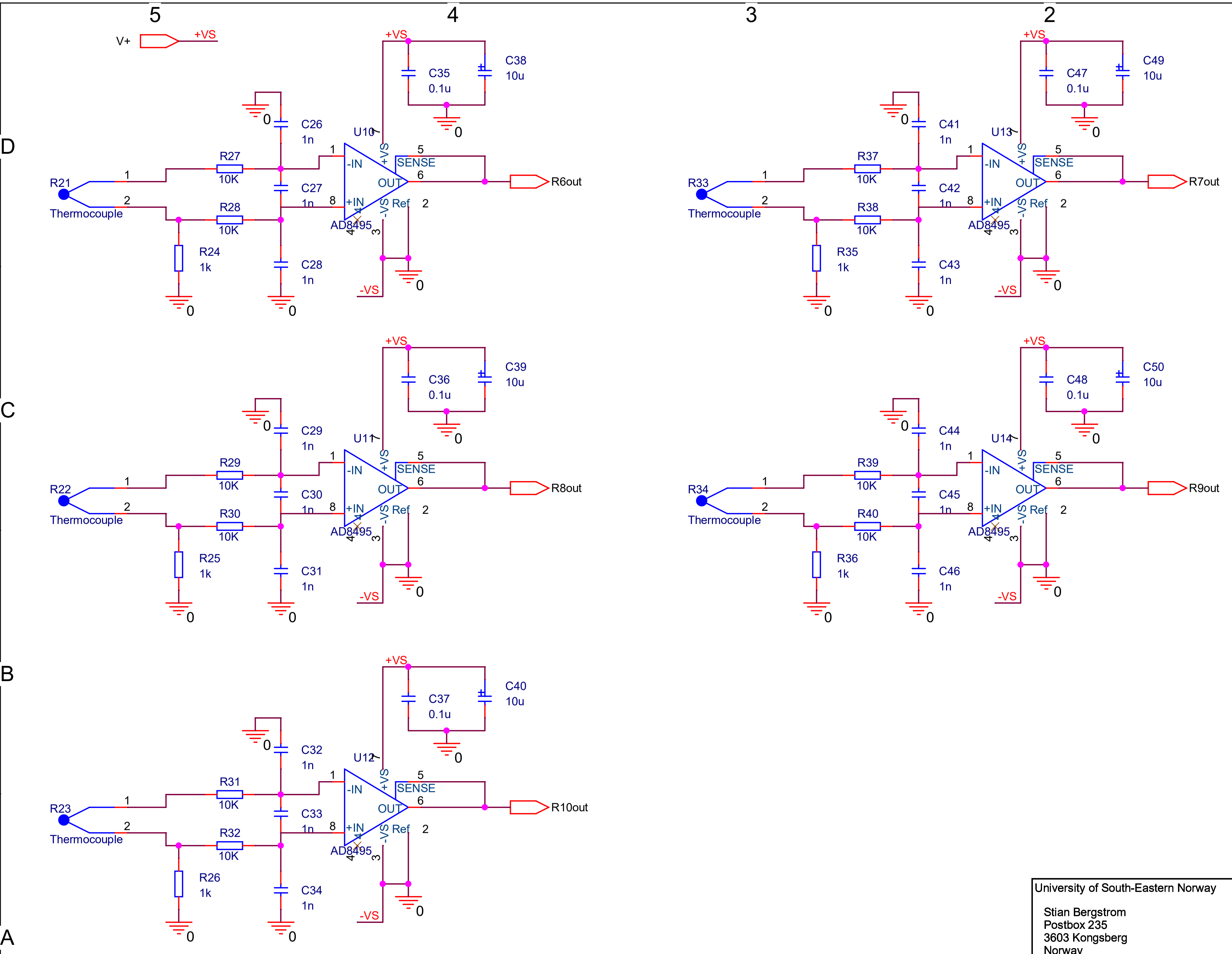
Date: Sunday, May 24, 2020
Sheet 1 of 4



University of South-Eastern Norway

Stian Bergstrom
Postbox 235
3603 Kongsberg
Norway

Title	Arduino Thermocouple shield	
Size A4	Document Number Input1	Rev ?
Date: Sunday, May 24, 2020	Sheet 2 of 4	



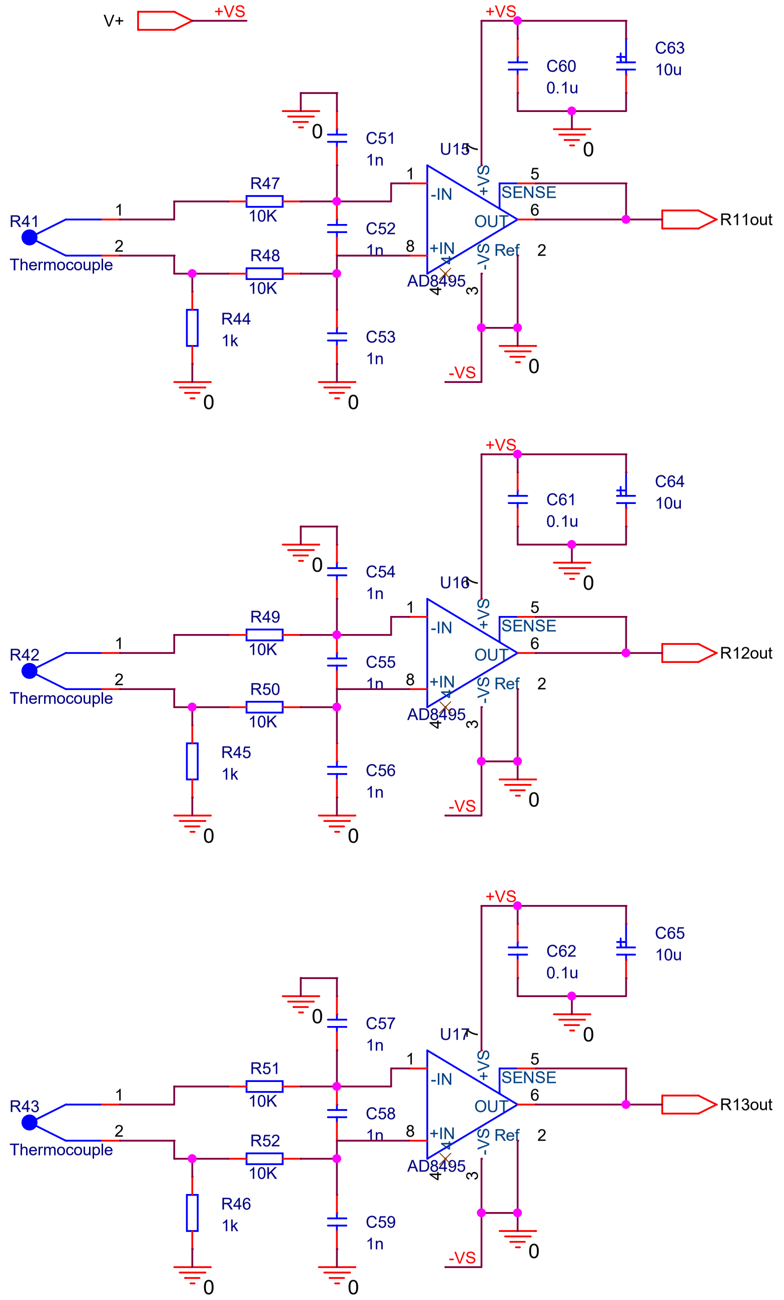
University of South-Eastern Norway

Stian Bergstrom
Postbox 235
3603 Kongsberg
Norway

Title
Arduino Thermocuple shield

Size	Document Number	Rev
A4	Input2	?

Date: Sunday, May 24, 2020 Sheet 3 of 4



University of South-Eastern Norway

Stian Bergstrom
Postbox 235
3603 Kongsberg
Norway

Title
Arduino Thermocuple shield

Size	Document Number	Rev
A4	Input3	?

Date: Sunday, May 24, 2020 Sheet 4 of 4

D.5 ATEX

One of the requirements for our system is that it needs to be ATEX certified. ATEX certification is a certification for an explosive atmosphere. ATEX is a certification made by EU and are split into three zones. These zones describes what conditions the equipment needs to handle. [67]

With the ATEX directive there is different requirements and demands for what situation the equipment are under. There is defined three different zones that describes what amount of hazardous gas there is, and how often there is gas present. In the section under there is described the three different zones for hazardous gas. After communication with the client, the equipment needs to be Ex zone 2 certified at least. It can strive to be in zone 1. [68]

Zone 0:
In zone 0 there is an exposure of an explosive atmosphere continuously or with long periods with high frequency.

Zone 1:
In zone 1 there is likely to occur an explosive atmosphere under normal circumstances.

Zone 2:
In zone 2 there is unlikely that an explosive atmosphere will take place. In the event of it happening it will be present for a short period.

D.6 PLC systems

On gas turbines the control system that are used in PLC(programmable logic controllers) systems.

For our client the manufactures for PLC modules are Allen Bradley and Siemens. When working on this tasks the electrical domain have looked into Siemens PLC systems. With the selection of the units Siemens TIA (Totally Integrated Automation) selection tool[69] has been used. After finding this platform the selection of modules become less troublesome than reading the product catalog. When finding the platform to build the system out from only the supported modules to this platform was available.

For the concept the system have thirteen thermocouples. Each of these sensors need to be processed and sent to software. With this the electrical system needs thirteen analog inputs that support thermocouples. There is also a need for to be able to communicate to send the data for the software. Each of the platform under have built in Ethernet communication.

With these requirement the electrical domain have worked on two PLC systems. There is two different systems with one advanced platform and one more basic platform. The two systems that the electrical domain are going to look into are the Siemens Simatic s7-1500 and Simatic s7-1200. Simatic s7-1500 is a more advanced platform with more functions to work upon. There is possible to use another CPU to be able to program in high level languages.

D.6.1 Simatic S7-1500

The first system that the electrical domain have looked into is the Simatic S7-1500. This is a advanced control from Siemens[70] which is compatible with allot of modules. One downside for this system is that it has relatively high purchasing cost. Therefore, it is not the most cost efficient solution.

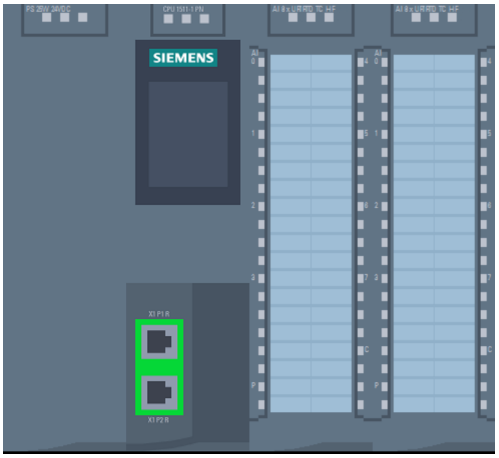


Figure D.8: Simatic1500 Graphic

Module type	Tittel	Article number
CPU	CPU 1511-1 PN	6ES7511-1AK02-0AB0
Thermocouple module	Analog input, AI 8xU/R/RTD/TC HF	6ES7531-7PF00-0AB0
Power supply	System power supply, PS 25W 24V DC	6ES7505-0KA00-0AB0

Table D.1: Simatic s7-1500 modules

CPU: The Simatic s7-1500 supports a good amount of processors. With 1511-1 PN which are meant for Small to medium application, it suits the electrical system. The processor is a standard processor and support up to four analog inputs.

ANALOG MODULE: The analog module that are chosen is a eight channel analog input that support thermocouples. This analog input have all integrated mathematics with temperature compensation. To be able to read all the sensors there is a need for two modules since there is not enough inputs on only one. The analog input in D.1 are chosen with the use of TIA selection tool compare mode.

Power Supply: With the analog modules and the CPU there is need of a power supply. The chosen power supply for this platform is a 24V, 25W supply. It is connected to the modules by the backplane Bus. There is a low amount of modules that are connected on the system that leads to not needing a high amount of Watts.

The cost of the system modules is around 30 000 NOK. The cost estimate are gathered from various third party retailers that have the pricing shown and may not be accurate when looking at prices directly from Siemens. It has a high purchasing system, but it can deliver high functionality.

D.6.2 Simatic S7-1200

The other system that the electrical domain shall present is based on the Simatic S7-1200. It is a basic controller that is mean to be used for low and middle performance range. It is not as powerful as the s7-1500, however it is priced lower. In comparison to the Simatic S7-1500, the Simatic s7-1200 is cheaper and not as powerful. With that in mind and that the process for our system is to measure temperature, there is not a need for the highest processing power. The s7-1500 becomes to expensive for the benefits it delivers in this system.

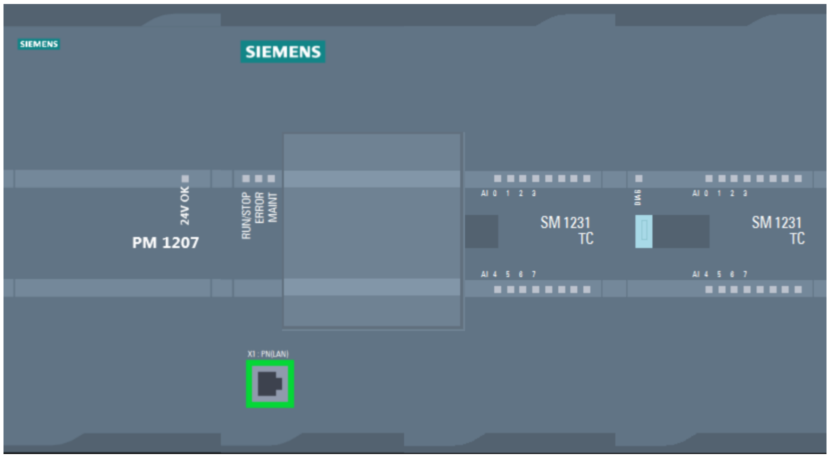


Figure D.9: Simatic1200 Graphic

Module type	**Tittel**	**Article number**
CPU	CPU 1212C	6ES7212-1AE40-0XB0
Thermocouple module	Analog input, SM 1231 TC, 8 AI thermocouples	6ES7231-5QF32-0XB0
Power supply	Power supply PM1207	6EP1332-1SH71

Table D.2: Simatic s7-1200 modules

CPU: The CPU for the Simatic s7-1200 is 1212C. With this CPU there is support to connect two signal modules that are the analog input modules. For our system that is sufficient and there is not that big of a difference on performance on the different processors.

ANALOG MODULE: For the Simatic s7-1200 the analog inputs are similar to the Simatic s7-1500. With one module available to chose from, that supports thermocouples, there was not much of a choice to be made. The module is an eight channel input module which means that there is a need for two modules to support thirteen thermocouples.

Power Supply: The power supply for the system is supplying the system with 24V and is the only available choice from the TIA selection tool.

With the Simatic s7-1200 being a basic controller platform, it is cheaper than the Simatic s7-1500. The cost estimate for the modules are taken from RS components prices. In table D.3 there are the prices for this system.

Module type	**Price**	**Rs stock number**
CPU	2 372,86Kr	862-4465
Thermocouple module	4 297,07Kr×2	810-4150
Power supply	737,78Kr	668-0554
Total	11704,78Kr	

Table D.3: Simatic s7-1200 prices from Rs components

E. Appendix - test documentation

E.1 Mechanical test documents

E.1.1 Tests for existing instrumentation solutions

Test ID	MT.1
Traceability	M.1
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Compare maximum temperature it is capable to withstand to one listed in the requirement.
Result Description	It is stated in documentation that the test duration is expected to be 4 hours, where average temperature experienced by the sensor will be $550°C$. Dantec dynamics provides equipment that is capable to withstand $700°C$ and is for repeatable use. (It can be re-calibrated and used several times).
Status	Passed
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.2
Traceability	M.1
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage id dialog with suppliers.
Result Description	It has been found that it is possible to purchase a prove from Airflow Sciences Equipment that would withstand required temperatures during time range of the test. However additional water-cooling (can be salt water) system is needed. This possibly will cause difficulties in offshore applications.
Status	Passed
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.3
Traceability	M.1
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method, evaluate.
Result Description	Has been proven to work for combustion processes in aerospace application. However requires carefully design shielding structures.
Status	Passed
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.4
Traceability	M.2
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Make assumptions based on existence of portable test kits for velocity data acquisition systems.
Result Description	Similar sensor by *Airflow Sciences Equipment* is possible to use a part of a portable test-kit, therefore making assumption that it would be possible to use hot-wire probe the same way.
Status	Pass
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.5
Traceability	M.2
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers.
Result Description	3DDAS data acquisition system can be supplied in a form of a complete test kit, that is portable an requires only one operator. Does only require an opening in the duct for insertion of the probe. Calibration however has to be done off-cite with special equipment.
Status	Pass
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.6
Traceability	M.2
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method, interview an external supervisor that used this technique in their research (Elisabet Syverud)
Result Description	After gathering all available information ([24],[6],[22]) and discussing the results obtained with external expert, the understanding is as such: PIV is a highly scientific laboratory method of investigation that may require days to build a testing rig.
Status	Failed
Test Performed	18.03.2020
Test Responsible	Iversen

Test ID	MT.7
Traceability	M.3
Test Method	Inspection, Analysis
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. See if data provided includes parameters that are possible to derive from CFD. Check if the sensor has high enough accuracy.
Result Description	Instruments parameters are compatible with current experiment set up. Until more up to date CFD report is provided the test is on hold.
Status	In progress
Test Performed	21.03.2020 - to date
Test Responsible	Iversen

Test ID	MT.8
Traceability	M.3
Test Method	Inspection, Analysis
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers.
Result Description	Instruments parameters are compatible with current experiment set up. Until more up to date CFD report is provided the test is on hold.
Status	In progress
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.9
Traceability	M.3
Test Method	Inspection, Analysis
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method, evaluate.
Result Description	Instruments parameters are compatible with current experiment set up. Until more up to date CFD report is provided the test is on hold.
Status	In process
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.10
Traceability	M.5
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Make assumptions based on existence of portable test kits for velocity data acquisition systems.
Result Description	Similar sensor by *Airflow Sciences Equipment* is possible to use as a part of a portable test-kit, therefore making assumption that it would be possible to use hot-wire probe the same way.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.11
Traceability	M.5
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers.
Result Description	After describing environment of both testing facility and oil platform to the supplier, they stated that their system is qualified to be used in both environments. However, there is not any portable computer that has been approved by the client that could be used off-shore. Therefore data treatment system shall be redesigned.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.12
Traceability	M.5
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method, interview an external supervisor that used this technique in their research (Elisabet Syverud)
Result Description	After gathering all available information ([24],[6],[22]) and discussing the results obtained with external expert, the understanding is as such: method is not possible to implement neither at testing facility in Drammen, nor at the final destination of the product.
Status	Failed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.13
Traceability	M.7
Test Method	Inspection, Analysis
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Compare the measured resolution to the CFD plot dimensions. Elaborate.
Result Description	Maximum measurable dimensions are of fractions of the millimeter. To cover a 2 m in diameter section the multi-probe assembly is required. To date there is no research found on an algorithm of placement of sensors for high temperature fluid motion. Otherwise if used downscaled low-temp model, log-Chebyshev method is applied. Combined purchase and calibration cost of a multi-sensor assembly is not justified.
Status	Failed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.14
Traceability	M.7
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers. Elaborate
Result Description	Maximum measurable dimensions are of fractions of the millimeter. To cover a 2 m in diameter section the multi-probe assembly is required. To date there is no research found on an algorithm of placement of sensors for high temperature fluid motion. Otherwise if used downscaled low-temp model, log-Chebyshev method is applied. Combined purchase and calibration cost of a multi-sensor assembly is not justified.
Status	Failed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.15
Traceability	M.7
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method. Compare the measured resolution to the CFD plot dimensions. Elaborate.
Result Description	PIV is a unique method that has the widest spatial resolution of three compared commercially available methods in this paper. With great assurance one can say that it can create an experimental model that would match almost any CFD analysis, since minimal scale of PIV is in a range of particle size (micro scale), and maximum scale corresponds to the size of a beam of light (macro scale).
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.16
Traceability	M.8
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Compare probe's dimensions to exhaust duct and flexible dimensions. Elaborate.
Result Description	Diameter of a probe is 45 mm, length of a pole is up to 2,2 m. Possible to rise flexible up to 50 mm. Diameter of the duct 2 m. Hence, probe fits. In theory for steady state fluid one probe can cover all points in cross section due to the feeding pole.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.17
Traceability	M.8
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers. Elaborate
Result Description	Diameter of a probe is up to 50,8 mm, length of a pole is up to 4,3 m. Possible to rise flexible up to 50 mm. Diameter of the duct 2 m. Hence, probe fits. In theory for steady state fluid one probe can cover all points in cross section due to the feeding pole.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.18
Traceability	M.8
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method. Compare probe's dimensions to exhaust duct and flexible dimensions. Elaborate.
Result Description	
Status	In progress
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.19
Traceability	M.9
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read data sheet for heavy-duty hot wire sensor provided by Dantec Dynamics. Check if the sensor is capable of measuring three components of the velocity.
Result Description	Three-wire heavy-duty high temperature gold plated probe by Dantec Dynamics is capable of measuring three components of velocity in a range of 70° flux.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.20
Traceability	M.9
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation and data spread-sheet for 3DP. Engage in dialog with suppliers. Elaborate
Result Description	3D probe has been designed by Airflow Sciences Equipment specifically for pressure based experiments on exhaust flows in 3 dimensions
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

Test ID	MT.21
Traceability	M.9
Test Method	Inspection
Priority	High
Test Iteration	1
Test Description	Read documentation available on PIV method. Check if the sensor is capable of measuring three components of the velocity.
Result Description	It has been found that this method originally was specifically designed as a data acquisition system for velocity vector field.
Status	Passed
Test Performed	21.03.2020
Test Responsible	Iversen

E.2 Software test documents

E.2.1 Construction 2

Test ID	ST.1
Traceability	S.1
Test Method	Demonstration
Priority	High
Test Iteration	#1
Test Description	Run application that writes data to file
Result Description	Software has not reached this stage.
Status	Fail
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.2
Traceability	S.2
Test Method	Inspection
Priority	High
Test Iteration	#1
Test Description	Test readability of stored data
Result Description	Software has not reached this stage.
Status	Fail
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.3
Traceability	S.3
Test Method	Instrumented
Priority	High
Test Iteration	#1
Test Description	Connect sensor to device to read input
Result Description	This will not be possible due to Covid-19
Status	Fail
Test Performed	
Test Responsible	

Test ID	ST.4
Traceability	S.4
Test Method	Demonstration
Priority	High
Test Iteration	#1
Test Description	Start/stop the system
Result Description	Software starts when start was pressed and goes back to the initial screen when stop is pressed.
Status	Pass
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.5
Traceability	S.5, S.6, S.7, S.8, S.9
Test Method	Demonstration
Priority	High
Test Iteration	#1
Test Description	Run software with incoming data to verify visualization methods.
Result Description	The application visualizes the values created within the software, and displays it with graphs and display fields. It updates sensor values at a given frequency. The application does not display values in figures. Currently the data is created within this software and not read from files.
Requirement Pass	N/A
Requirement Progress	S.5, S.6, S.7, S.9
Requirement Fail	S.8
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.6
Traceability	S.10, S.11
Test Method	Inspection
Priority	High
Test Iteration	#1
Test Description	Validate test result.
Result Description	Software has not reached this stage.
Requirement Pass	N/A
Requirement Progress	N/A
Requirement Fail	S.10, S.11
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.7
Traceability	S.12, S.13
Test Method	Demonstration
Priority	Medium
Test Iteration	#1
Test Description	While running software, compare visualization to CFD.
Result Description	Software has not reached this stage.
Requirement Pass	N/A
Requirement Progress	N/A
Requirement Fail	S.12, S.13
Test Performed	01.05.2020
Test Responsible	Ole Henrik and Helge Sondre

E.2.2 Construction 3

Test ID	ST.1
Traceability	S.1
Test Method	Demonstration
Priority	High
Test Iteration	#2
Test Description	Run application that writes data to file
Result Description	When ”Store data” is pressed the software creates a table in the database. Table name is the location input and the operator gets a column in the table. All sensors gets automatically one column and sensor information is inserted to the corresponding columns.
Status	Pass
Test Performed	15.05.2020
Test Responsible	Ole Henrik and Sondre

Test ID	ST.2
Traceability	S.2
Test Method	Inspection
Priority	High
Test Iteration	#2
Test Description	Test readability of stored data
Result Description	The readability of the stored data can be improved, but is at an acceptable stage.
Status	Pass
Test Performed	15.05.2020
Test Responsible	Ole Henrik and Sondre

Test ID	ST.5
Traceability	S.5, S.6, S.7, S.8, S.9
Test Method	Demonstration
Priority	High
Test Iteration	#2
Test Description	Run software with incoming data to verify visualization methods.
Result Description	The application reads data from a database, and visualizes correctly according to requirements.
Requirement Pass	S.5, S.6, S.7, S.8, S.9
Requirement Progress	N/A
Requirement Fail	N/A
Test Performed	15.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.6
Traceability	S.10, S.11
Test Method	Inspection
Priority	High
Test Iteration	#2
Test Description	Validate test result.
Result Description	Currently the data is stored at the computer because there is no physical storage device. The application does not store data according to high and low velocity.
Requirement Pass	S.10
Requirement Progress	N/A
Requirement Fail	S.11
Test Performed	15.05.2020
Test Responsible	Ole Henrik and Helge Sondre

Test ID	ST.7
Traceability	S.12, S.13
Test Method	Demonstration
Priority	Medium
Test Iteration	#2
Test Description	While running software, compare visualization to CFD.
Result Description	The software visualizes data with the same color coding as the CFD, but the resolution is low compared to CFD.
Requirement Pass	S.12, S.13
Requirement Progress	N/A
Requirement Fail	N/A
Test Performed	15.05.2020
Test Responsible	Ole Henrik and Helge Sondre

F. Appendix - figures and models

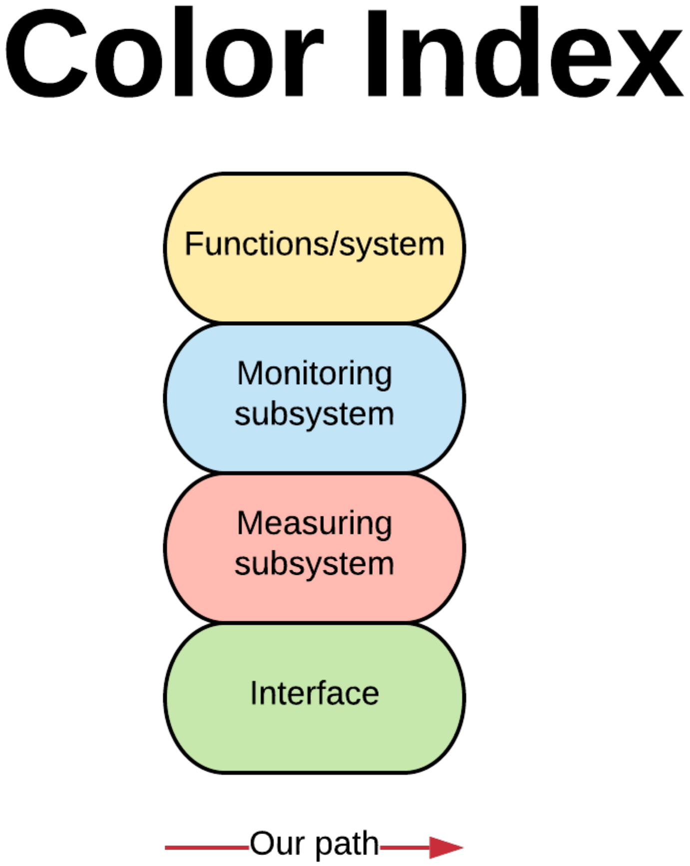


Figure F.1: Color Index

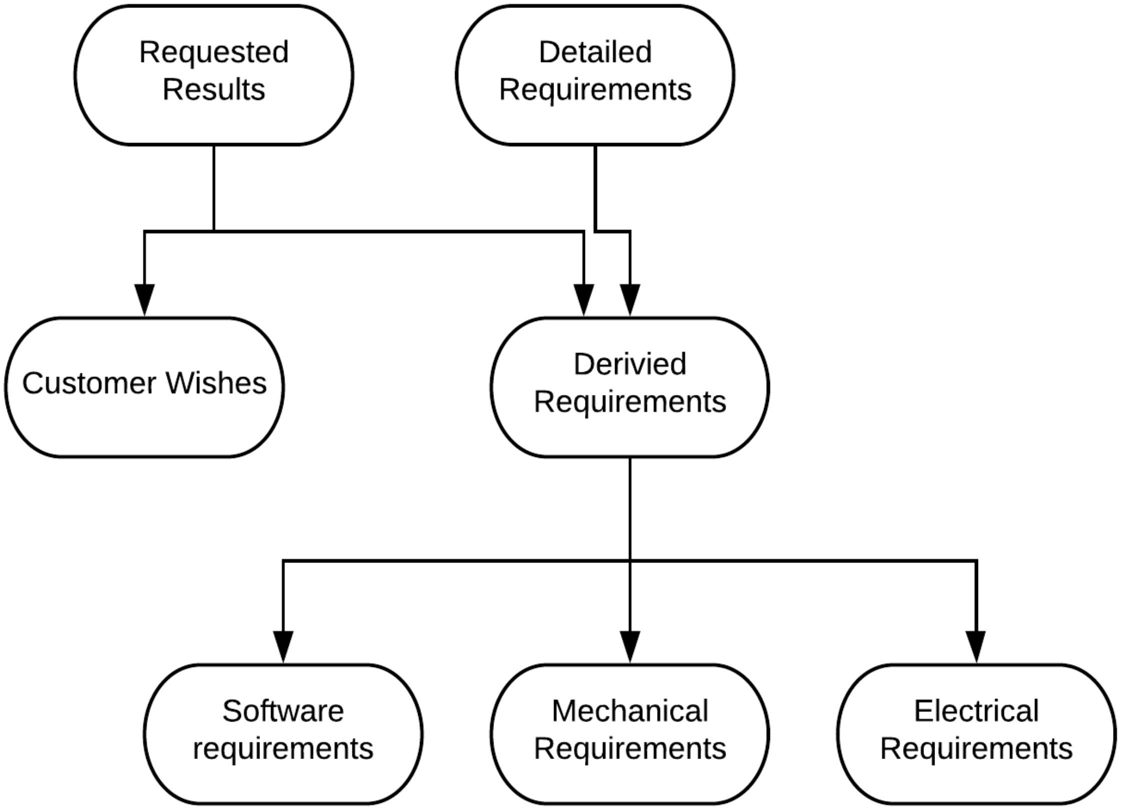


Figure F.2: Requirements Hierarchy

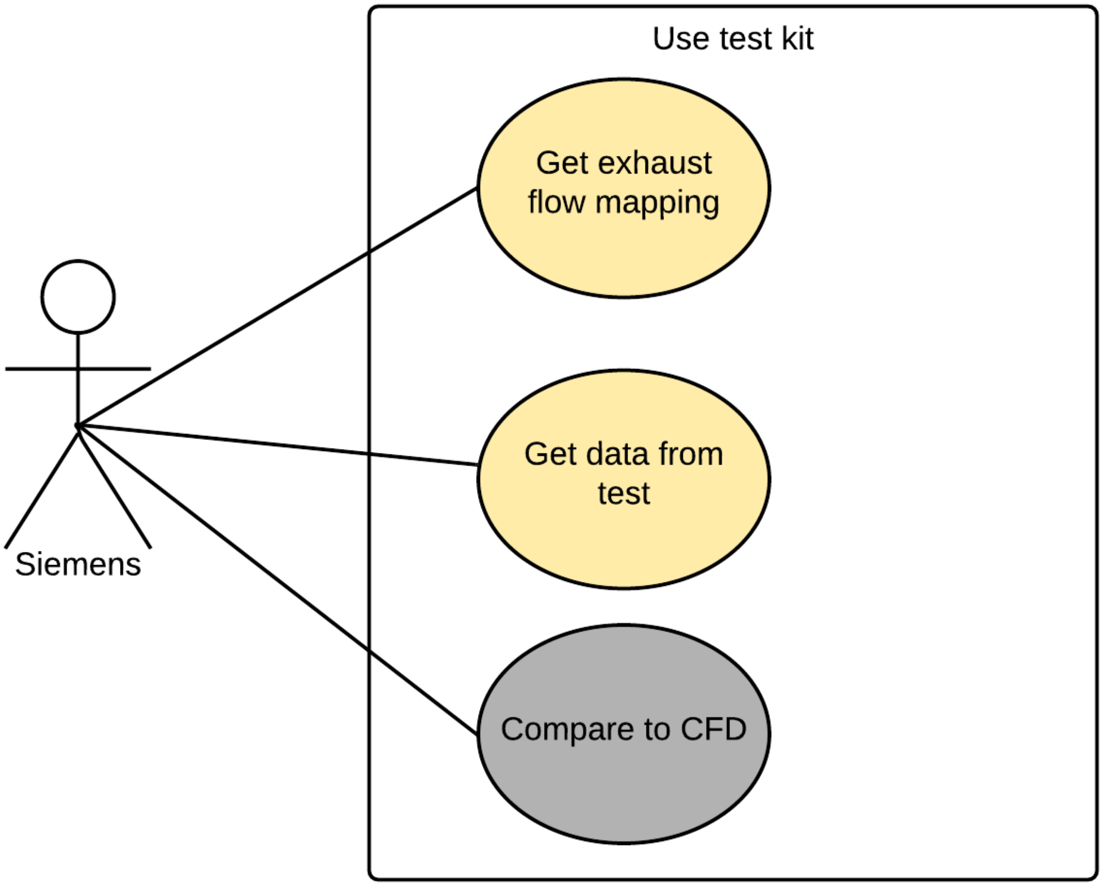


Figure F.3: UI test kit use case

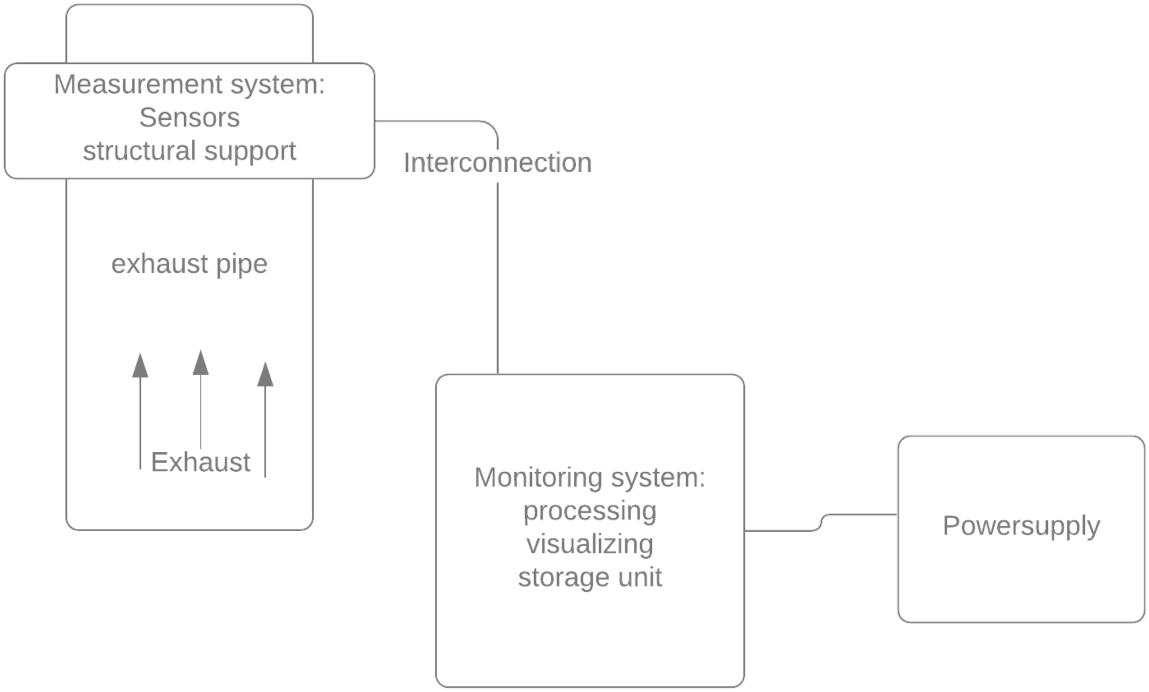


Figure F.4: Concept of operation

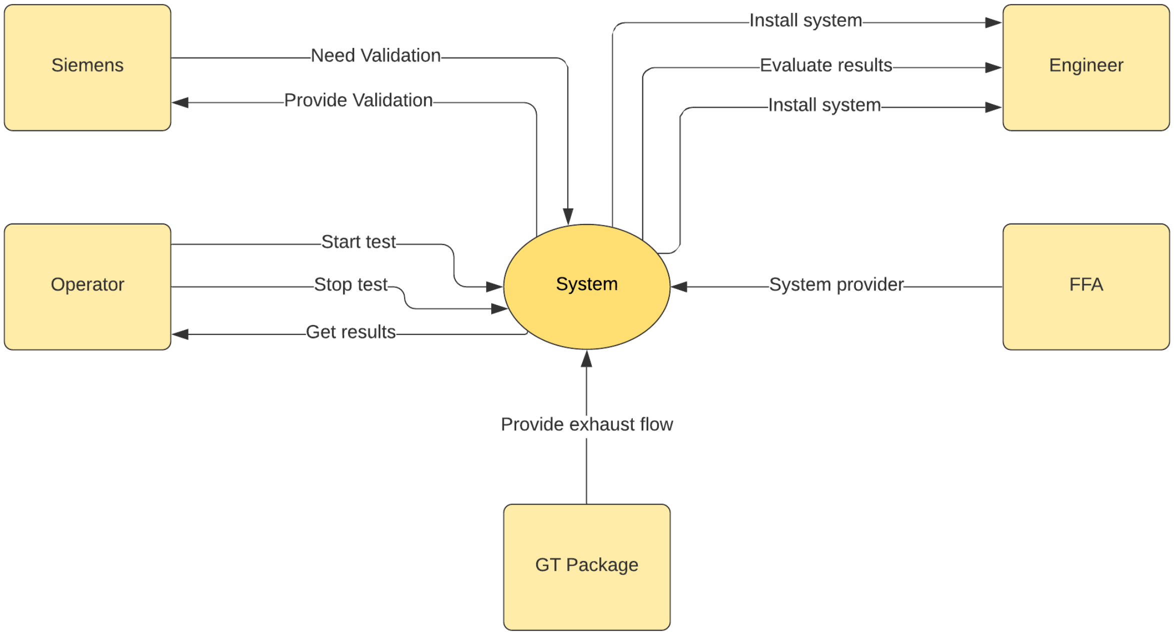


Figure F.5: Actors Context Diagram

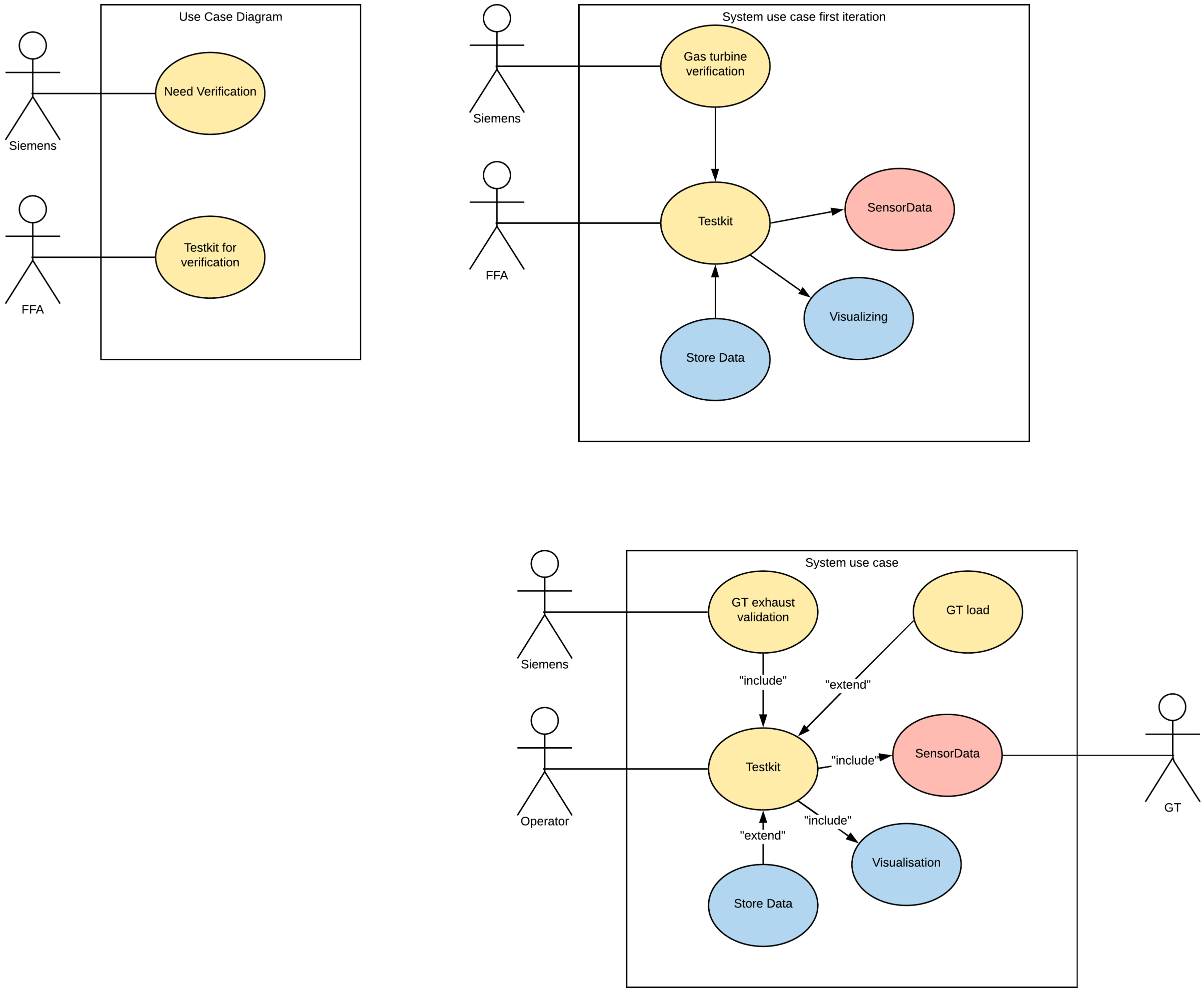


Figure F.6: Use case StakeHolder

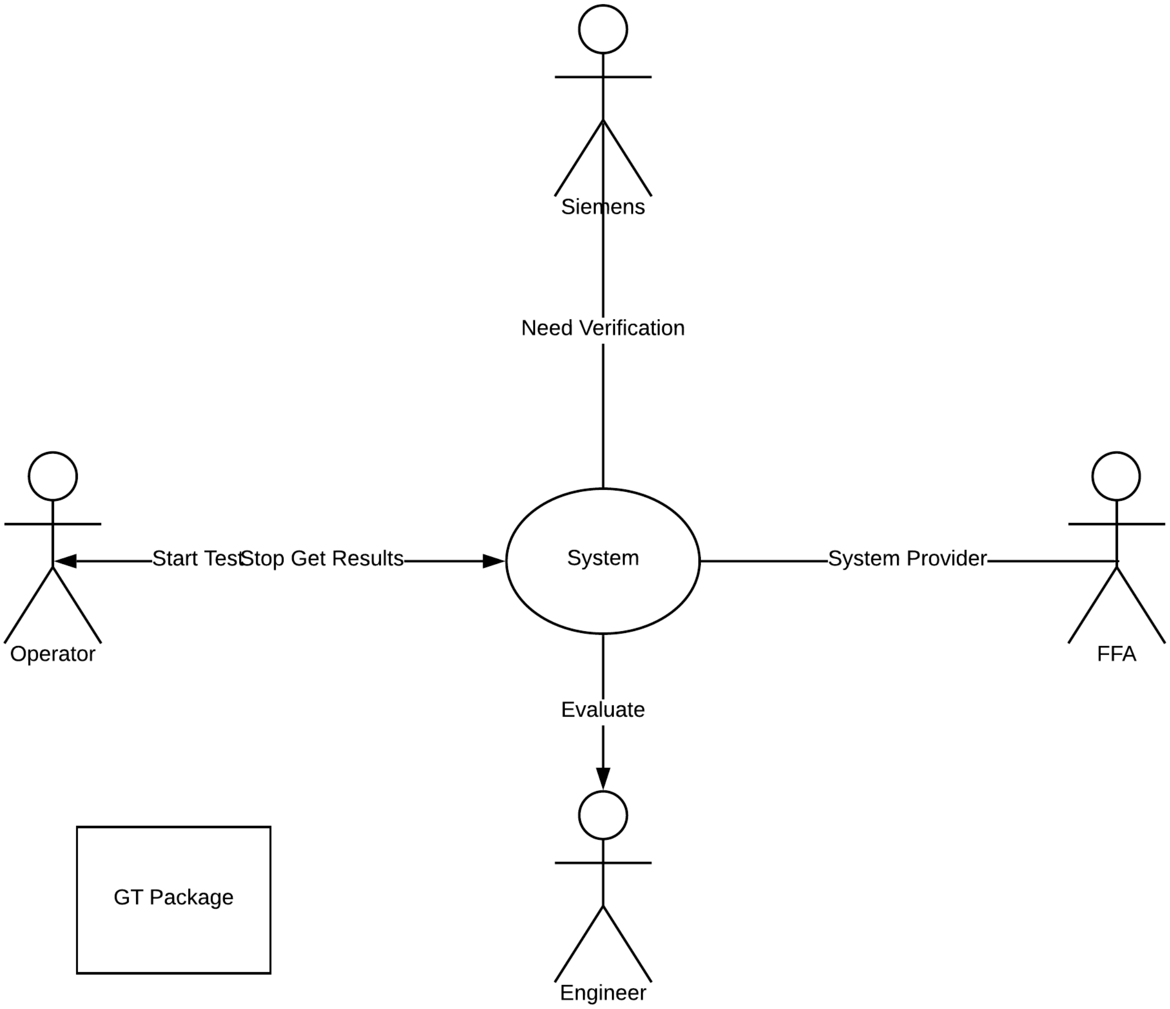


Figure F.7: Actors

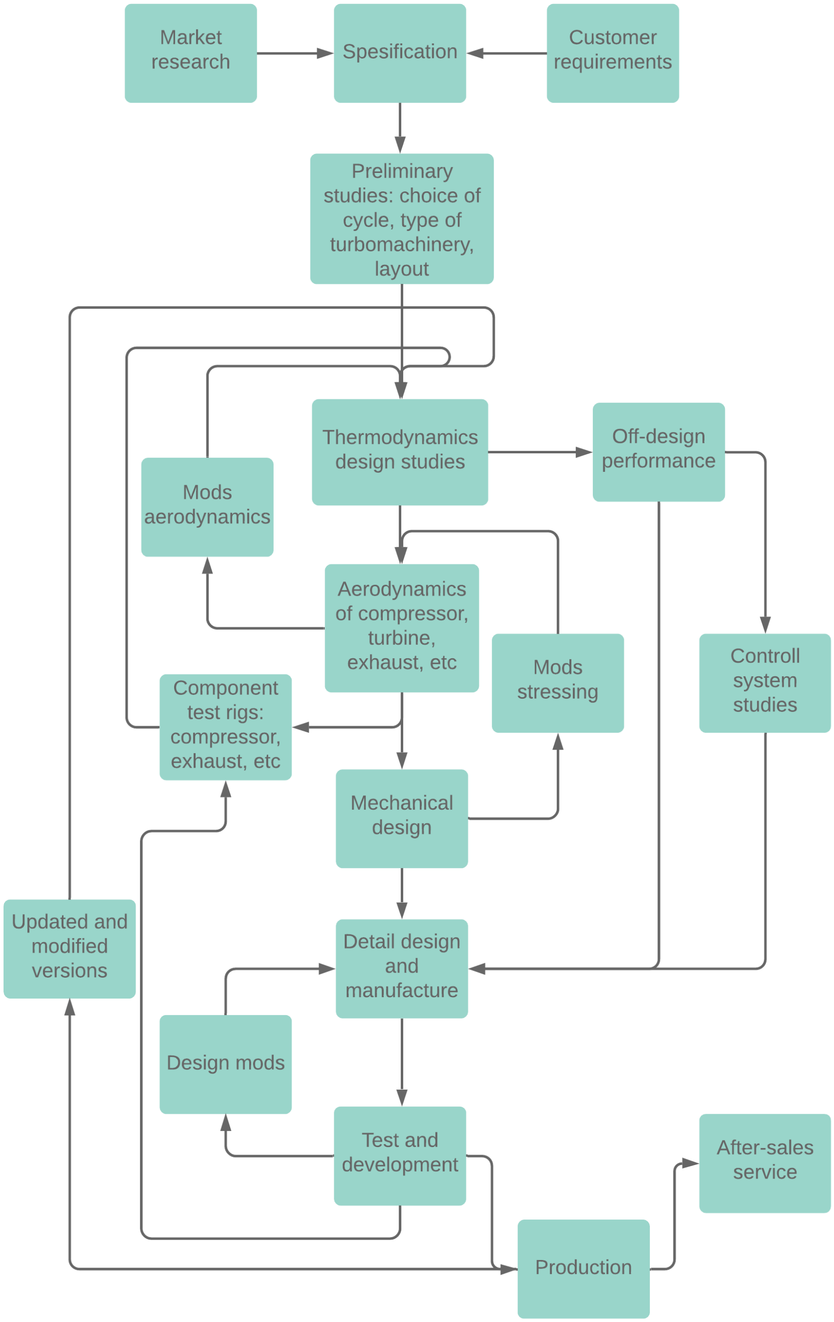


Figure F.8: Typical gas turbine design procedure [2]

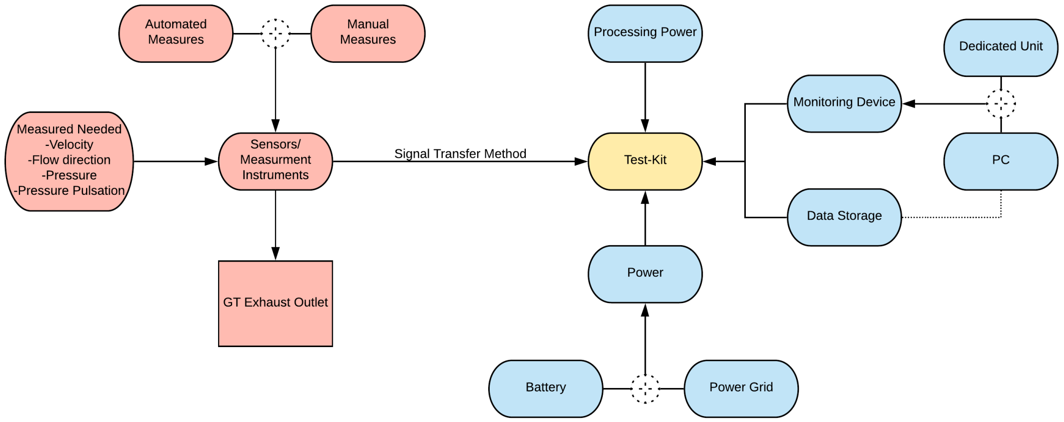


Figure F.9: System Conceptional

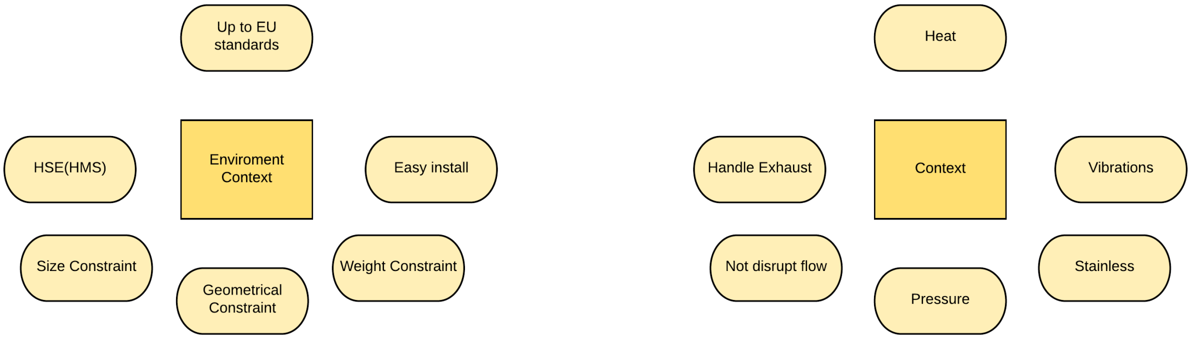


Figure F.10: Context Diagram

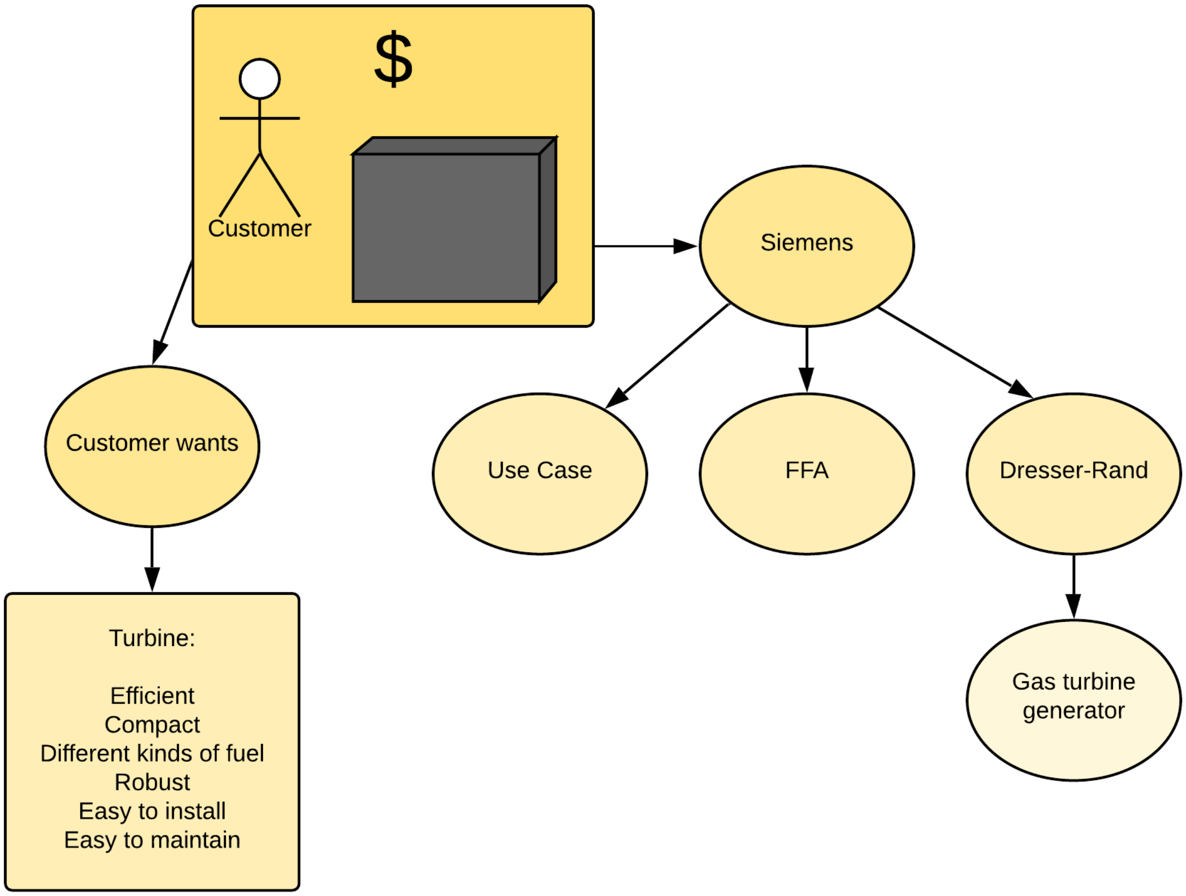


Figure F.11: Customer need diagram

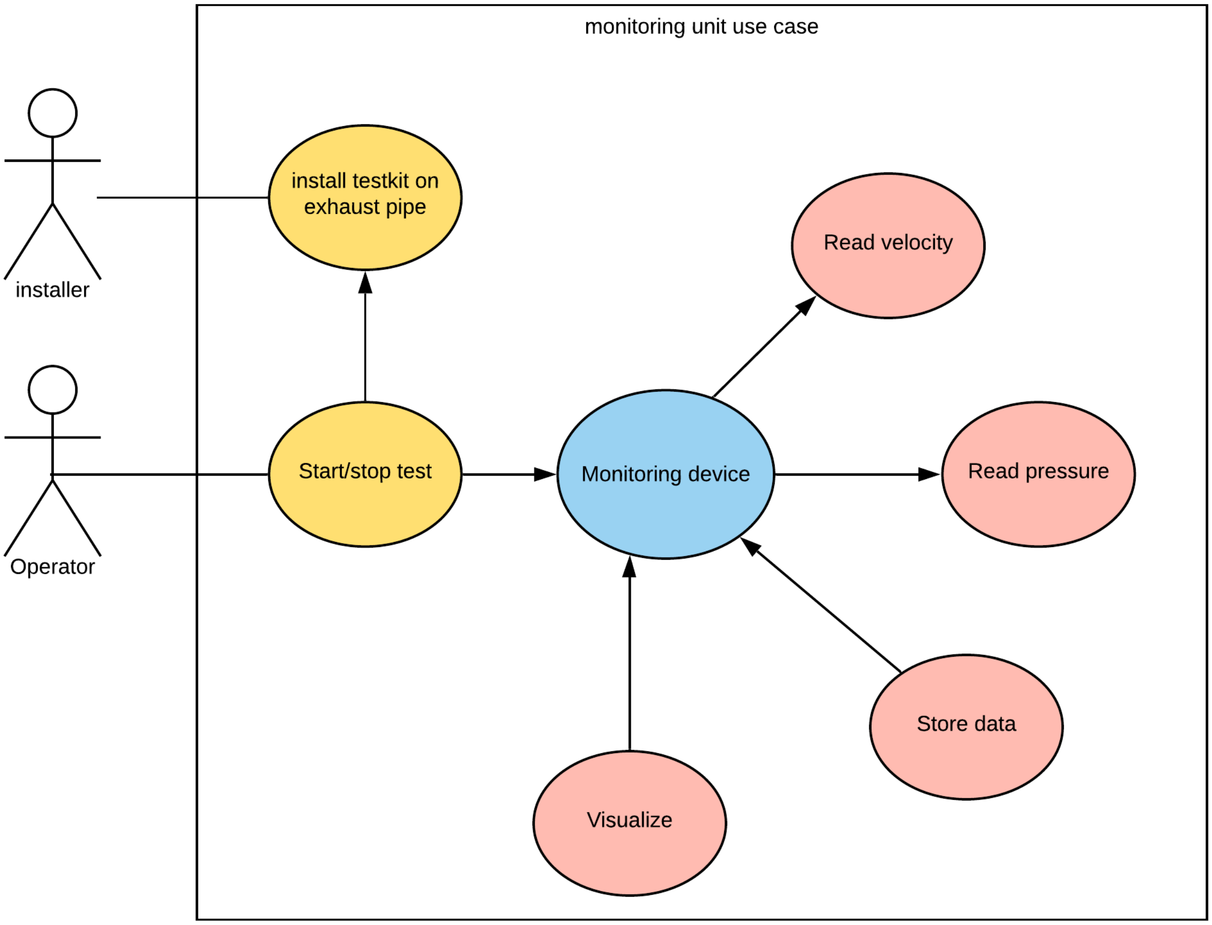


Figure F.12: Use Case 3

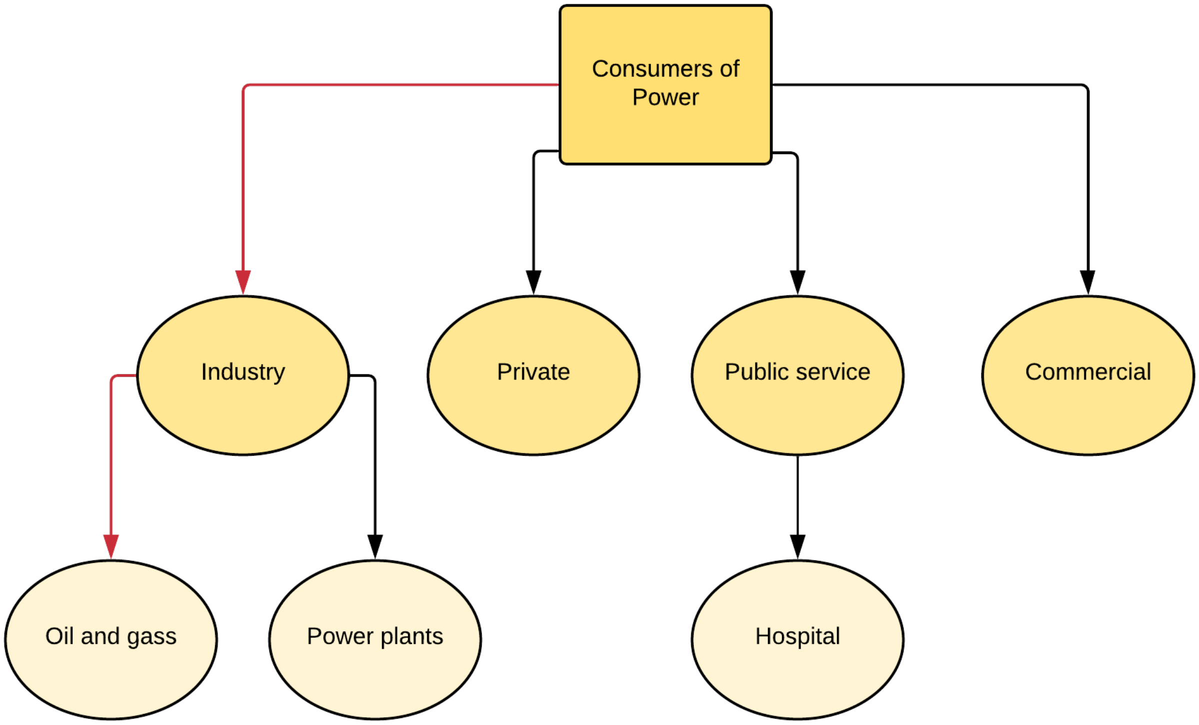


Figure F.13: Customer

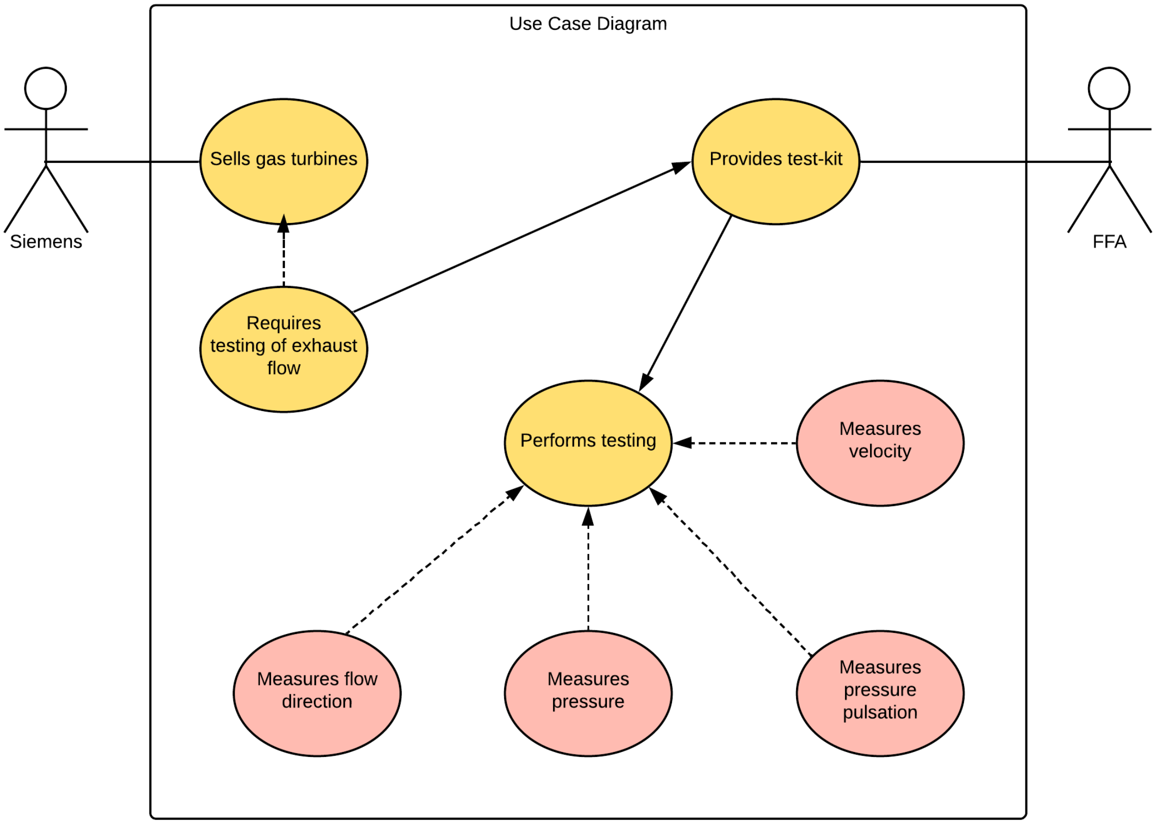


Figure F.14: Use Case 1

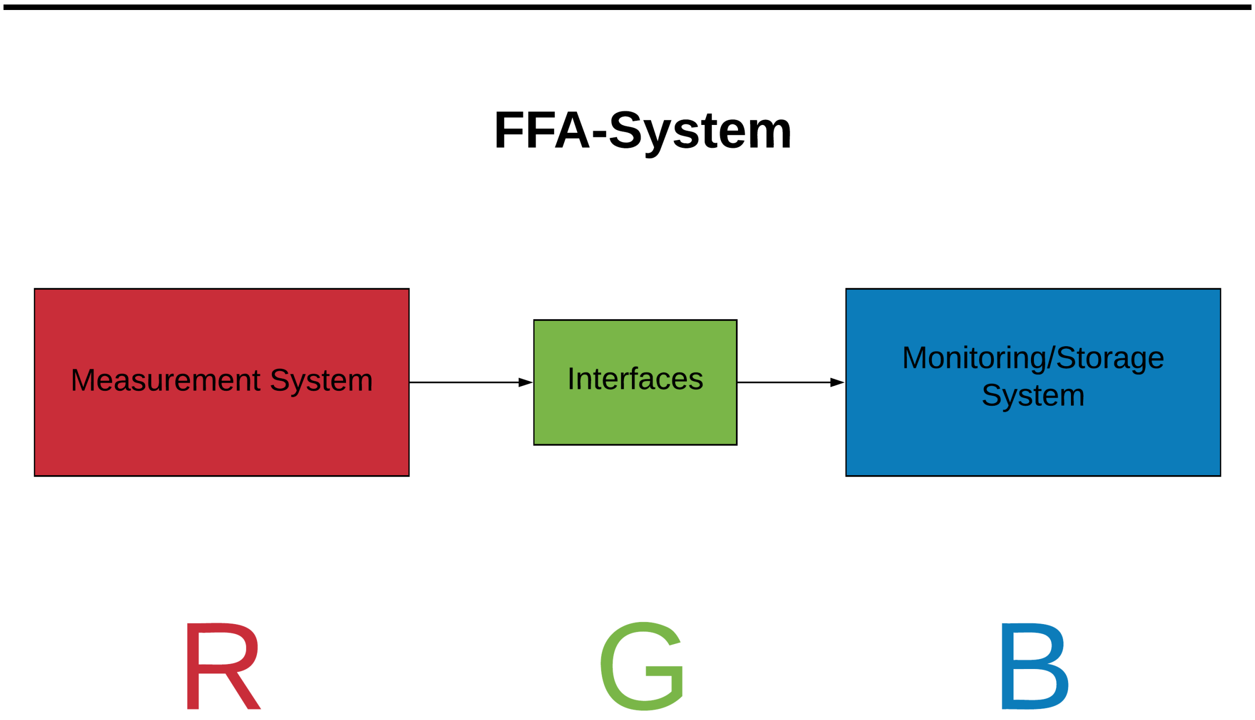


Figure F.15: Systems RGB

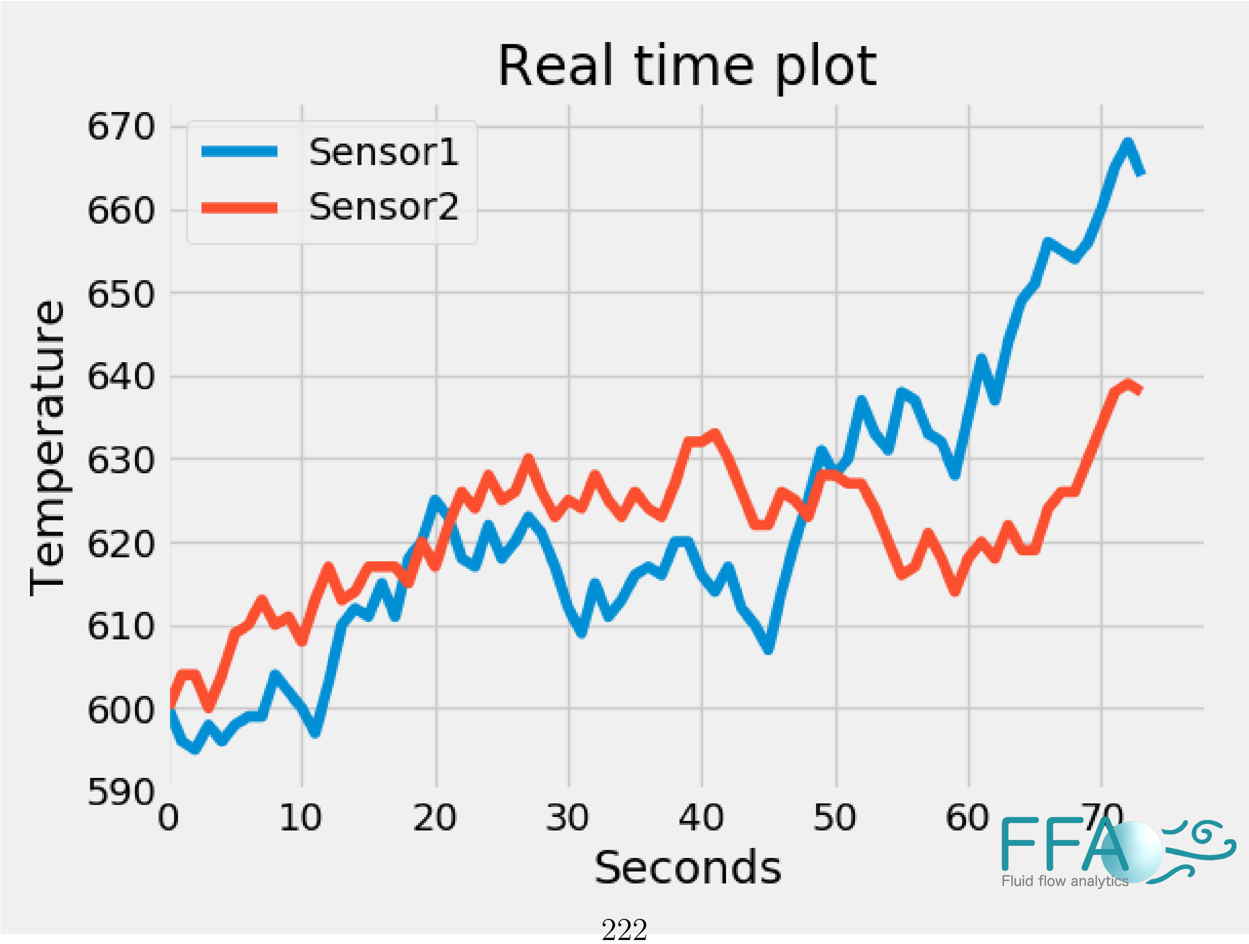


Figure F.28: Real time plotting with random generated values in Python

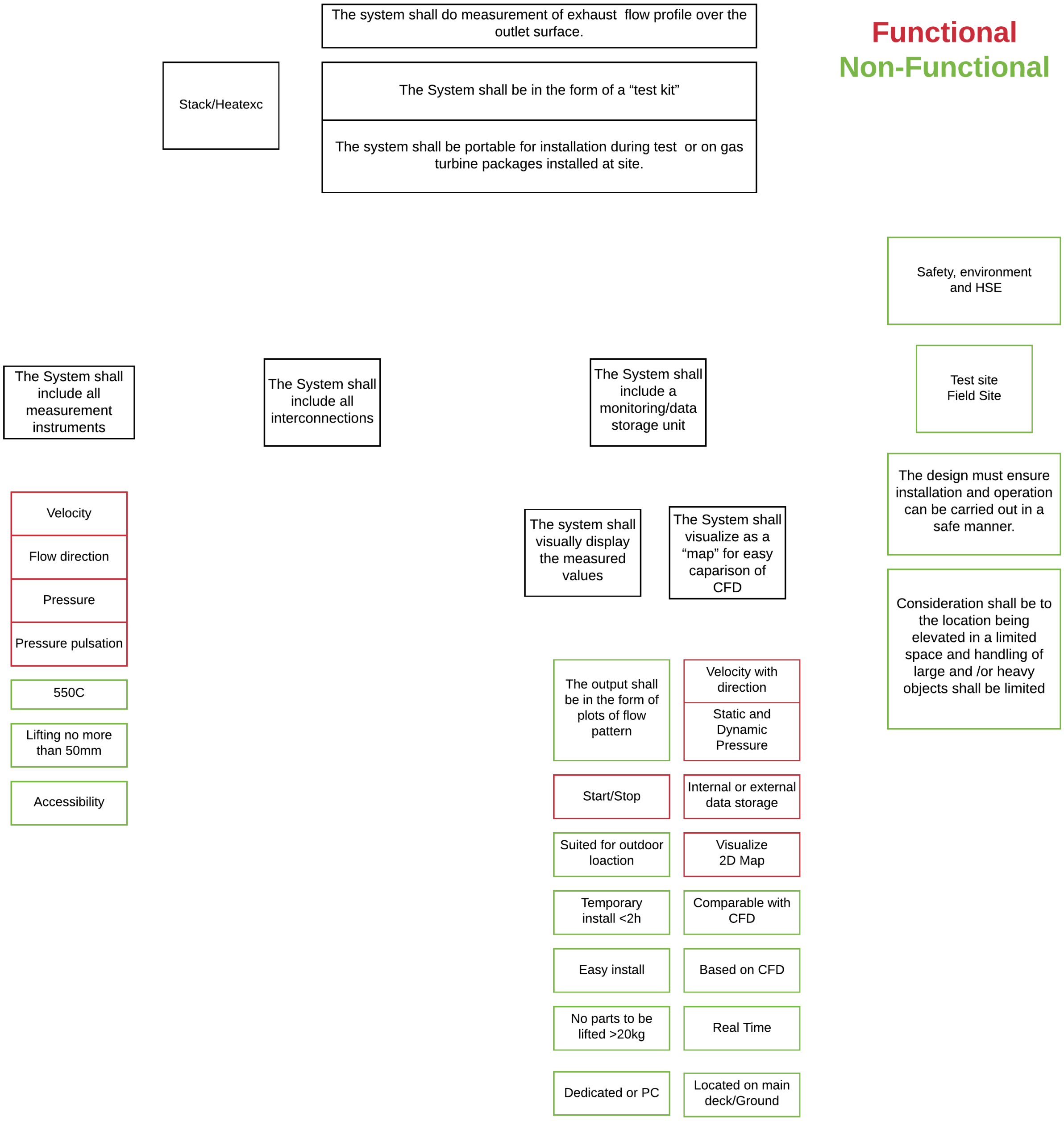


Figure F.16: System Requiremts 0.2

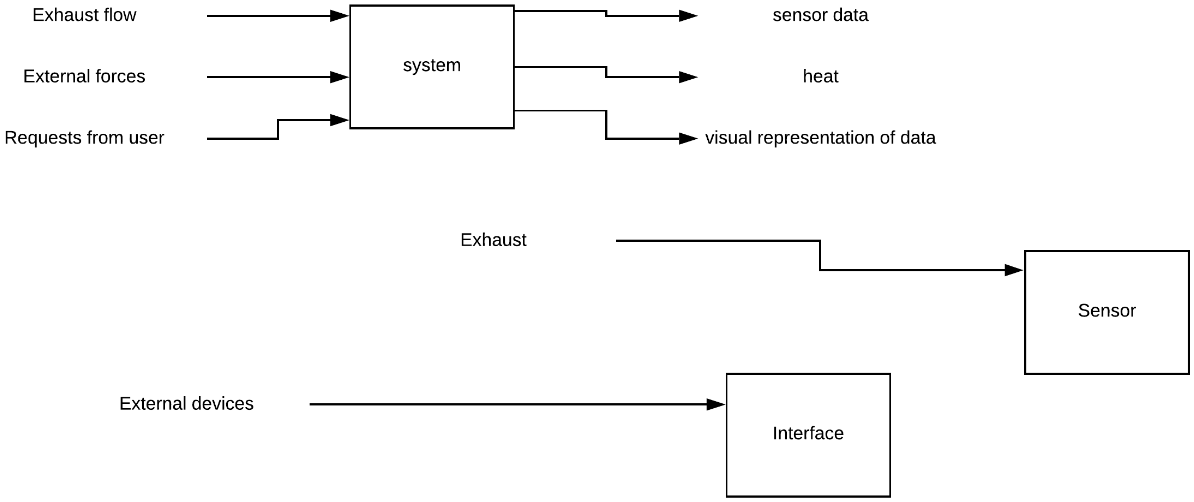


Figure F.17: CONOPS

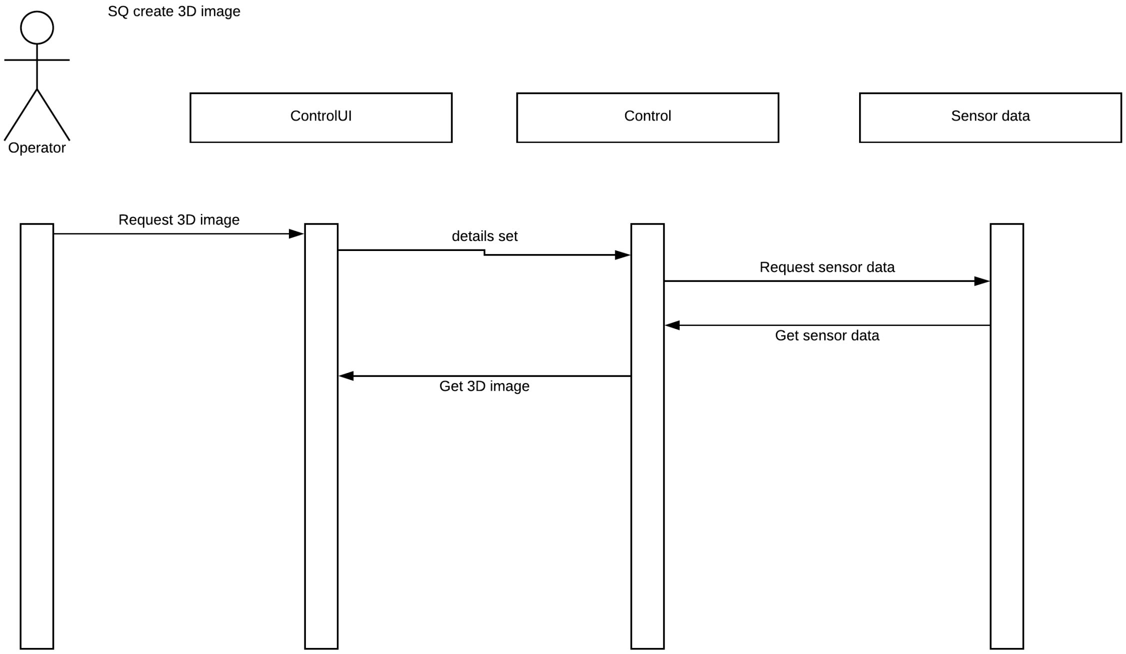


Figure F.18: SQD1

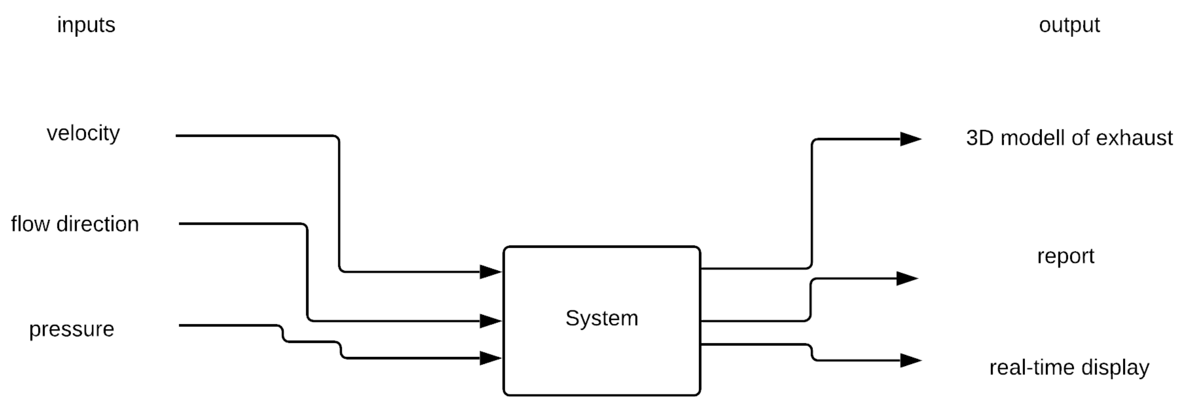
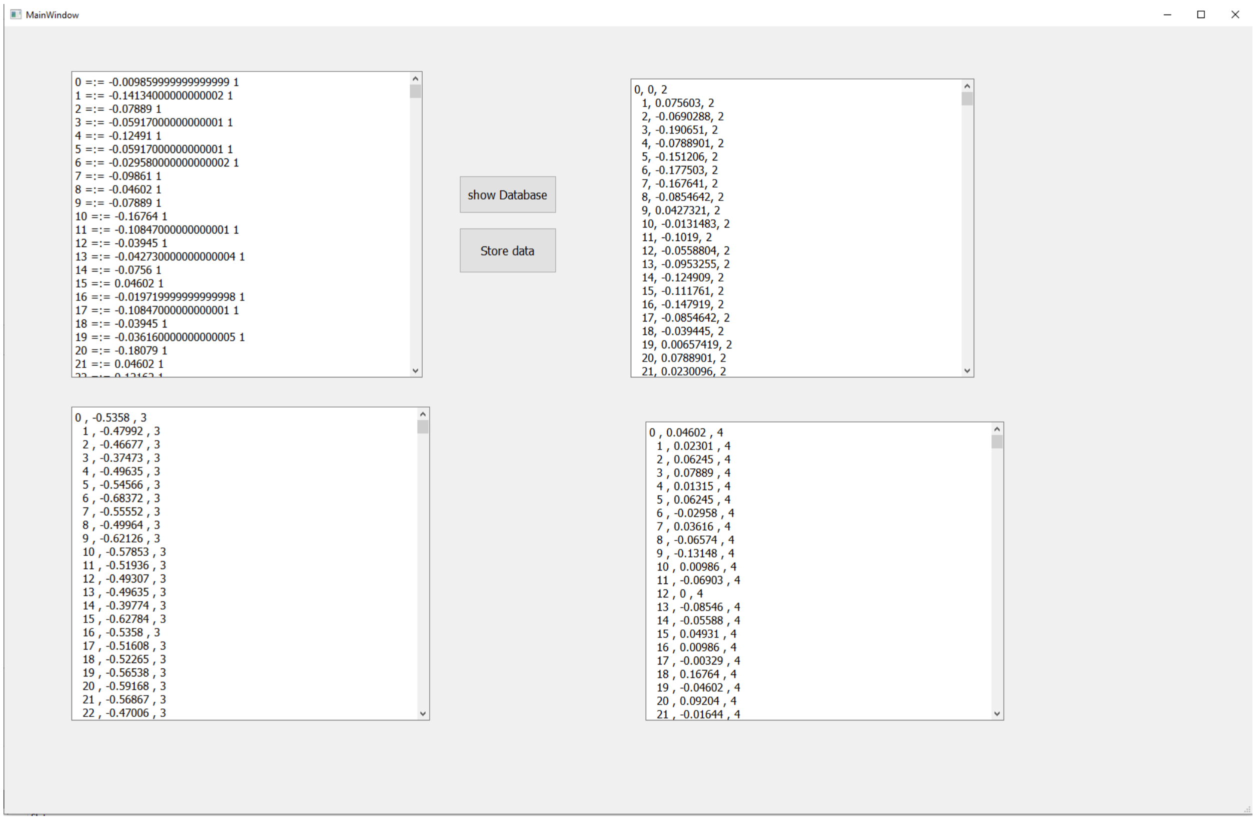


Figure F.19: input output system



```
0 =:= -0.009859999999999999 1
1 =:= -0.14134000000000002 1
2 =:= -0.07889 1
3 =:= -0.05917000000000001 1
4 =:= -0.12491 1
5 =:= -0.05917000000000001 1
6 =:= -0.029580000000000002 1
7 =:= -0.09861 1
8 =:= -0.04602 1
9 =:= -0.07889 1
10 =:= -0.16764 1
11 =:= -0.10847000000000001 1
12 =:= -0.03945 1
13 =:= -0.042730000000000004 1
14 =:= -0.0756 1
15 =:= 0.04602 1
16 =:= -0.019719999999999998 1
17 =:= -0.10847000000000001 1
18 =:= -0.03945 1
19 =:= -0.036160000000000005 1
20 =:= -0.18079 1
21 =:= 0.04602 1
```

```
0, 0, 2
 1, 0.075603, 2
 2, -0.0690288, 2
 3, -0.190651, 2
 4, -0.0788901, 2
 5, -0.151206, 2
 6, -0.177503, 2
 7, -0.167641, 2
 8, -0.0854642, 2
 9, 0.0427321, 2
 10, -0.0131483, 2
 11, -0.1019, 2
 12, -0.0558804, 2
 13, -0.0953255, 2
 14, -0.124909, 2
 15, -0.111761, 2
 16, -0.147919, 2
 17, -0.0854642, 2
 18, -0.039445, 2
 19, 0.00657419, 2
 20, 0.0788901, 2
 21, 0.0230096, 2
```

```
0 , -0.5358 , 3
 1 , -0.47992 , 3
 2 , -0.46677 , 3
 3 , -0.37473 , 3
 4 , -0.49635 , 3
 5 , -0.54566 , 3
 6 , -0.68372 , 3
 7 , -0.55552 , 3
 8 , -0.49964 , 3
 9 , -0.62126 , 3
 10 , -0.57853 , 3
 11 , -0.51936 , 3
 12 , -0.49307 , 3
 13 , -0.49635 , 3
 14 , -0.39774 , 3
 15 , -0.62784 , 3
 16 , -0.5358 , 3
 17 , -0.51608 , 3
 18 , -0.52265 , 3
 19 , -0.56538 , 3
 20 , -0.59168 , 3
 21 , -0.56867 , 3
 22 , -0.47006 , 3
```

```
0 , 0.04602 , 4
 1 , 0.02301 , 4
 2 , 0.06245 , 4
 3 , 0.07889 , 4
 4 , 0.01315 , 4
 5 , 0.06245 , 4
 6 , -0.02958 , 4
 7 , 0.03616 , 4
 8 , -0.06574 , 4
 9 , -0.13148 , 4
 10 , 0.00986 , 4
 11 , -0.06903 , 4
 12 , 0 , 4
 13 , -0.08546 , 4
 14 , -0.05588 , 4
 15 , 0.04931 , 4
 16 , 0.00986 , 4
 17 , -0.00329 , 4
 18 , 0.16764 , 4
 19 , -0.04602 , 4
 20 , 0.09204 , 4
 21 , -0.01644 , 4
```

Figure F.29: Database Testing

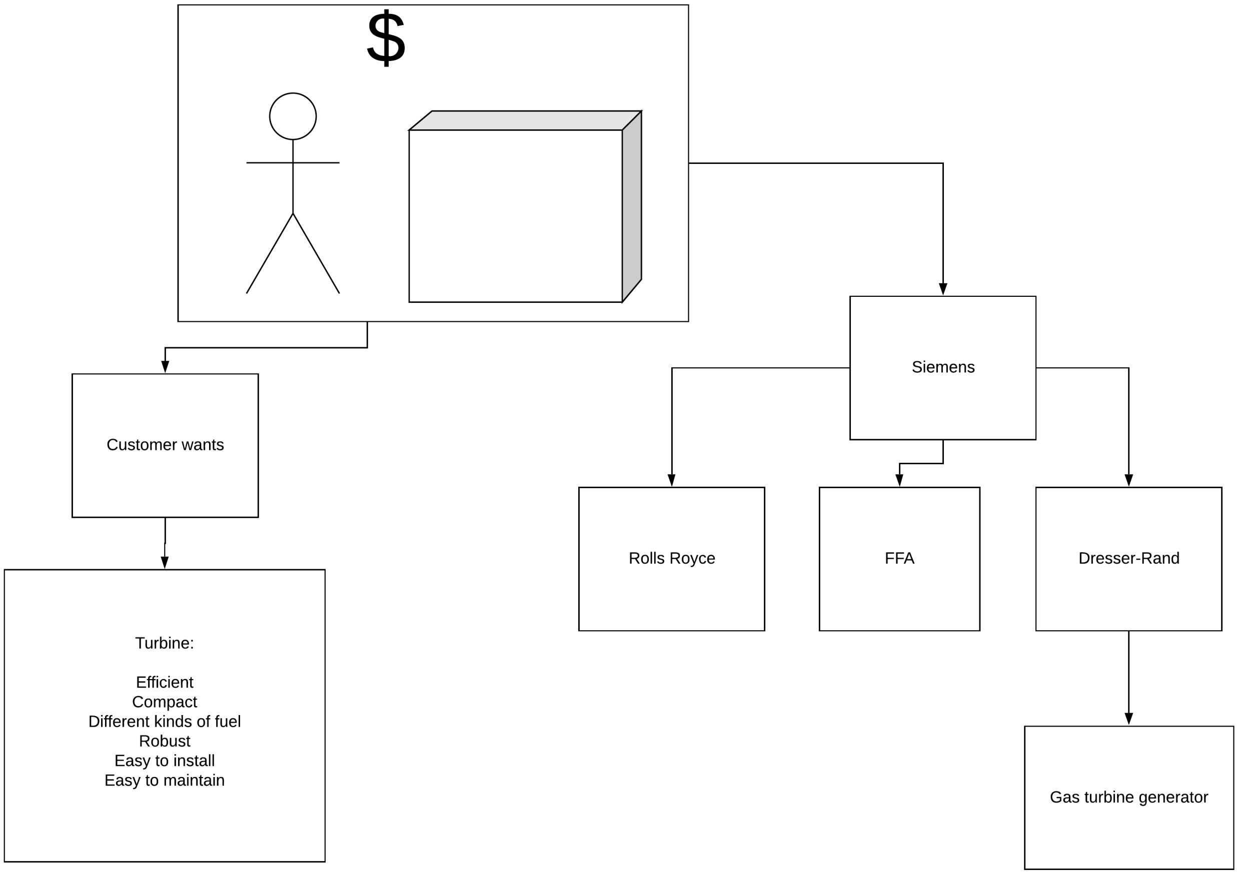


Figure F.20: Diagram

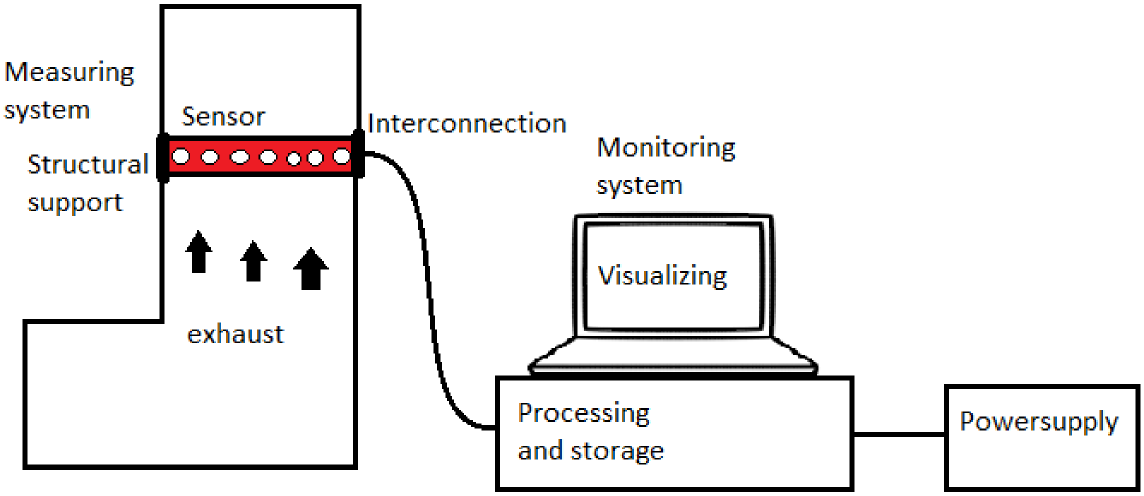


Figure F.21: Concept of operation 2

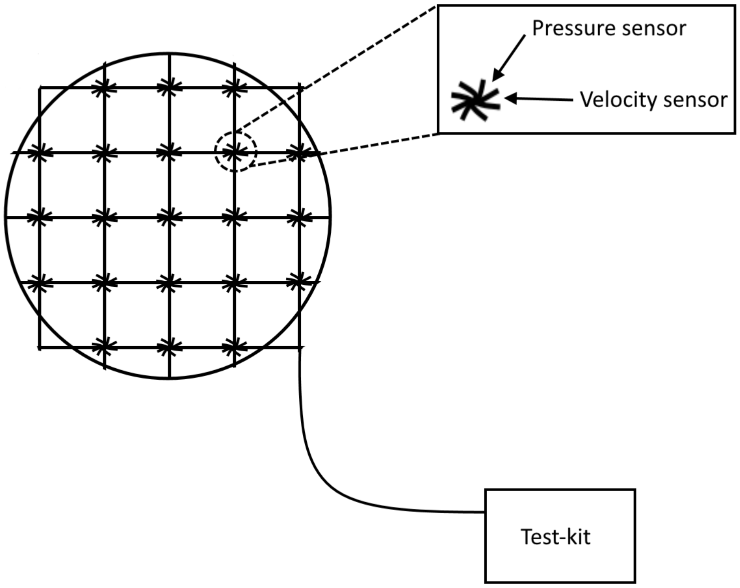


Figure F.22: Fan and pressure

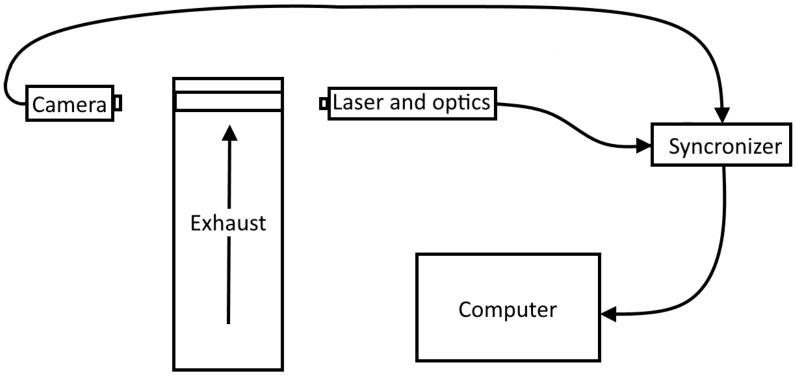


Figure F.23: Laser and optic

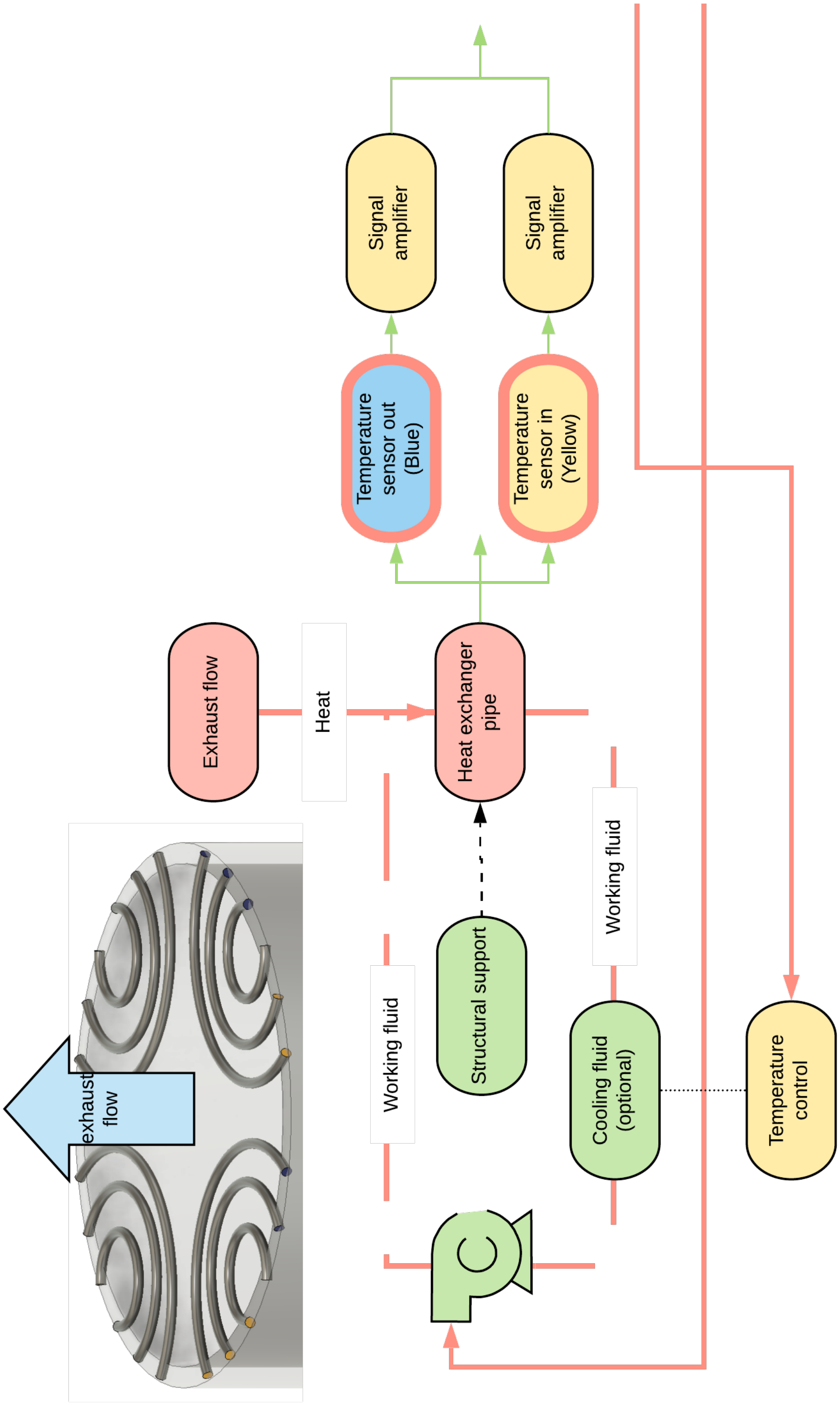


Figure F.24: Concept of operation 1

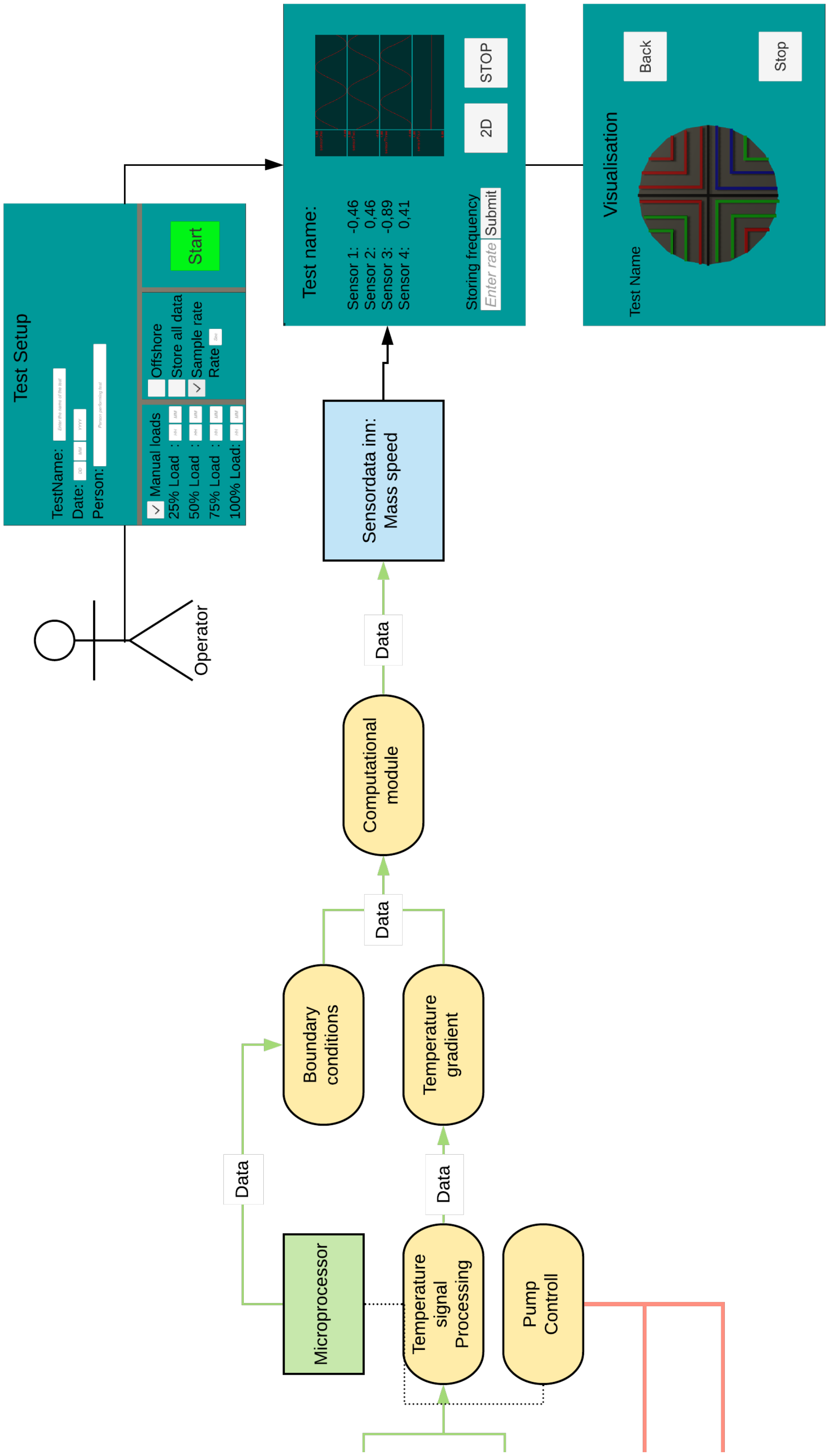


Figure F.25: Concept of operation 2

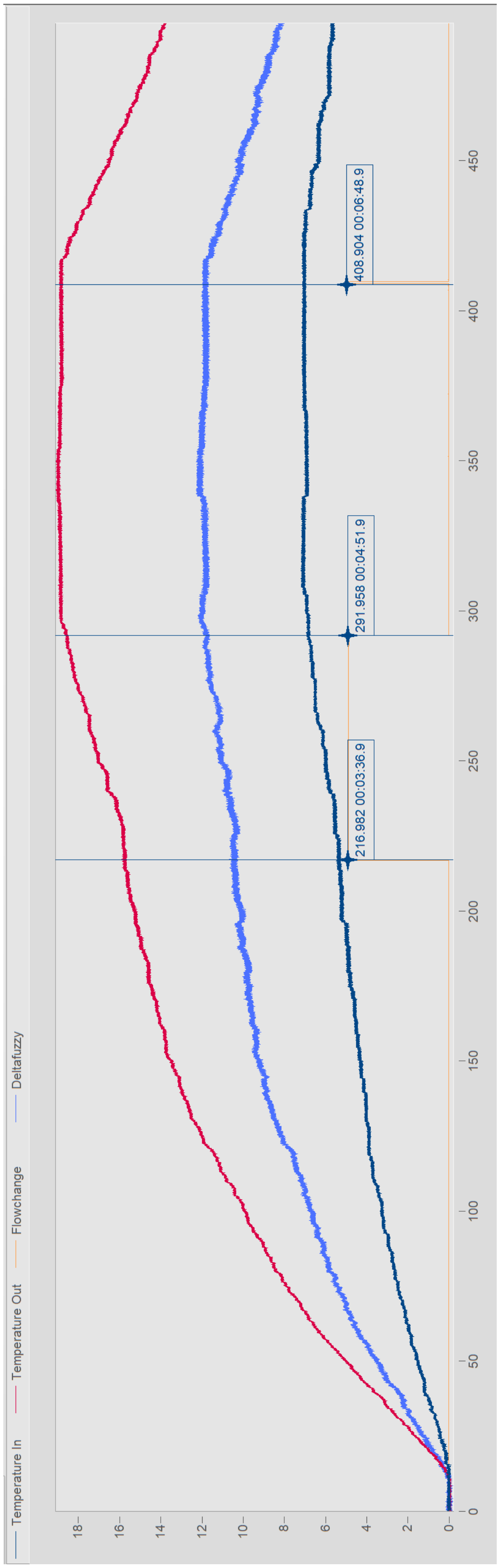


Figure F.26: IMC Famos: unconditioned data representation full scale

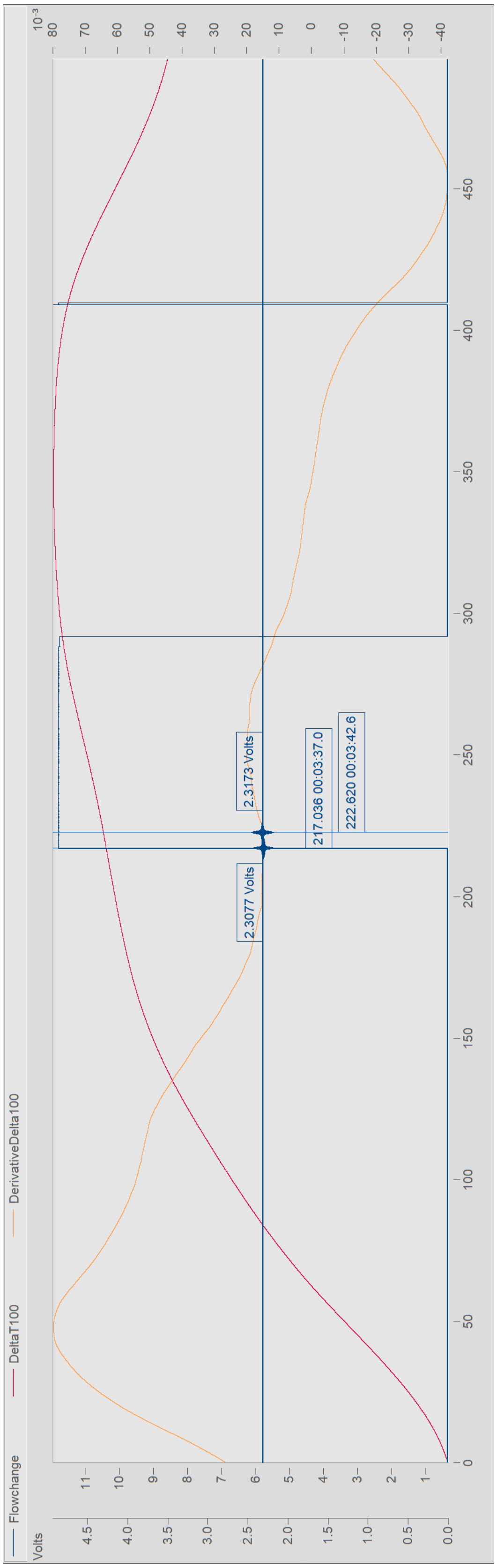


Figure F.27: IMC Famos: smoothed change in temperature with derivative full scale

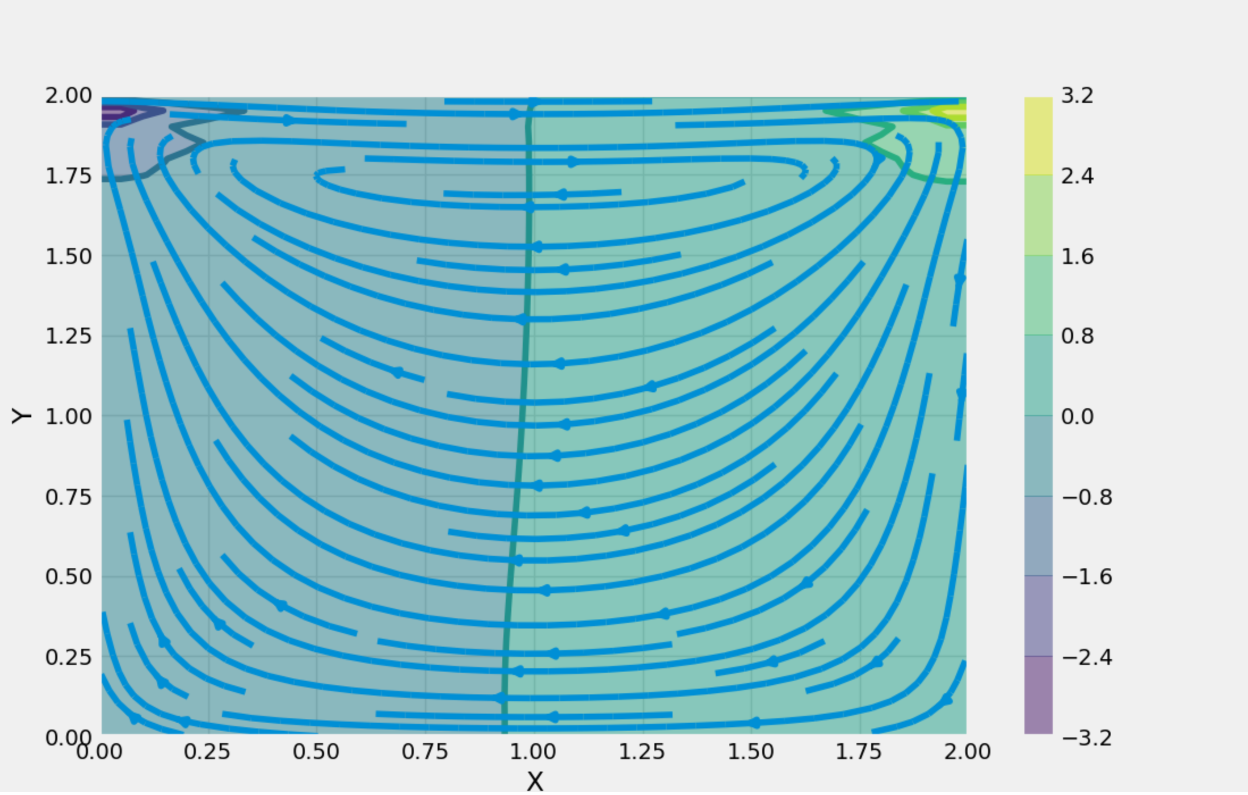


Figure F.30: 2D CFD made in Python

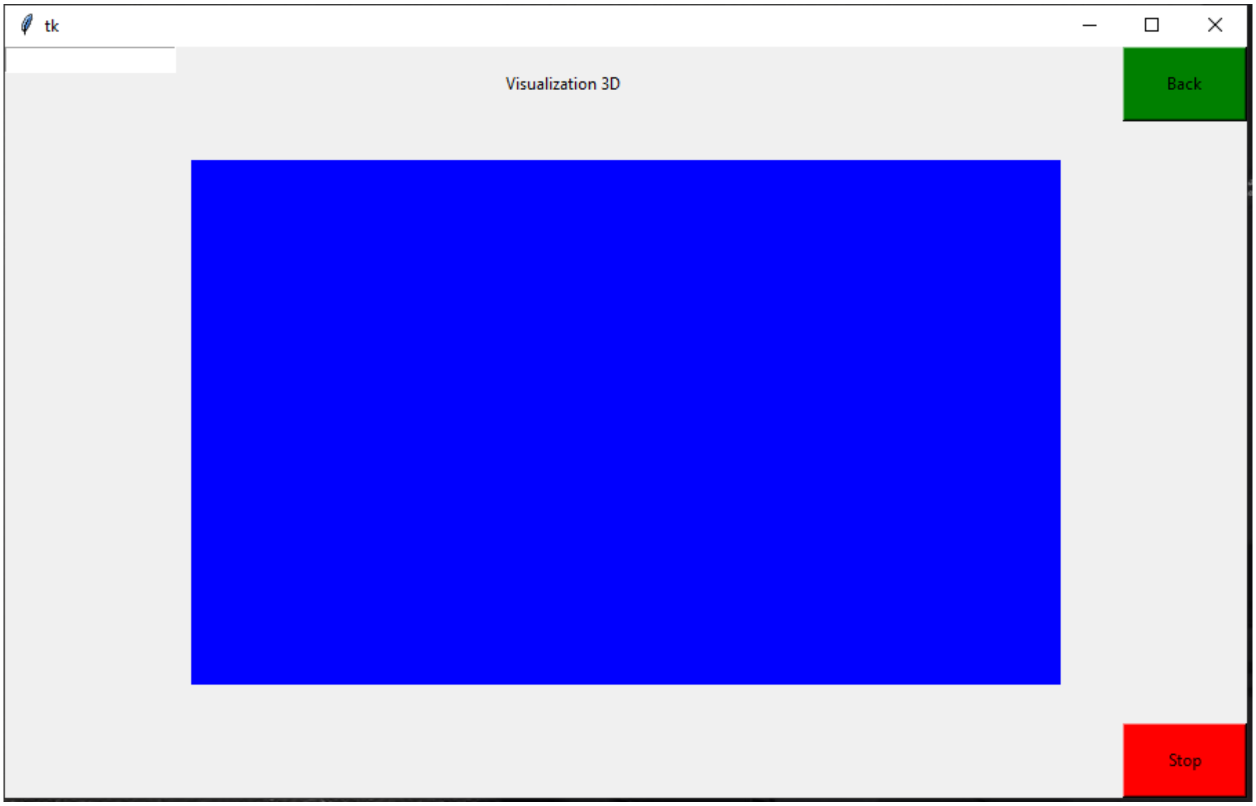


Figure F.31: Interface for visualization in Python