DESIGN, PRODUCTION AND TESTING OF A MICRO SCALE ORC

by

Berkhan Bayraktar

B.S., Mechanical Engineering, Boğaziçi University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfilment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering
Boğaziçi University

2022

ACKNOWLEDGMENTS

First of all, I would like to thank my thesis advisor Assoc. Prof. Hasan Bedir and co-advisor Prof. Günay Anlaş for their great support during my study. Their encouragement and knowledge allowed me to finish my research.

I would like to thank my BURET lab mates Ertuğrul Altun, Alpay Asma, and Hasan Eren Bekiloğlu for their great support, friendship, and great memories.

I would like to thank all the people in student clubs BUDS and BUHAK for their great memories and for keeping me sane especially in the last several months.

Finally, I would like to thank my mother Günay, my father Erdal, and his wife Cemile, for their support without any questions and for letting me be whoever I am.

ABSTRACT

DESIGN, PRODUCTION AND TESTING OF A MICRO SCALE ORC

Energy efficiency and renewable energy technologies are becoming more and more important every day. Organic Rankine Cycle is a proven way of generation of power from low temperature heat sources. In this study a 1 kW Organic Rankine Cycle is designed, produced and tested at BURET Laboratory on Boğaziçi University Sarıtepe Campus. A preliminary thermodynamic analysis is done at MATLAB to select the components. After the selection of the components, the test set-up connections are designed. In order to measure temperature, pressure and flow rate data from the system, a data acquisition system is also designed and programmed. The existing oil heating and chiller systems in BURET laboratory are used as the heat source and heat sink of the system. As the expander, an automobile turbocharger is used. Due to unavailability of very high-speed generators, the compressor of the turbocharger is used to measure the power output. Finally, all of the components are connected, the cycle is built, and testing is done. Due to the unexpected behavior of the pressure sensors, limited data are taken at the most efficient part of the experiment. A maximum turbine power output of 1.55 kW is measured at the turbine output. However, due to the low rotational speed of the turbine, the mechanical power generated by the turbine is not used effectively by the compressor, and a maximum compressor power of 33.15 W is obtained.

ÖZET

MİKRO SKALA ORGANİK RANKİNE ÇEVRİMİ SİSTEMİ TASARIMI, ÜRETİMİ VE TESTİ

Enerji verimliliği ve yenilenebilir enerji teknolojileri her geçen gün daha önemli hale geliyor. Organik Rankine Çevrimi düşük sıcaklıklardaki ısı kaynaklarından enerji üretmek için ispatlanmış bir yöntemdir. Bu çalışmada 1 kW güç üreten Organik Rankine Çevrimi tasarlandı, üretildi ve Sarıtepe Kampüsü'nde bulunan BURET laboratuvarında test edildi. Üretim öncesi yapılan analiz için temel olarak termodinamik analiz üzerine yoğunlaşıldı ve bu analiz MATLAB üzerinde yapıldı. Parçalar seçildikten sonra konumları belirlendi ve gerekli bağlantı elemanları tasarlandı. Ayrıca sistemden veri alabilmek için bir veri toplama sistemi tasarlandı ve programlandı. Halihazırda BURET laboratuvarında bulunan yağ ısıtıcı ve su soğutma sistemleri, kurulan systemin ısı kaynağı ve ısı emicisi olarak kullanıldı. Bu ısıtıcı ve soğutucuları sisteme bağlamak için bazı değişiklikler yapıldı. Ekspander olarak bir araba turboşarjının türbin kısmı kullanıldı. Yüksek hızlarda çalışan jeneratör bulunmaması sebebiyle turboşarjın kompresör tarafı mekanik güç çıkışını hesaplamak için kullanıldı. Tüm parçalar birleştirildikten sonra sistem çalıştırıldı ve test edildi. Sensörlerin beklenmeyen davranışı sebebiyle systemin en verimli çalıştığı aralıkta kısıtlı veri toplanabildi. Buna rağmen ve maksimum türbin gücü 1.55 kW olarak ölçüldü. Ancak, türbinin düşük hızda dönmesi sebebiyle, türbin tarafından üretilen bu güç kompresör tarafından etkin bir şekilde kullanılamadı ve maksimum kompresör gücü 33.15 W olarak ölçüldü.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

A	Area, m^2
act	(subscript) Actual
c	Specific Heat, (kJ/kgK)
CV	(subscript) Control Volume
cw	(subscript) Cooling Water
g	Gravitational Constant, m/s^2
gen	(subscript) Generated
in	(subscript) Inlet
oil	(subscript) Thermal Oil
out	(subscript) Outlet
p	Pressure, Pa
P	(subscript) Pump
Q	Heat Transfer, (W)
s	(subscript) Isentropic
T	Temperature, (K)
T	(subscript) Turbine
U	Overall Heat Transfer Coefficient
v	Specific Volume
wf	(subscript) Working Fluid
Δ	Difference
η	Efficiency
ρ	Density, (kg m^{-3})

LIST OF ABBREVIATIONS AND ACRONYMS

BURET	Boğaziçi University Renewable Energy Technologies
CHP	Combined Heat and Power
CSP	Concentrated Solar Power
DAQ	Data Acquisition System
IEA	International Energy Agency
LMTD	Logarithmic Mean Temperature Difference
MAF	Mass Air Flow
ORC	Organic Rankine Cycle
PTFE	Polytetrafluoroethylene
P&ID	Piping and Instrumentation Diagram
WHR	Waste Heat Recovery

1. INTRODUCTION

The global surface temperature increased by 1.0°C in the last century due to human activities. The main reason for the temperature rise is the increase in emission of CO_2 and other greenhouse gases. Although 1°C may look insignificant, it already resulted in important glacier loss and many climate extremities. According to the IPCC report, going beyond 1.5°C will result in irreversible climate change and extremities [1]. And the temperature rise should be limited below 1.5°C.

According to International Energy Agency [2], 42% of CO_2 emission is from electricity and heat generation, and 64.5% of the electricity production is based on fossil fuels [2]. The data shows that there is huge potential to decrease CO_2 emissions by changing sources of electricity production.

Hydropower plants are being widely used for power generation since 19th century. Although they do not emit any greenhouse gasses directly, generally a huge dam area is required, and they damage the ecosystem considerably and should be planned wisely. Alternatively, renewable energy sources like wind, geothermal, solar energy, and industrial waste heat recovery (WHR) may be used for power generation. For example, the amount of waste heat generated in cement or steel industry is significant. Even though most of the industrial waste heat and some renewable energy sources like geothermal are low to medium temperature sources they can still be converted to electricity and contribute to reduction of greenhouse gas emissions. Another renewable energy source is geothermal. Geothermal plants take little space and require less investment; however, they can only be built in specific places since hot water sources are not available everywhere.

A huge heat source is the sun. Solar power is generally harvested through photovoltaic panels and converted directly into electricity. But, solar radiation is not constant, changes throughout the day, and goes to zero at night. The deviation in daytime can be very rapid when clouds appear. This makes fluctuations in electric current which can be dangerous and can even lead to blackouts. In addition to daytime fluctuations, zero

production at night is an important problem. Storage batteries can solve both problems, but they are expensive, and they increase the initial investment significantly. Because of their thermal inertia, scncentrated solar power (CSP) system can be a solution to this problem. In a CSP system, the solar radiation is concentrated via mirrors (curved or planar) to heat a fluid which then is used to drive a thermodynamic power cycle, generally Rankine or Stirling. Extra heat can be stored in storage tanks and can be used at night as well. There are several options in concentration systems: solar tower, parabolic trough, Fresnel collectors and parabolic dish. Parabolic trough systems are easy to operate and efficient enough to be used in electricity generation. Rankine or Organic Rankine Cycles can be used as power cycles to generate electricity, since large range of temperatures can be obtained by changing the collector type and size [3].

The Rankine Cycle is a very well-known power cycle, which uses water as a working fluid. Water needs to be heated to temperatures higher than 500°C. However, small parabolic trough systems or low temperature heat sources cannot always reach those temperatures. Therefore, organic fluids such as refrigerants may are used to run a Rankine Cycle at lower temperatures, which is then called an Organic Rankine Cycle (ORC). Low boiling point temperature of organic fluids enable ORC systems to be used at lower. Refrigerants are organic fluids that have higher molecular weight and lower boiling point than water. They are generally non-poisonous and non-flammable, and since they diffuse easily in air, they can safely be used. However, their environmental effect should be considered, because some of the gasses are harmful to ozone layer and may contribute to global warming as greenhouse gas.

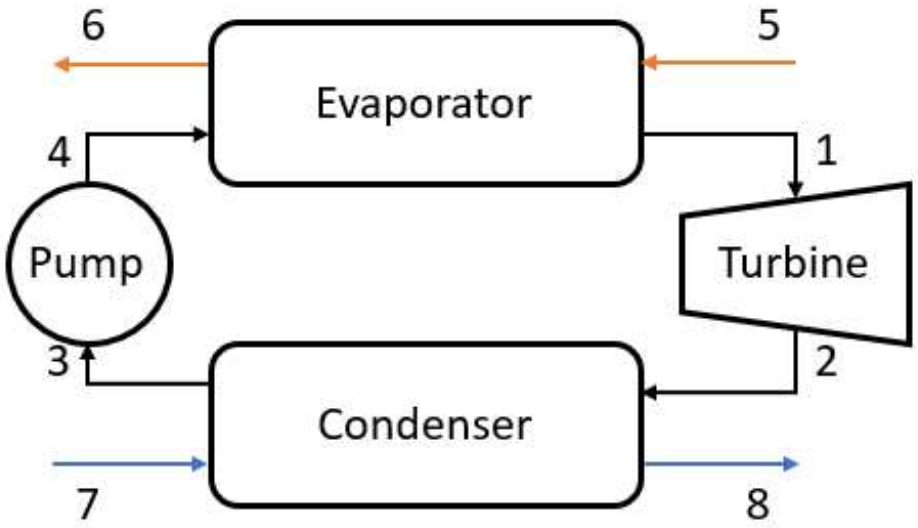

Figure 1.1. Simple ORC Scheme.

A schematic view of simple ORC is shown in Figure 1.1. The main components of ORC are turbine, evaporator, condenser, and pump. The pump is used to increase the pressure of the organic fluid; the phase of the fluid in the pump is liquid. After the pump, the fluid enters the evaporator, where, the fluid is heated, and its phase is changed to vapor; the pressure remains almost constant in the evaporator. The main energy input to the working fluid of the cycle is achieved in the evaporator. Although some energy is added to fluid by pump as well, it is much smaller in comparison to the heat received in the evaporator. In large facilities, evaporator may be separated into different parts as economizer, boiler, and superheater. Their geometries are optimized for different fluid phases and the heat transfer efficiency is increased. However, in smaller systems, a single heat exchanger is sufficient. After the evaporator, the fluid enters the expander, which converts the fluid's internal energy to mechanical work. That mechanical work generally is converted to electricity by a generator. Different type of expanders, such as radial inflow turbine, scroll expander etc. may be used in a cycle, depending on the capacity and pressure difference. The temperature and the pressure of the fluid decreases considerably in the expander; in an ideal case, fluid is a saturated vapor at the exit of the expander. However, in real life, due to possible damage

from liquid droplets to the turbine, fluid is kept at a few degrees above the condensation temperature in superheated form. The fluid leaving the expander enters the condenser, where the pressure remains almost constant just like in the evaporator. But the temperature and the enthalpy of the fluid decreases since it transfers heat to a cooling medium and condenses. The fluid leaving the condenser is a few degrees below of saturation temperature in a sub-cooled liquid form to eliminate the chance of vapor entering the pump.

In this thesis, a small (1 kW) lab scale ORC is designed, built and tested, The purpose is to build a portable small and cheap ORC to produce electricity.

1.1. Literature Review

In the literature, there are many studies on ORCs. Some relevant studies are summarized below:

Georges et al. [4] modeled and designed an ORC system coupled to a solar thermal system. The system is designed for 140°C heat source and 35°C condensing temperature. The power output of the system is 3 kWe. The study analyzes 6 fluids and R245fa is the selected fluid. There are two serially connected scroll expanders. Because water is not available at the experiment site, fan cooling is used. The thermal efficiency of the system is 11%.

Dickes et al. [5] built the system that Georges et al. designed. The ORC system is investigated experimentally under design point and off design conditions. 110 unique steady state conditions were selected and experimented. Evaporation temperature from 111°C to 135°C and condensing temperature from 25.8°C to 61.8°C were observed; other parameters are evaporation and condensing pressures, working fluid mass flow rate, cooling, and heating capacities by changing oil and condenser fluids' flow rate. The final net power output is 915 W and thermal efficiency of the system is 4.5%.

Bekiloğlu et al. [6] optimized an ORC system for 90, 120 and 150°C heat sources. In the study, components of basic ORC system are designed and working fluid is selected. As

the expander, radial inflow turbine is designed iteratively with 1-D mean line approach. Condenser and evaporator are designed for each case, and they are the economic indicator since they are the most expensive part of the system. The turbine design formulas used in this paper can be used to estimate the performance of existing turbochargers.

Da Lio et al. [7] designed a radial inflow turbine for ORC with working fluid R245fa. They inspected the effects of volumetric expansion ratio and specific speed on efficiency. They generated efficiency maps and loss graphs to explain the losses of the turbine. They found that optimum volumetric expansion ratio is around 3.7 to 4.

Le Roux et al. [8] used a turbocharger as the turbine of a recuperated solar dish Brayton cycle. The performance of three different turbochargers and different recuperator geometries are analyzed to determine the maximum thermal efficiency of the system. Maximum thermal efficiencies between 20.2% and 34.2%, and solar conversion efficiencies between 13.5% and 21% are obtained. It is concluded that small-scale power systems can provide electricity for small communities that have no access to the grid. Large scale production possibility of small scale power systems can decrease the overall cost.

Declaye et al. [9] designed and tested an ORC system. The uses R245fa as a working fluid, and an open drive scroll expander as the expander. The expander is converted from a commercially available compressor. To eliminate leakage, the expander is put in an airtight box. The expander shaft output is connected to the torque meter via magnetic coupling. Design heat input is 20 kW, and power output is 1.8 kW. Maximum cycle efficiency of 8.5% is reached. The system can also be used as a combined heat and power (CHP) system. In this configuration, power to heat ratio varies between 1.9% and 11.8%. In addition to ORC testing, performance map of the turbine is obtained.

Bracco et al. [10] designed, built and tested an ORC prototype. An inverter-driven diaphragm pump is used as the pump. The expander of the system is a scroll expander, and it is connected to a 1 kW generator. Expander outlet temperature of the fluid varies between 90°C and 130°C. Measured efficiency is between 7% and 9%. In addition to the prototype, a simulation model is proposed, all the experimental data consistent with the simulation

except boiler outlet pressure and cycle efficiency. It is proposed that the main problem is the vapor quality at the boiler outlet is unknown. This decreases the cycle efficiency.

As it can be seen, small scale power generation can decrease the overall cost of the electricity for remote residential areas. Depending on the temperature of the heat resource in those areas, different power cycles can be used. ORC is one of them and it is suitable for small scale power generation. In small scale ORC, generally scroll expanders are used but it is shown that radial inflow turbines are also suitable.

1.2. Objectives

At Boğaziçi University Renewable Energy Technologies Laboratory (BURET), there is an Organic Rankine Cycle (ORC) test system designed and built by Günay Anlaş and Hasan Bedir using funding from ISTKA and Boğaziçi University. The system is designed for a power generation capacity of 10 kW, and it employs a scroll type expander. The test system allows the measurement of pressure and temperature at many points in the cycle with a significant number of sensors and transmitters installed. This increases the overall system size and heat loss since some of the sensors require a certain lead pipe length. The measurements generate valuable information about the cycle and their presence is justifiable. Furthermore, the increased heat loss is still not high in comparison to the cycle power output (in the range of 10 kW). Radial turbines are viable alternatives to scroll expanders and they can be scaled up to higher capacity more readily than scroll expanders. Although it is possible to install and test the cycle with a radial turbine on BURET ORC test system, a radial turbine with a power rating of 10 kW is not readily available. A new one should be designed and produced. A prototype radial turbine will have a significant cost and it is very likely that more than one prototype is necessary to finalize the design. A cost effective and a practical solution could be the use of turbines from automotive industry. In this thesis, an automobile turbocharger will be tried as the ORC expander. The average power output of a small vehicle turbocharger is around 1 kW. And this power output level is not suitable for testing on BURET ORC. As a result, design and production of a smaller system with a 1 kW power output is aimed. The purpose of the thesis is to design and build,

a small-scale portable simple ORC system using an automobile turbocharger as the expander. Power rating will be 1 kW. This ORC system will be coupled to a low temperature heat source of the existing ORC system in BURET. After building the system, experiments will be done to determine the performance of the system. Net power output, heat and electricity inputs will be measured. Then, power output and thermal efficiency of the system will be evaluated.

The working fluid of the ORC will be R134a because of its suitable properties such as the values of saturation pressure, saturation temperature, specific heat, thermal diffusivity, global warming potential etc. for the application and its low cost. The system will be built as compact as possible to minimize heat and pressure losses from pipes. Another feature will be the portability of the system; a small portable system can be used with different heat sources. In BURET Laboratory, there is an attached oil heating unit to operate the ORC. This heating unit is capable of heating thermal oil up to 140°C temperature. This heating unit will be used to operate the ORC that will be designed and built.

2. PRELIMINARY ANALYSIS AND COMPONENTS

As stated in the objectives part, the main goal is to get 1 kW mechanical work from the turbine. At the end, the system will be tested in the laboratory. However, before building the system, some preliminary analyses will be done. In this section, before selecting the parts, a simple ORC cycle analysis is carried out with four main components: pump, evaporator, turbine, and condenser. Evaporator and condenser are both heat exchangers and their equations are similar.

2.1. Thermodynamic Analysis

For the thermodynamic analysis the following assumptions are made:

- All the components work at steady state,
- Heat losses from the components are neglected,
- Pressure drop in pipes and the components are neglected,
- Flow is fully developed and one dimensional in the pipes.

The power generated by the turbine is calculated as follows

$$\dot{W}_T = \dot{m}_{wf}(h_{in} - h_{out}) \tag{1.1}$$

where $\dot{W}_T$ is turbine power, $\dot{m}_{wf}$ is mass flow rate of the working fluid and h is the enthalpy at the stated location. The pump power input to the system is calculated as follows

$$\dot{W}_P = \dot{m}_{wf} v_{wf}(P_{out} - P_{in}) \tag{1.2}$$

where $\dot{W}_P$ is pump power, v_{wf} is specific volume of the working fluid and P is the pressure at the specified point. Evaporator energy balance is

$$\dot{m}_{wf}(h_{out,wf} - h_{in,wf}) = \dot{m}_{oil} c(T_{in,oil} - T_{out,oil}). \tag{1.3}$$

where c is specific heat of the oil, and T is the temperature of the oil at the specified locations. Condenser energy balance is

$$\dot{m}_{wf}(h_{in,wf} - h_{out,wf}) = \dot{m}_{cw}(h_{out,cw} - h_{in,cw}). \quad (1.4)$$

2.2. Turbine and Compressor

The turbine generates the output power of the cycle; it converts internal energy of the fluid to mechanical work. Normally, this mechanical work is converted to electricity with a generator. However, in this setup, an existing automobile turbocharger will be used as a turbine. The turbocharger's rotational speed increases as its size decrease [11] [12], the relatively small turbocharger that is used in the study is manufactured by Supsan and its rotational speed is around 80,000 rpm at full speed. The maximum rotational speed of a commercially available generator is limited to 24,000 rpm [13]. Designing a generator for this application with such a high speed is beyond the scope of this study, and the power output from the turbine will not be converted to electricity. Instead, a compressor will be used. The power output from the turbine will be measured from the existing compressor of the turbocharger. By measuring the outlet pressure and the flow rate of the compressor, it is possible to calculate the shaft power consumed by the compressor (generated by the turbine); the energy equation used is shown below [14]

$$\frac{p_{out}}{\rho} + \frac{V_{out}^2}{2} + gz_{out} = \frac{p_{in}}{\rho} + \frac{V_{in}^2}{2} + gz_{in} + \frac{\dot{W}_{shaft\ \ net\ in}}{\dot{m}}. \quad (1.5)$$

2.3. Heat Exchangers

There are two heat exchangers in the ORC system: the evaporator and the condenser. One of them is used to heat the working fluid, the other is used to cool it. They are of counter flow type, which makes the hot side of the cold fluid is closer to the hot side of the hot fluid and vice versa. This way, the temperature difference between the two fluid streams in the

heat exchanger is more uniform, and the heat transfer rate is similar at all heat transfer surfaces. In the system, soldered plate type heat exchangers are used because their pressure ratings are usable for the system, their heat transfer performances are good, and they can be connected easily using threaded connections. Similar heat exchangers are already used in the existing ORC system in BURET Laboratory and their experimental data are available to estimate performances.

In the analysis of heat exchangers energy balance of the evaporator (equation 1.3) and that of the condenser, (equation 1.4) and following formula are used [15]:

$$\dot{Q} = UA \cdot LMTD \tag{1.6}$$

where U is overall heat transfer coefficient, A is heat transfer area, and the log mean temperature difference, $LMTD$ is:

$$LMTD = \frac{\left(T_{ho\ ,in} - T_{cold,out}\right) - \left(T_{ho\ ,out} - T_{cold,in}\right)}{\ln\left(\frac{T_{hot,in} - T_{cold,out}}{T_{hot,out} - T_{cold,in}}\right)}. \tag{1.7}$$

U is taken from the experimental results of BURET with similar heat exchangers the tested with similar fluids. $LMTD$ is calculated with design conditions. As a result, only areas of the heat exchangers are selected.

3. EXPERIMENTAL SETUP

3.1. BURET Laboratory

In the first part of this section, the existing ORC system in BURET and the components that will be used is explained. Then the design of the microscale ORC, its components, its manufacturing, the data acquisition system will be presented.

BURET was founded in 2015 at Boğaziçi University Sarıtepe Campus by Günay Anlaş and Hasan Bedir using funds from ISTKA. The main objective of this laboratory is developing renewable energy technologies and their applications. An oil heating system, a chiller and an industrial compressor are available in the laboratory. A picture of BURET laboratory can be seen in Figure 3.1.

Figure 3.1. External View of BURET Laboratory.

In this study, a micro scale ORC is built separately but connected to the existing heater, chiller, and compressor systems in BURET. The oil heater system is used as the low temperature heat source. Connections are designed to connect the new ORC system to heater pipes. High pressure air from the compressor is used to operate a pneumatic pump to supply lubricant to the turbocharger. For the heat sink in the cycle cold water from the chiller is used. The existing oil heater, the chiller and the compressor are explained in the next section.

3.2. Existing BURET Laboratory Components

In this section, existing heater, chiller and compressor are explained including the changes and fixes.

3.2.1. Heater

Organic Rankine Cycles are used with low temperature heat sources such as concentrated solar power, geothermal power, waste heat etc. In the BURET laboratory a controllable electric heater is used as the heat source to run the ORC system.

Figure 3.2. Electric Oil Heater.

The electric heater is used to heat a thermal oil; the maximum power of the electric heater is 107 kW and maximum oil outlet temperature is 140 °C [16]. The heater has a thermocouple inside and with a controller connected to it, it can keep the oil temperature around a set temperature with ±2.5 °C accuracy. The thermal oil in the system is driven by an internal gear pump which can pump up to a mass flow rate of 1 kg/s. In addition to which, there is also an expansion vessel in the system with a volume of 60 L which allows the oil to expand to avoid increase in the pressure of the system. The electric oil heater is shown in Figure 3.2.

The heater system is connected to the evaporator of the existing ORC test set-up. In order to connect it to the new micro ORC system, two connection ports are added, and a ball valve is installed at the end of each port. With those changes, it is possible to do experiments with the new ORC set-up developed in this thesis or with the old ORC set-up; the heater system can be utilized by either systems.

3.2.2. Chiller

In ORC heat should be released after the expansion, so that the fluid phase changes to liquid before entering the pump. This is done by the cooling water, which leaves the ORC a few degrees higher temperature than it enters. The chiller cools the heated cooling water coming from the ORC with a refrigeration cycle. The captured heat from the cooling water is released to outside air.

Figure 3.3. Chiller.

The chiller in the Figure 3.3 has a cooling capacity of 107 kW [16]. Currently the minimum temperature that is possible to get with this chiller is 4 °C. If an antifreeze solution is used, lower temperatures are also possible. During this study, some problems in the chiller system were encountered, the water inside the chiller system was cleaned and some electronic components were changed and the problems were solved, and the chiller operated as it should.

3.2.3. Pneumatic Compressor

Normally, there is no need of a compressor in an ORC system other than pressurization of the components for leakage detection, however, in this study, a pneumatic pump is used to pump the lubricant oil for the turbocharger. Further details of the oil pumping system will be explained in the Oil Feeding System section. The existing

compressed air at the BURET laboratory is at 7 bars maximum. However, for higher pressures, a pneumatic booster also exists in the laboratory. With the booster, a maximum pressure of 20 bars can be supplied. The compressed air is firstly filled in a pressure vessel then transferred to the different parts of the laboratory via PVC pipes.

3.3. Micro Scale ORC System

In this section, the components of the ORC system that is designed are explained. The piping and instrumentation diagram is given in Figure 3.4.

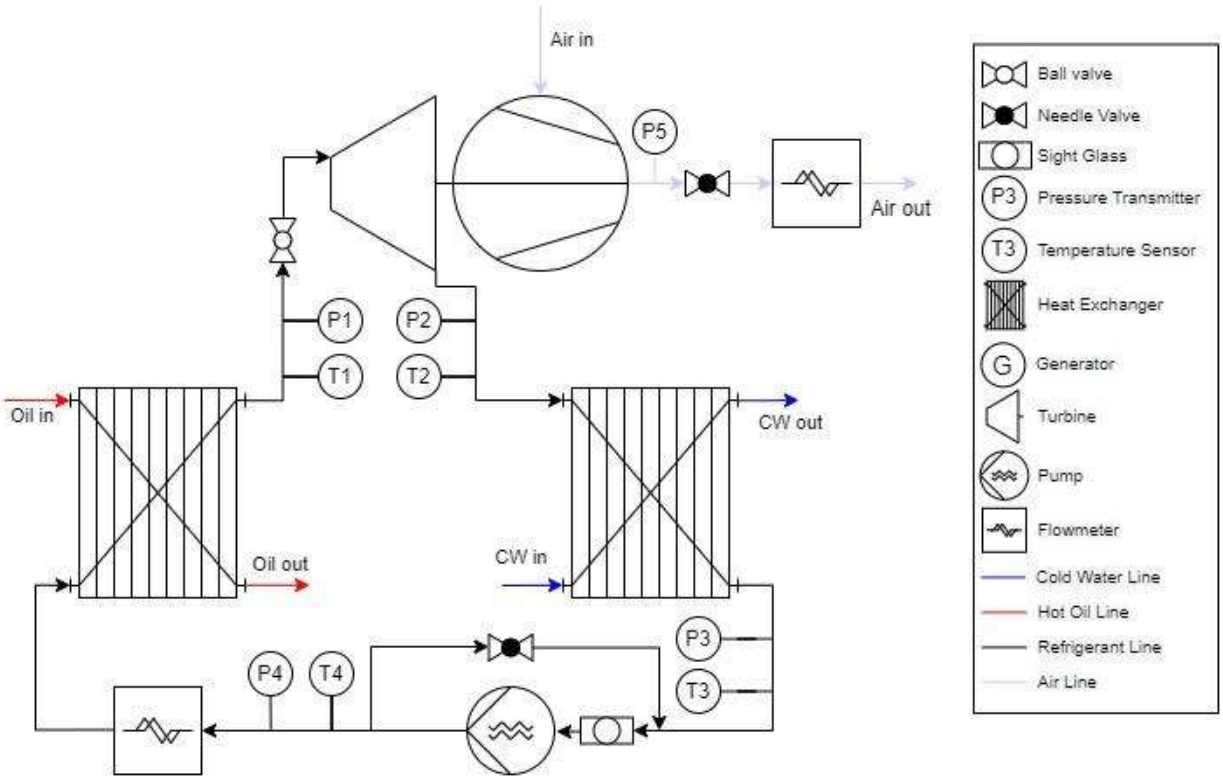

Figure 3.4. P&ID of the Micro ORC System.

3.3.1. Pump

The pump pressurizes the fluid and keeps the fluid in motion. The desired pressure difference in the system is between 4 and 6 bar. The mass flowrate of the working fluid will be between 0.1 and 0.4 kg/s. The pump for the system is SUMAK SM15 peripheral pump. The performance graph of the pump can be seen in Appendix D. In the preliminary analysis, the pump power is found to be around 200 W. This pump is too powerful for the system, but smaller pumps aren't rated for desired pressures. The used pump on the system can be seen in Figure 3.5. The isentropic efficiency, of the pump is calculated as follows

$$\eta_{pump} = \frac{W_s}{W_{act}} = \frac{h_{in} - h_{out,s}}{h_{in} - h_{out}} \tag{3.1}$$

where $h_{out,s}$ is the isentropic enthalpy which has the same entropy with the inlet.

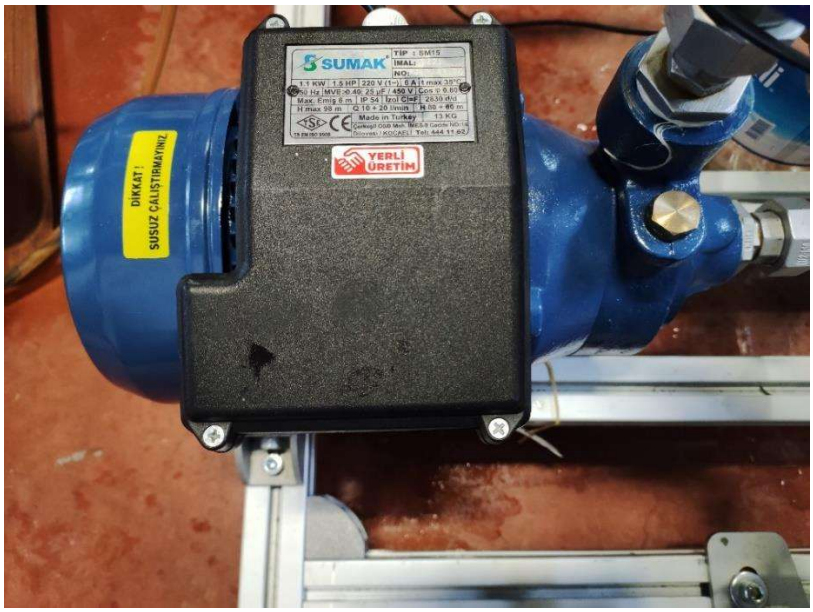

Figure 3.5. SUMAK SM15 Pump.

3.3.2. Turbocharger

The turbocharger consists of a turbine and compressor. The main performance parameter is the power output of the turbine which will be calculated with the steady state energy balance given Equation 1.1. The entropy change for the control volume around the turbine is written as follows

$$\frac{dS_{CV}}{dt} = \frac{\dot{Q}}{T} + \dot{m}_{in}s_{in} - \dot{m}_{out}s_{out} + \dot{S}_{gen}. \tag{3.2}$$

In the isentropic case the equation above simplifies to

$$\dot{m}_{in}s_{in} = \dot{m}_{out}s_{out} \tag{3.3}$$

and the exit state of the turbine has the same entropy with the inlet state of the turbine. However, due to irreversibilities, $\dot{S}_{gen}$ is positive and entropy is higher at the turbine exit. The exit enthalpy of the fluid h_{out}, is also different than the exit enthalpy of the isentropic turbine $h_{out,s}$. The isentropic properties are denoted with the subscripts. The ratio of the actual power output to the isentropic one is defined as the isentropic efficiency of the turbine, which is calculated as shown

$$\eta_T = \frac{W_{act}}{W_s} = \frac{h_{in} - h_{out}}{h_{in} - h_{out,s}}. \tag{3.4}$$

In the turbocharger, the compressor is connected to the turbine directly with a shaft. The power input of the compressor can be calculated using Equation 1.5. Since there is no generator or torquemeter connected to the turbine, the mechanical power output of the system will be calculated from the compressor power. The ratio of compressor power output to the power of the turbine, calculated by Equation 1.1, gives the mechanical efficiency of the turbocharger used in this study.

Turbocharger's turbine part is normally connected to car's air inlet via a metal connection with metal gasket between them. The technical drawings of the connections were

not available, as a result, their connection geometry is obtained by measuring and 3D printing the designed connection. After few iterations a good connection geometry is obtained, and two connection parts are designed to connect the surfaces of tge turbine to 1/2-inch NPT fittings. The connection parts have three main parts: male threaded pipe, butt weld pipe reducer and planar connection surface. The plane connection surface is laser cut from a 10 mm AISI 304 stainless steel sheet, and all parts are welded together. As expected, the welding process bent the planar connection by few millimeters, and although the metal gasket was used, the connection leaked. This is solved by using a thread sealant. Planar parts can be seen in Figure 3.6. Connected form of them can be seen in Figure 3.7.

Figure 3.6. Planar Connection Surfaces.

On the turbocharger, there is a wastegate which is a hole to let some of the exhaust gas go through without going into the turbine to decrease the turbocharger's speed. This way the power generation of the turbine is reduced, and compressor outlet pressure higher than what is demanded are avoided. In ORC experiments because this was not needed and to avoid leakage, the two holes of the wastegate are closed with epoxy.

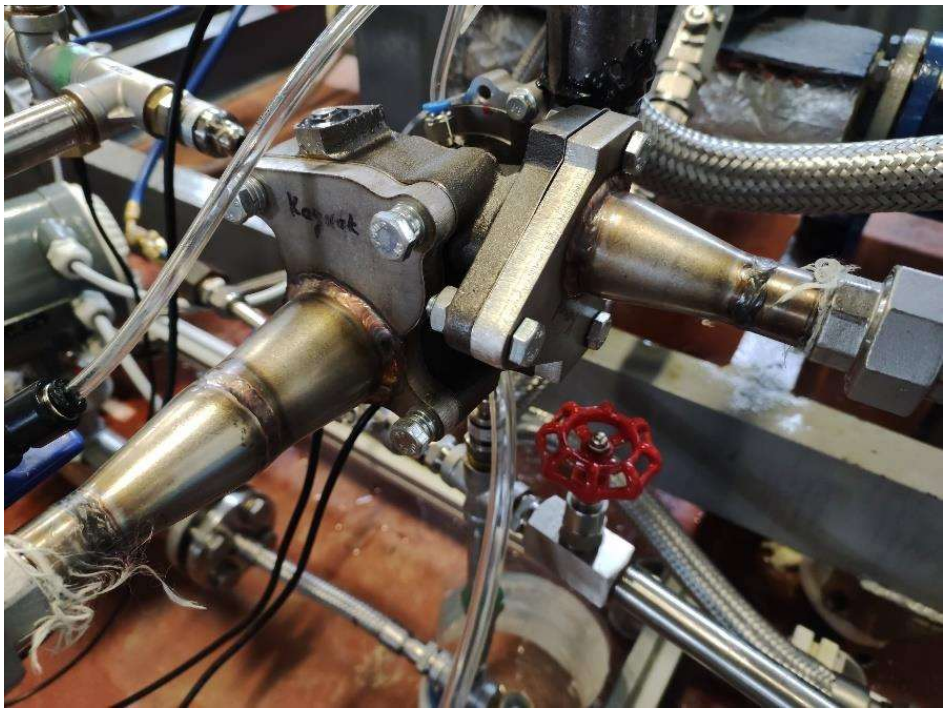

Figure 3.7. Supsan U003 Turbocharger Connected to the ORC.

A pressure sensor, a valve and an air mass flowmeter are added to the compressor part of the turbocharger. This is done via metal connection part, which is produced from a steel rod in the machine shop in the department. One side of the connection part is a hole with 37 mm diameter and the other side is female threaded with G1/2 thread cutter. 37 mm side is connected to the compressor with epoxy. With two nipples and a tee fitting, the outlet is connected to a pressure sensor and a needle valve. This connection is shown in the Figure 3.8.

In addition to the two main parts of the turbocharger, the shaft connecting them needs to be lubricated. An oil feeding system is designed and built in the laboratory which is explained in the Oil Feeding System section.

Figure 3.8. Compressor Outlet Connection.

3.3.3. Heat Exchanger

In the evaporator, pressurized liquid is heated, and it vaporizes. The heat comes from the heat transfer oil which is heated in the oil heater. There is some pressure loss due to viscosity of the fluid, however, it is still considered as high pressure at the outlet.

The condenser cools the working fluid to a sub condensation temperature. The heat is transferred to the cooling water which is non-processed tap water. The water enters the chiller after leaving the condenser.

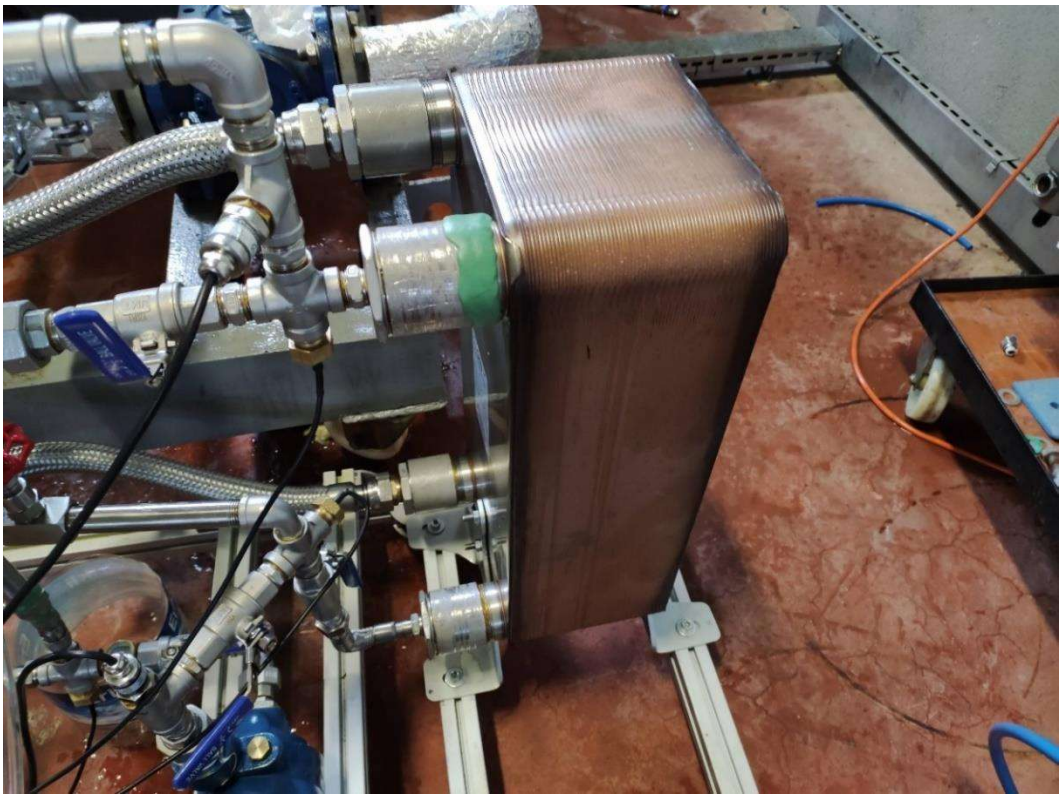

Figure 3.9. MIT MB-07-70 Heat Exchanger as the Evaporator.

Selected heat exchangers for the evaporator and condenser are soldered plate type MIT MB-07-70. A photo of the evaporator can be seen Figure 3.9. There are 70 plates soldered together, the working and heat exchanger fluids flow between those plates. Evaporator and condenser energy balance equations were given in Equations 1.3 and 1.4.

3.3.4. Sensors

The properties of the working fluid at four main points should be measured to calculate the performance of the system. Since sensors that can read both of them are very expensive, two different sensors are used to measure pressure and temperature. In addition to the temperature and pressure, mass flow rate of the fluid should also be known. The Coriolis flowmeter of the existing ORC system is used to measure the flow rate. Finally, on the compressor part of the turbocharger, a mass air flow (MAF) sensor is used. All the measurements are done with a frequency of 1 Hz.

Figure 3.10. DS18B20 Sensor with Brass Casing Connected to the ORC.

To measure temperature, DS18B20 temperature sensor with waterproof casing is used. DS18B20 uses 1-Wire communication interface that makes it possible to connect more than one sensor with a single pin. It measures temperatures from -55°C to 125°C with an accuracy of ±0.5 °C and it is pre-calibrated in the factory. There are four DS18B20 sensors used in the system. A small circuit board is designed to collect their data from a single pin which connects all the identical type pins of the sensors together to the Arduino board. The waterproof metal case of the sensor is 6 mm steel rod, and an additional casing is made using a hexagonal brass rod. A 6 mm hole is drilled at the middle and G1/2 thread is cut at one end, and the original casing is glued to the brass casing with epoxy. Casing, sensor assembly and combined sensor can be seen in the Figure 3.10.

For measuring pressure DFRobot's analog water pressure sensor is used. Although the name indicates that it is a water pressure sensor, it can measure pressure of all non-corrosive fluids. The pressure measurement range is given as 0-1.6 MPa, however, calibration shows that the maximum readable pressure is around 1.1 MPa. Measurement accuracy is 0.5% of full range, so it can detect changes of 5.5 kPa. Sensor works with 5V voltage input and, 0-5V output analog arej used.

Figure 3.11. Pressure Sensor.

The Coriolis flowmeter from ABB of the main ORC system is used to measure the flow rate of the working fluid. This sensor can measure up to 1 kg/s independent of the fluid type. Its accuracy is ±0.4% of measured value. The connection type is DN15 flange. The measurement from this sensor is obtained by the Data Acquisition System (DAQ) of the existing ORC. So, it is not connected to the Raspberry Pi-Arduino DAQ, which will be explained in Data Acquisition System section.

Figure 3.12. MAF Sensor with 3D Printed Connector.

The mass flow rate of the air at the compressor outlet is measured with a Bosch HFM7-4.7 R5 MAF sensor. It has a maximum measuring capacity of 480 kg/h (0.133 kg/s) with an accuracy of ±3% of the measured value. The calibration data are given in the manual, a third-degree polynomial is fitted to the data, and using the fitted curve, voltage values are converted to kg/h. The sensor has five pins and four of them are used for the flow rate measurement. Two of them are for powering the sensor and two are for signal output. However, if the ground of the reference voltage and the power input is not same, the reading is wrong, and the connection is done accordingly. Connection of the MAF sensor to the compressor is done via a 3D printed conical shaped connector which can be seen in the Figure 3.12.

3.3.5. Piping and Fittings

The working fluid is transferred in stainless steel pipes and fittings. Although it seems straightforward, connection of the equipment should be done carefully. Otherwise, leakage occurs, and pressure might be lost.

21.4 mm diameter stainless steel pipe with 2 mm thickness is used. This pipe can be connected with both NPT and BSP type threaded fittings if threaded properly. Even though piping tried to be made with minimum length to avoid pressure drop and heat loss, some threaded long pipes are used in the system. They are cut with pipe cutter and threaded with pipe thread cutter seen in the Figure 3.13. If thread cutting is not done carefully, the pipe may be bent and sealing of the pipe might be impossible.

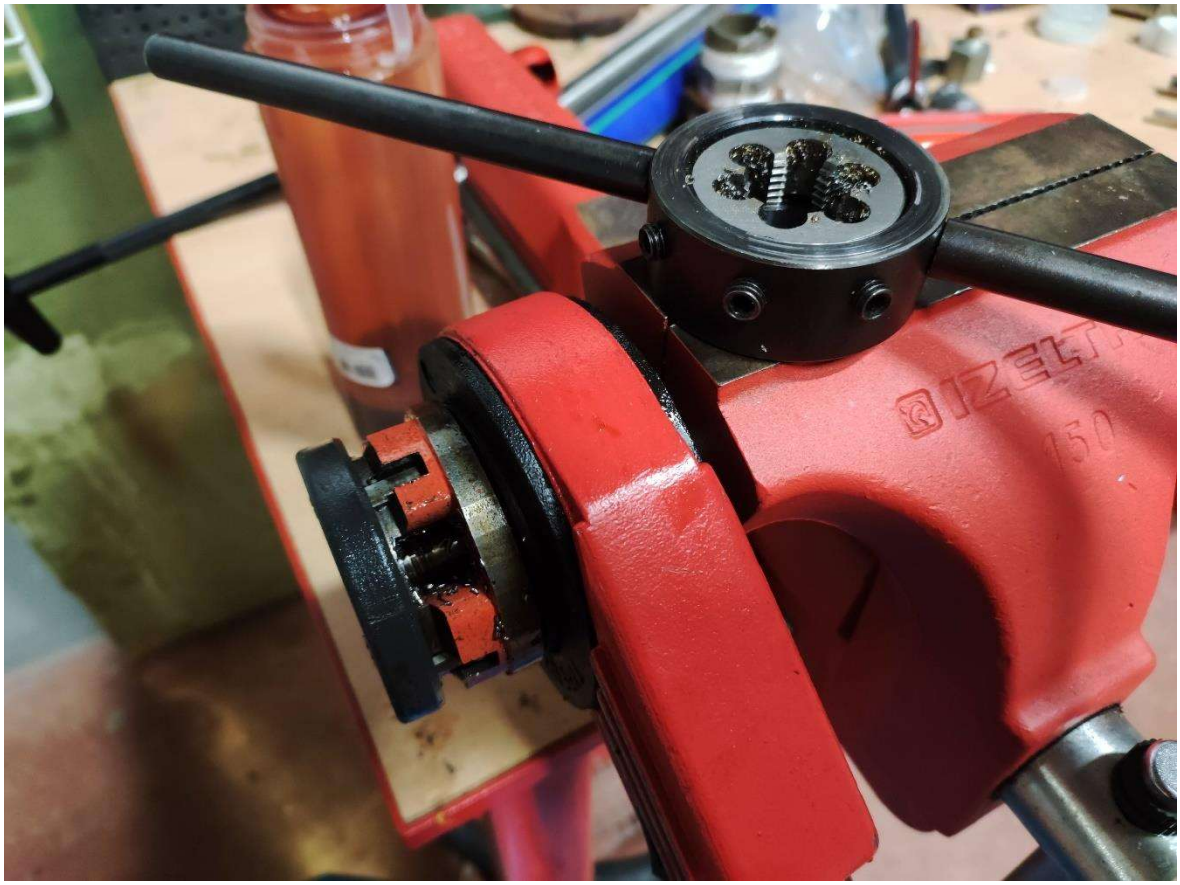

Figure 3.13. Pipe Threader and Threading Die.

Along with the long pipes, some fittings are used to connect the components to pipes. Used fittings can be seen in Figure 3.14.

Figure 3.14. Stainless Threaded Fittings.

In addition to the pipes and fittings, several flexible pipes are used in the system. Steel pipes are rigid and very strong, so they can hold the parts in place and offer great sealing if connected properly. However, if everything is rigid, some parts like Coriolis flowmeter cannot be connected or removed. A flexible pipe is shown in Figure 3.15.

Figure 3.15. Flexible Pipe on Oil Line.

3.3.6. Sealing

Threads on the pipes and fittings connects the parts mechanically but they don't offer perfect sealing. Without using a proper sealant, it is impossible to hold the refrigerant in the system. Before starting any piping, different sealant types are tried, and best ones are selected to use in the actual system. Three types of sealants are used: band type sealants, glue like sealants and hole closers. From band type sealants; PTFE tape, hemp-paste, and sealant cord; Loctite 55 sealant cord is used. Because it can be used safely on properly cut threads, fast application, and it doesn't leak over time with possible readjustments up to 45°. However, the pipes threaded in the laboratory are bent slightly, so the gap on the connection is larger than the sealant cords can seal. It is only used on some nipples on the water and oil lines. On the refrigerant line, it is used only on outlets of the pump because they are painted. Glue like sealants are chemicals that reacts with metals and bonds the parts together. Therefore, they can only be used on metal connections. They are generally classified with their glue strength. However, in our system, the parts should be removable with hand tools if any modifications are needed; only low strength sealants can be used. The more important factor is the gap

filling, rated pressure and temperature rating. The most important factor is gap filling capacity of the sealant. Three different sealants were tried in the laboratory, and it is seen that EMS 5577 pipe sealant is the best for the system with gap filling up to 0.6 mm and easy readjustment in 10 minutes. All of the connections are made with this sealant other than the ones sealed with cord. The last sealant type is hole closer. Two different hole closers are used in the system. Between temperature sensors and their casings, between compressor and connector part, and on the holes of the turbocharger epoxy is used to close the gap between them and glue them together. These connections never leaked but it is impossible to separate them again without breaking. In addition to the epoxy, adhesive putty is used on the holes that were not closed after the whole system is connected. After the connection, it is not possible to remove parts and seal them again. Adhesive putty contains two different colored putties, and they are mixed to start reaction between them. It should be spread over the leak and in three hours, it hardens. Figure 3.16 shows two connections that are sealed with adhesive putty. Since it is nearly impossible to remove it without damaging the original part, its use is minimal, but seal is very strong.

Figure 3.16. Adhesive Putty Over Two Connections.

3.4. Oil Feeding System

Turbochargers rotate at very high rotational speeds of up to 90000 rpm [17]. At these speeds, the operation life of ball bearings shortens. Therefore, in all turbochargers oil fluid bearings are used. The pressure of the oil is higher than the operating pressures of the turbine and the compressor and some of the oil leaks slowly to both sides; this leakage is expected to be minimal. The oil of the turbocharger is pumped with an oil pump which is powered by the engine in automobiles.

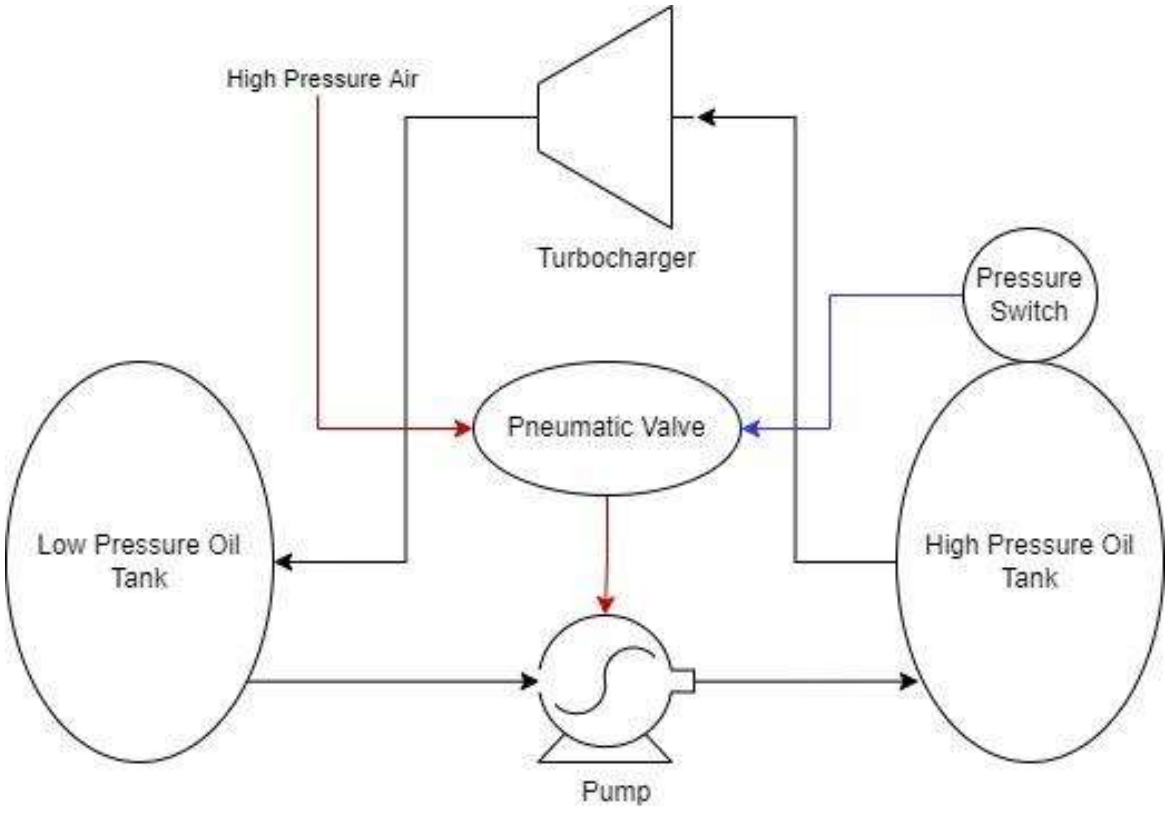

Figure 3.17. P&ID of Oil Feeding System.

In an ORC system, due to low amount of oil requirement of the turbocharger, the oil cannot be directly transferred by a pump, it is kept in a pressurized tank, and it flows through the bearing of the turbine. The oil then flows into a second tank which has less pressure than the first. When the pressure of the first tank is decreases below a certain level, pressure

switch on it activates a pneumatic switch. This starts the pneumatic pump, and it starts to pump the oil from second tank to high pressure tank. With this system, pressure of the oil bearing is kept constant, and the oil pump doesn't work at all time. P&ID of the oil feeding system is shown in Figure 3.17.

3.5. Data Acquisition System

In order to collect data from the system a new Data Acquisition System (DAQ) is prepared. Existing DAQ of the ORC system is very complex and it will be expensive to modify. Similarly, commercially available solutions are too expensive to use. So, a separate unique DAQ hardware and software is built for the micro ORC.

For the hardware, mainly an Arduino Mega development board and Raspberry Pi 4 2GB minicomputer are used. The data are collected with Arduino, processed, and sent to Raspberry Pi via the serial port. The data received are held at the buffer memory of Raspberry Pi. Each second, Raspberry Pi reads the data and sends received info to Arduino. Then Arduino collects data again. Arduino and sensor shield are shown in Figure 3.18.

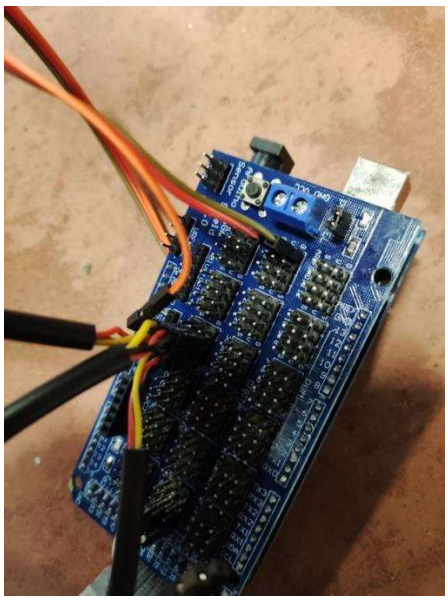

Figure 3.18. Arduino and Sensor Shield.

As softwares two main codes are used, C++ of Arduino and Python Dash in Raspberry Pi. C++ is capable of reading from four DS18B20 sensors, five pressure sensors and a MAF sensor in half a second. If desired, it is possible to calibrate and zero trim the pressure readings. On the Raspberry side, the Dash Environment is used to have a graphical user interface. Dash is a low-code framework that can build a data app on a web browser. It has different components such as graph, interval, dropdown, gauge, indicator, led display, power button and thermometer to show the data read from sensors or calculated in the code. These are coming from Arduino with serial interface and saved in Raspberry Pi.

The web app that can be seen in Figure D.1, has two graphs to plot the pressure and temperature at the inlet and outlet of the turbine. Under the graphs, the pressure and temperature values are shown separately on analog looking gauges to read values easily without using the axes on the graph. On the righthand side, at the top, a power button is used to start and stop data logging. Under that, the temperature and the pressure values at the inlet and the outlet of the pump are shown. Below, two indicator lights are used to show if the fluid is liquid or not. The code behind, uses CoolProp to get fluid properties. At the bottom of everything, a text shows the raw data, taken from the Arduino. Arduino and Python codes can be seen in Appendix B and C.

4. RESULTS AND DISCUSSION

After building the system, several trial experiments are done to check if there is a leakage problem from any component of the set-up. During the trial runs, it is seen that the working fluid leaks through the bearing of the turbocharger more than expected. This leads to further pressurization of the oil feeding system. Above 7 bar, the oil pump starts to leak; several trials are made with a closed oil return line. But oil didn't fill the bearings adequetely. Finally, to solve the problem, opening the air release valve on the low pressure oil tank, allowed oil to flow through the bearing but increased the R134a leakage to the atmosphere.

The experiment is run for 3000 seconds. In the beginning, a total mass 9.5 kg of R134A is filled into the system and at the end of the experiment, 4.5 kg of R134a is recaptured. On average, the working fluid leaked at a rate of 0.0017 kg/s. When compared with the average mass flow rate of 0.2662 kg/s, the leakage is negligible for instantaneous calculations.

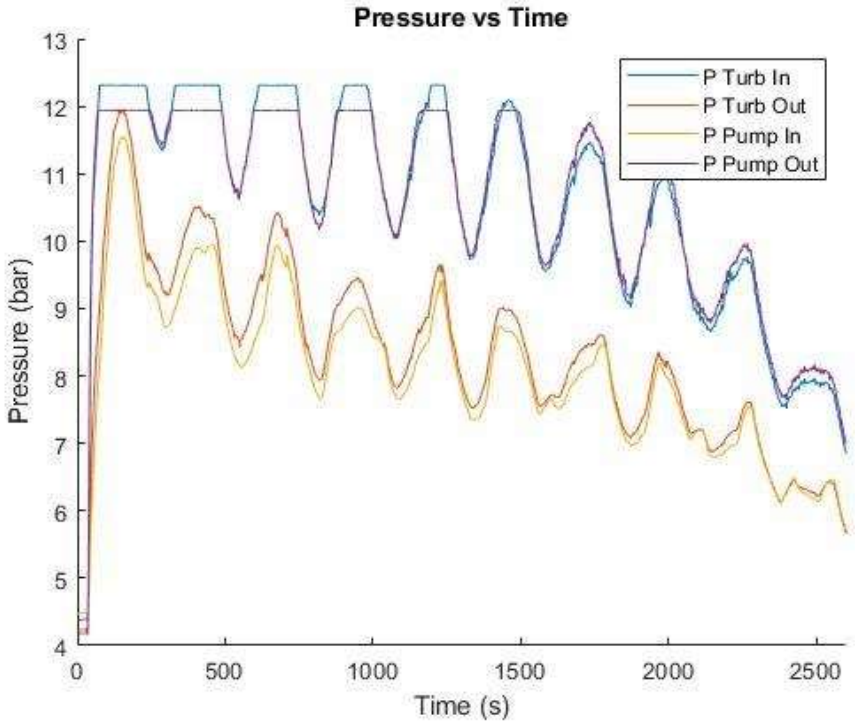

Figure 4.1. Pressure Values (Absolute).

Raw measured pressure values can be seen in Figure 4.1. The pressure sensors are rated for 16 bar gauge pressure, but it is seen in the experiments that they are not able to measure more than 11 bar. Therefore, parts of the experiment, having absolute pressures higher than 11.8 bar are omitted since the pressure output is saturated at that level. This restricted the working range significantly since maximum efficiency and power output are expected to be obtained in that range. However, it was still possible to analyze the performance of the system with the rest of the data. As it can be seen in Figure 4.1, pressure values are oscillating. The reason for the oscillation is the oscillating temperature of the oil heating system (Figure 4.4). Increasing the temperature of the superheated gas with a warmer oil inlet increases the pressure of the system since the refrigerant is ideally recirculated in a closed system. Set temperature for the oil heating system is 70°C. So, when the oil temperature decreases to 70°C, the oil is heated back to 80°C. The general decreasing in pressure motion is due to the leakage from the system. With less amount of fluid in the system, the pressure decreased.

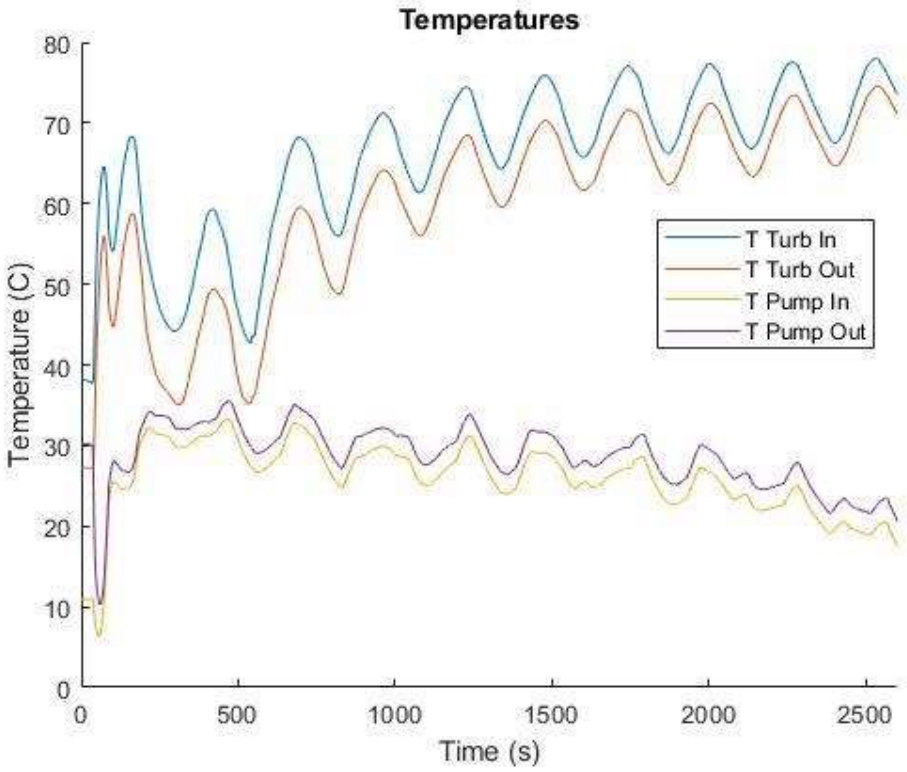

Figure 4.2. ORC Temperatures.

Temperatures measured at turbine and pump inlet and outlets are plotted Figure 4.2. At the beginning, it can be seen that the temperature difference between liquid and vapor side is very different to the thermal inertia of the system. However, after 400 seconds, they show a similar pattern. Power analyses are plotted after that point in Figures Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9.

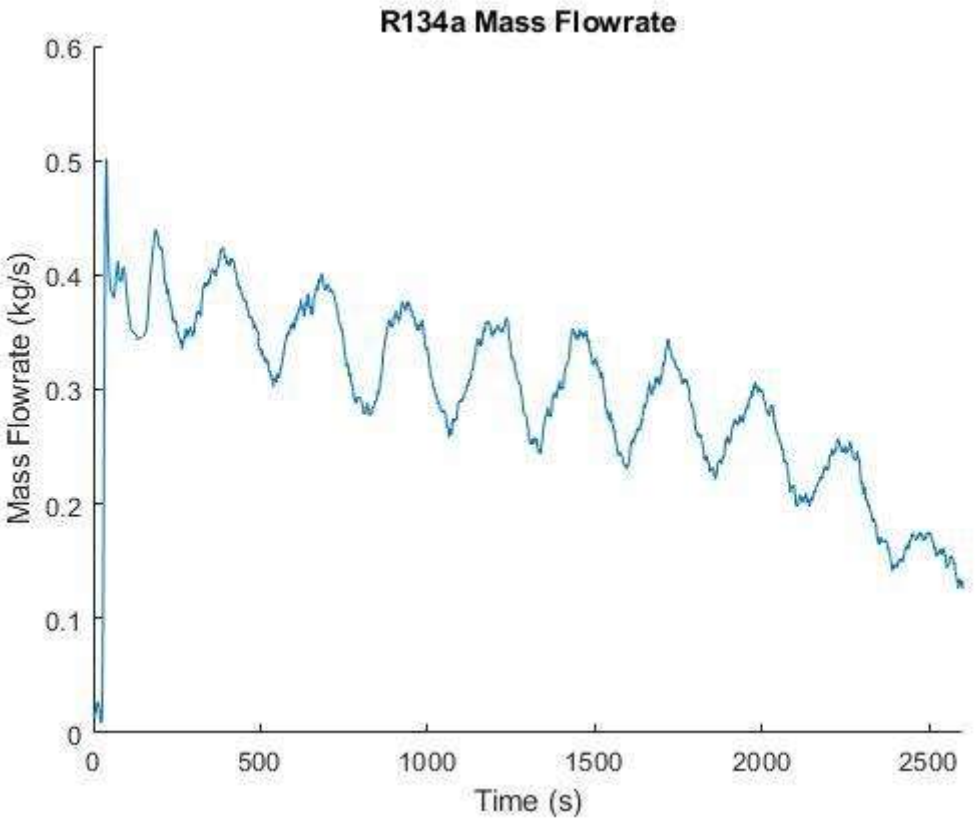

Figure 4.3. R134a Mass Flow Rate.

Measured mass flow rates of the working fluid can be seen in Figure 4.3. Similar to the pressure data, mass flow rates also oscillate. Leakage leads to lower flow rate.

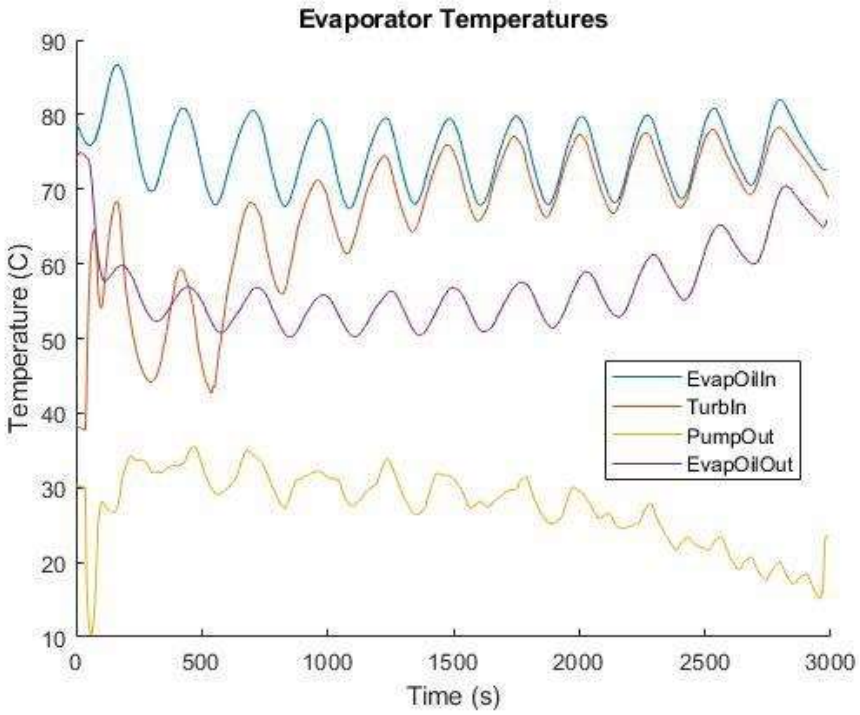

Figure 4.4. The Temperatures at the Evaporator.

Figure 4.4 shows the measured temperatures values at the inlet and exit of the evaporator as a function of time. The oscillation at the oil side temperatures causes the oscillations at all other results. Figure 4.5 shows the calculated heat transfer rate in the evaporator and the condenser from the measured temperature and flow rate data. Their difference is not easy to see, so rate of evaporator heat transfer minus the rate of condenser heat transfer is plotted in Figure 4.6. The unplotted parts are the saturated parts for the pressure sensors; the enthalpy calculations aren't done in those regions. Normally, the rate of heat transfer in the evaporator would be more than the rate of heat transfer in the condenser. But, due to the low efficiency of the pump, pumping power is more than the turbine power. So, more cooling is needed.

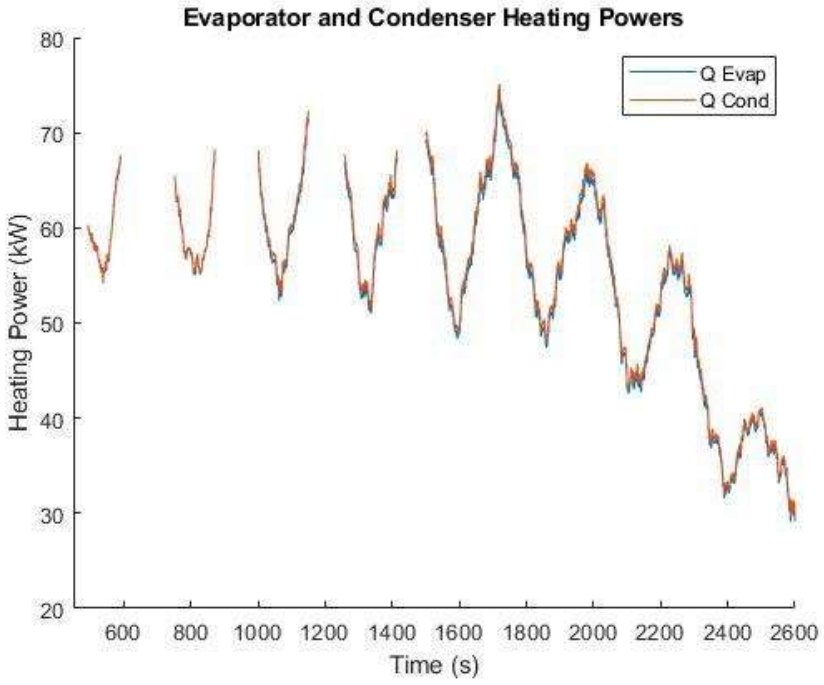

Figure 4.5. Evaporator and Condenser Heating Powers.

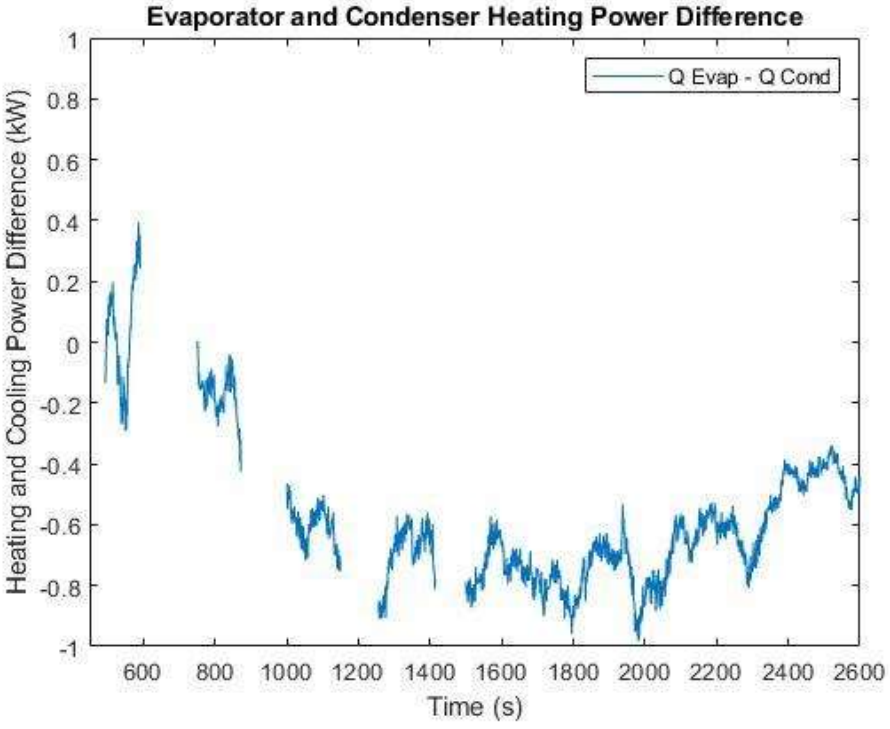

Figure 4.6. Evaporator and Condenser Heat Transfer Rate Difference.

Figure 4.7 shows the power output of the turbine and the power input of the pump to the system. As it can be seen, in the beginning, the turbine power is very close to the pump output and peaks up to 1.55 kW at 587th second. Figure 4.8 and Figure 4.9 show the isentropic efficiencies of the turbine and the pump. It can be seen that the pump's isentropic efficiency is too low, and it leads to a high power input from to the system. The isentropic efficiency of the turbine is very high at the beginning; however, it decreases with time. It may be caused by decreasing flow rate of the working fluid.

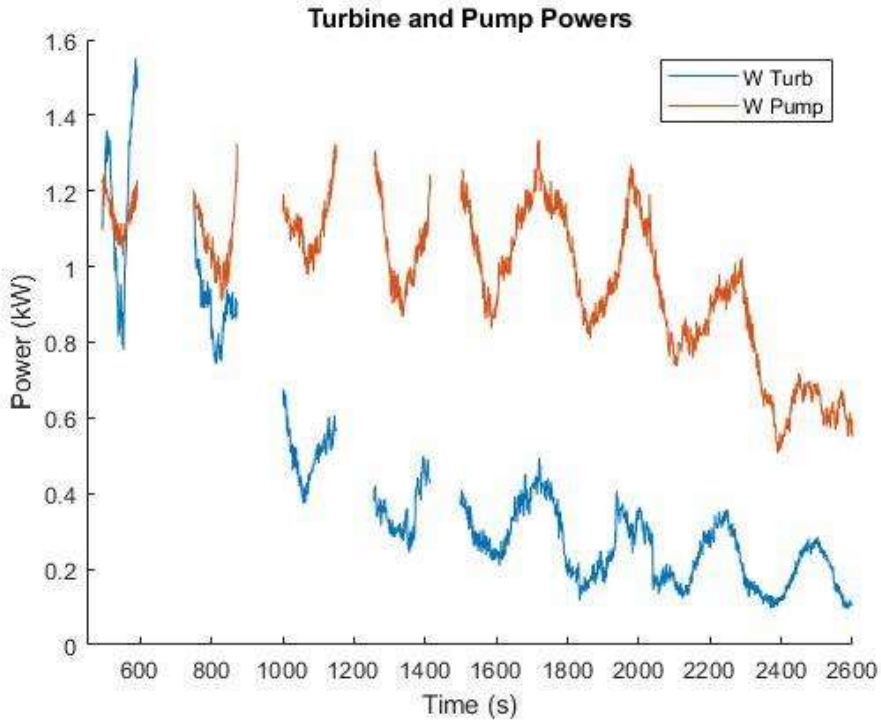

Figure 4.7. Turbine and Pump Powers.

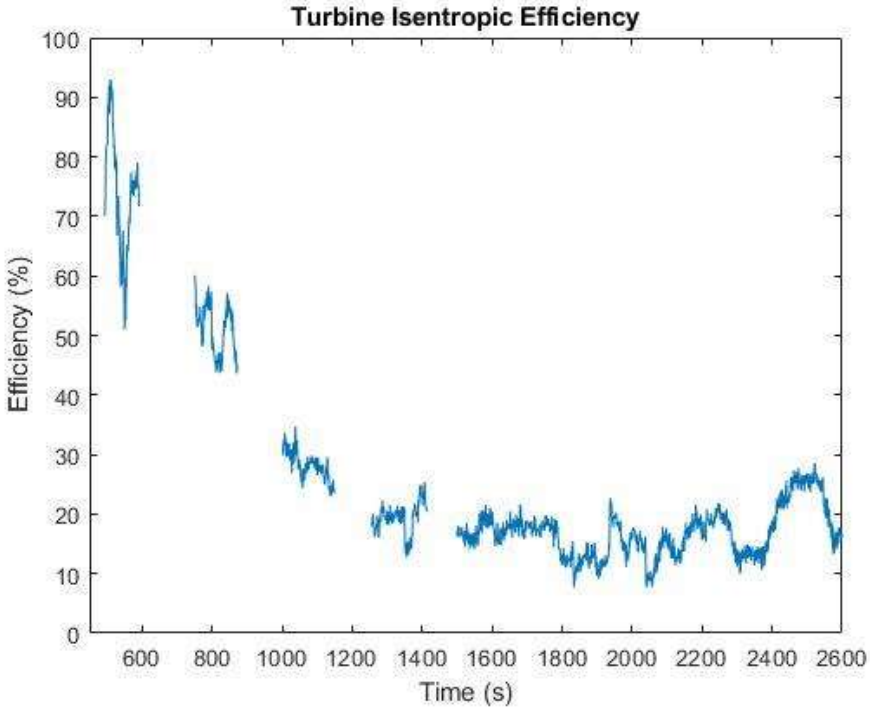

Figure 4.8. Isentropic Efficiency of the Turbine.

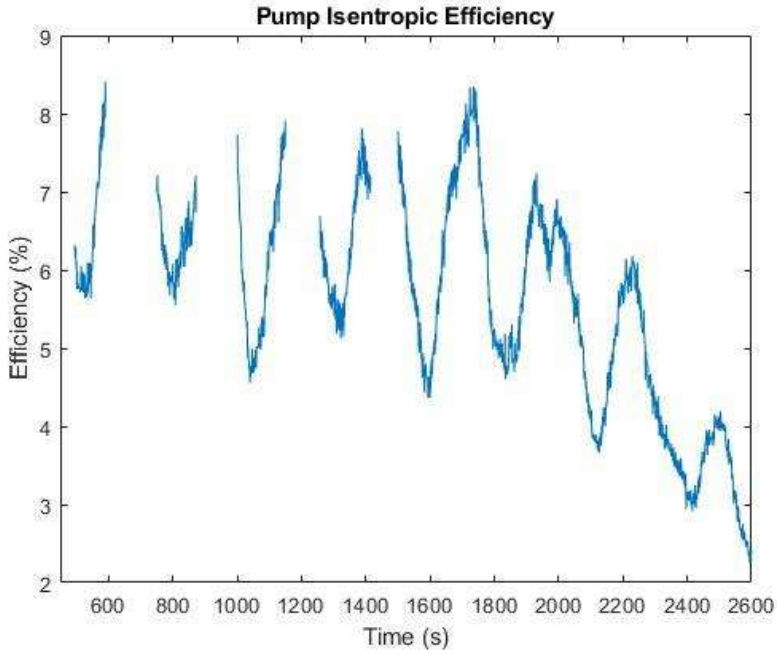

Figure 4.9. Isentropic Efficiency of the Pump.

The Ts diagram of the system at that peak turbine output time can be seen in Figure 4.10. On the Ts diagram, the area below yellow curve shows the liquid-vapor region; righthand side of it is vapor, lefthand side of it shows the liquid region. Above the curve, the fluid becomes supercritical so that there is no separation between liquid and gas. The numbers from 1 to 4 represents the sensor points for the working fluid, and points 5-6 is the heating oil, which is only represented by its temperatures, the entropy is not important for it. The horizontal lines in the liquid-vapor region are the saturation lines at constant pressure.

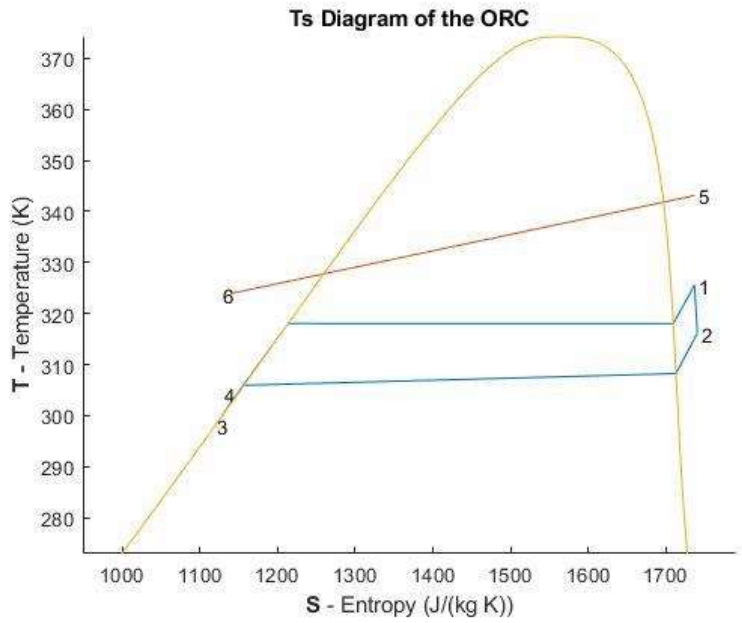

Figure 4.10. Ts Diagram of the System at Peak Turbine Output Point (587th second).

Compressor output pressure, compressor air flow rate, and calculated compressor power of the turbocharger can be seen in Figures Figure 4.11, Figure 4.12, and Figure 4.13. As it can be seen, the compressor output gauge pressure is nearly constant around 0.015 bar; compressor air flow rate, however, oscillates between 0.015 and 0.035 kg/s. Although it is noisier than the other data, it can be seen that the compressor airflow also oscillates with the temperature of the oil heater. Although turbine power output is around expected values, the power output of the compressor is very, low around 30W with a maximum of 33.51 W

(Figure 4.13). As stated in the Turbine and Compressor section, this power is calculated by using energy equation (1.5) with data collected at the compressor outlet. This may be caused by the low volumetric flow rate of the turbine, causing it to rotate at much smaller speeds than it is designed for. Therefore, even though the compressor turns steadily, it cannot create enough back pressure. Although there wasn't any temperature sensor at the compressor outlet, it can be felt that it was around body temperature. So, the power loss heats the compressor air.

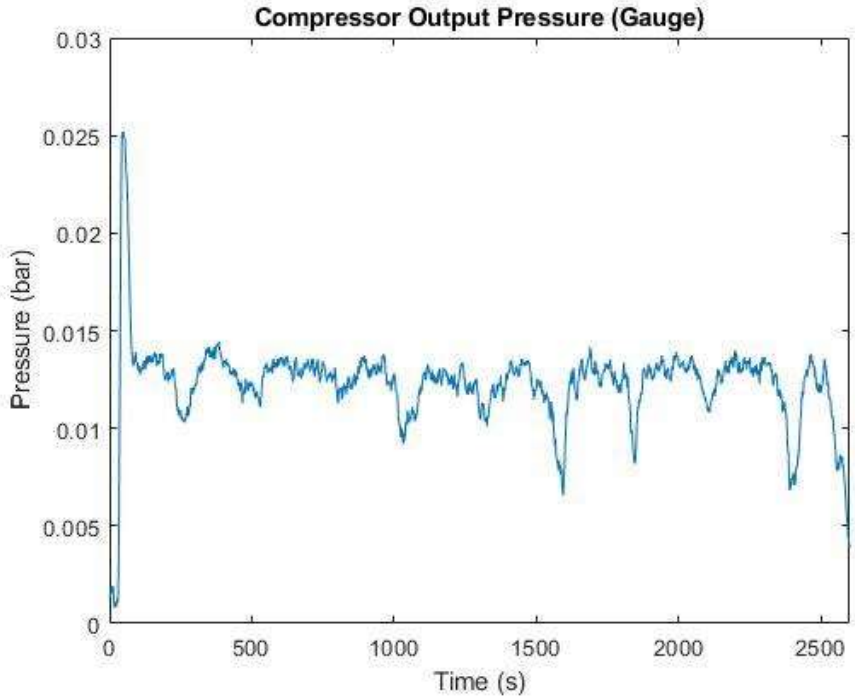

Figure 4.11. Pressure at the Compressor Outlet.

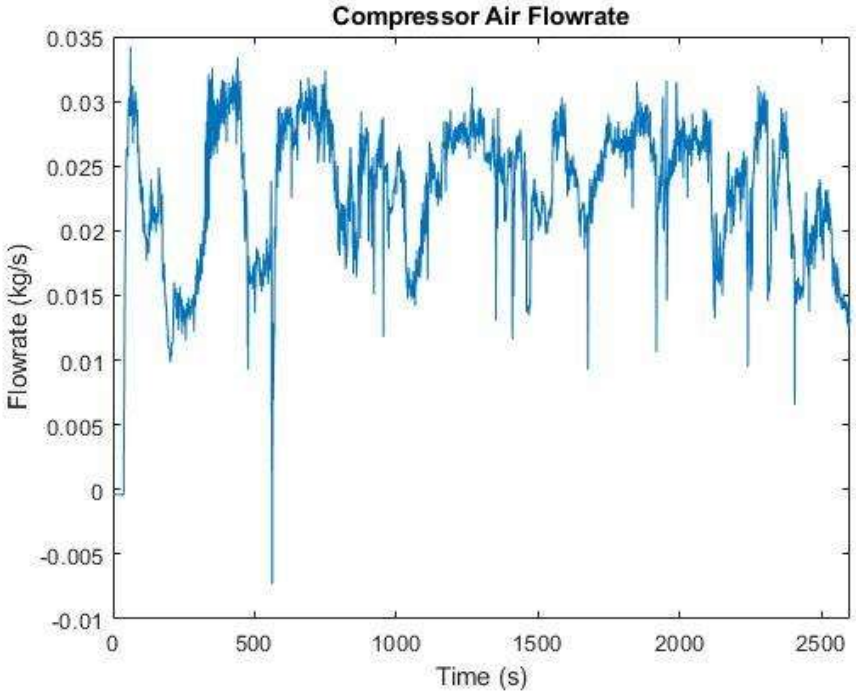

Figure 4.12. Compressor Air Flow Rate.

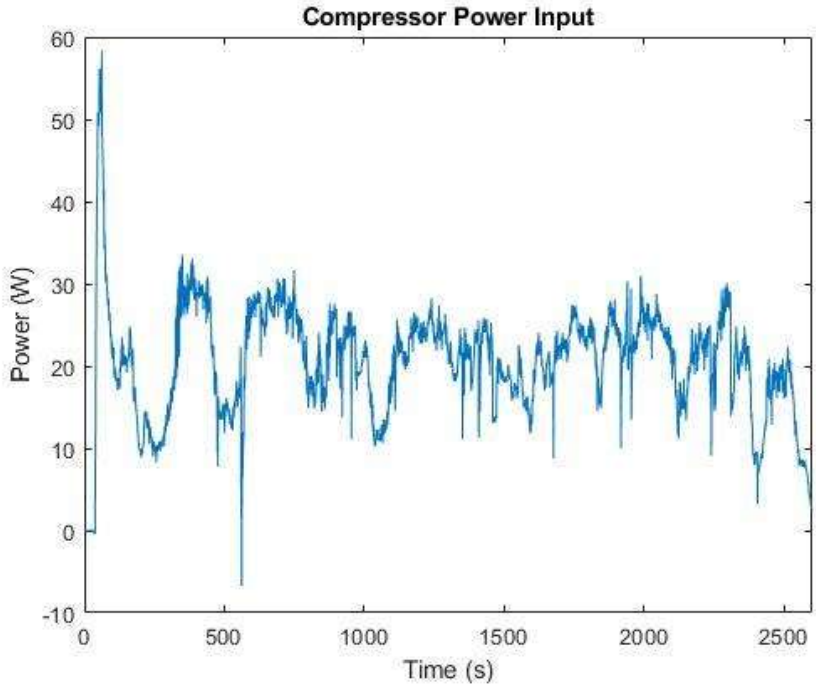

Figure 4.13. Compressor Power Input.

In the experiments, the obtained maximum turbine power is 1.55 kW, calculated by enthalpy differences. If the pressure sensors were working properly, higher values could be measured since the maximum compressor power outputs are seen at the highest-pressure regions. In addition to that, the pump is more powerful than it should be as indicated in the pump section. This leads to a negative cycle efficiency. If a more suitable pump were available and the pressure data were taken in high pressure zones, higher efficiencies and more optimal turbine power measurements would be possible to obtain. The obtained maximum turbine power is higher than the objective of the system. However, the compressor power output is much lower than the turbine power, due to low rotational speeds. There was no temperature sensor at the compressor outlet, and the heat loss to the air through the compressor is not calculated. But it could've been felt that the output temperature was around body temperature; the power loss leads to heated compressor output.

5. SUMMARY AND CONCLUSIONS

In this study, a micro scale ORC was designed, produced, and tested. To select the components, a preliminary analysis with thermodynamic calculations was done using MATLAB. After selecting the components, their 3D locations were determined in Solidworks.

A data acquisition system was designed, programmed, and built to get data from the system. The main two components were Arduino and Raspberry Pi. The data acquisition codes were written in C++ and Python. To see the system behavior instantaneously, a web application was designed using Dash framework using Python. To connect the sensors to Arduino, a sensor hat and a self-made circuit board were used.

As the heat source and the heat sink, the oil heating system and the chiller of the ORC testing rig in the BURET laboratory were used. To connect them to the system, several modifications were made and flexible high pressure pipes were used.

An internal combustion engine's turbocharger was used as the expander of the system. Connection interfaces were obtained by reverse engineering and suitable connection parts were built to connect it both to the ORC system and to the oil feeding system. Due to the low oil flow rate, a custom oil feeding system was designed with pneumatic components. As the power output, the same turbocharger's compressor was used due to expected high rotational speed of the compressor. However, due to the low volumetric flow rate of the working fluid rotational speed of the turbocharger rotates at much lower speeds than its normal speed on an internal combustion engine.

Due to the unexpected behavior of the pressure sensors, limited data were taken at the most efficient part of the experiment. Still, obtained data show a maximum turbine power output of 1.55 kW which is the beyond expected. However, the mechanical power output of the system is not fully consumed by the compressor. The compressor only consumes a maximum power of 33.15 W due to the low rotational speed of the turbine and the compressor together. The difference in the power generated by the turbine and the power consumed by the compressor goes out from the compressor outlet in the form of heat.

As a result, to use an internal combustion engine turbocharger as the expander of an ORC system, leakage and lubrication problems should be solved with more robust solutions along with the connection to a generator. The lubrication problem might be solved by a dynamically controlled oil feeding system along with added an oil separator on the ORC. Other than those, connecting and obtaining power from the turbine is possible, and the turbine can withstand high pressures of the cycle. However, using the compressor to measure the power output is not suitable due to the low speed. Connecting the turbine shaft to a proper power measurement unit or a generator is a must to measure the mechanical output of the turbine.

6. REFERENCES

1. Masson-Delmotte, V., P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J. Matthews, Y. Chen, X. Zhou, M. Gomis, E. Lonnoy, T. Maycock, M. Tignor and T. Waterfield (eds.), "IPCC, 2018: Global Warming of 1.5°C", In Press, 2018.

2. IEA, "IEA Electricity Information", 2019, https://www.iea.org/statistics/electricity/, accessed 26 September 2019.

3. IEA, "Technology Roadmap-Concentrating Solar power", https://www.iea.org/publications/freepublications/publication/csp_roadmap.pdf, accessed 26 September 2019.

4. Georges, E., S. Declaye, O. Dumont, S. Quoilin and V. Lemort, "Design of a Small-Scale Organic Rankine Cycle Engine Used in a Solar Power Plant", *International Journal of Low-Carbon Technologies,* 2013.

5. Dickes, R., O. Dumont, S. Declaye and S. Quoilin, "Experimental Investigation of an ORC System for a Micro-Solar Power Plant", *International Compressor Engineering Conference*, 2014.

6. Bekiloğlu, H., H. Bedir and G. Anlaş, "Multi-Objective Optimization of ORC Parameters and Selection of Working Fluid Using Preliminary Radial Inflow Turbine Design", *Energy Conversion and Management,* Vol. 183, pp. 833-847, 2019.

7. Luca Da, L., M. Giovanni and L. Andrea, "A Mean-Line Model to Predict the Design Efficiency of Radial Inflow Turbines in Organic Rankine Cycle (ORC) Systems", *Applied Energy,* Vol. 205, pp. 187-209, 2017.

8. R. Willem G. Le and S. Adriano, "Recuperated Solar-Dish Brayton Cycle Using Turbocharger and Short-Term Thermal Storage. Solar Energy", *Solar Energy,* Vol. 194, pp. 569-580, 2019.

9. Declaye, S., S. Quoilin, L. Guillaume and V. Lemort, "Experimental Study on an Open-Drive Scroll Expander Integrated Into", *Energy,* Vol. 55, pp. 173-183, 2013.

10. Bracco, R., S. Clemente, D. Micheli and M. Reini, "Experimental Tests and Modelization of a Domestic-Scale ORC (Organic Rankine Cycle)", *Energy,* Vol. 58, pp. 107-116, 2013.

11. Dixon, S. and C. Hall, *Fluid Mechanics and Thermodynamics of Turbomachinery*, 2013.

12. Aungier, R. H., *Turbine Aerodynamics*, New York: ASME Press, 2005.

13. Rowlands, A. and E. Sauret, "Candidate Radial-Inflow Turbines and High-Density Working Fluids for Geothermal Power Systems", *Energy,* Vol. 36, No. 7, pp. 4460-4467, 2011.

14. Munson, B. R., T. H. Okiishi, W. W. Huebsch and A. P. Rothmayer, *Fundamentals of Fluid Dynamic Mechanics*, John Wiley & Sons, Inc., 2013.

15. Çengel, Y. A., A. J. Ghajar, *Heat and Mass Transfer Fundamentals & Applications*, New York: McGraw-Hill Education, 2015.

16. Kardaş, O., *Design, Production and Testing of a Laboratory Scale Organic Rankine System*, M.S. Thesis, Boğaziçi University, 2017.

17. Moraal, P. a. K. I., "Turbocharger Modeling for Automotive Control Applications", *SAE Technical Paper 1999-01-0908,* 1999.

APPENDIX A: TECHNICAL DRAWING OF THE HEAT EXCHANGER

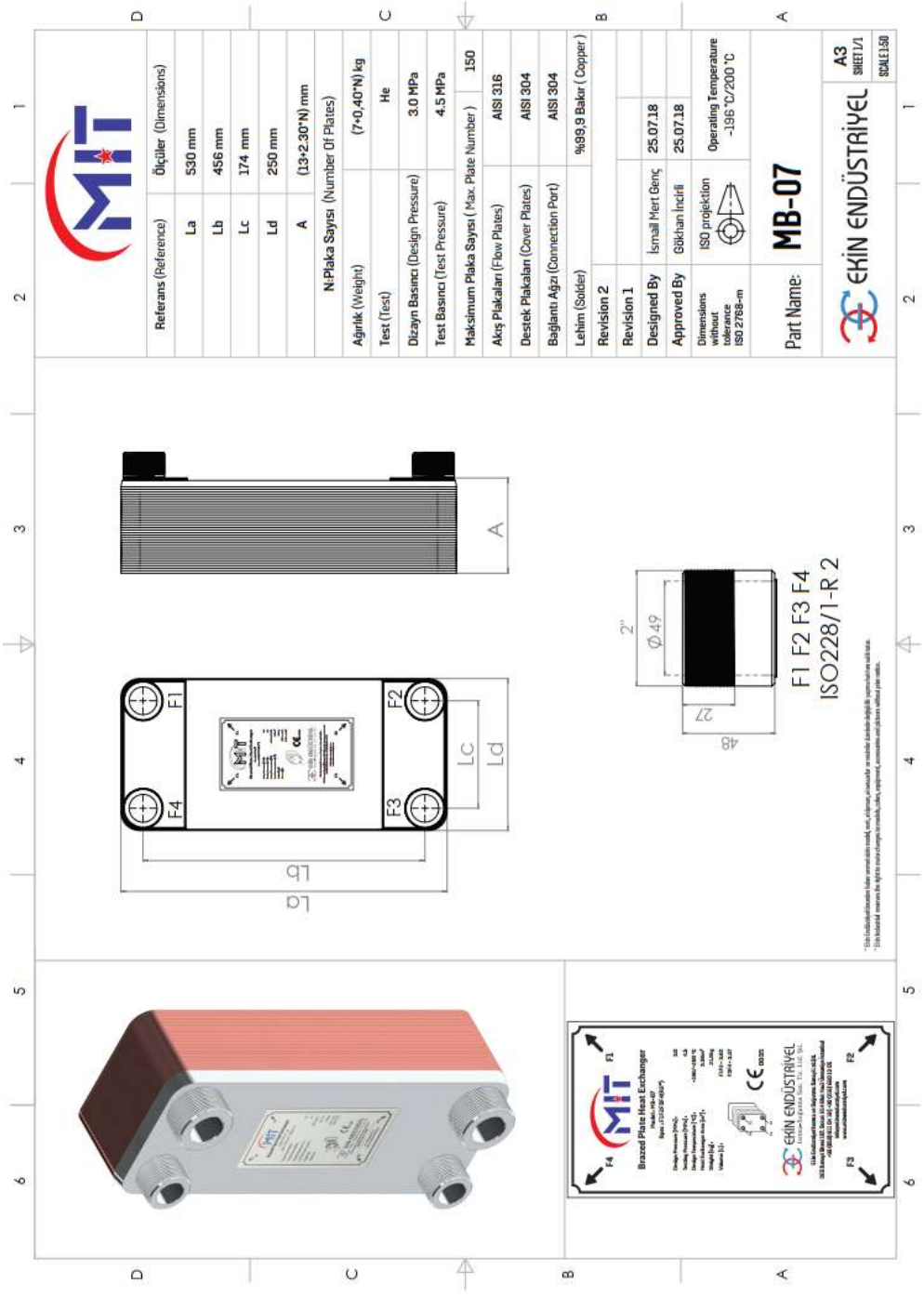

Figure A.1. Technical Drawing of the Heat Exchanger.

APPENDIX B: ARDUINO CODE

```
#include <EEPROM.h>   //Needed for writing pressure calibration data to EEPROM!!!

/*-----( Import needed libraries )-----*/
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS_PIN 45
OneWire oneWire(ONE_WIRE_BUS_PIN);
DallasTemperature sensors(&oneWire);
const int n_of_T_sensors = 4;
DeviceAddress ProbeAdd[n_of_T_sensors] = {{ 0x28, 0x4C, 0x97, 0x76, 0xE0, 0x01,
0x3C, 0x4F },
{ 0x28, 0xEA, 0x78, 0x76, 0xE0, 0x01, 0x3C, 0x14 },
{ 0x28, 0x9E, 0x6A, 0x76, 0xE0, 0x01, 0x3C, 0xAF },
{ 0x28, 0xCA, 0x6D, 0x79, 0xA2, 0x19, 0x03, 0x8D }};



int stepTime = 5;     //Pressure reading step time
int stepNumber = 50;
int calibrationStepTime = 5;
int calibrationStepNumber = 500;

float v_zero_max = 0.6;
float p_zero_min = 2.00;
const int n_of_p_sensors = 5;   // Cannot be larger than 12 for Arduino Mega (Analog
input limit)!!!
float  OffSet[n_of_p_sensors];
float OffSetStd = 4.8;
float  PressureSlope[n_of_p_sensors];
float PressureSlopeStd = 250;
float V[n_of_p_sensors+1], P[n_of_p_sensors]; //MAF Sensor also needs a Voltage
reading. It is done at the same function.
int buttonPin = 22;

//MAF SENSOR
#define RANGE           150    //Measurement Range
#define ZEROVOLTAGE       0.5    //Zero Voltage
#define FULLRANGEVOLTAGE   4.5    //Full scale voltage
#define VREF            5      //Reference voltage

int MAFPin = 0;    // select the input pin for the air meter
float MAFValue = 0;
```

```
void setup()
{
 Serial.begin(9600);       // open serial port, set the baud rate to 9600 bps
 for (int i=0; i<n_of_p_sensors; i++){
  float offset_temp;
  EEPROM.get(i*4, offset_temp);
  if (offset_temp == offset_temp){
   OffSet[i]=offset_temp;
  }
  else {
   OffSet[i]=OffSetStd;
  }
 }
 for (int i=0; i<n_of_p_sensors; i++){
  float preang_temp;
  EEPROM.get((i+n_of_p_sensors)*4, preang_temp);
  if (preang_temp == preang_temp){
   PressureSlope[i]=preang_temp;
  }
  else {
   PressureSlope[i]=PressureSlopeStd;
  }
  //Serial.println(PressureSlope[i]);
 }


//  Serial.println("/** Water pressure sensor demo **/");

 //PrintOffSet();
 pinMode(buttonPin, INPUT_PULLUP);
 voltageReading(stepTime,stepNumber);

 sensors.begin();

 // set the resolution to 12 bit (Can be 9 to 12 bits .. lower is faster)
 for (int i=0; i < n_of_T_sensors; i++){
  sensors.setResolution(ProbeAdd[i], 12);
 }
}

void loop()
{
 sensors.requestTemperatures();
```

```
  voltageReading(stepTime,stepNumber);
  VoltageToPressure();
  //PrintOffSet();
  //PrintVoltages();

  while(Serial.available() == 0) {}
  String Raspinput = Serial.readStringUntil('\n');

//  Serial.print(Raspinput);
  //PrintVoltages();
  if (Raspinput == "Sensor"){
    //printTemperature(ProbeAdd[0]);
    PrintTemps(); //Print previous results
    Serial.print(",");
    PrintPressures();
    VoltageToKGH();
    //Serial.println();
  }
  else if (Raspinput == "1"){
    Serial.println("Zero Calibration Button is pressed");
    zeroCalibration();
  }
  else if (Raspinput == "2"){
    Serial.println("Pressure Calibration Button is pressed");
    pressureCalibration();
  }
  else if (Raspinput == "off"){
    PrintOffSet();
  }
  else {
    Serial.println("Wront input is:");
    Serial.println(Raspinput);
  }

//  Serial.println(V[0],4);

  if (digitalRead(buttonPin) == LOW){
    delay(50);
    if (digitalRead(buttonPin) == LOW){
      Serial.println("Zero Calibration Button is pressed");
      zeroCalibration();
    }
  }
  delay(stepTime);
}
```

```
void voltageReading(int stptm, int stpnmb){
  for (int j = 0; j < n_of_p_sensors+1; j++){
    V[j] = 0;
  }

  for (int i = 0; i < stpnmb; i++) {
    for (int j=0; j < n_of_p_sensors+1; j++){

      V[j] += analogRead(j) * 5.00 / 1024;   //Sensor output voltage


    }
    delay(stptm);
  }

  for (int j=0; j < n_of_p_sensors+1; j++){
    V[j] = V[j]/stpnmb;
    //Serial.println(V[j]);
  }

}

void VoltageToPressure(){
  for (int j=0; j < n_of_p_sensors; j++){
    //Serial.println(V[j+1]);
    P[j] = (V[j+1] - OffSet[j]) * PressureSlope[j]/100+1;
  }
}

void PrintVoltages(){
  Serial.println("Voltages:");
  for (int j=0; j < n_of_p_sensors+1; j++){
    Serial.print(V[j], 4);
    if (j != n_of_p_sensors){
      Serial.print(", ");
    }
  }
  Serial.print(" ");
  Serial.println("V");
}

void PrintPressures(){
//  Serial.println("Pressures:");
  for (int j=0; j < n_of_p_sensors; j++){
```

```
   Serial.print(P[j], 3);
   Serial.print(",");
  }
/*  Serial.print(" ");
  Serial.println("kPa");
  Serial.println();*/
}

void PrintTemps(){
//  Serial.println("Pressures:");
  for (int j=0; j < n_of_T_sensors; j++){
    printTemperature(ProbeAdd[j]);
    if (j != n_of_T_sensors-1){
      Serial.print(",");
    }
  }
/*  Serial.print(" ");
  Serial.println("kPa");
  Serial.println();*/
}

void PrintOffSet(){
  Serial.println("OffSet Voltages:");
  for (int j=0; j < n_of_p_sensors; j++){
    Serial.print(OffSet[j], 4);
    if (j != n_of_p_sensors-1){
      Serial.print(", ");
    }
  }
  Serial.print(" ");
  Serial.println("V");
  Serial.println();

}

void zeroCalibration(){
  float max_v=0;
  for (int j=0; j<n_of_p_sensors; j++){
    if (V[j+1] > max_v) {
    max_v = V[j+1];
    }
  }
  if (max_v>v_zero_max){
    Serial.println("Zero calibration should be done at atmospheric pressure!");
    return;
```

```
  }
  Serial.println("Zero calibration is started");
  voltageReading(calibrationStepTime, calibrationStepNumber);
  Serial.print("Calibrated, Calibration ");
  PrintVoltages();
  for (int i=0; i<n_of_p_sensors; i++){
   OffSet[i] = V[i+1];
   EEPROM.put(i*4, OffSet[i]);
  }
  return;
}

void pressureCalibration(){
  Serial.println("Pressure calibration is started, set the pressure:");
  while(Serial.available() == 0) {}
  String serinput = Serial.readStringUntil('\n');
  double calPres = ::atof(serinput.c_str());

  float max_v=0;
  for (int j=0; j<n_of_p_sensors-1; j++){
   if (V[j+1] > max_v) {
   max_v = V[j];
   }
  }

  if (max_v<p_zero_min){
   Serial.println("Zero calibration should be done at atmospheric pressure!");
   return;
  }

  voltageReading(calibrationStepTime, calibrationStepNumber);
  Serial.print("Calibrated, Pressure Slopes: ");
  //PrintVoltages();
  for (int i=0; i<n_of_p_sensors-1; i++){
   PressureSlope[i] = calPres/(V[i+1]-OffSet[i]);
   EEPROM.put((i+n_of_p_sensors)*4, PressureSlope[i]);
   Serial.print(PressureSlope[i]);
   Serial.print(", ");
  }
  Serial.println();
  return;
}

void printTemperature(DeviceAddress deviceAddress){
  float tempC = sensors.getTempC(deviceAddress);
```

```
  Serial.print(tempC,1);

}

void VoltageToKGH() {
  // read the value from the sensor:
  MAFValue = 13.77*pow(V[0],3) - 63.646*pow(V[0],2) + 133.22*V[0] - 86.755;
  Serial.println(MAFValue);
}
```

APPENDIX C: PYTHON CODE

```
# ORC Dash Site

#imports
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input,Output,State
import dash_daq as daq
import plotly.graph_objs as go
import os
import random
import numpy as np
import dash_bootstrap_components as dbc
import time
# from w1thermsensor import W1ThermSensor
import CoolProp
from CoolProp.CoolProp import PhaseSI, PropsSI, get_global_param_string
from datetime import datetime
import serial

# App Start
app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

#Sensor Start
# sensor = W1ThermSensor()

try:
   ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
except:
   ser = serial.Serial('/dev/ttyUSB1', 9600, timeout=1)
ser.reset_input_buffer()

#This will be true when data is being collected
writing_state = False



def random_color():
   colors=["red", "blue","green"]
   return random.choice(colors)

data_t=[i for i in range(0,51)]
```

```
data_p1=np.random.randint(100000,105000,50)
data_p2=data_p1+1000
data_dict={'P_Turb':data_p1,'T_Turb':data_p2}

options = ["P_Turb","T_Turb"]

ledsize = 24

#Layout ------------------------------------------------------------
app.layout = dbc.Container([
   dcc.Store(id='writing-state', storage_type='session', data=writing_state),
   dcc.Store(id='csv-name', storage_type='session',data=False),
   dbc.Row(html.H1("Berkhan's Data Acquisition System"),style={'width':'90%', 'text-
align':'center'}),

   dbc.Row([
      dbc.Col([dcc.Dropdown(id='dropdown',
         options=[{'label':option,'value':option} for option in options],
         multi=True,
         value=[option for option in options]
         ),
      ],style={'width':'30%','display':'inline-block',
'verticalAlign':'top',"backgroundColor": random_color()})
   ]),

   dbc.Row([
      dbc.Col(dbc.Row(id='graphs'),width=8,),

      dbc.Col([
         dbc.Row([
            daq.PowerButton(
            id='the-power-button',
            on=False
            )
         ],justify="center", align="center"),

         dbc.Row([
            dbc.Col([
               daq.LEDDisplay(
                  id='pump_in_T',
                  label="Pump Inlet T",
                  labelPosition='bottom',
                  value='12:34',
                  size=ledsize
               ),
```

```
            daq.LEDDisplay(
              id='pump_in_P',
              label="Pump Inlet P",
              labelPosition='bottom',
              value='12:34',
              size=ledsize
            ),
            daq.Indicator(
              id='pump_in_Phase',
              label="Pump In",
              labelPosition='bottom',
              value=False
            )
          ]),
          dbc.Col([
            daq.LEDDisplay(
              id='pump_out_T',
              label="Pump Outlet T",
              labelPosition='bottom',
              value='12:34',
              size=ledsize
            ),
            daq.LEDDisplay(
              id='pump_out_P',
              label="Pump Outlet P",
              labelPosition='bottom',
              value='12:34',
              size=ledsize
            ),
            daq.Indicator(
              id='pump_out_Phase',
              label="Pump Out",
              labelPosition="bottom",
              value=True
            )
          ])
        ]),
        #dbc.Row()
    ],width=4,style={"border-style": "none"})

],style={"border-style": "none"},justify='center',no_gutters=True),


dbc.Row([html.Pre(id='live_temp'),dcc.Interval(
    id='inter_comp',
```

```
        interval=1*1000, # in milliseconds
        n_intervals=0)]
    )

])

Row_P = dbc.Row([
    daq.Gauge(
        id='turb_in_P',
        showCurrentValue=True,
        units="Bar",
        value=5,
        label='Turbine Inlet Pressure',
        max=13,
        min=0,
        size=150
    ),

    daq.Gauge(
        id='turb_out_P',
        showCurrentValue=True,
        units="Bar",
        value=5,
        label='Turbine Outlet Pressure',
        max=13,
        min=0,
        size=150
    )],justify="center", align="center"
)

Row_T = dbc.Row([
    daq.Thermometer(
        id='turb_in_T',
        min=5,
        max=120,
        value=20,
        showCurrentValue=True,
        label='Turb In Temp',
        units="C",
        height=150,
        style ={'margin':'15px'}
    ),

    daq.Thermometer(
        id='turb_out_T',
```

```
        min=5,
        max=120,
        value=20,
        showCurrentValue=True,
        label='Turb Out Temp',
        units="C",
        height=150,
        style ={'margin':'15px'}
    )],justify="center", align="center"
)


session_data = [[] for i in range(10)]


#callbacks ----------------------------------------------

#Interval
@app.callback(
    [Output('live_temp', 'children'),
    Output('turb_out_T', 'value'),
    Output('turb_in_T', 'value'),
    Output('turb_out_P', 'value'),
    Output('turb_in_P', 'value'),
    Output('pump_in_T', 'value'),
    Output('pump_out_T', 'value'),
    Output('pump_in_P', 'value'),
    Output('pump_out_P', 'value'),
    Output('pump_out_Phase', 'value'),
    Output('pump_in_Phase', 'value'),
    Output('writing-state','data'),
    Output('csv-name','data'),
    Output('T_Turb','figure'),
    Output('P_Turb','figure')],
    [Input('inter_comp','n_intervals'),
    Input('the-power-button','on'),
    Input('writing-state','data'),
    Input('csv-name','data')])

def tempread(intermilan,pwrbttn,wrt_stt,csv_nm):
    ser.write(b"Sensor\n")
    line = ser.readline().decode('utf-8').rstrip()

    ValuesSTR = line.split(",")
    SensorValues = [float(x) for x in ValuesSTR]
```

```
    for i in range(len(SensorValues)):
        session_data[i].append(SensorValues[i])


    session_x=[i for i in range(len(session_data[0]))]

    data_T1 = go.Scatter(
            x=session_x,
            y=session_data[0],
            name='Turbine_Inlet',
            fill="tozeroy",
            fillcolor="red"
            )
    data_T2 = go.Scatter(
            x=session_x,
            y=session_data[1],
            name='Turbine_Outlet',
            fill="tozeroy",
            fillcolor="blue"
            )

    fig1={'data': [data_T1,data_T2],'layout' :
go.Layout(xaxis=dict(range=[min(session_x),max(session_x)]),
                                        yaxis=dict(range=[0,120]),
                                        title='{}'.format('T_Turb'),
                                        showlegend=False)}

    data_P1 = go.Scatter(
            x=session_x,
            y=session_data[4],
            name='Turbine_Inlet',
            fill="tozeroy",
            fillcolor="red"
            )
    data_P2 = go.Scatter(
            x=session_x,
            y=session_data[5],
            name='Turbine_Outlet',
            fill="tozeroy",
            fillcolor="blue"
            )
```

```
   fig2={'data': [data_P1,data_P2],'layout' :
go.Layout(xaxis=dict(range=[min(session_x),max(session_x)]),
                                        yaxis=dict(range=[0,13]),
                                        title='{}'.format('P_Turb'),
                                        showlegend=False)}

   if pwrbttn==True:
      if wrt_stt==False:
         csv_nm='/home/pi/ORC/Experiment
Logs/'+str(datetime.now().strftime('%Y_%m_%d_%H_%M_%S'))+'.csv'
         csv_file = open(csv_nm,'a')
         csv_file.write("Time,T1,T2,T3,T4,P1,P2,P3,P4,P5,MAF\n")
         csv_file.close()
         wrt_stt=True
      else:
         csv_file = open(csv_nm,'a')
         csv_file.write("{0},{1}\n".format(datetime.now().strftime("%d-%m-%Y
%H:%M:%S"),line))
   else:
      wrt_stt = False

   pump_in_press=(100*SensorValues[6])*1000
   Phase_in=PhaseSI("P", pump_in_press, "T", SensorValues[2]+273.15, "R134a")
   pump_out_press=(100*SensorValues[7])*1000
   Phase_out=PhaseSI("P", pump_in_press, "T", SensorValues[2]+273.15, "R134a")



   return 'Sensor Reading is: {}, Temperature 1 is: {}, Phase is liquid {}'.format(line,
SensorValues[0], Phase_in=="liquid"), SensorValues[1], SensorValues[0],
SensorValues[5], SensorValues[4], SensorValues[2], SensorValues[3],
SensorValues[6], SensorValues[7], Phase_in=="liquid", Phase_out=="liquid", wrt_stt,
csv_nm, fig1, fig2

#dropdown
@app.callback(
   Output('graphs', 'children'),
   [Input('dropdown', 'value')]
   )
def output(data_names):
   graphs = []


   for data_name in data_names:
```

```
        data = go.Scatter(
            x=data_t,
            y=data_dict[data_name],
            name='Scatter',
            fill="tozeroy",
            fillcolor="#6897bb"
            )

        graphs.append(dbc.Col([
            dbc.Row(dcc.Graph(
                id=data_name,
                animate=True,
                figure={'data': [data],'layout' :
go.Layout(xaxis=dict(range=[min(data_t),max(data_t)]),

yaxis=dict(range=[min(data_dict[data_name]),max(data_dict[data_name])]),
                                            title='{}'.format(data_name))},
                                            style={'width': '90%'}
                )),
            Row_T if data_name=='T_Turb' else Row_P
        ],style={"border-style": "none",'width':'30%'}))

    return graphs



#run_server
if __name__ == '__main__':
    app.run_server()
```

APPENDIX D: DASH INTERFACE

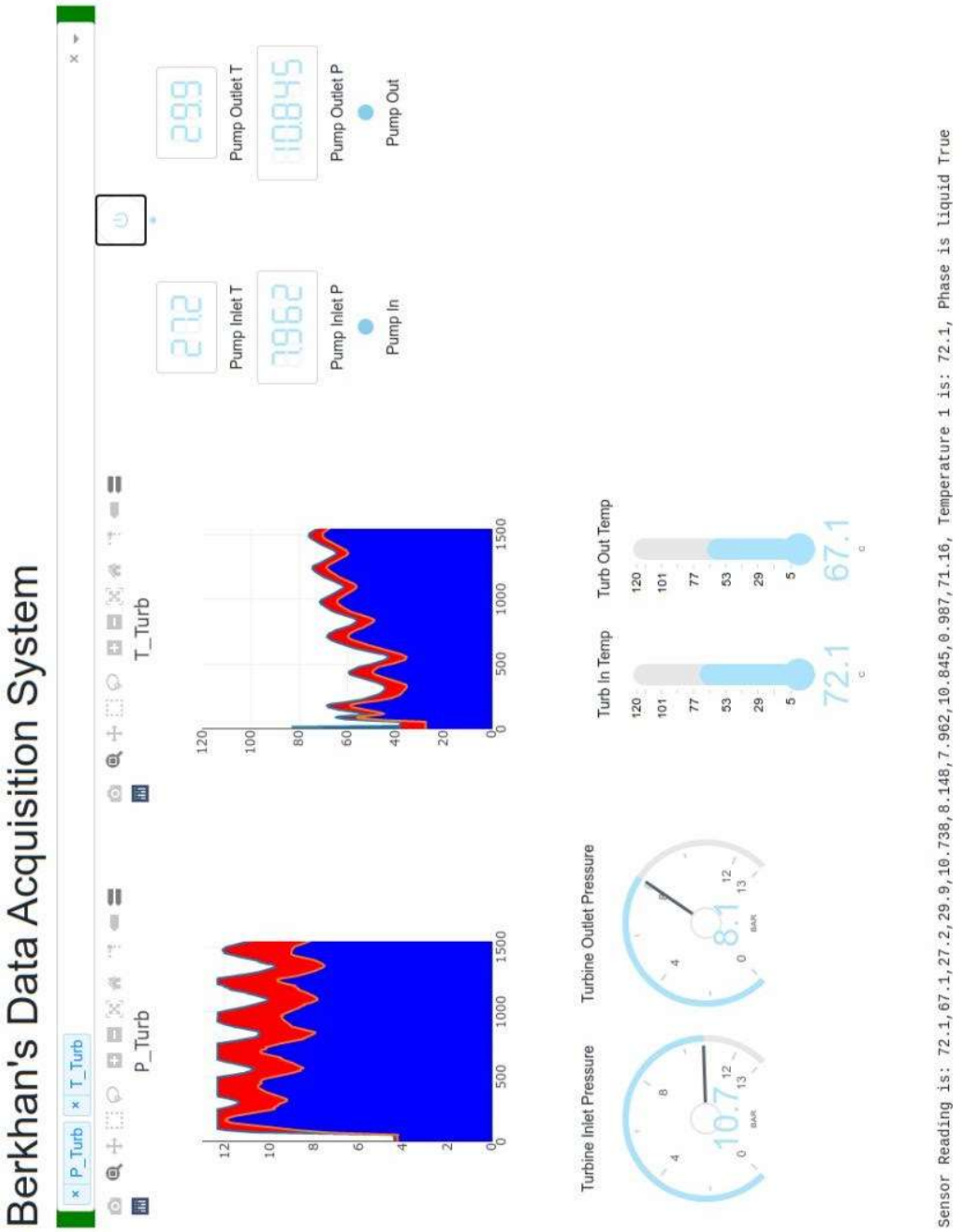

Figure D.1. Dash Web App.

APPENDIX E: PERFORMANCE CURVE OF THE PUMP

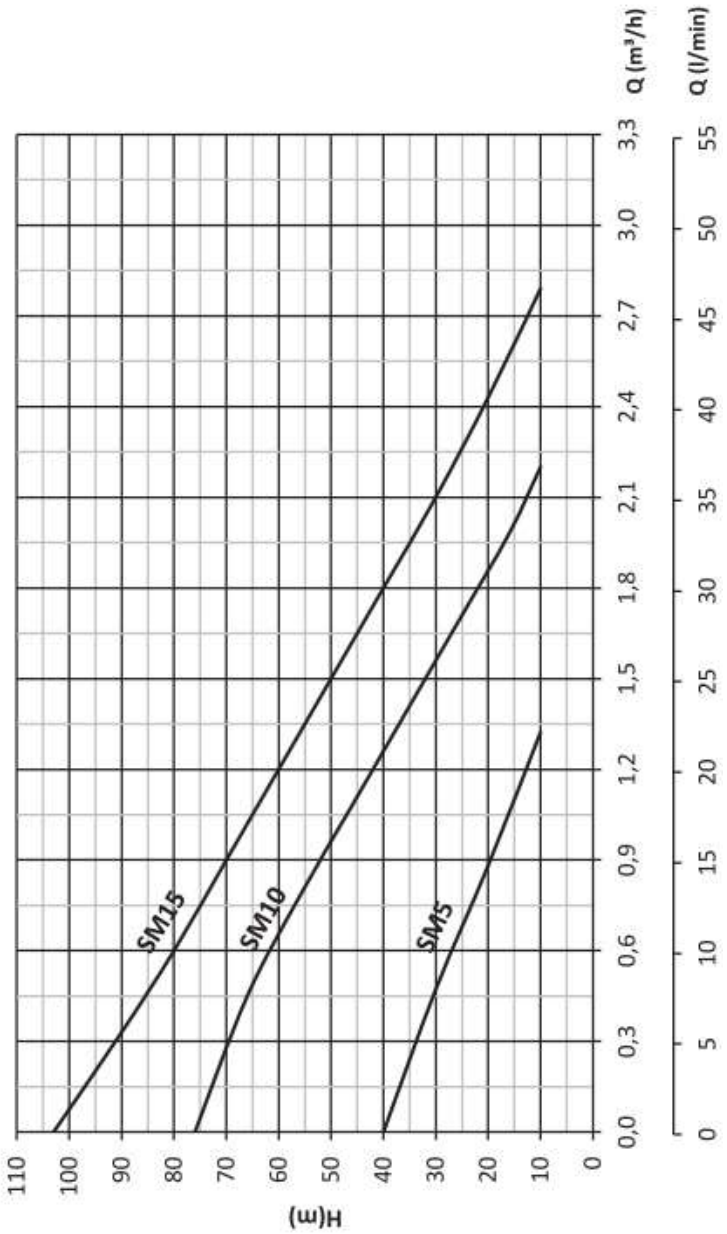

Figure E.1. Pump Performance Curve