Marmara University

Faculty of Engineering

Mechanical Engineering Department

ENGINEERING PROJECT

FINAL

1-D Flow Calculation with Complex Flow Systems

Instructor: Prof Dr. M. Zafer GÜL

Duhan ÇATAL : 15018002

Ömer Şükrü ŞENEL: 150420509

June 2023 Istanbul

ACKNOWLEDGEMENT

Firstly we would like to thank to our supervisor Prof. Dr. Mehmet Zafer GÜL for his guide and support throughout the duration of this study.

We are grateful to Modelon for letting us to use libraries we need in our study. We are particularly grateful to Martin Horacek and Daniel Andersson for their exceptional support and assistance.

Their contributions have significantly enriched the quality and outcomes of this study, and we are thankful for their involvement.

June 2023 Ömer Şükrü ŞENEL, Duhan ÇATAL

CONTENTS

FIGURES LIST

ABSTRACT

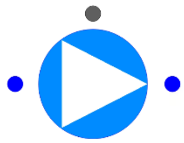
This thesis includes the development of a Modelica-based engine liquid cooling model at Dymola and the simulation of the model to collect data. The Liquid Cooling and Heat Exchanger Library from Modelon was used to build the model.

This thesis focuses on the development and simulation of a Modelica-based engine liquid cooling model using Dymola software. The aim of the study is to create a detailed and accurate model that can simulate the behavior of an engine cooling system and test the model in different engine speeds and in different cooling conditions such as overheating and over cooling and analyze the results.

This paper is also a guide for people who have no experience in Modelica so it contains basic Modelica examples to our model.

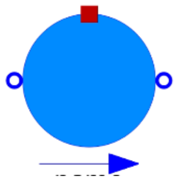
SYMBOLS

Pump



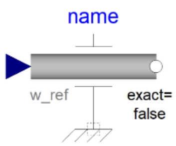
This model describes a centrifugal pump with a mechanical rotational connector for the shaft, to be used when the pump drive has to be modelled explicitly. It uses the Modelon PumpShaft model. In the case of Np pumps in parallel, the mechanical connector is relative to a single pump.

Liquid Volume



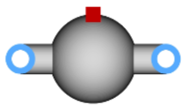
Two-port model of an ideally mixed volume of fixed size. It contains dynamic energy and mass balances, but no hydraulic resistance. Heat capacity and heat transfer to the wall can be included. This is controlled by parameters.

Constant Speed



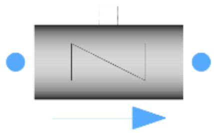
Model of fixed angular velocity of flange, not dependent on torque.

Air Two Port



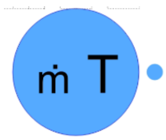
Two-port model of an ideally mixed volume of fixed size. It contains dynamic energy and mass balances, but no hydraulic resistance. Heat capacity and heat transfer to the wall can be included. This is controlled by parameters.

Simple Air Pipe



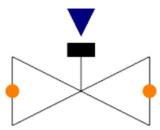
Lumped model of a pipe without storage of mass and energy. A dynamic momentum balance (which usually makes massflow a state) is optional. The flow resistance model is replaceable. Heat transfer is optional and replaceable.

Air Flow Source



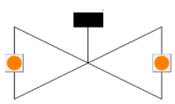
Ideal source of air flow. Prescribes a mass or volume flow rate, temperature and air composition flowing out of the component (positive m_flow) or into the component (negative m_flow). The properties can be assigned fixed parameter values or set from real input signals. When using input signal sources, some different signal units are supported. The unit of the signal must be selected in the parameter dialog.

Exhaust Valve



Valve model for gas flow.

Valve Incompressible

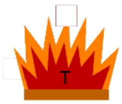


Valve model according to the IEC 534/ISA S.75 standards for valve sizing, incompressible fluids. This model can be used with any low compressibility fluids, such as liquids or gases at very low pressure drops.

Thermal Conductor

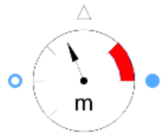
This is a model for transport of heat without storing it. It may be used for complicated geometries where the thermal conductance G (= inverse of thermal resistance) is determined by measurements and is assumed to be constant over the range of operations.

Temperature Source



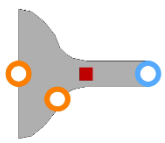
Ideal temperature source. Prescribes a temperature in the heat connector either from a fixed parameter value or from an input signal.

Air Flow Sensor



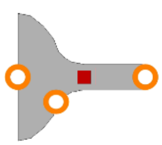
Air flow sensor. Sends the mass flow rate [kg/s] to a Real Output connector. Does not affect the fluid flow rate or properties.

Inlet Manifold



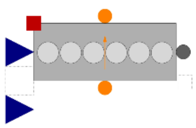
Three-port model of an ideally mixed volume of fixed size. Contains dynamic energy and mass balances, but no hydraulic resistance. This model instantiates MultiportVolume. The airIntakePort connector represents the inlet flow in the Air model representation.

Exhaust Manifold



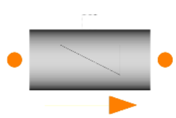
Three-port model of an ideally mixed volume of fixed size. Contains dynamic energy and mass balances, but no hydraulic resistance.

Cylinder



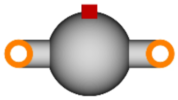
Mean value combustion model for spark ignited engines. The characteristics is mostly governed by equations proposed in Guzzella & Onder (2010), which require relatively few parameters to be set. Volumetric efficiency has a engine speed dependent part, which has to be provided as a table.

Simple Exhaust Pipe



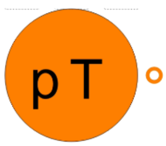
Lumped model of a pipe without storage of mass and energy. A dynamic momentum balance (which usually makes massflow a state) is optional. The flow resistance model is replaceable. Heat transfer is optional and replaceable.

Gas Two Port



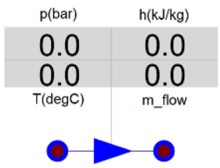
Two-port model of an ideally mixed volume of fixed size. It contains dynamic energy and mass balances, but no hydraulic resistance.

Gas Pressure Source



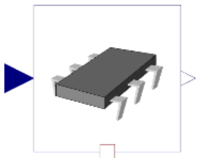
Ideal air pressure source. Prescribes an absolute pressure, temperature and air composition flowing out of the component. The properties can be assigned fixed parameter values or set from input signals. This component can be used either as source (outgoing flow) or sink (incoming flow).

MultiDisplaySensor



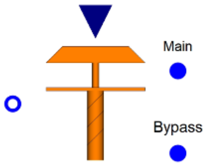
This is a sensor model that visualizes the fluid pressure, specific enthalpy, mass flow rate and temperature in the diagram layer.

TempPIDControl



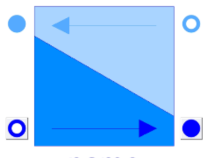
This is a temperature controller, created using the PID controller from the Modelica Standary Library and a temperature sensor.

ControlValve_ThreePort



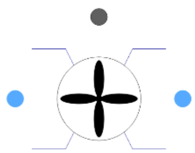
Externally controlled main valve and bypass valve which opens when the main closes. The component splits the inlet flow into two branches determined by two valves. The main valve opening is controlled directly by the input control signal u_main and the bypass valve opens as the main valve closes.

Static Effectiveness Table



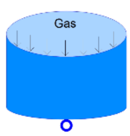
Quasi-static heat exchanger model. The heat exchanger effectiveness is mapped directly from the mass flow rates of both flow channels using a look-up table. Linear interpolation is used.

Fan



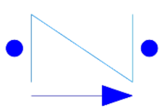
The model describes a centrifugal fan. The fan model is based on the theory of kinematic similarity: the fan characteristics are given for nominal operating conditions (rotational speed and gas density), and then adapted to actual operating condition, according to the affinity laws. Note that the fan affinity laws are not identical to those of the pump.

Expansion Volume



This model of an expansion vessel contains an incompressible liquid phase which is connected to the port and a trapped air mass. Pressure is determined by the gas phase depending on its compression by the liquid.

Liquid Resistance



Flow resistance model for liquids

Quasi-static isenthalpic flow, mass and energy storage are neglected

Pressure difference between the two ports (dp) is caused by friction losses only

1. INTRODUCTION

Internal combustion engines are used in many areas from automobiles to ships, from agricultural machinery to generators. It is very important that the cooling systems work properly so that these engines, which are used in machines that make human life easier in many areas, can give better performance and work efficiently.

Overheating can cause damage to engine components, loss of power and even engine failure. By providing this temperature control, the engine cooling system maintains the optimal operating temperature of the engine and prevents overheating. This increases the durability of the engine and ensures a longer life.

In this thesis we aim to develop a car engine and a cooling circuit of this engine and analyze the act of the component in this circuit by making simulations in the parameters we want. Thanks to Dymola program we are able to analyze how the circuit behaviors in different working conditions.

This study aims to bring a better understanding and development to engine cooling systems. Understanding the operating dynamics of engine cooling circuits can increase engine performance, improve fuel efficiency and extend the life of engine components. It is possible to provide more information by optimizing and improving the model on coming up with new solutions to design motor cooling systems.

2. LITERATIVE REVIEW

ENGINE COOLING SYSTEM

Air-Cooled System

Many small engines and some medium-sized engines are air cooled. This includes most small-engine tools and toys like lawn mowers, chain saws, model airplanes, etc. This allows both the weight and price of these engines to be kept low. Some motorcycles, automobiles, and aircraft have air-cooled engines, also benefitting from lower weight.

Air-cooled engines rely on a flow of air across their external surfaces to remove the necessary heat to keep them from overheating. On vehicles like motorcycles and aircraft, the forward motion of the vehicle supplies the air flow across the surface. Deflectors and ductwork are often added to direct the flow to criticallocations. The outer surfaces of the engine are made of good heat-conducting metals and are finned to promote maximum heat transfer. Automobile engines usually have fans to increase the air-flow rate and direct it in the desired direction. Lawn mowers and chain saws rely on free convection from their finned surfaces. Some small engines have exposed flywheels with air deflectors fastened to the surface. When the engine is in operation, these deflectors create air motion that increases heat transfer on the finned surfaces.

It is more difficult to get uniform cooling of cylinders on air-cooled engines than on liquid-cooled engines. The flow of liquid coolants can be better controlled and ducted to the hot spots where maximum cooling is needed.

Figure 1: Air cooled engine example

Liquid-Cooled System

The engine block of a water-cooled engine is surrounded with a water jacket through which coolant liquid flows. This allows for a much better control of heat removal at a cost of added weight and a need for a water pump. The cost, weight, and complexity of a liquid coolant system makes this type of cooling very r~e on small and/or low-cost engines.

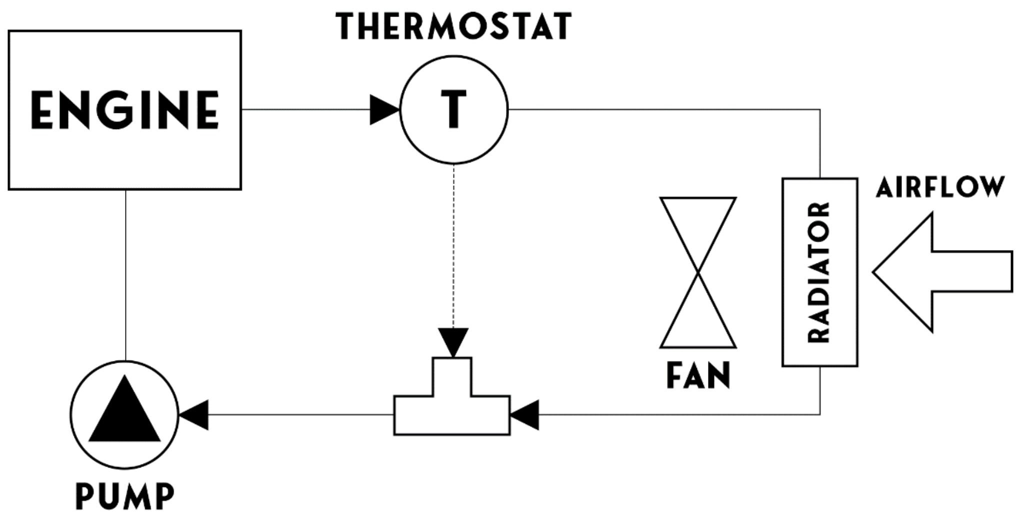


Figure 2: Diagram of an liquid engine cooling system

Very few water-cooled engines use just water as the coolant fluid in the water jacket. The physical properties of water make it a very good heat transfer fluid, but it has some drawbacks. Used as a pure fluid it has a freezing point of O°C,unacceptable in northern winter climates. Its boiling temperature, even in a pressurized cooling system, is lower than desired, and without additives it promotes rust and corrosion in many materials. Most engines use a mixture of water and ethylene glycol, which has the heat transfer advantages of water but improves on some of the physical properties. Ethylene glycol (C2H602), often called antifreeze, acts as a rust inhibitor and a lubricant for the water pump, two properties not present when water is used alone. When added to water, it lowers the freezing temperature and raises the boiling temperature, both desirable consequences. This is true for mixtures with ethylene glycol concentrations from a very small amount up to about 70%. Due to a unique temperature-concentration-phase relationship, the freezing temperature again rises at high concentrations. The desirable heat transfer properties of water are also lost at high concentrations. Pure ethylene glycol should not be used as an engine coolant. Ethylene glycol is water soluble and has a boiling temperature of 197°C and a freezing temperature of -11°C in pure form at atmospheric pressure. Table 10-1 gives properties of ethylene glycol-water mixtures. When ethylene glycol is used as an engine coolant, the concentration with water is usually determined by the coldest weather temperature which is expected to be experienced. Engine coolant cannot be allowed to freeze. If it does, it will not circulate through the radiator of the cooling system and the engine will overheat. A more serious consequence is caused when the water in the coolant expands on freezing and cracks the walls of the water jacket or water pump. This destroys the engine. Even in climates where there is no danger of freezing water, some ethylene glycol should be used because of its better thermal and lubricating properties. In addition to good thermal properties, a coolant should satisfy the following requirements:

Chemically stable under conditions of use

1. Non-foaming
2. Non-corrosive
3. Low toxicity
4. Non-flammable
5. Low cost

Most commercial antifreezes satisfy these requirements. Many of them are basically ethylene glycol with small amounts of additives. A hydrometer is used to determine the concentration of ethylene glycol when it is mixed with water. The specific gravity of the mixture is determined by the height at which the calibrated hydrometer floats. Charts such as Figure 3 can be used to determine the concentration needed. The coolant in the model is Ethylene Glycol 40% for an efficient heat transfer.

ETHYLENE GLYCOL–WATER MIXTURES					
% ETHYLENE GLYCOL by Volume	SPECIFIC GRAVITY at 101 kPa and 15°C	FREEZING POINT at 101 kPa		BOILING POINT at 101 kPa	
		°C	°F	°C	°F
0	1.000	0	32	100	212
10	1.014	−4	24		
20	1.029	−9	15		
30	1.043	−16	3		
40	1.056	−25	−14		
50	1.070	−38	−37	111	231
60	1.081	−53	−64		
100	1.119	−11	12	197	386

Figure 3: Properties of Antifreeze Solutions

Liquid Cooling Systems		Air Cooled Systems	
Advantages	Disadvantages	Advantages	Disadvantages
• More suitable for engines that require high power • More effective cooling	• Complex structure • More components. • Fluid leakage or pump failure problems.	• Simpler structure • Fewer components. • Lighter and takes up less space.	• Less effective cooling • Risk of engine overheating at high temperatures.

Examples for usage

- Modern cars, trucks and motorcycles with powerful engine
- Vintage motorcycles, Small power units

EXHAUST GAS RECIRCULATION

In all modern automobile engines and many other engines, some of the exhaust gas (EGR) is recycled back into the intake system to dilute the incoming air. This reduces the amount of nitrous oxide in the exhaust by reducing combustion temperatures inside the engine. Depending on how the engine is running, about 20% of the exhaust gases are routed back to the intake manifold. This method not only reduces the incoming air, but also heats the incoming air, reducing its density.

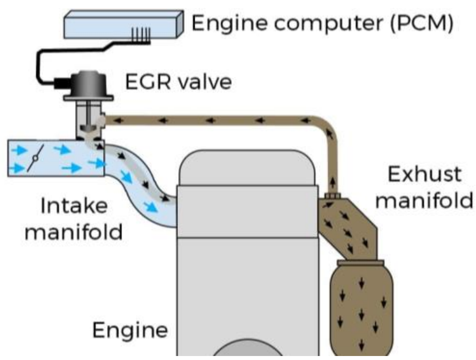


Figure 4: EGR Diagramh

DUAL LOOP COOLING SYSTEM

Some vehicles use dual loop cooling systems for better thermal management and more efficient cooling performance. Instead of the normal cooling cycle, 2 cycles are used. High temperature radiator only cools the engine. the engine coolant will be cooled in the high temperature radiator and will returned to the engine

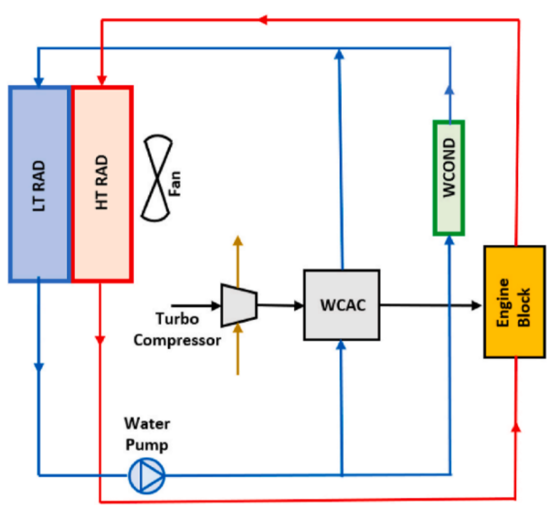


Figure 5: Dual Loop Cooling System Diagram

On the other hand, the low temperature cycle cools subsystems such as intercooler condenser and transmission oil, which were previously cooled by air flow and fan, with water cooled charge air cooler.

Engine is heating the coolant at nearly 90 ◦C on the high temperature cycle but in the low temperature cycle it remains around 60-70 C. So this system allows us to achieve more efficient cooling without mixing the coolant of the two systems. In new model vehicles, there is a subsystem that needs to be cooled more in the in-vehicle systems (battery, electronic components, etc.). These subsystems are connected to the Low Temperature Cycle, providing more efficient cooling and increasing the efficiency of the vehicle. At the same time, it provides a design advantage since low temperature connected systems do not need to be placed in front of the radiator.

WHAT IS DYMOLA

Dymola is a commercial modeling and simulation environment based on the open Modelica modeling language. Large and complex systems are composed of component models; mathematical equations describe the dynamic behavior of the system.

The automotive applications fall into three main categories. The engine and drive train are modeled using the Engines and Powertrain libraries. The flexibility of the open Modelica language is particularly suitable for modeling hybrid or alternative drive trains using the Battery, Brushless DC Drives and Electrified Powertrains libraries. Modal bodies or flexible shafts are available through the Flexible Bodies library. Engine and battery cooling is supported by the Cooling library, which can be combined with the HVAC library. The Human Comfort library adds models of occupant comfort for complete vehicle thermal modeling. Controller components are available in the Modelica Standard Library. [7]

3. EXAMPLES

3.1) HEATING SYSTEM

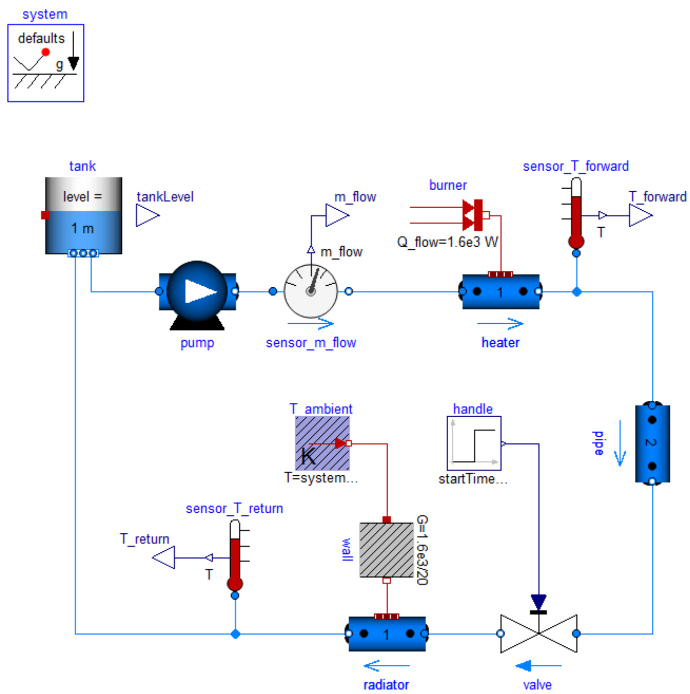
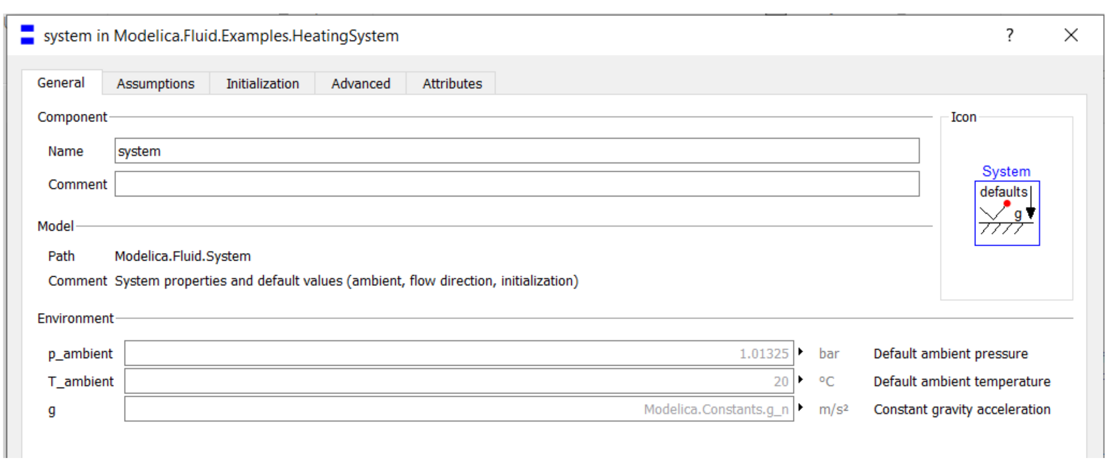


Figure 6: Heating System Example

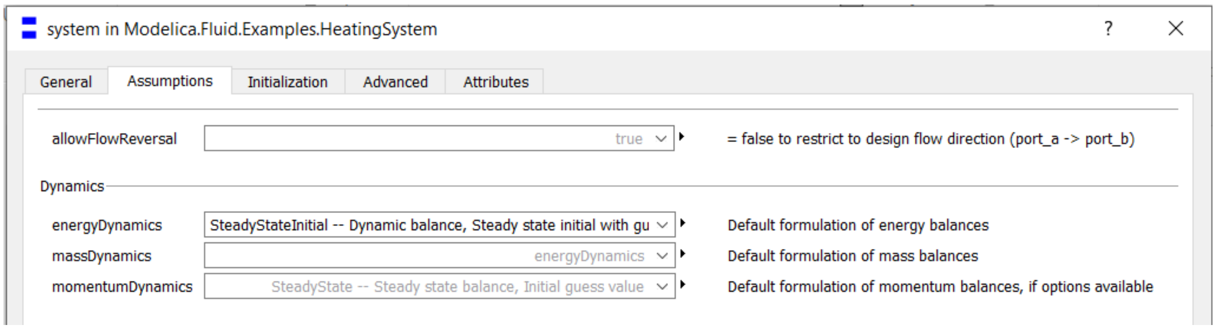
This example is simplified version of an engine cooling system engine is simplified as burner giving a constant heat to the system and radiator pipe cool the liquid with conduction.

BUILDING THE SYSTEM

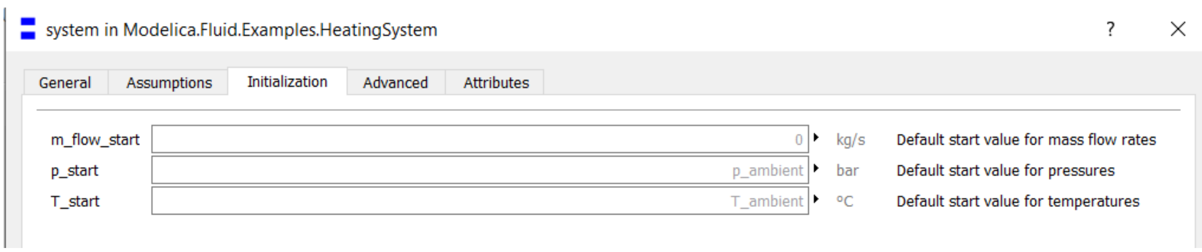
First, we determine the system properties. For this, we place the system block on the diagram from Modelica>Fluid>System path.



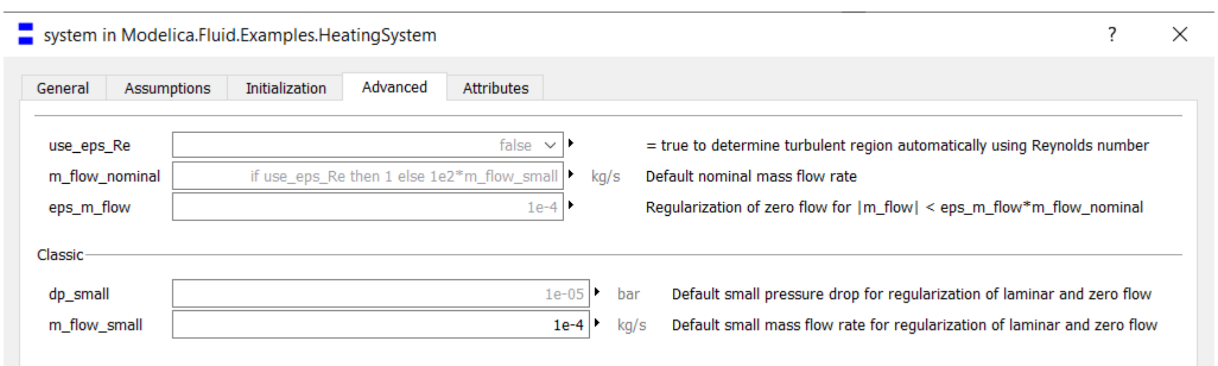
In General tab we take p_ambient, g and T_ambient values as default. The units of these values are also given next to them. Here we can also change the name of the block. We will use the parameters we set in the system block in other blocks by referring to the system block directly.



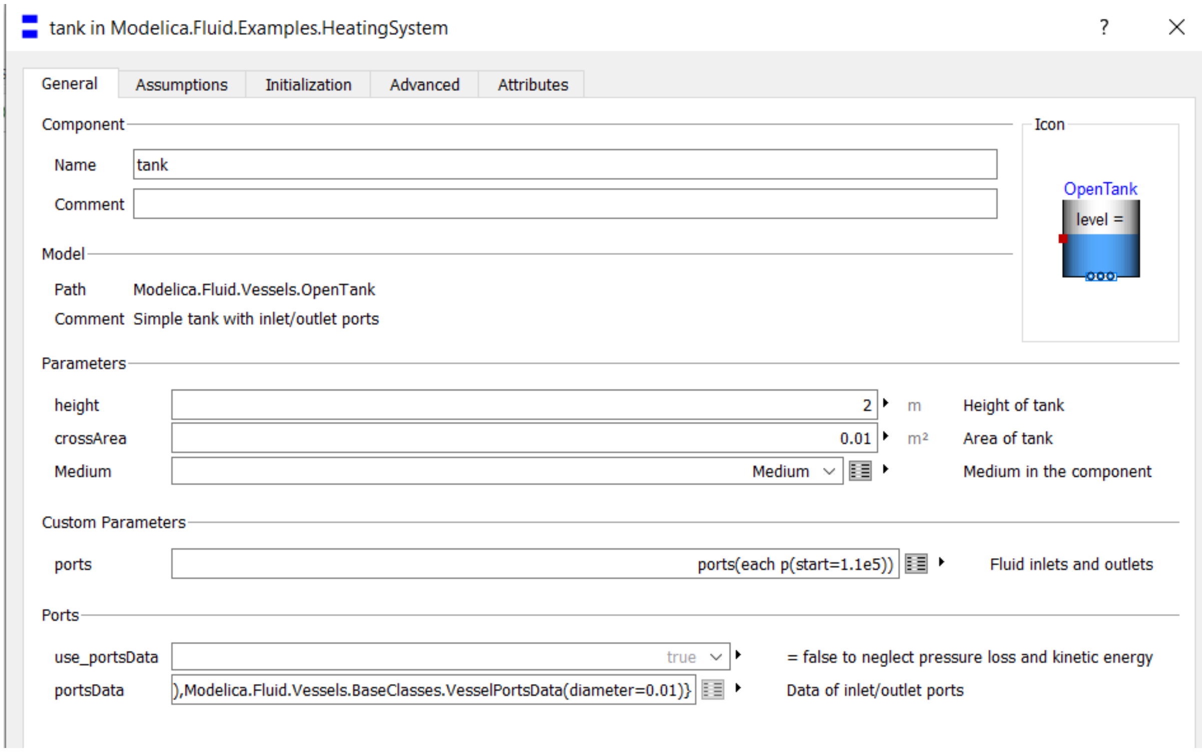
Initialization tab lists the initial values in the component. p_start and T_start taken as p_ambient and T_ambient.



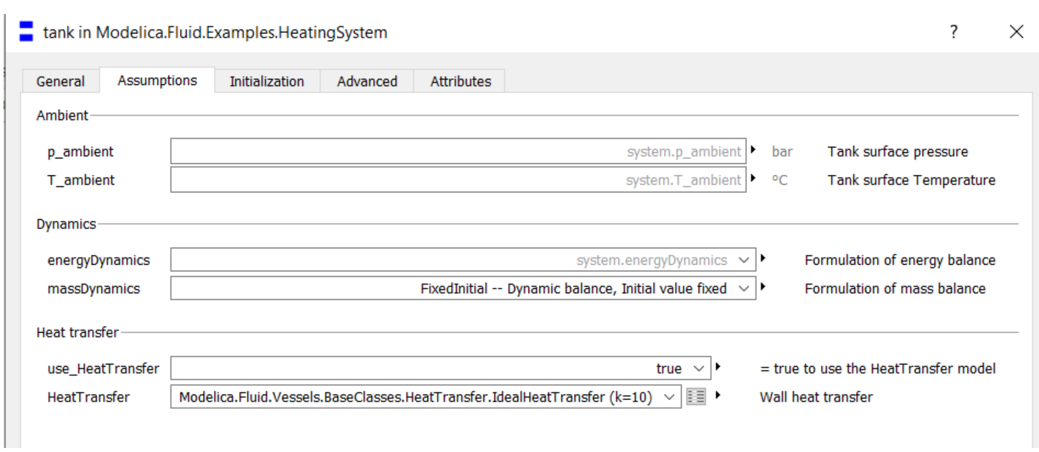
In Advance tab we also define default parameters as dp_small and m_flow_small which we will use at other blocks.



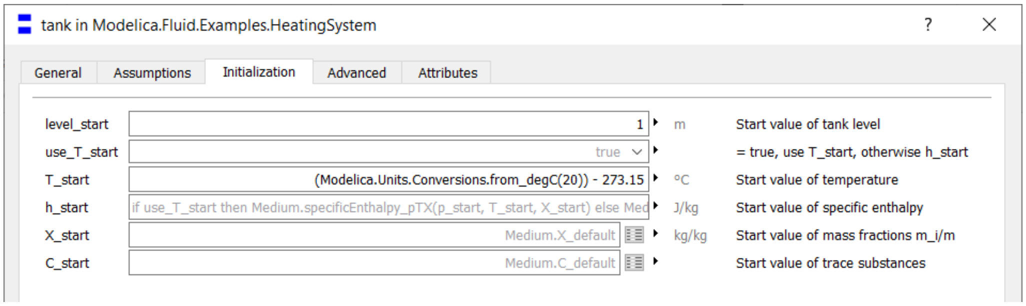
The first block we will use in the heating system diagram is the tank. We will use this tank as a resource. The pump takes liquid from here to pump to the pipes.



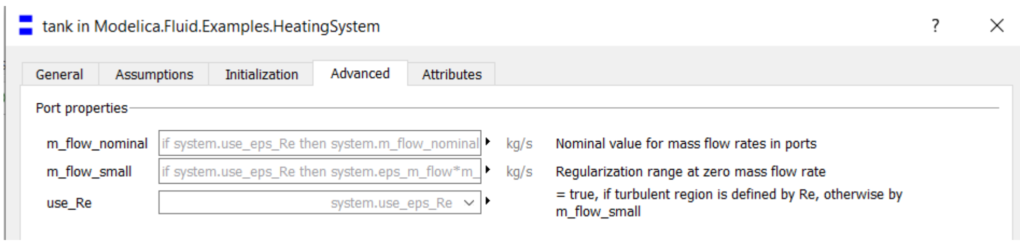
In the Parameters section, we can determine the area and height of the tank and thus determine how much water it can take. There are inlet and outlet ports to make connections. We can measure the temperature and pressure differences between these ports. In the assumption section, you can see that we get the T_ambient and p_ambient values from the system block. We can also determine whether there will be heat transfer in the tank by assigning true, false values. We will set the value as true because we want to see the temperature difference.



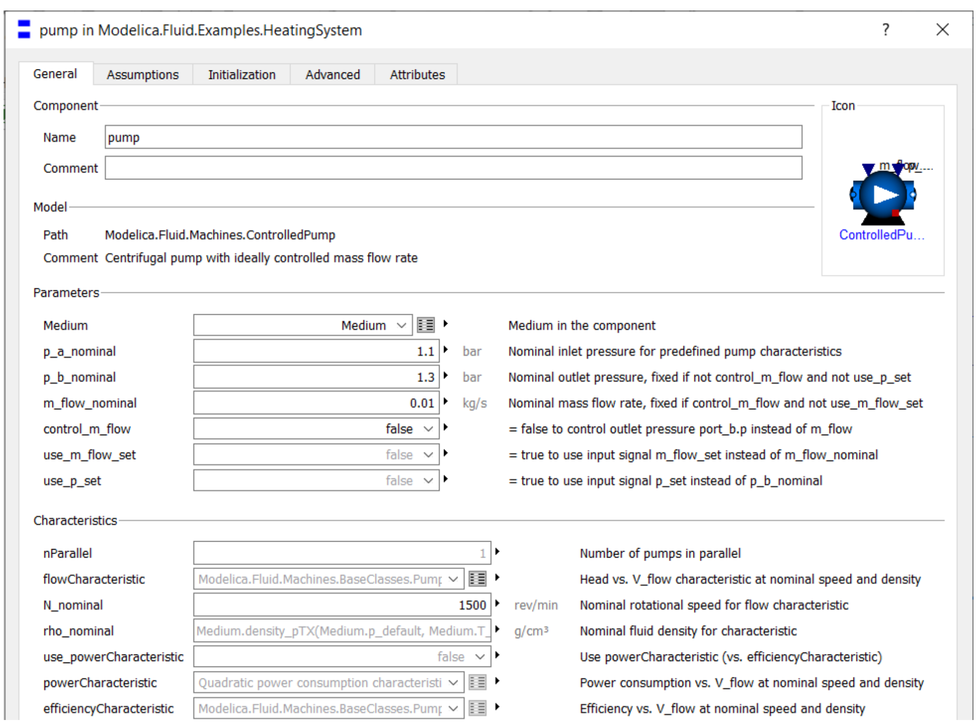
Initialization tab we see the initial values again.



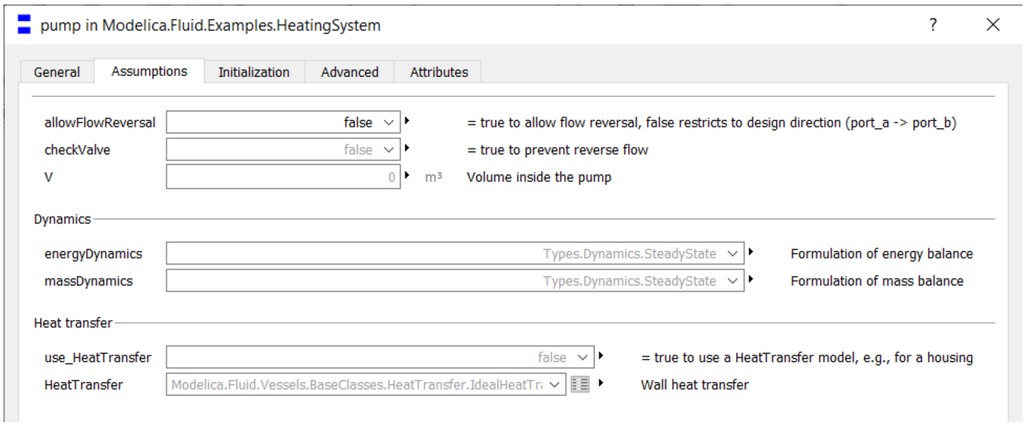
The properties in the ports are still taken from the system.



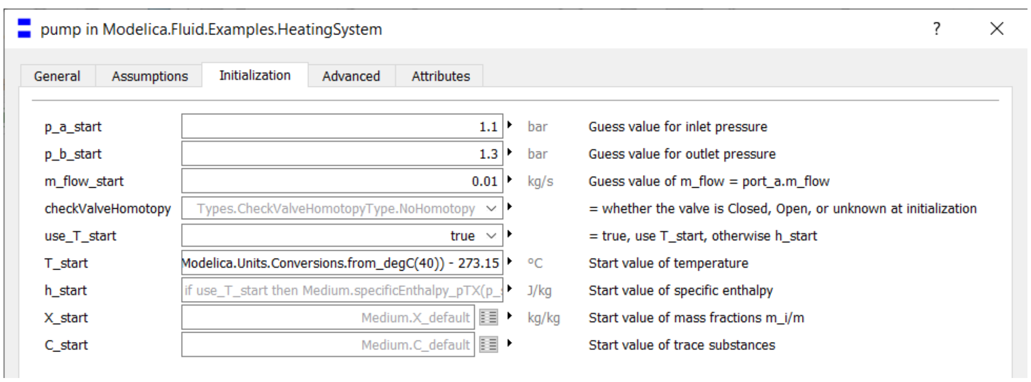
We convey the water we take from the tank to the pipes with the pump. While doing this, we can also make adjustments to whatever value we want the mass flow rate to be. There are inlet and outlet ports in the pump, we determine the pressure values of these ports from the parameters section. We can specify the liquid or gas we will use from the medium section.



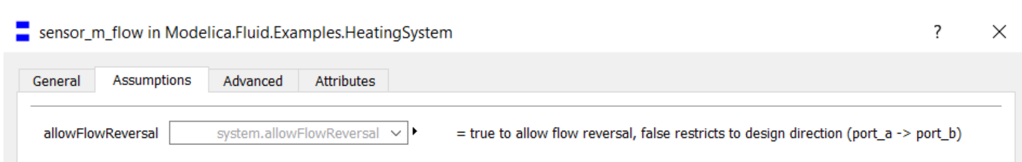
Here we can reverse the flow direction of the flow in the pump, but we want it to move in the design direction, so we choose the false parameter. At the same time, we choose false again because we do not want heat transfer.

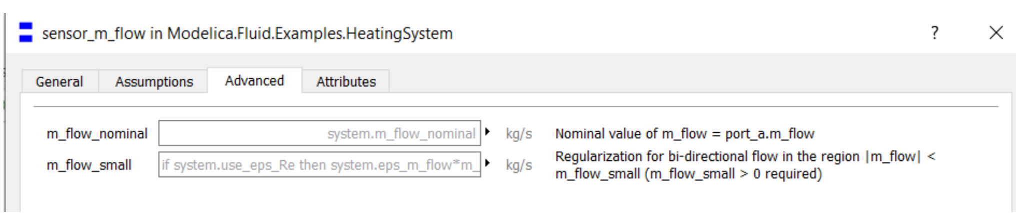


We can also see the values we gave in the General tab here. These are the initial values of the pump. Since we want a stable diagram, the default values are taken from the system again.

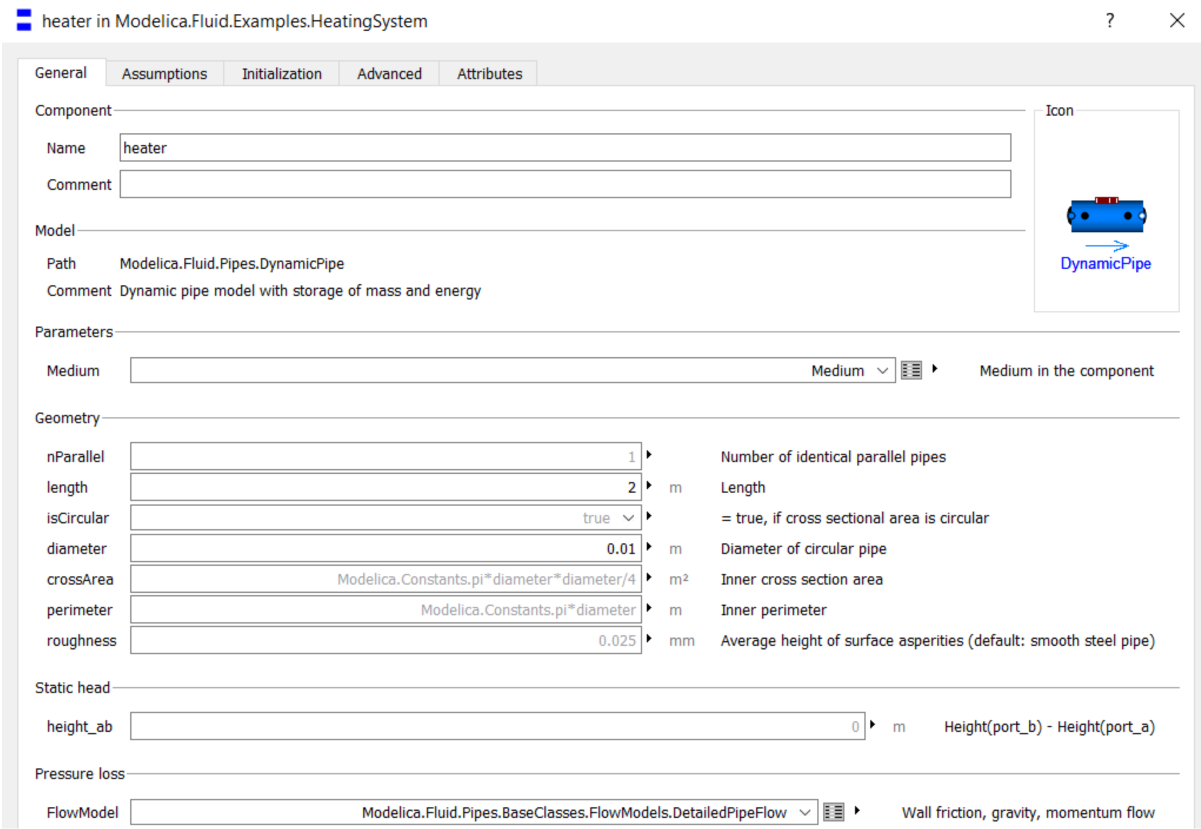


Sensors allow us to read the output value of the pump on the graph.

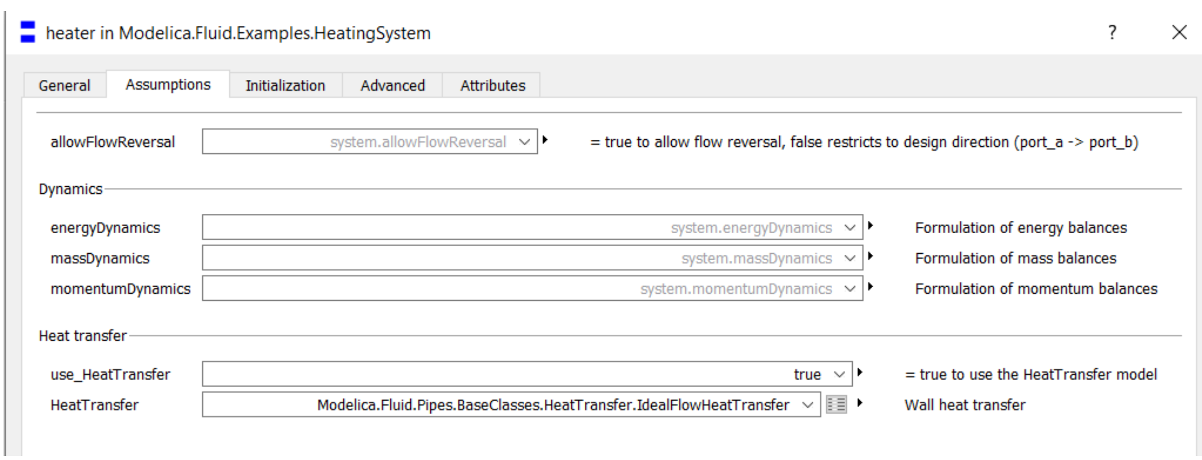


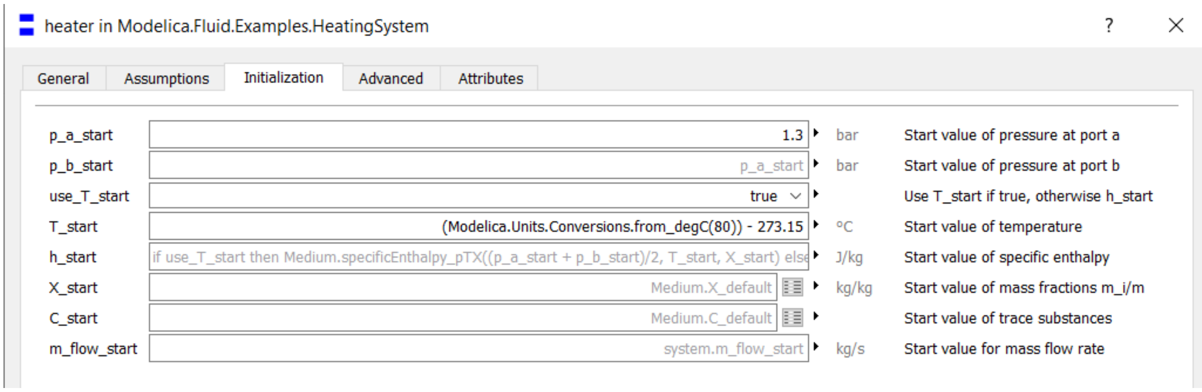


Values such as pipe's length, diameter, whether it will be connected in parallel, cross-section and perimeter are calculated automatically from the following parameters. Since we will heat this pipe with the heat source, we named the pipe heater and connected it to the Heat transfer source.

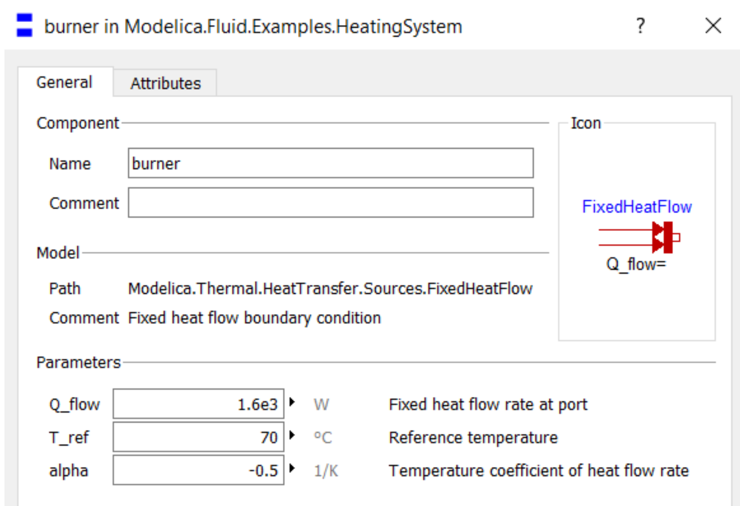


As in other components, it is taken from the assumption and initialization system block. Since we want to transfer heat, we also activate the wall heat transfer feature.

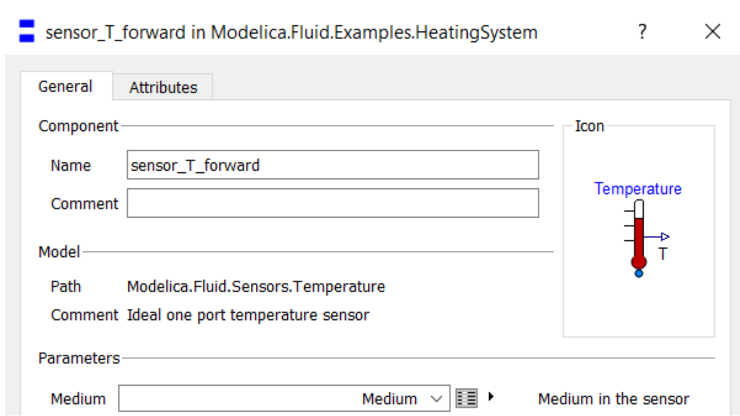




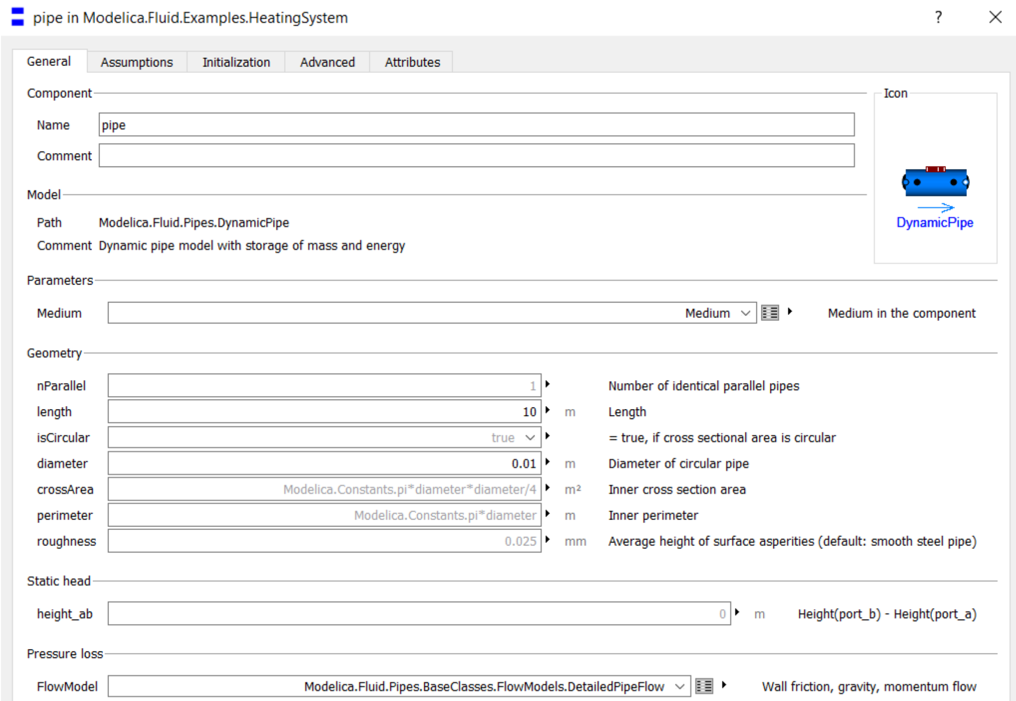
The heat source we use to heat the Pipe. There is heat transfer between the Reference temperature system block and the incoming T_ambient. Depending on the value you give, the temperature of the liquid in the pipe may increase or decrease. We can also determine the rate of Flow from here.



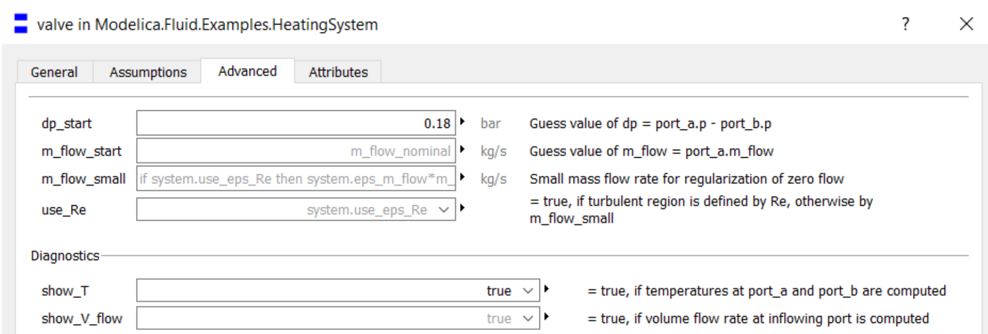
We need the sensor to measure and compare the temperature after the heat transfer process. By adding the temperature sensor, we can compare the changing temperature values on the graph.

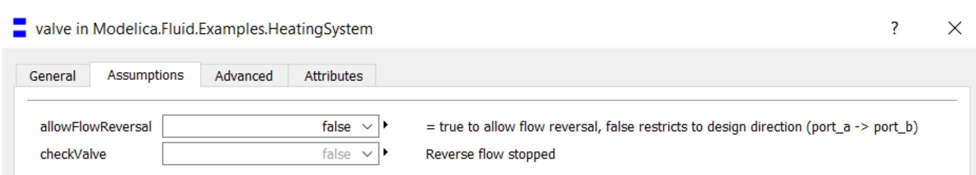


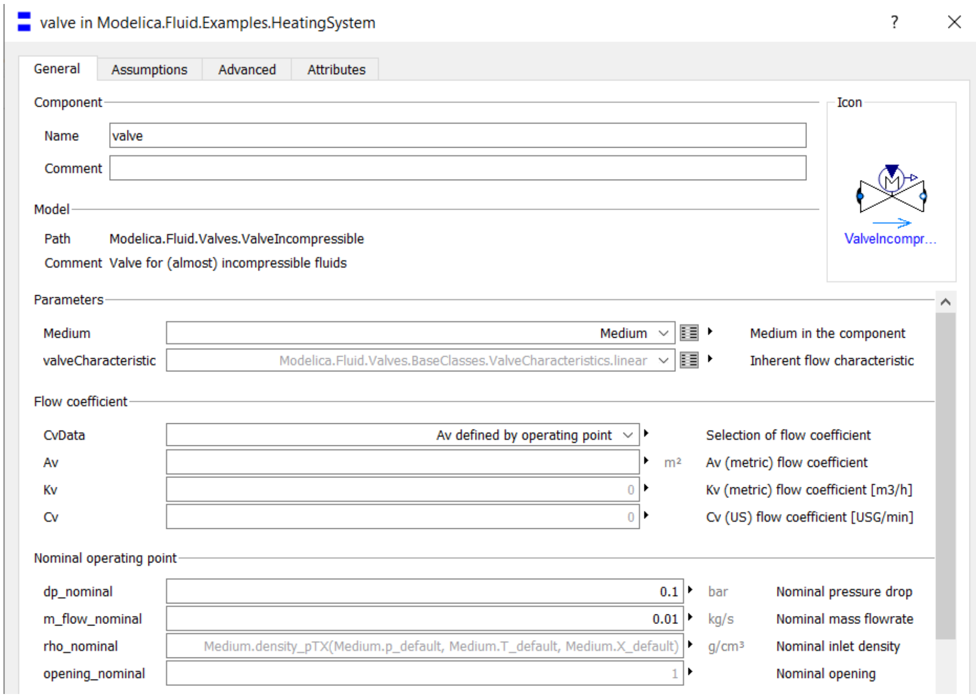
We use a pipe again to connect the liquid we heated to the valve. This time this pipe is longer, we do not change the cross-sectional area for a stable system, its diameter remains constant. Assumptions and initializations are taken from the system block like any other pipe.



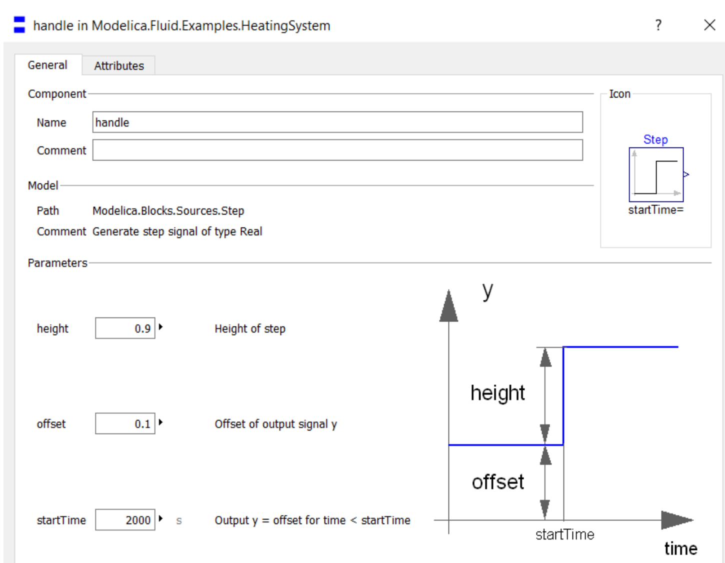
We use valve to control fluid flow. If the fluid has a coefficient, it can also be added here. We can also determine the pressure drop value at the inlet and outlet of the valve. we can also do reverse flow control like in pump.



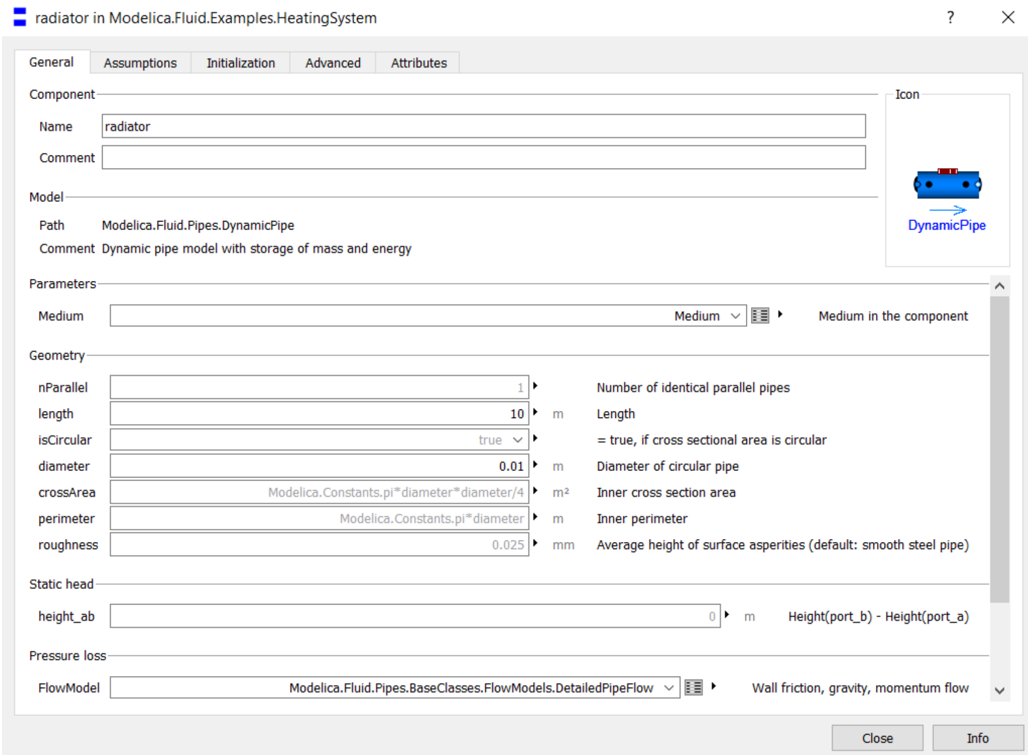




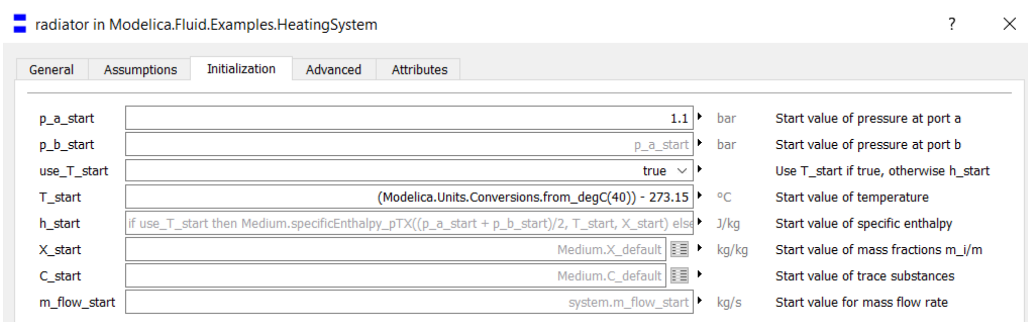
There is a signal connected to the valve. You can add any type of signal you want according to the flow you want to perform. Here we see step signal. With the signal we added, you can determine when the valve will open and how it will open.

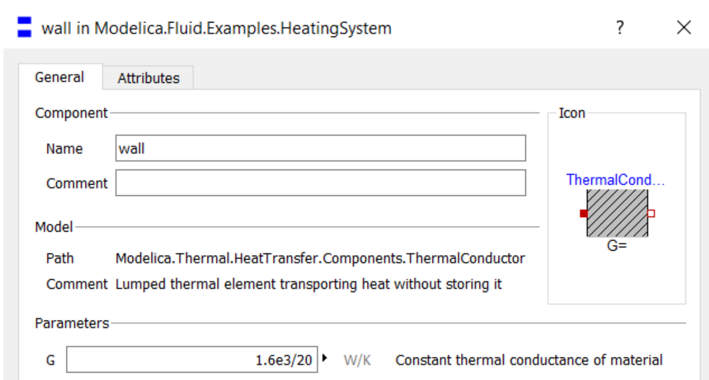


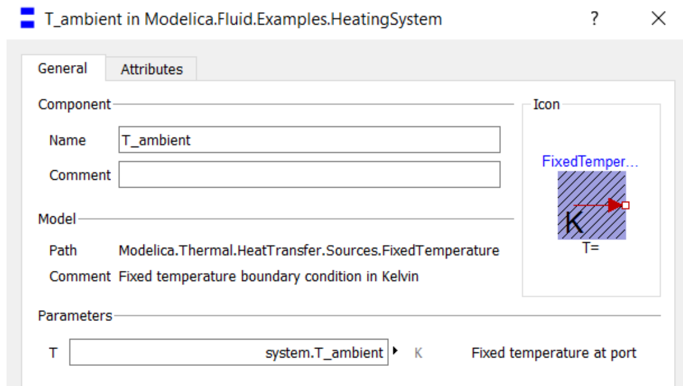
We use pipe again after the valve. In this pipe, we again use T_ambient to ensure heat transfer over the wall with conduction. Here we cool the fluid we heated again.



We can see that the pressure drop we made in the valve reduces the pressure in the pipe.

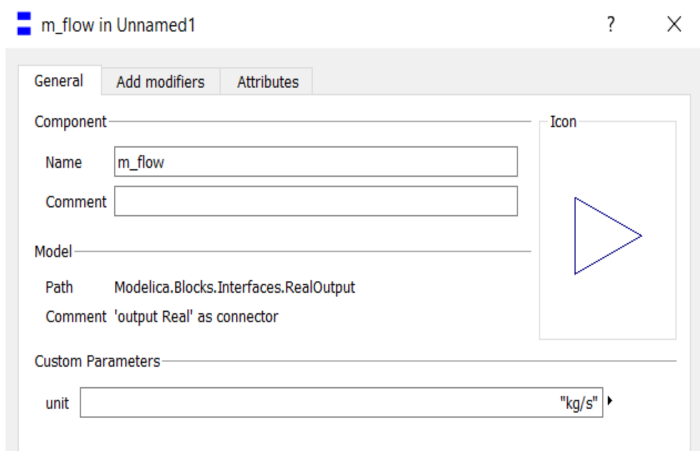


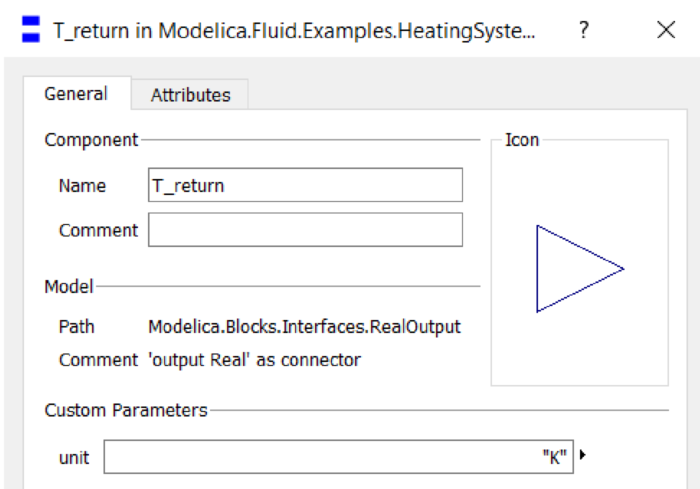




After the cooling process, we can read the temperature value on the graph by putting the temperature sensor again. Then we connect the pipe back to the tank and complete the cycle.

In order to read the values measured by the sensors on the graph, we need to connect these output components to the sensors.

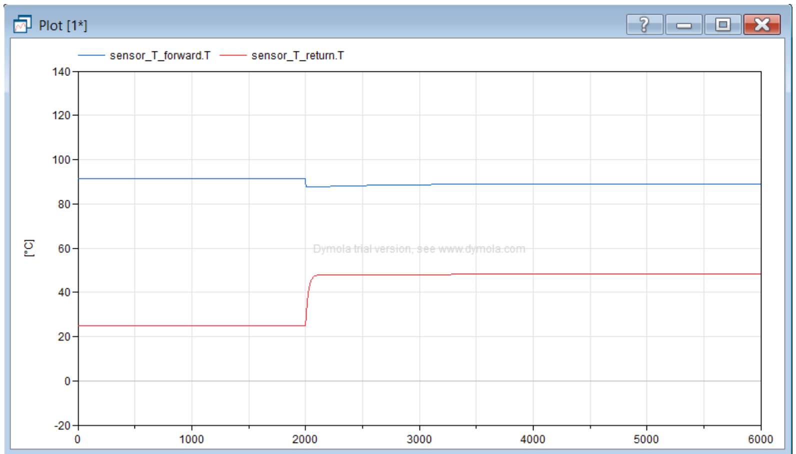




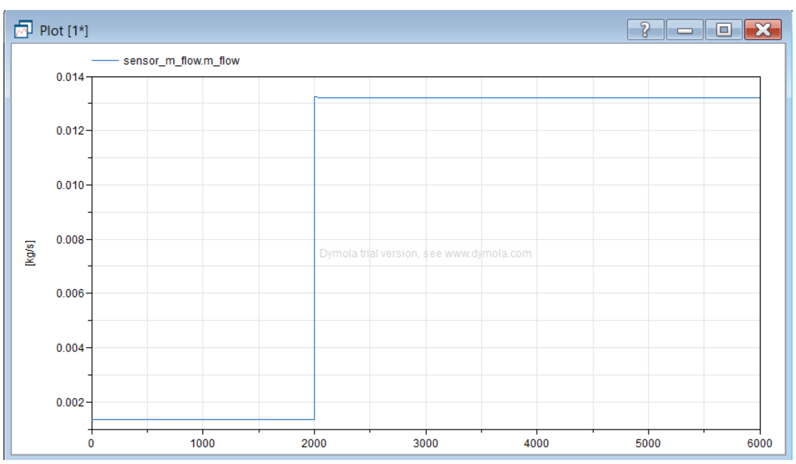
RESULTS

Now we will simulate the system and examine the pressure and flow changes.

If we examine the T values of the temperature sensors we put on the graph, we can see that there is heat transfer on both sides after the valve is opened. After reaching a certain degree, the temperature remains constant.



We can see how the flow changes depending on the valve opening.



3.2) INCOMPRESSIBLE FLUID NETWORK

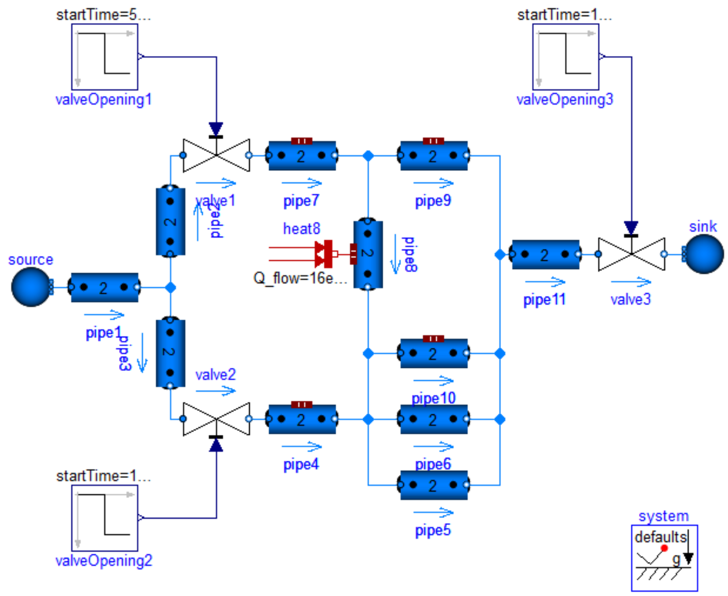
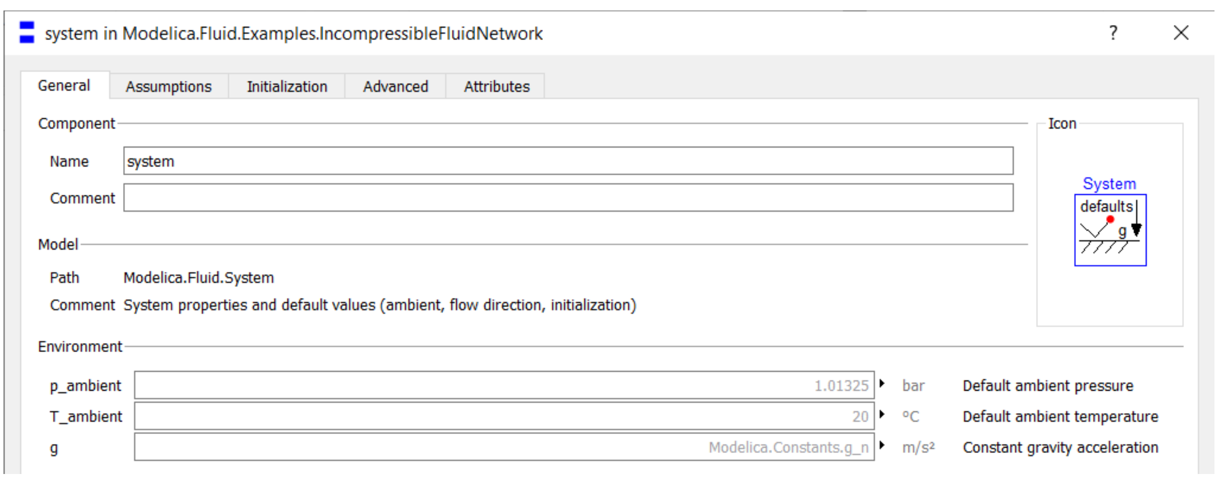


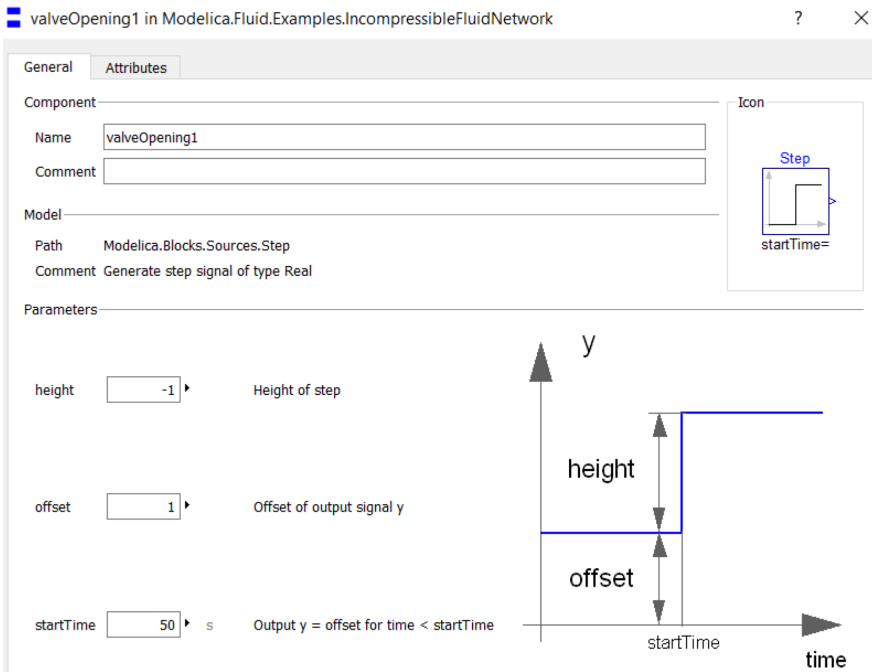
Figure 7: Incompressible Fluid Network Diagram

BUILDING THE SYSTEM

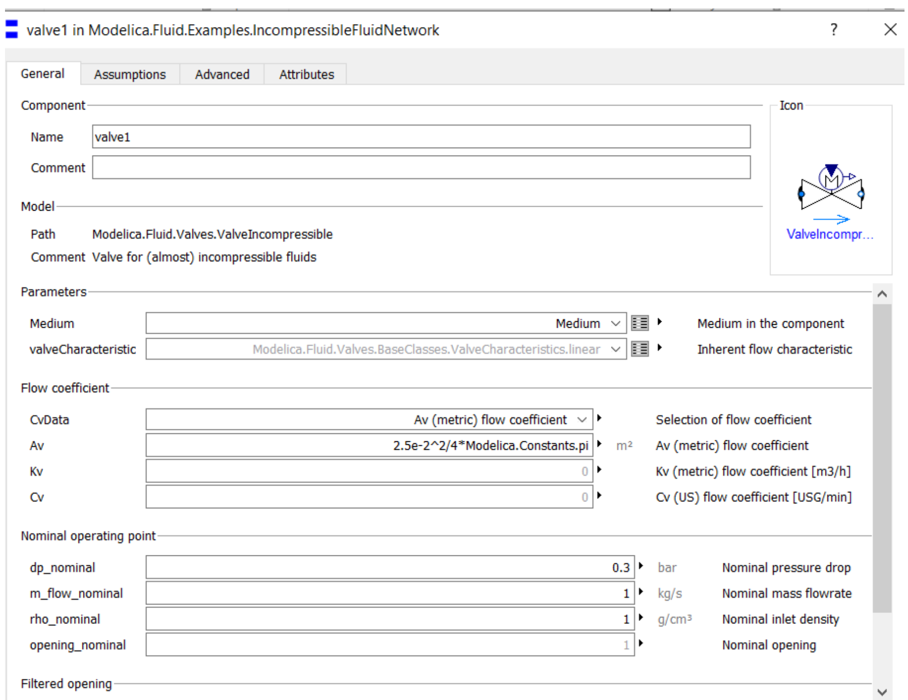
First, we determine the system properties. For this, we place the system block on the diagram from Modelica>Fluid>System path.



To control the valves with time we use step function block. You can find it following Block>Sources>Step path. For first valve at start valve is fully open when time reaches fifty seconds, step function closes the valve completely.



All three valves have identical properties. Just step functions are different.



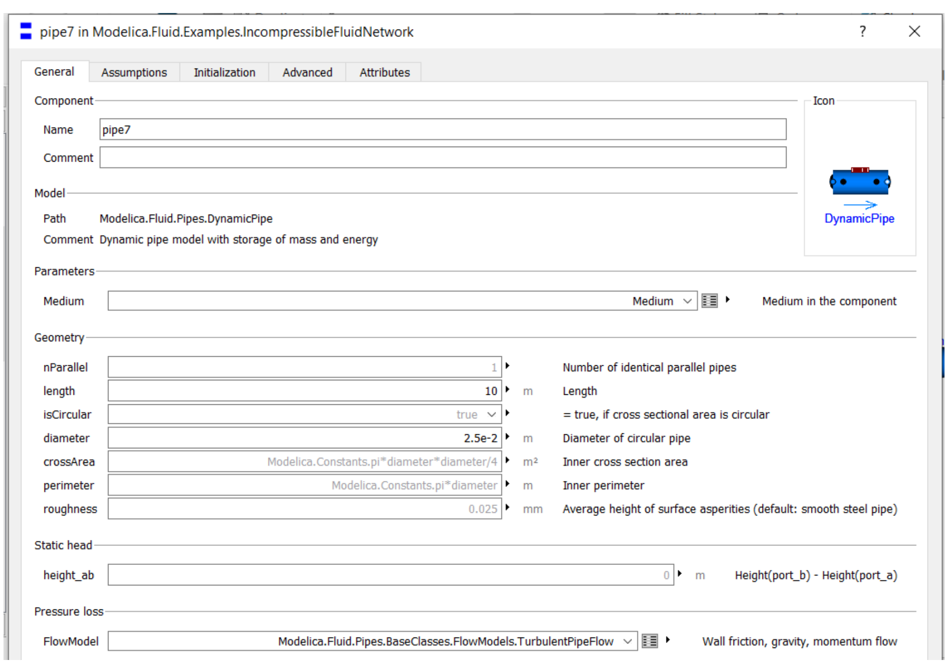
For second valve step function close the valve from fully open to half open at 100th seconds.

height	-0.5		Height of step
offset	1		Offset of output signal y
startTime	100	s	Output y = offset for time < startTime

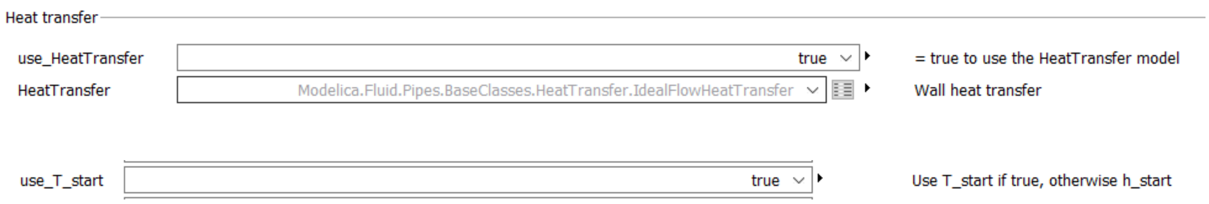
For third valve step function close the valve from fully open to half open at 150th seconds.

height	-0.5		Height of step
offset	1		Offset of output signal y
startTime	150	s	Output y = offset for time < startTime

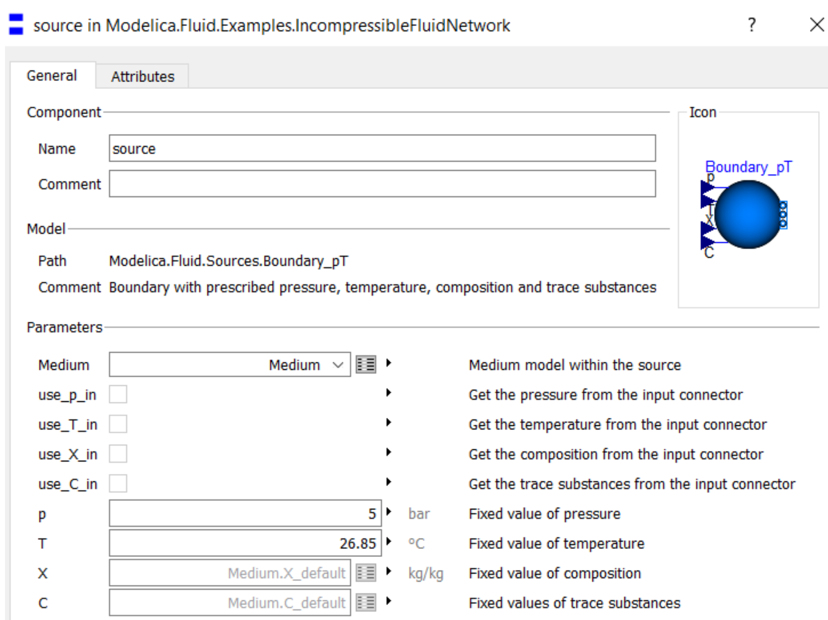
All of our pipes in this example are dynamic pipe and all of them have same diameter. Their length are respectively 10,0.5,0.5,2,20,20,10,10,10,10,0.5 m.



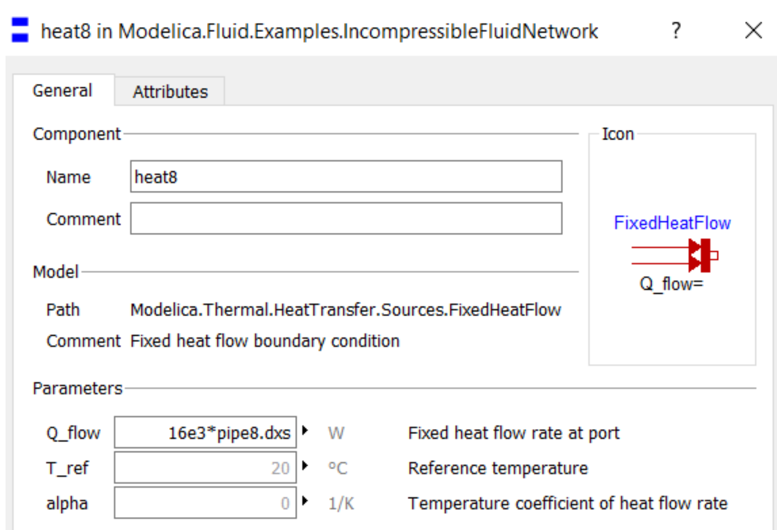
All pipes have use_HeatTransfer assumption and use_T_start initialization.



Properties of boundary source with fixed pressure and temperature.



We heat up the pipe8 with fixed heat flow.



After finish everything we can simulate the example

RESULTS

At t=0 all valves are fully open at t=50s valve one fully closed at t=100s valve two is half closed at t=150s valve 3 is half closed.

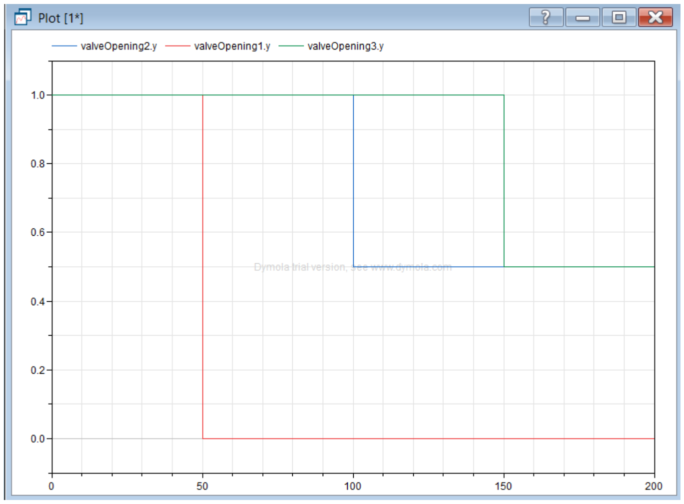


Figure 8: Valve opening timings

As we can see when first valve is fully closed mass flow at second valve is increased. Likewise, after second valve is half closed third and after third valve is half closed mass flows are decreasing.

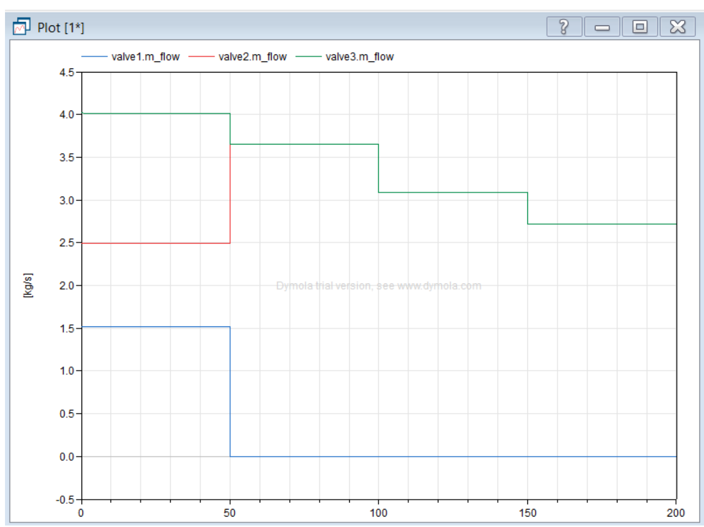


Figure 9: Mass flow rate through valves

At t is between 0 and 50 $\dot{m}_7 = \dot{m}_8 + \dot{m}_9$ after valve closed at t=50 $\dot{m}_7$ is going to be zero. At t is between 50 and 100 $\dot{m}_9 = -\dot{m}_8$, flow at pipe 8 change direction.

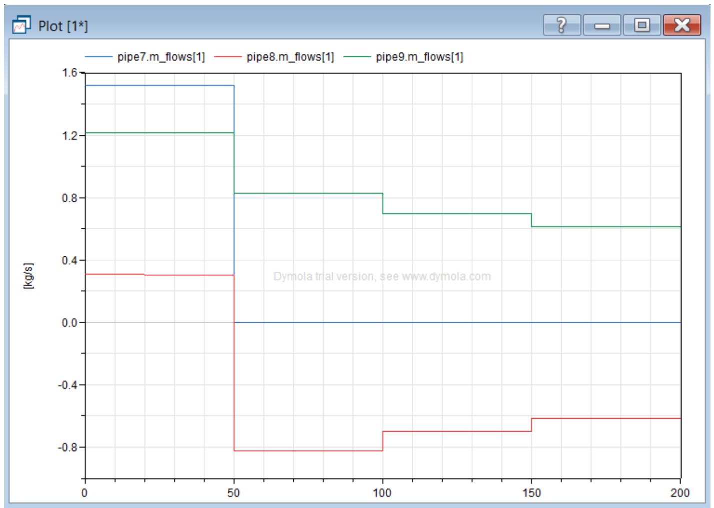


Figure 10:Mass flow rate through pipe 7,8,9

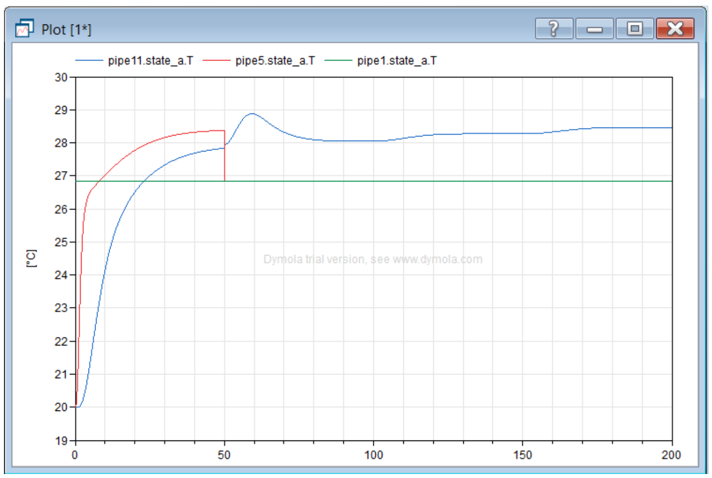


Figure 11: Time vs temperature graph

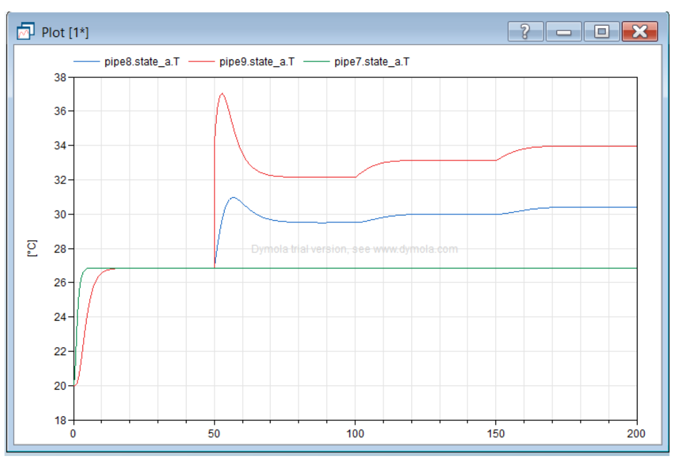


Figure 12: Time vs temperature graph

3.3) BRANCHING PIPES

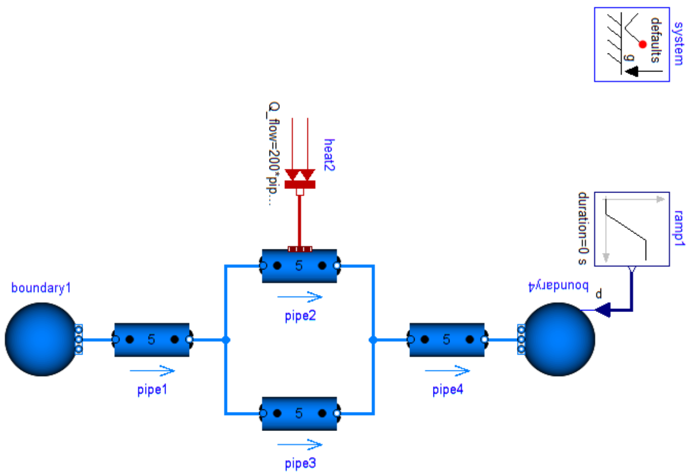
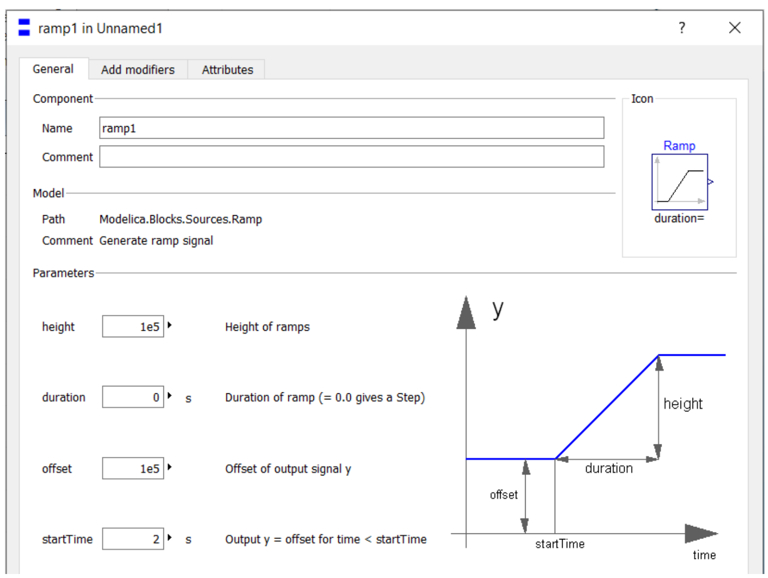
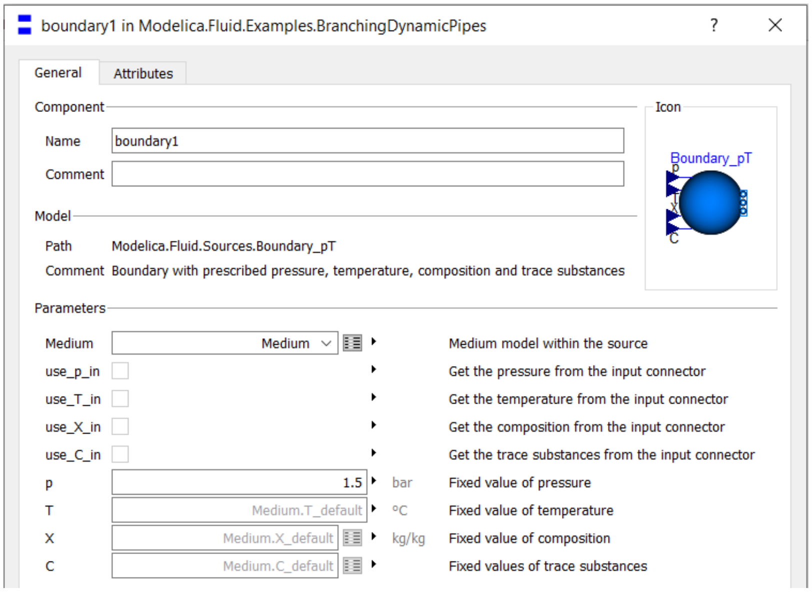


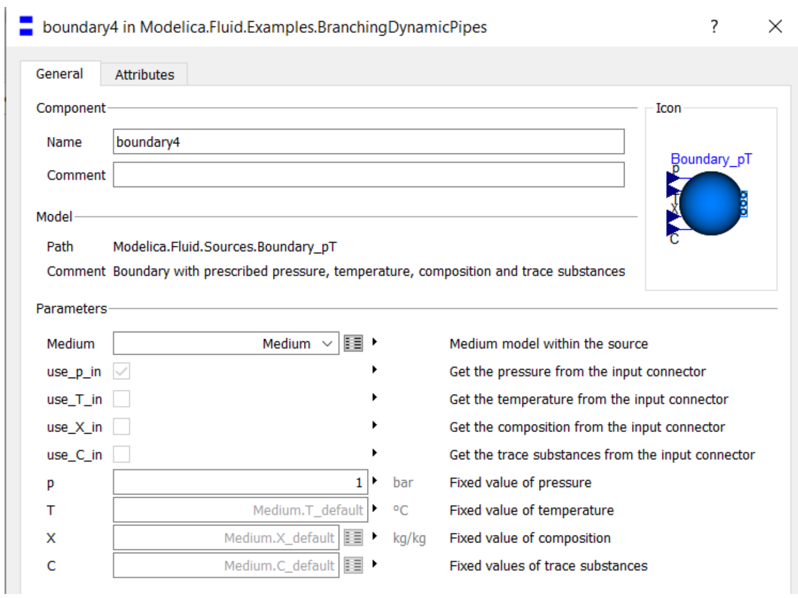
Figure 13: Branching Pipes Example

In this system there is 2 boundary source which contains simple water liquid. We can use the fluids in the library at these boundaries. We set the signal input for the boundary and set it to 2 seconds.

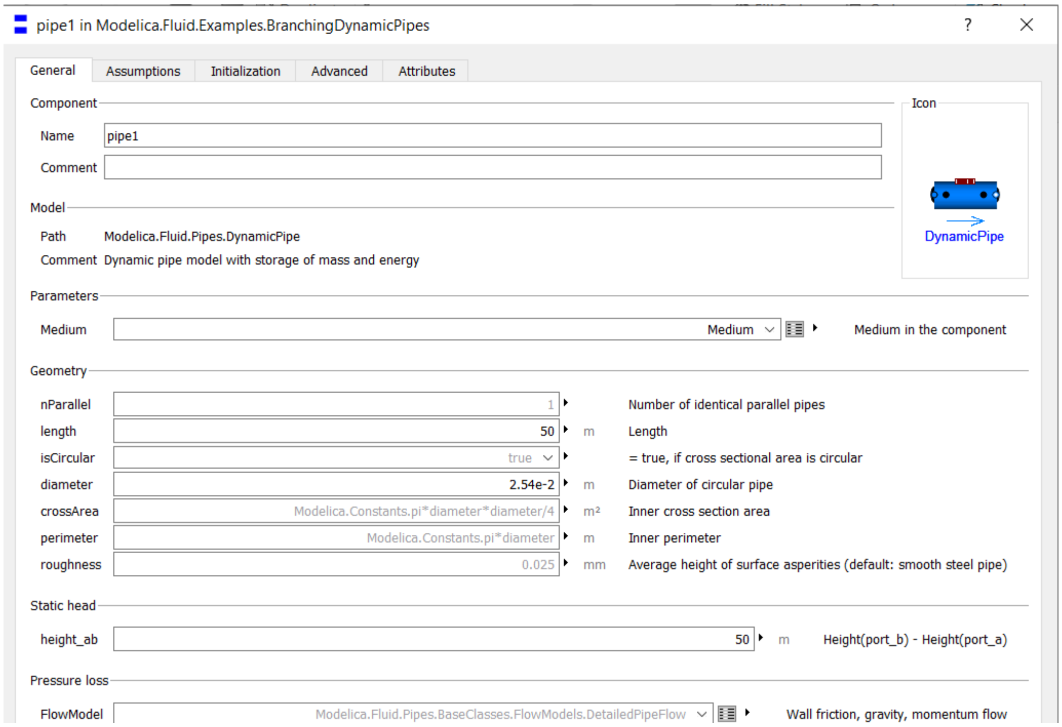


There are 2 boundaries in the system. Both have different pressures. In the simulation we will examine the pressure drop between the boundaries.

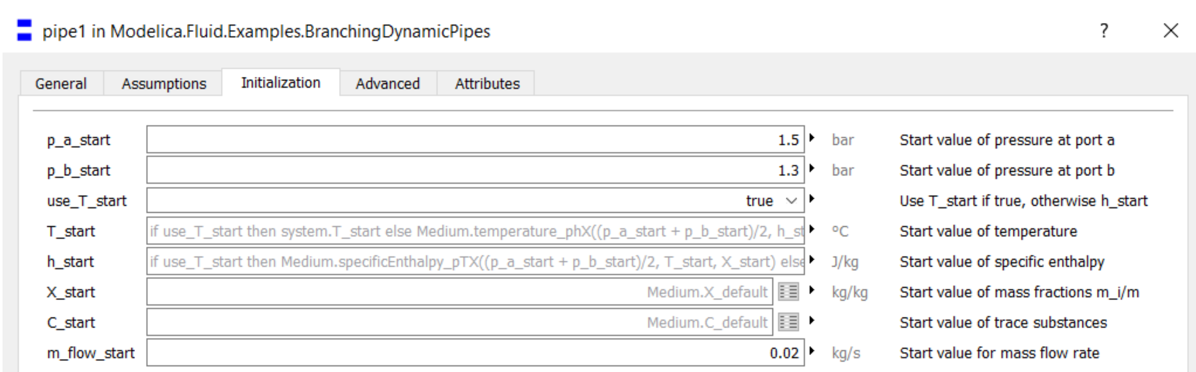




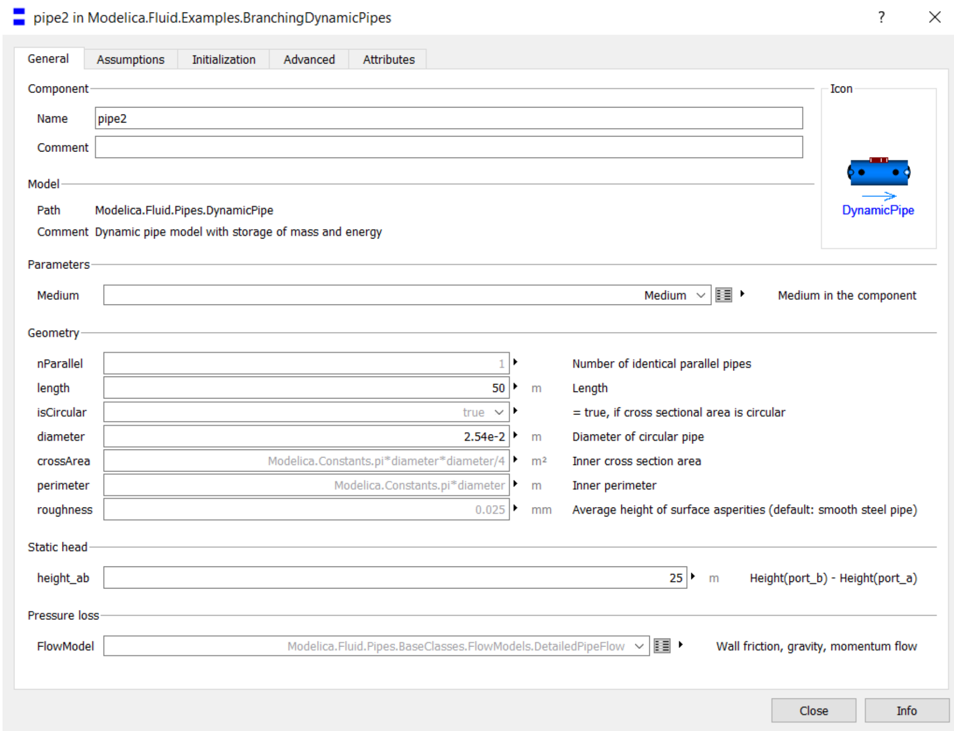
There are 4 dynamic pipes in the system. Each of these pipes consists of 5 nodes. We will examine the flow changes at the entrance and exit of the pipes according to the pipe properties. we will examine the flow of pipe of different length at the same pressure or the flow of pipe with heat input.

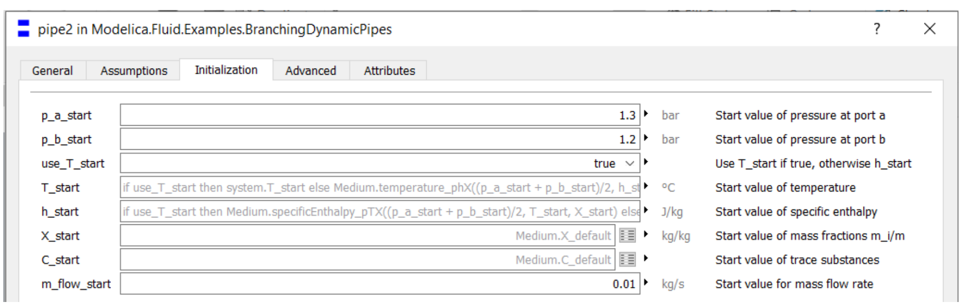


This pipe is the pipe connected to boundary 4. Its initial pressure is the same as the pressure it receives from the boundary to which it is attached. and then it will experience a pressure drop.

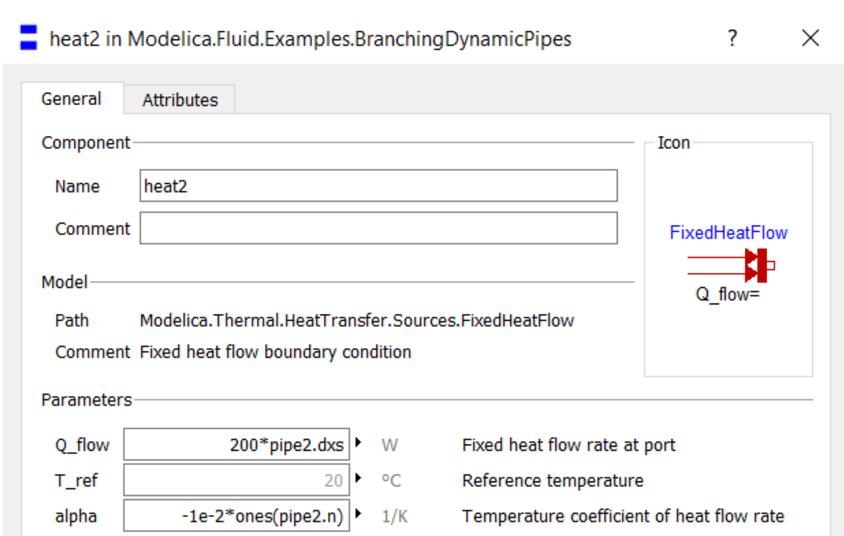


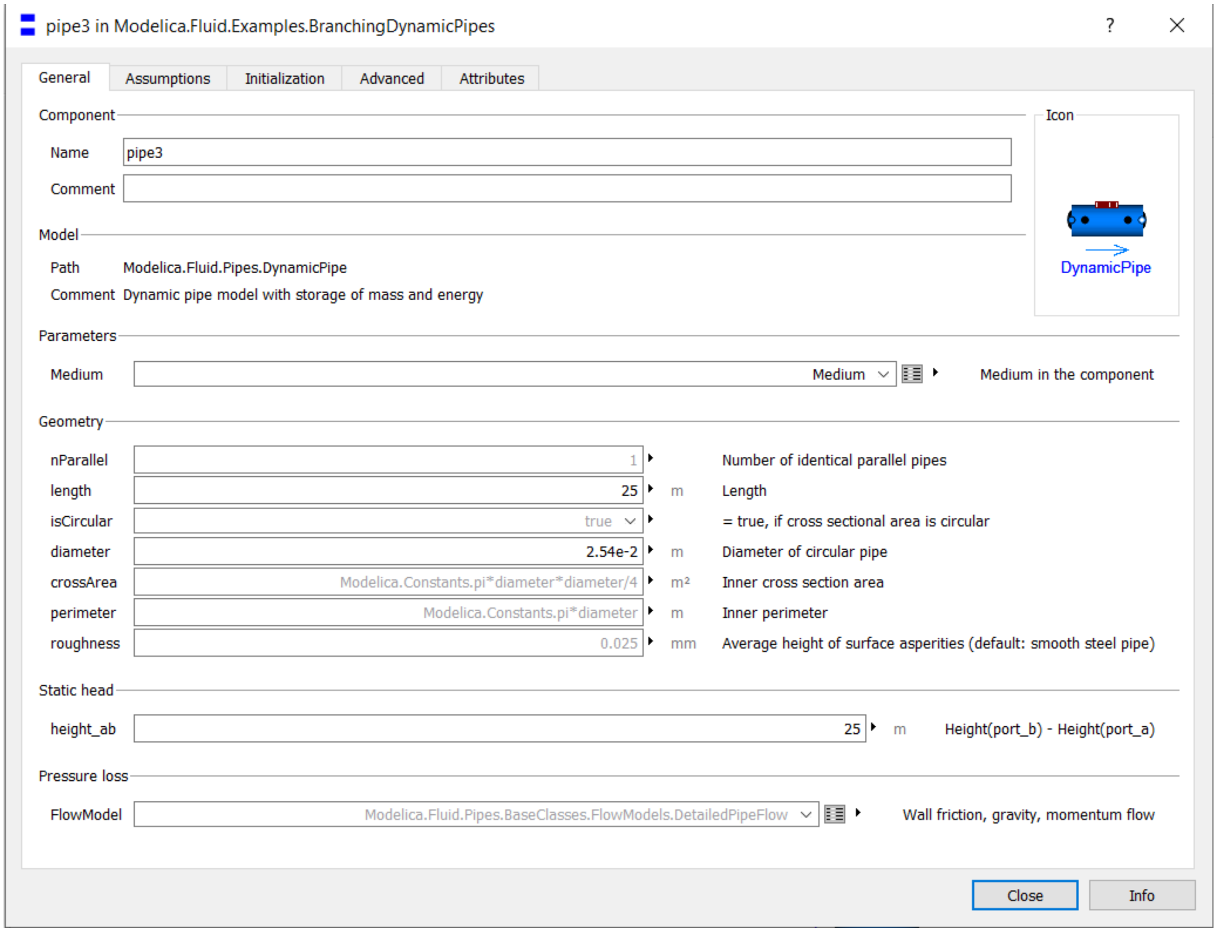
After pipe 1 we see two pipes connected in parallel. There is heat input to pipe 2. pipe 3 is half as short as pipe 2.

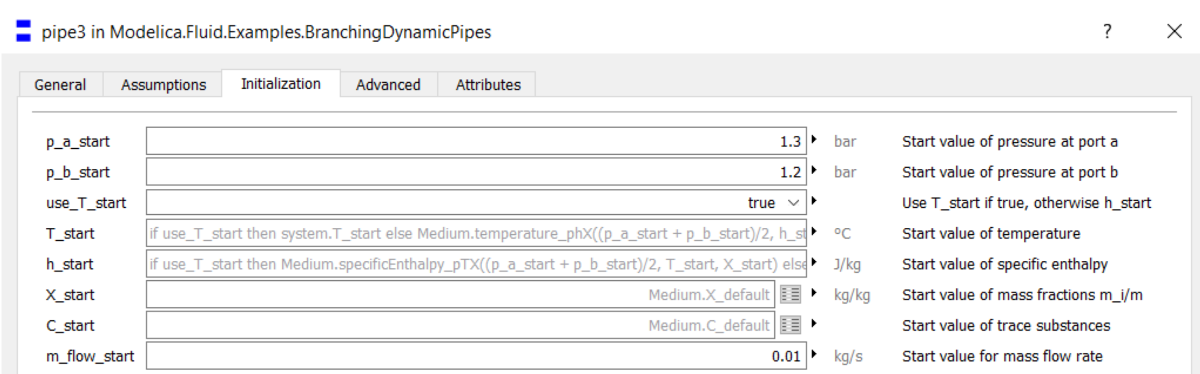




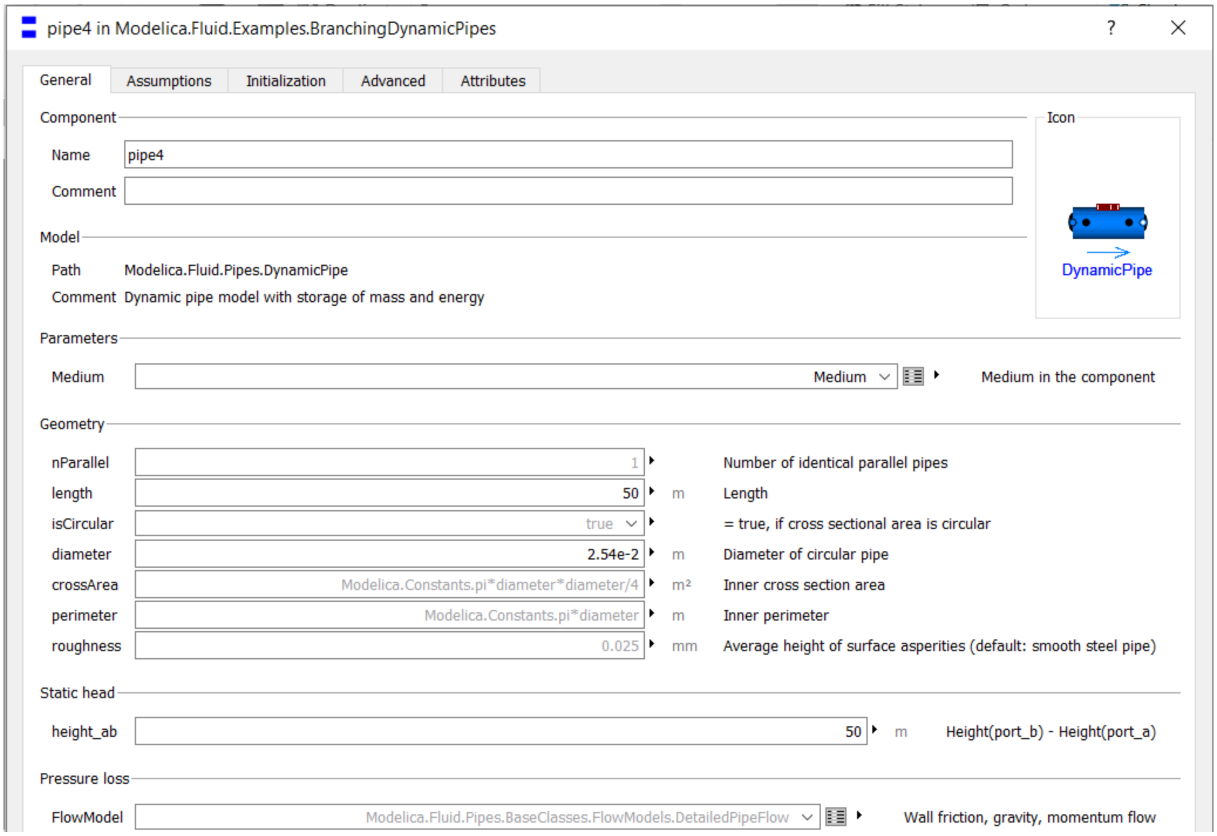
The pressures at the outlet of pipe 1 are the inlet pressures of pipe 2 and pipe 3.

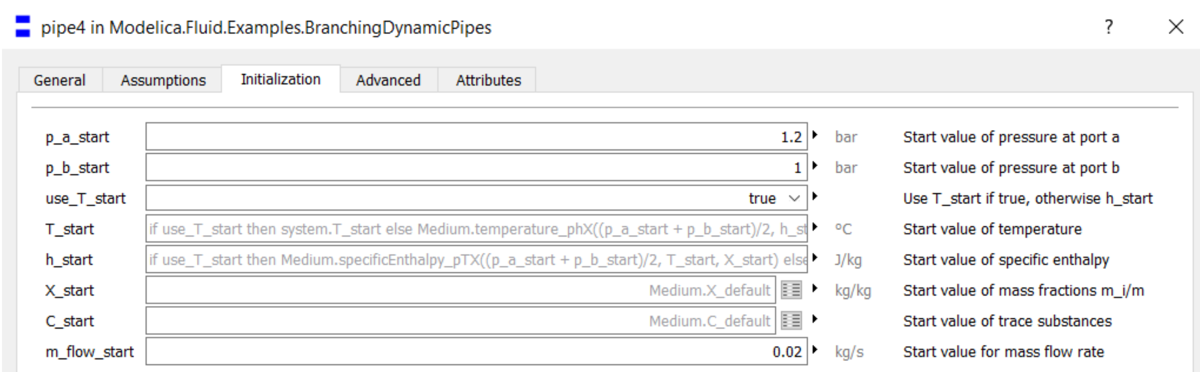






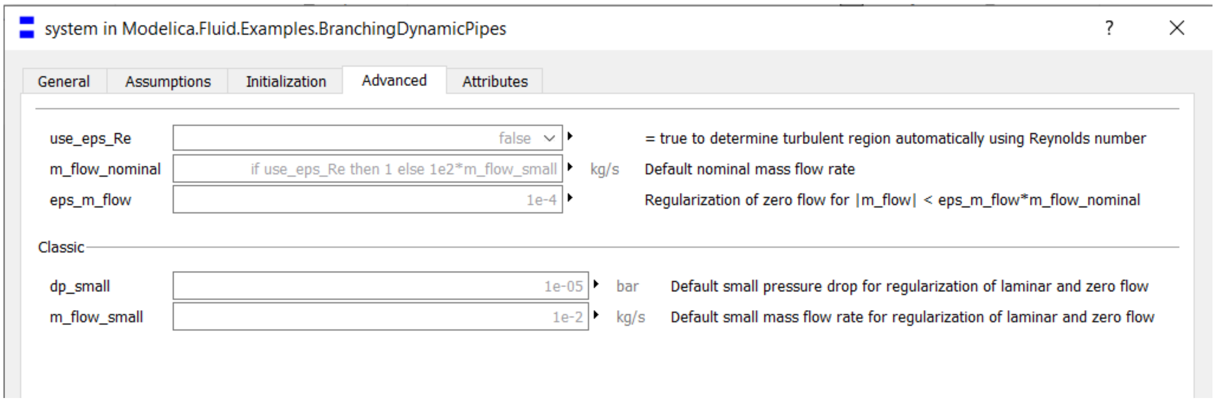
We see that pipe 3 is shorter than other pipes. pressure drop is still the same. Since there are pipes connected in parallel, the flow in pipe 1 is divided into two and evenly distributed to pipe 2 and pipe 3.

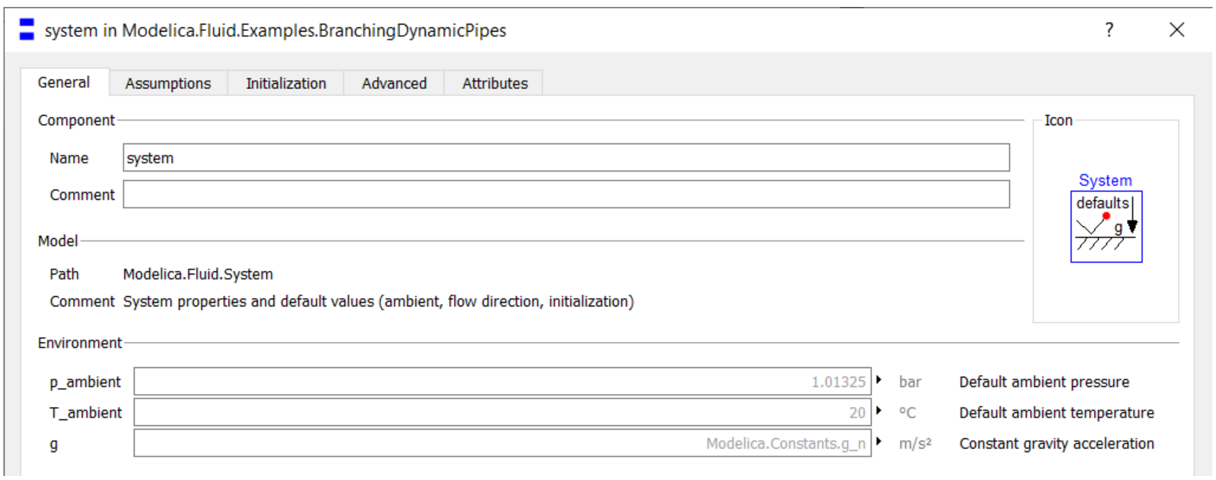




The outlet pressure of pipe 2 and 3 continues as the starting pressure of pipe 4 and it is experiencing pressure drop again. In this way, the boundary reaches the 1 level and the transfer between the two boundaries is completed. In addition, the m flow that has been dispersed merges here and doubles again.

The system default properties that apply to each pipe and boundary are as follows.





RESULTS

Now we will simulate the system and examine the pressure and flow changes.

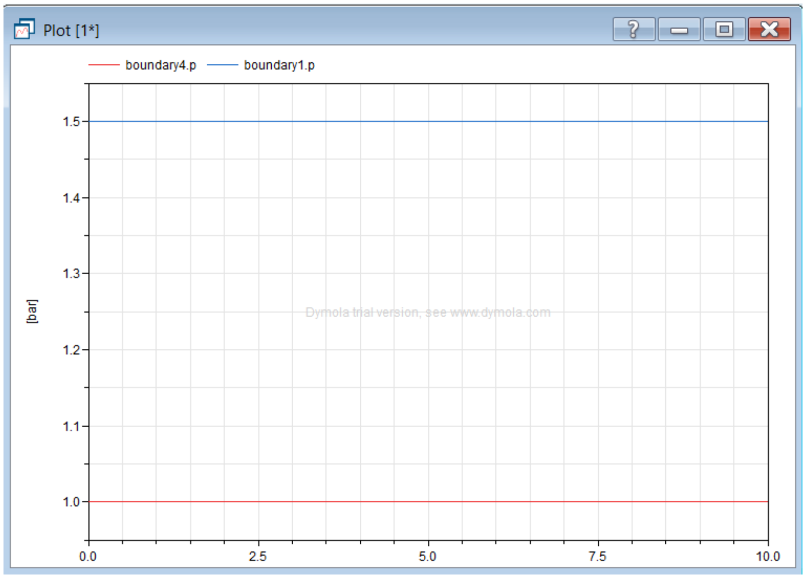


Figure 14:The pressure difference between the two boundaries.

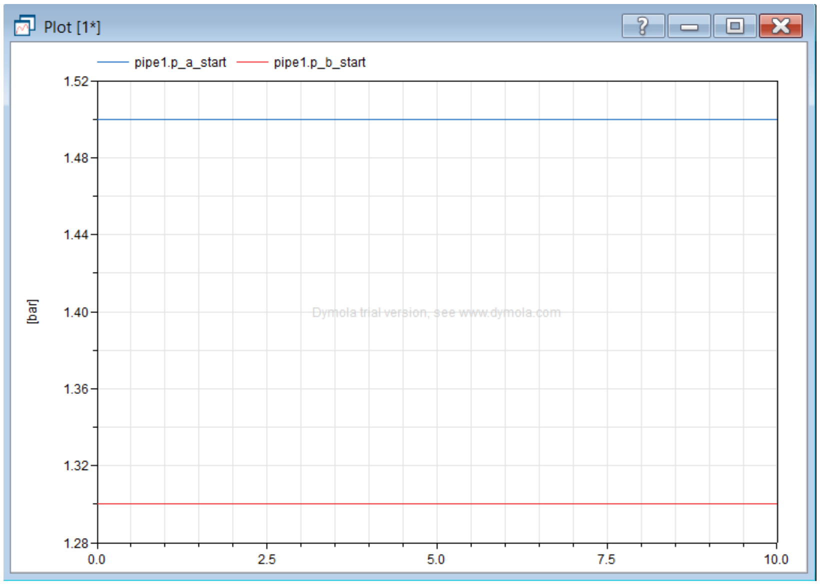
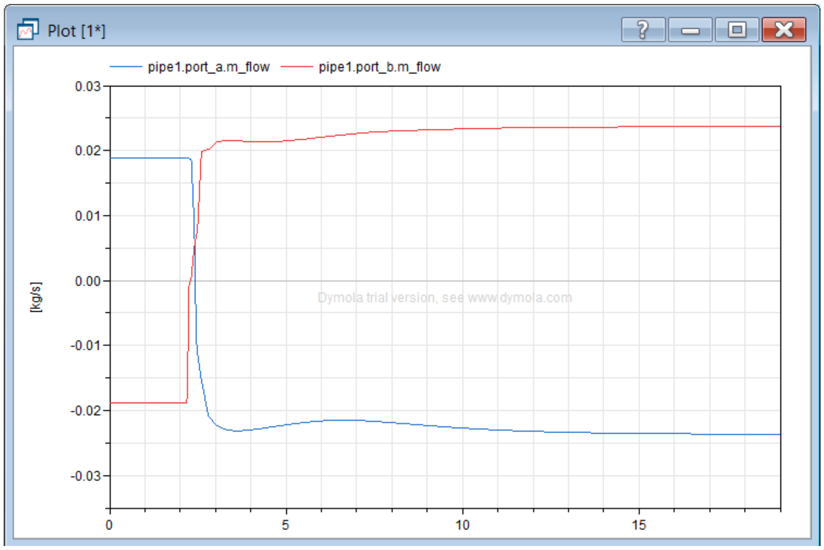
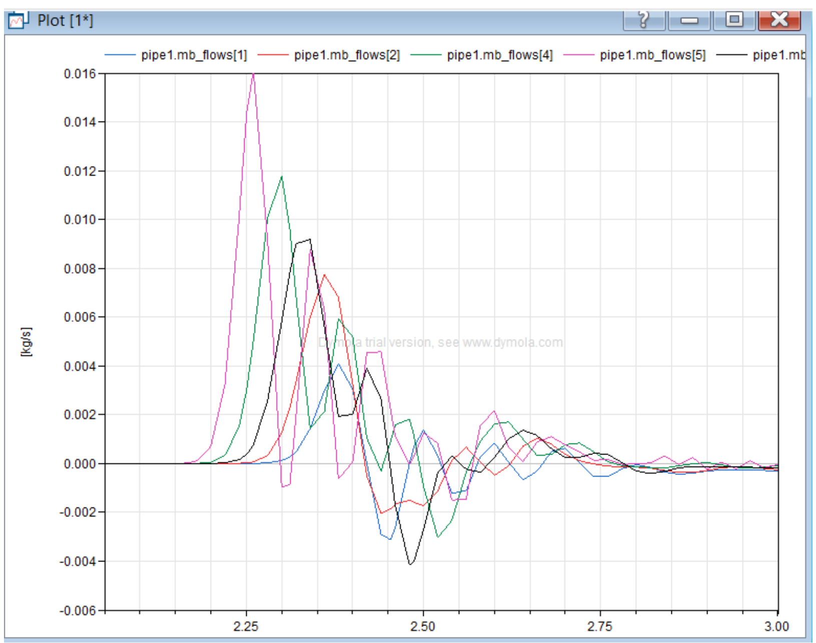


Figure 15:Pressure difference between pipe 1's inlet and outlet ports.

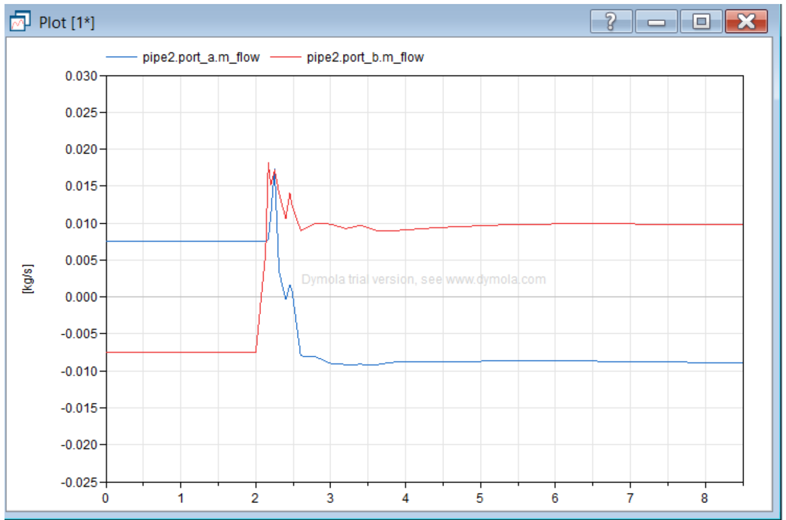
This graph shows us the flow change in the start and exit ports of the pipe during the elapsed time. We can see how long it takes for the fluid to move between the ports.



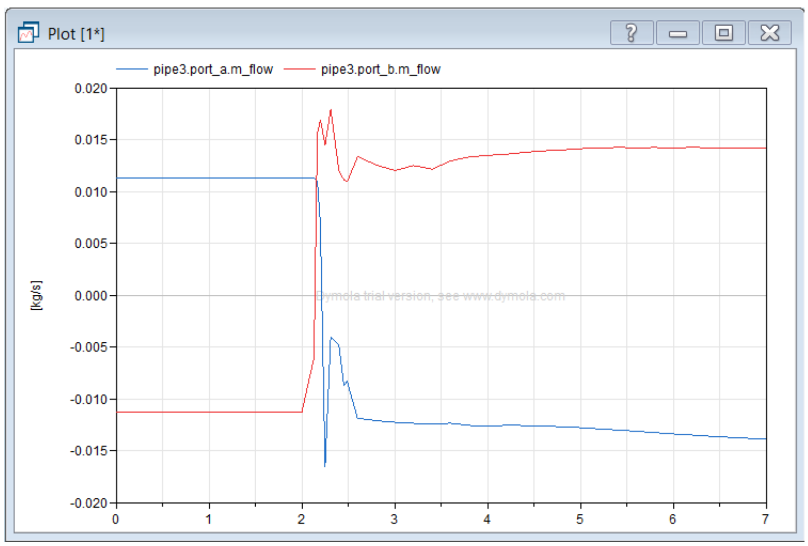
Here, we can examine how flow rate makes a change in which node.



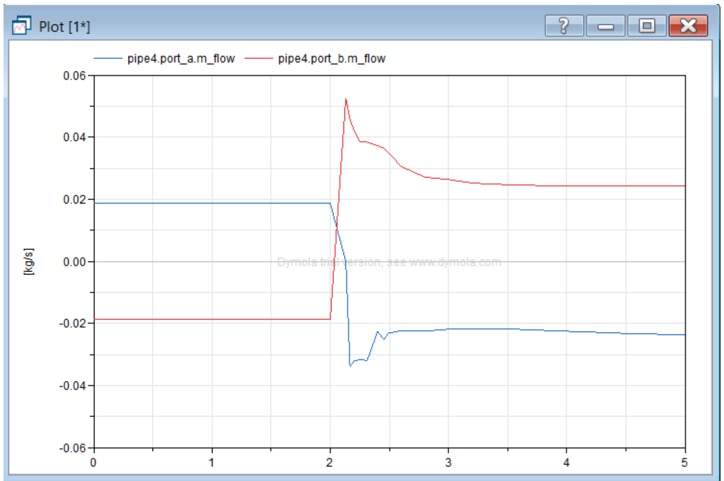
In pipe 2, instead of a direct drop at the inlet port (as we saw in pipe 1 and we will see in pipe 3), we first see a spike and then drop. This may be because of the heat we give to pipe 2.



In pipe 3, we can see a slight increase again after the flow rate completely decreases at the inlet port.



In pipe 4, we see that the flow rate in port b increases much faster than other pipes.



3.4) HEAT EXCHANGER

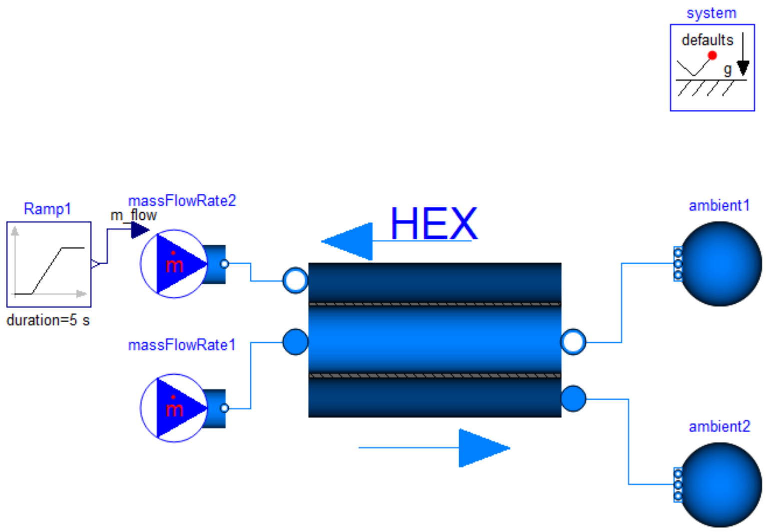


Figure 16: Heat Exchanger Diagram

In this system, there are 2 boundaries, 2 m flow sources, and a heat exchanger. Ramp is used as signal input.

If we examine the HEX(Heat Exchanger) shown in the system in detail:

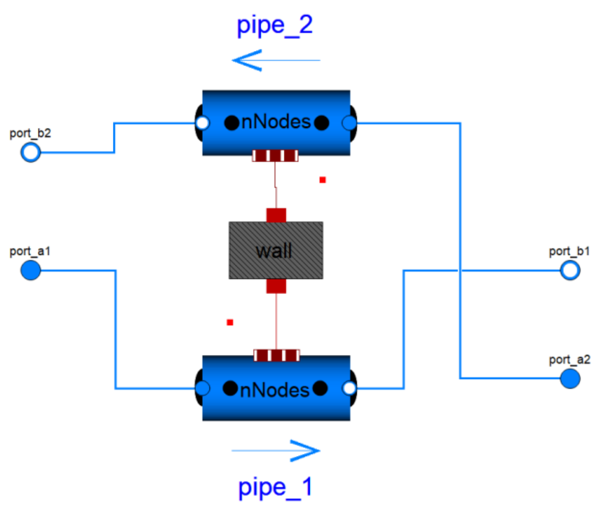
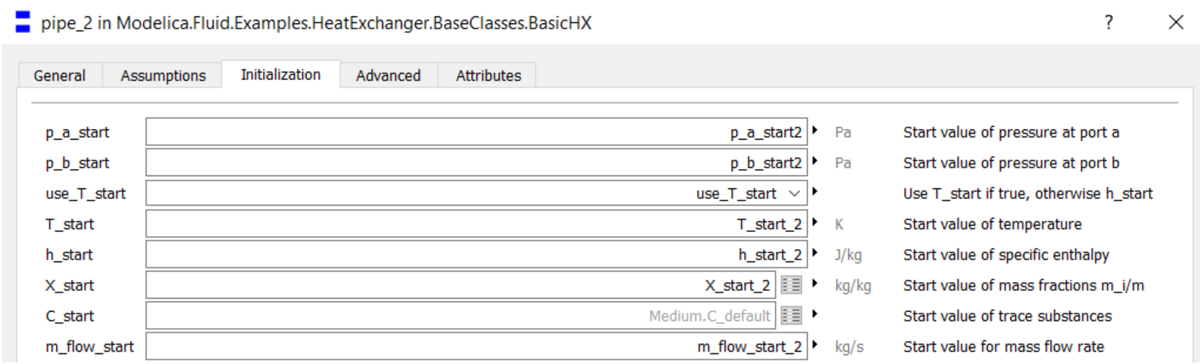


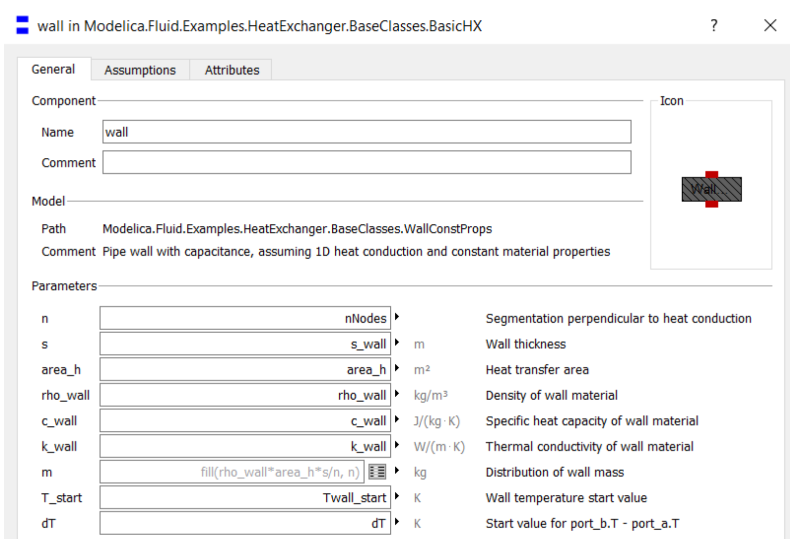
Figure 17: Basic Heat Exchanger Diagram

Although this heat exchanger is a simple heat exchanger, it is formed with 2 dynamic pipes and a wall. The wall between dynamic pipes helps heat transfer.

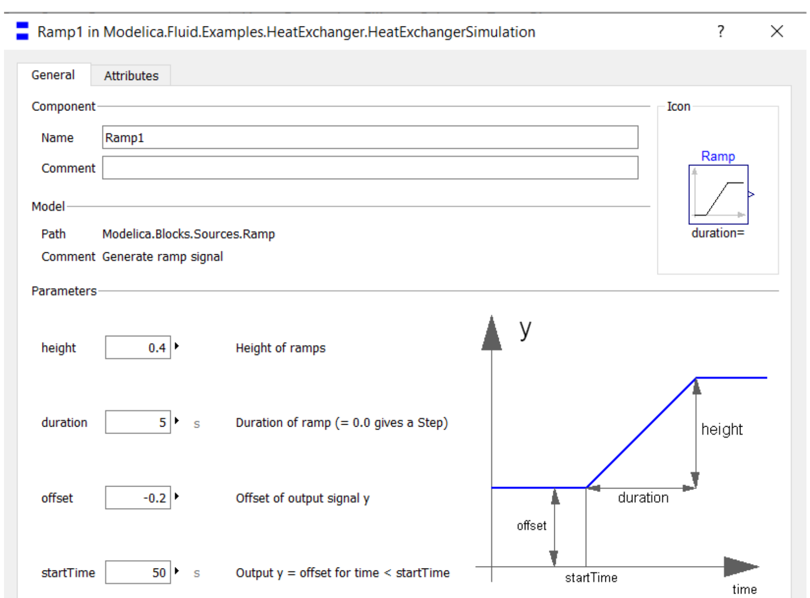
This is how we can determine the pipe's values such as temperature and pressure.



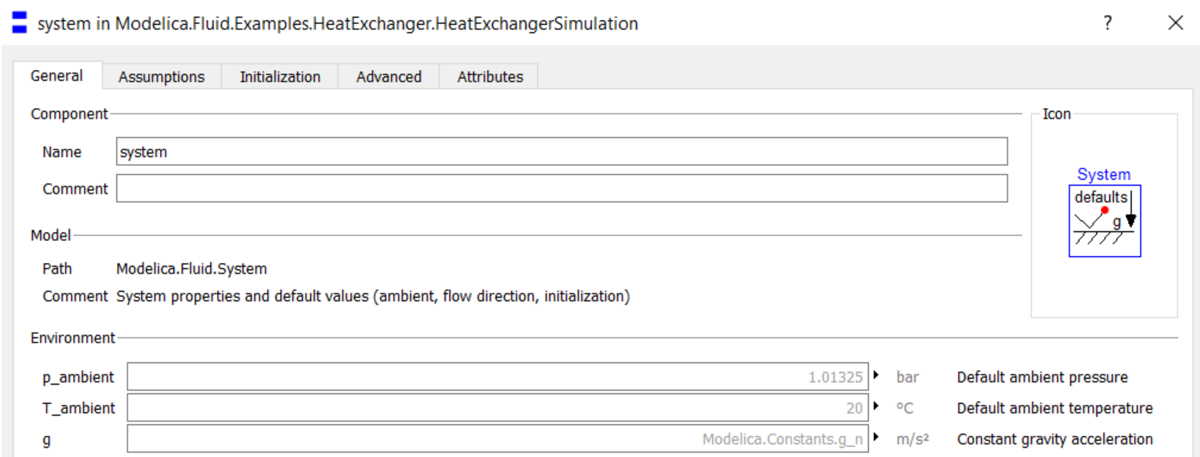
We can also give the necessary properties of the wall.



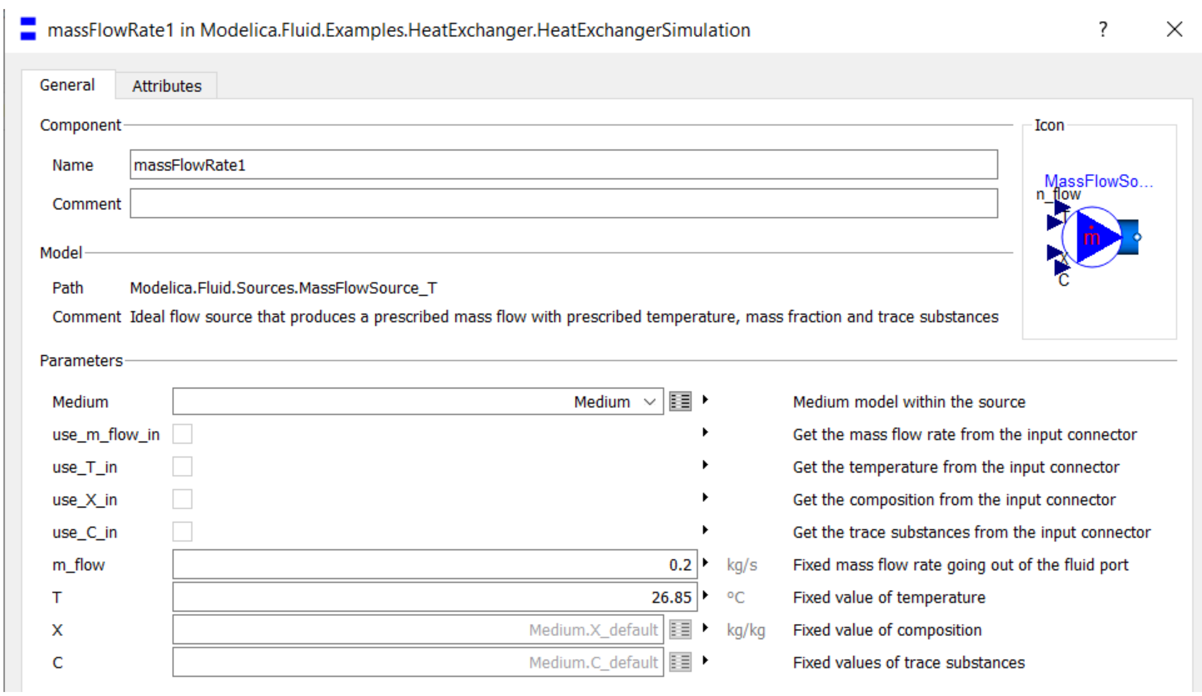
Signal values were determined as on the side.



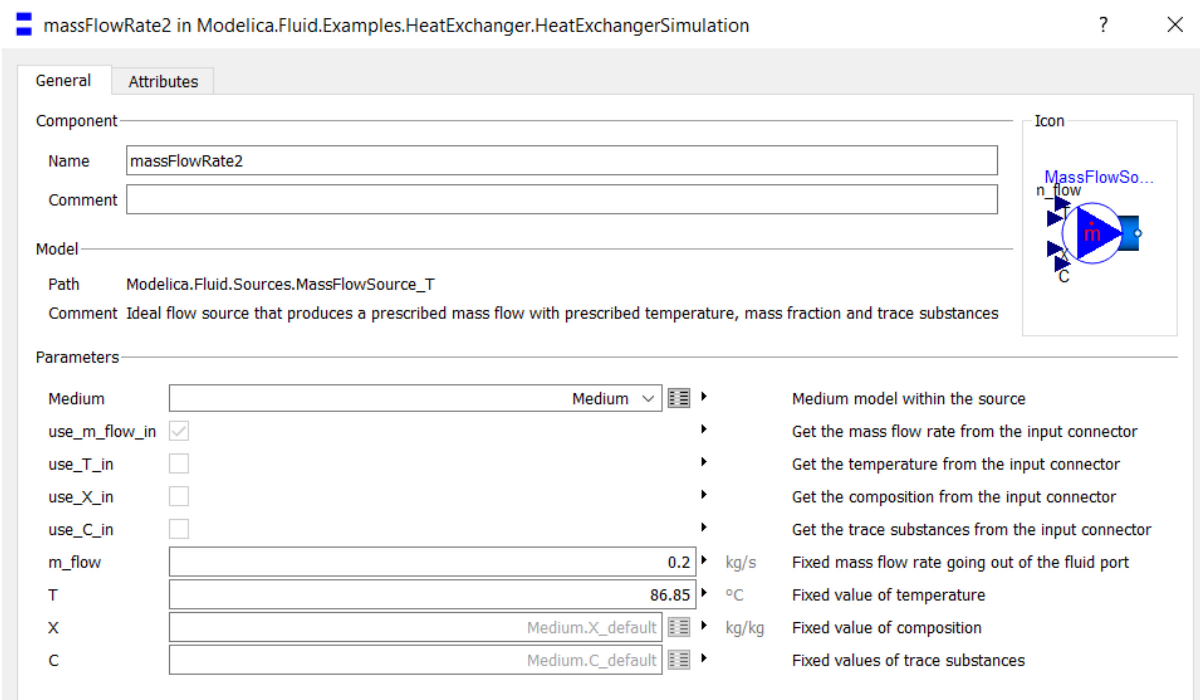
The system specifications we have determined for the diagram are as follows.



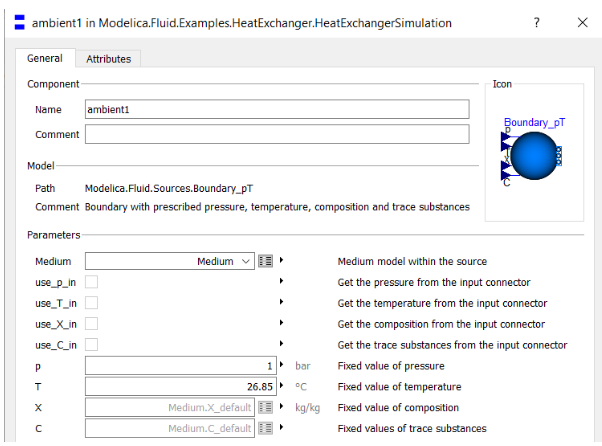
This is an m flow source. We can enter the m flow rate and temperature values that we want the source to create into the parameters.

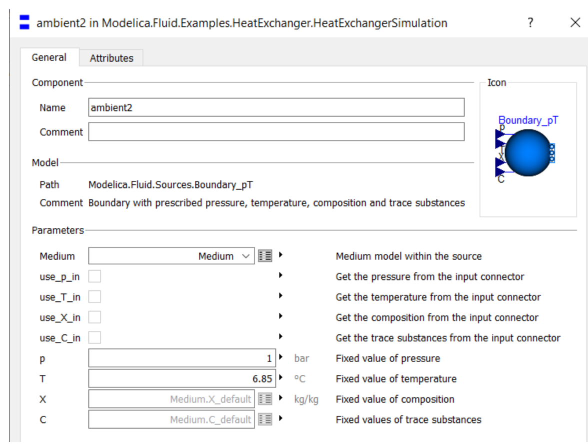


While using the same m flow value in our second m flow source, we used a higher temperature value.

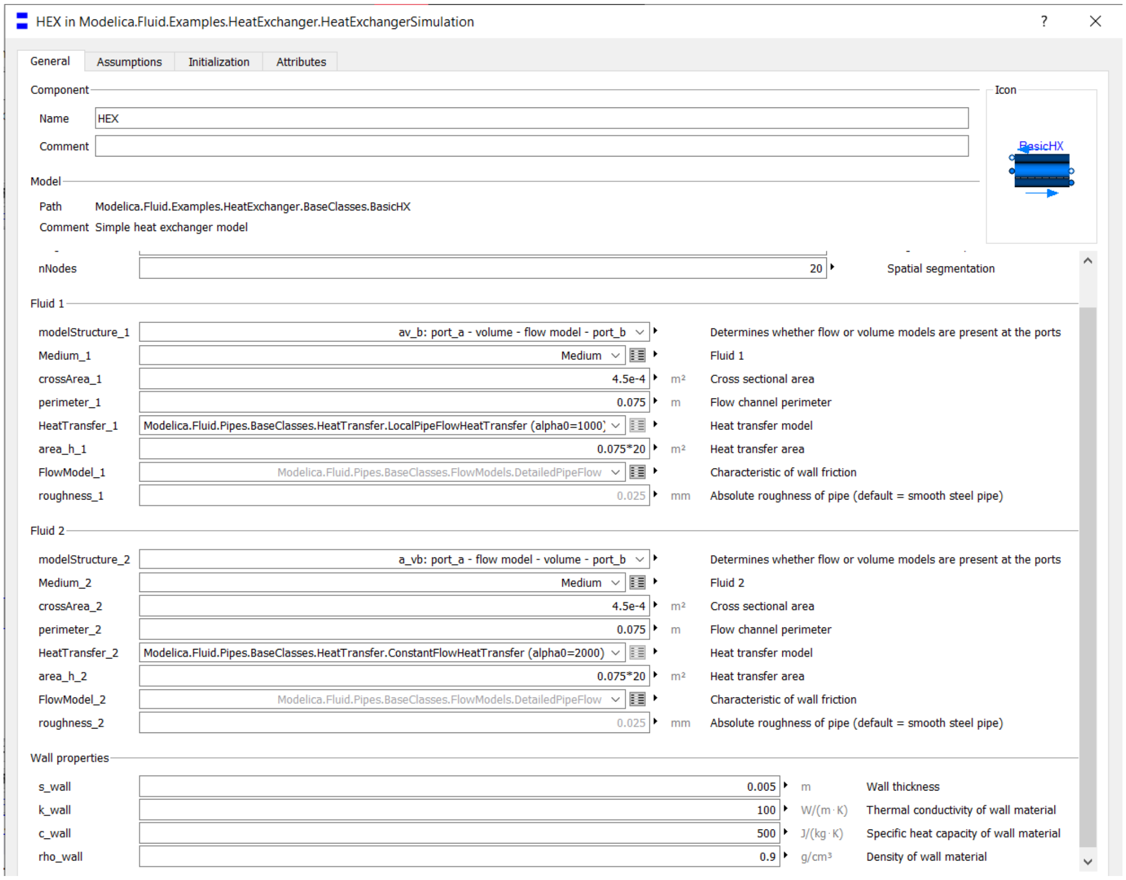


We use a different temperature value for each boundary to examine the temperature change while the pressures remain constant at the boundary.

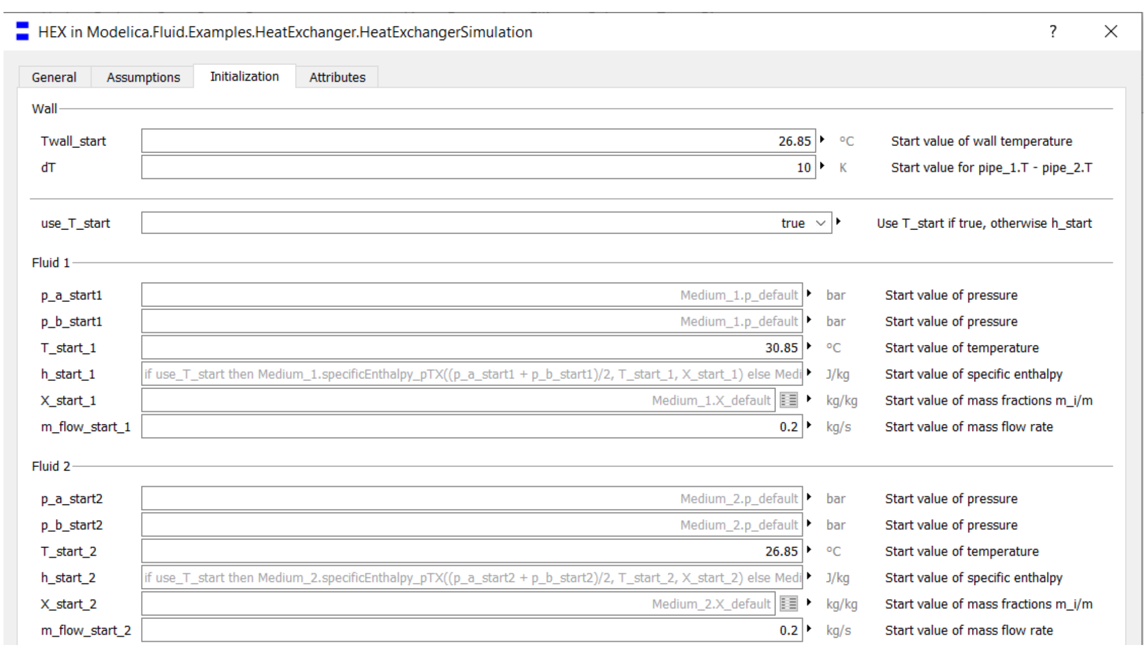




Due to the structure of the heat exchanger, there are two different fluids. and there are pipes through which fluids flow. We write the cross-sectional area, heat transfer type and roughness values of these pipes. We can also determine the constant values (wall thickness, thermal conductivity, specific heat capacity, density of the wall etc.) that we use in heat transfer processes.



For heat transfer, we determine the initial temperatures of the fluids and the wall. We will read the graphs over the change of these values.



RESULTS

Now we will simulate the system and examine the pressure and flow changes.

Here we see the flow rate change.

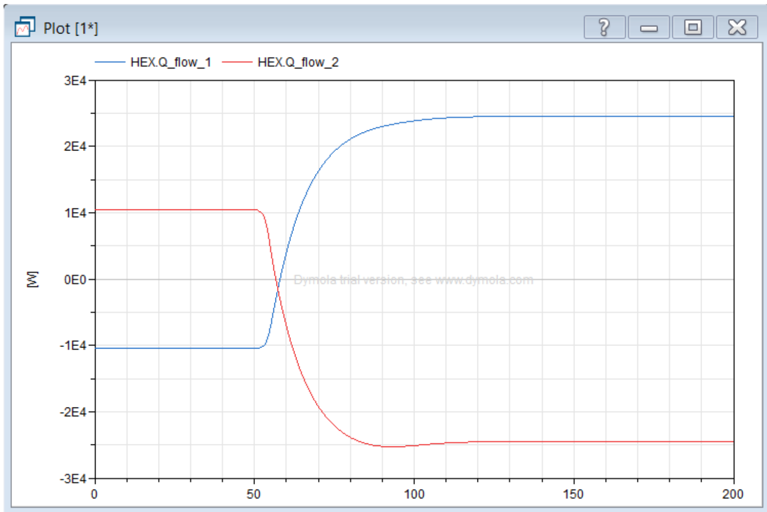


Figure 18: Total Heat Flow Rate of Pipe 1 and Pipe 2

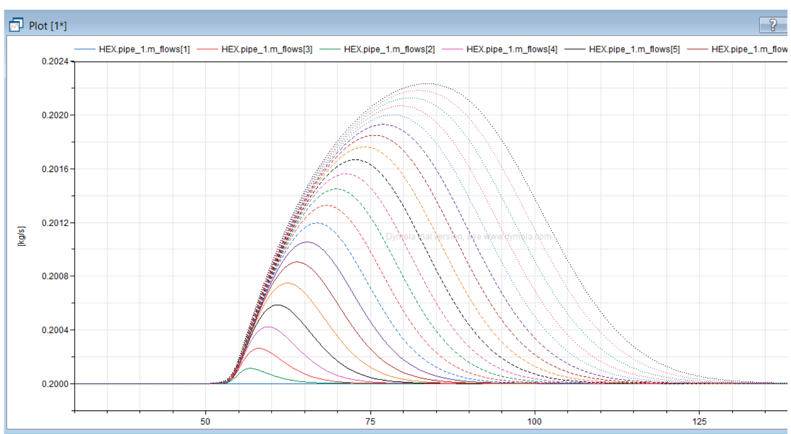


Figure 19:Mass flow rates of fluid across boundaries for pipe 1

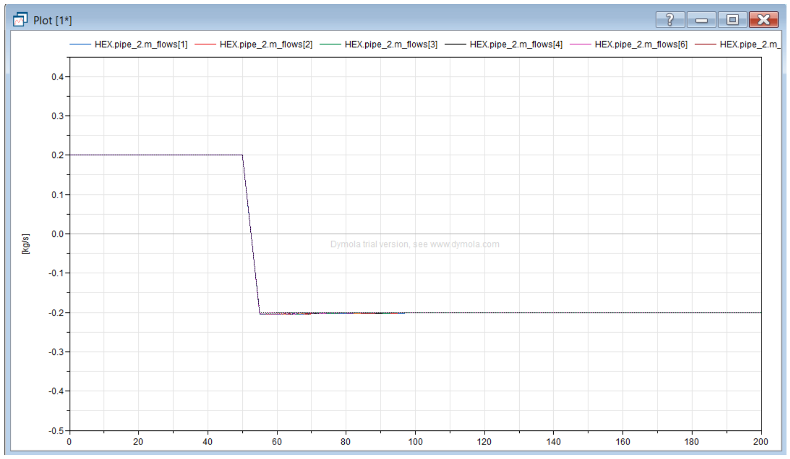
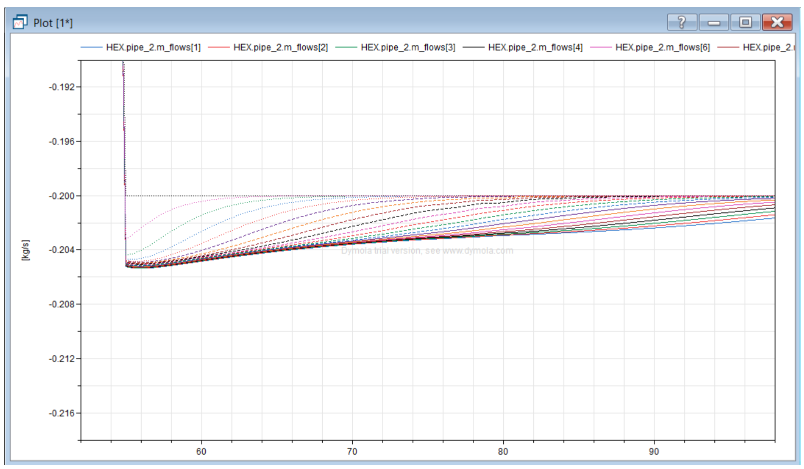


Figure 20: Mass flow rates of fluid across boundaries for pipe 2

If we zoom in to the above graph



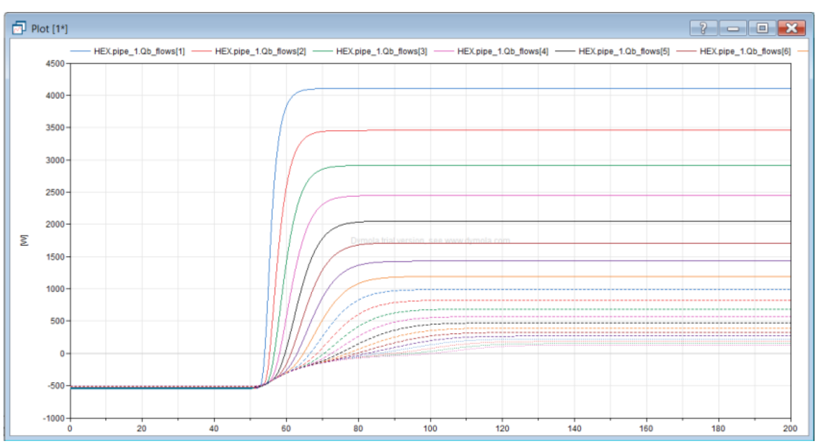


Figure 21:Heat flow rate for pipe 1

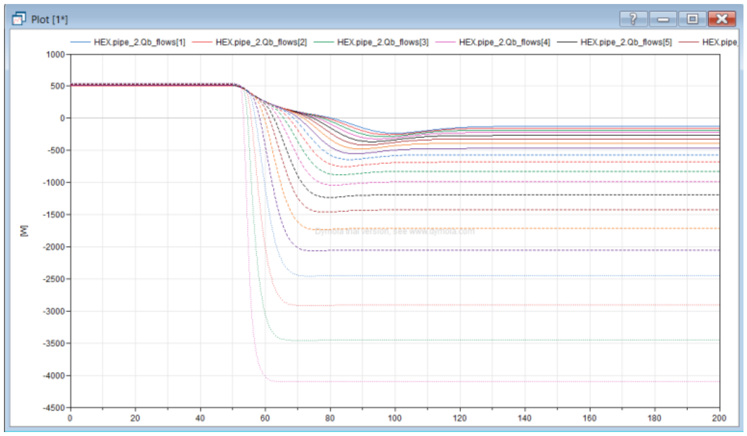


Figure 22:Heat flow rate for pipe 2

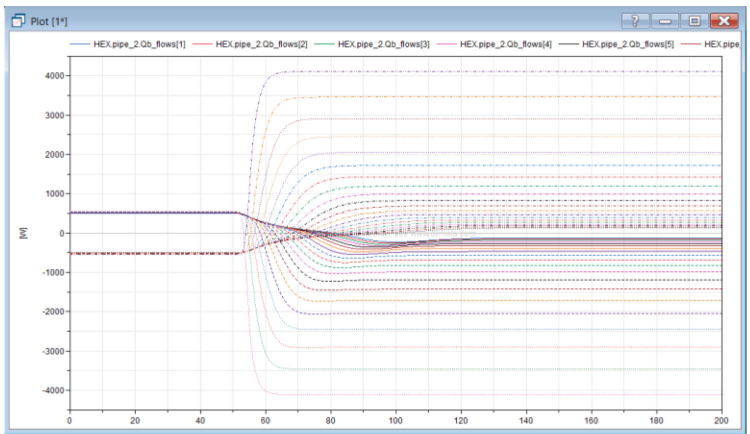


Figure 23: Heat flow rate of the both pipes in the same graph

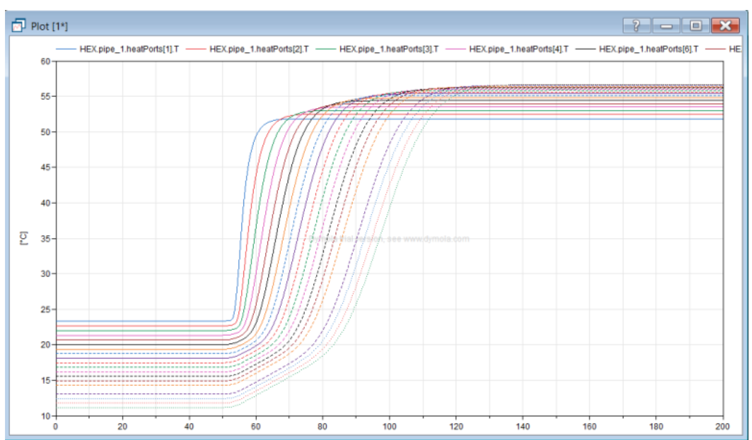


Figure 24: Port Temperatures for Pipe 1

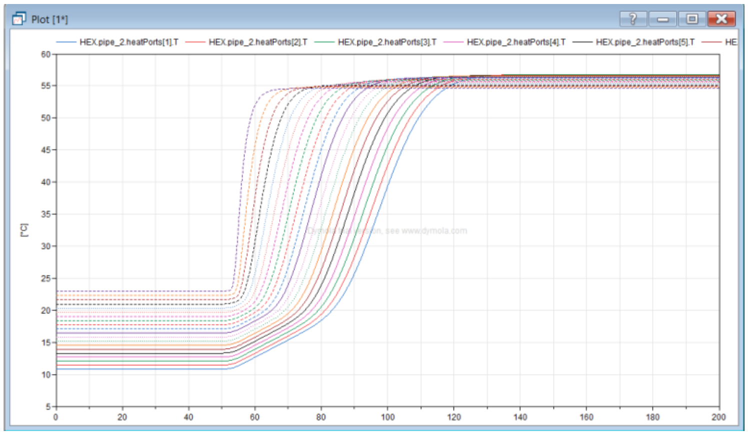


Figure 25: Port Temperatures for pipe 2

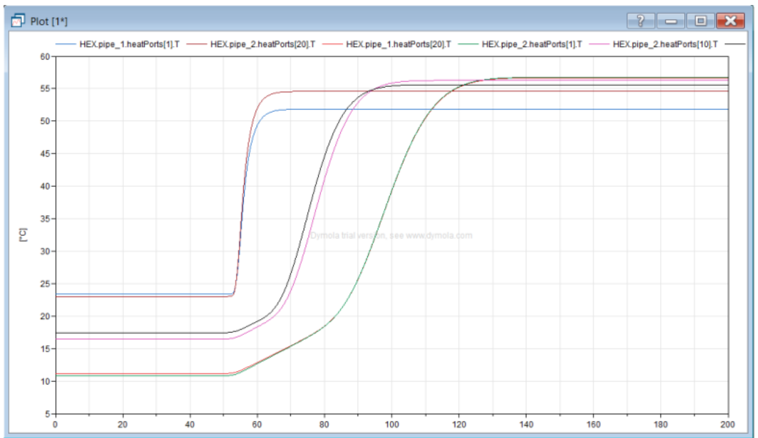


Figure 26: Port temperature comparison of pipe 1 and pipe 2 on the same ports

3.5) SIMPLE COOLING

This is a simple cooling example using flow between two identical ambients and a convectional heat transfer against a prescribed heat flow.

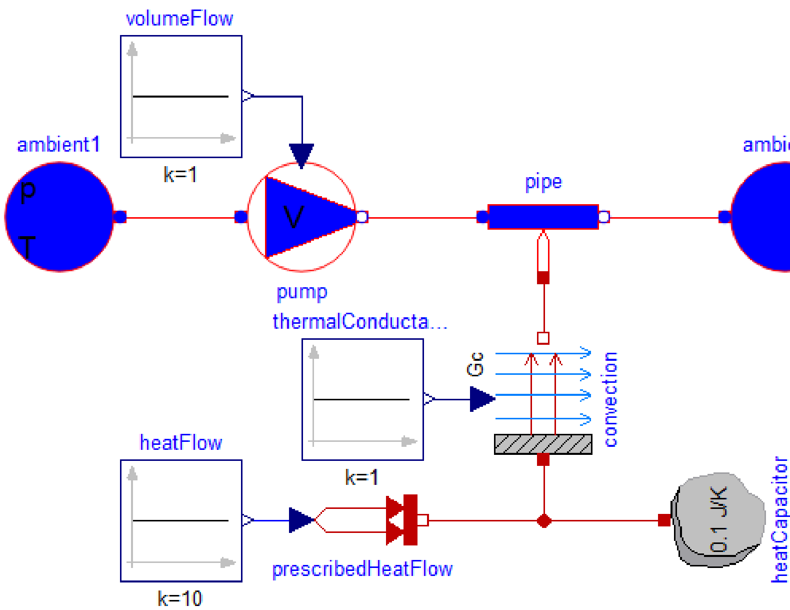
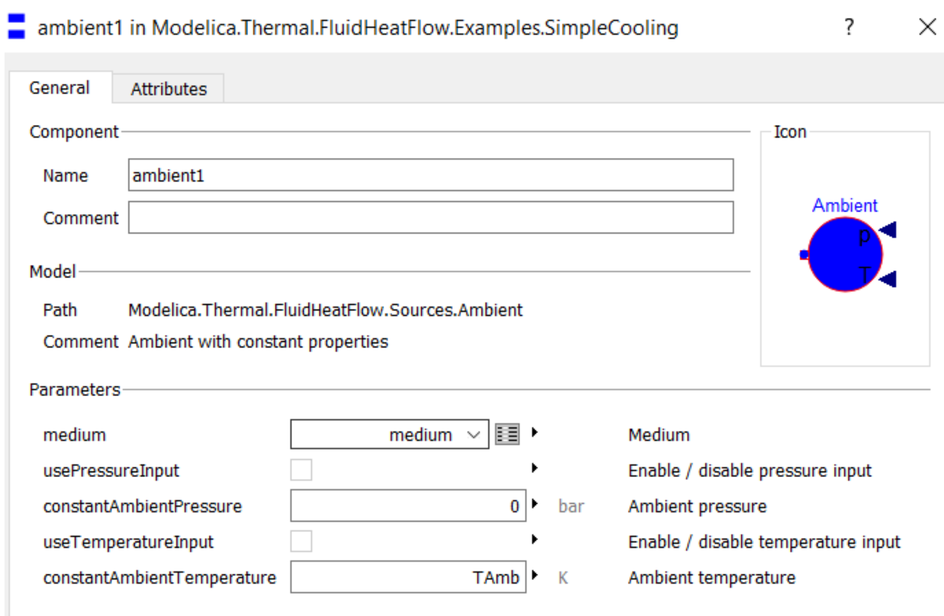
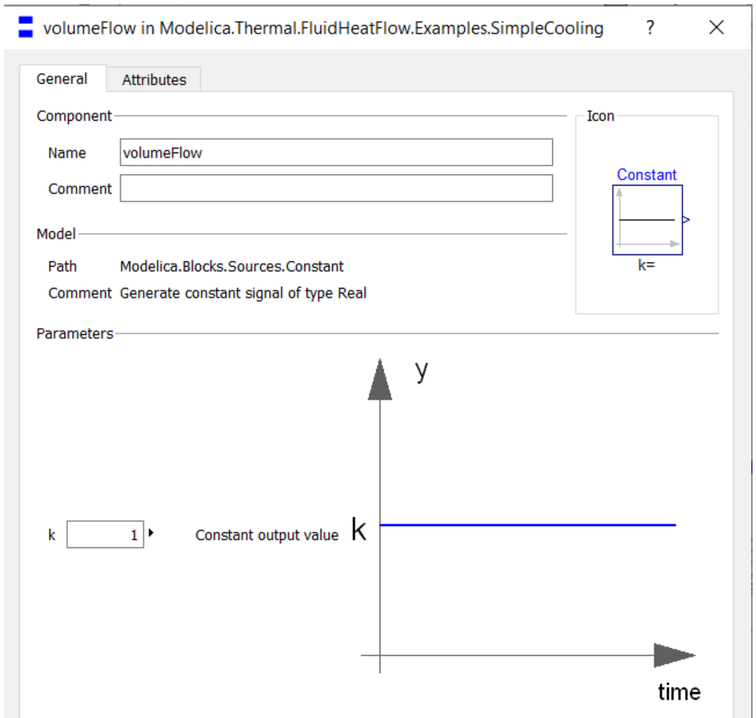


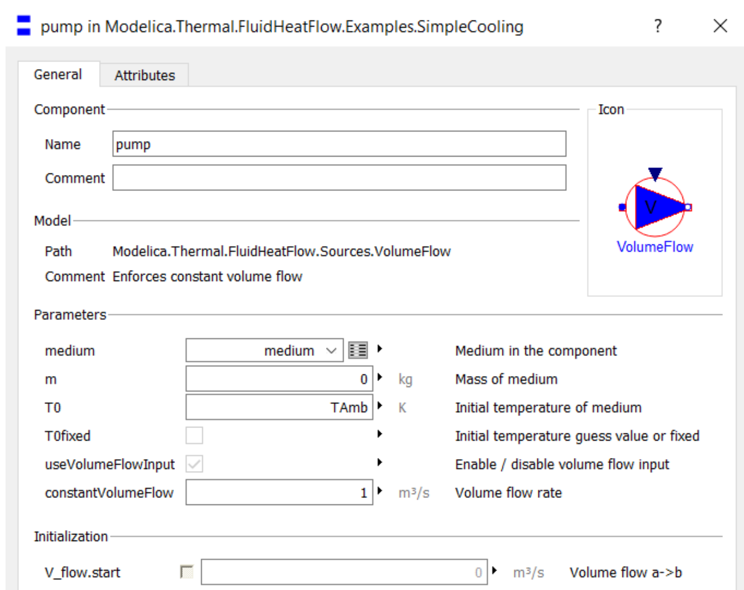
Figure 27: Simple Cooling Diagram

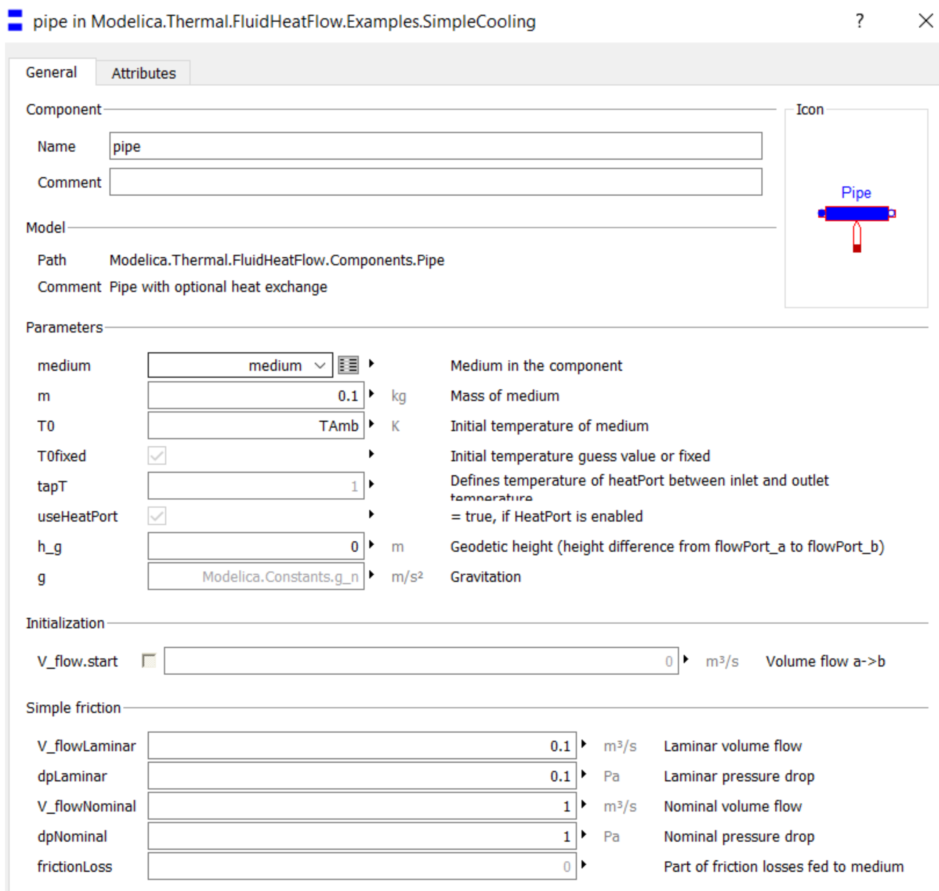


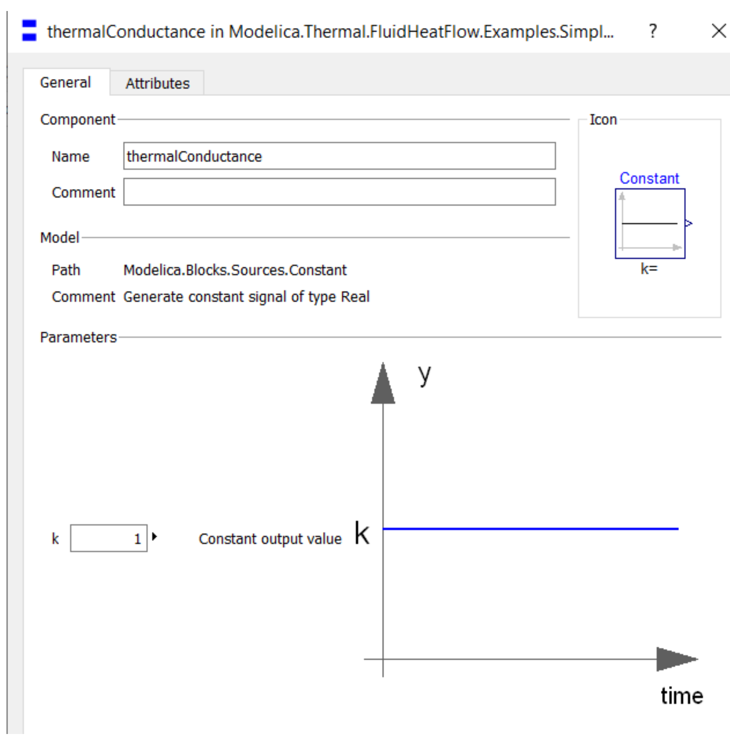
Volume flow is constant at pump.

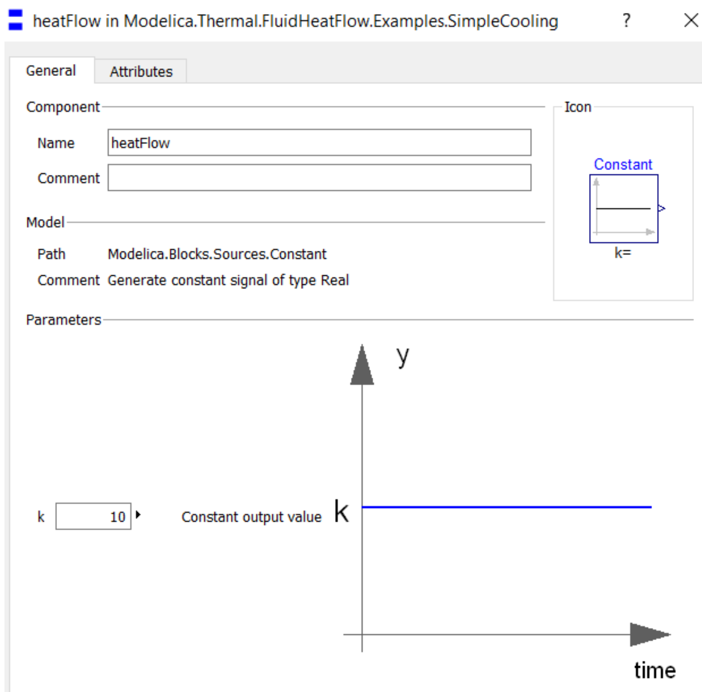


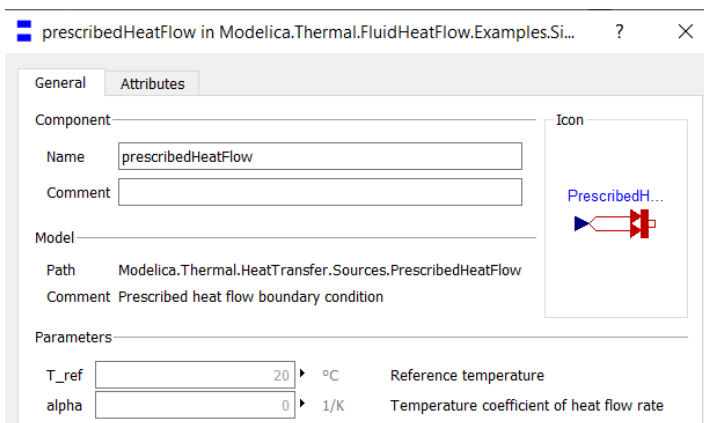
Constant volume flow.

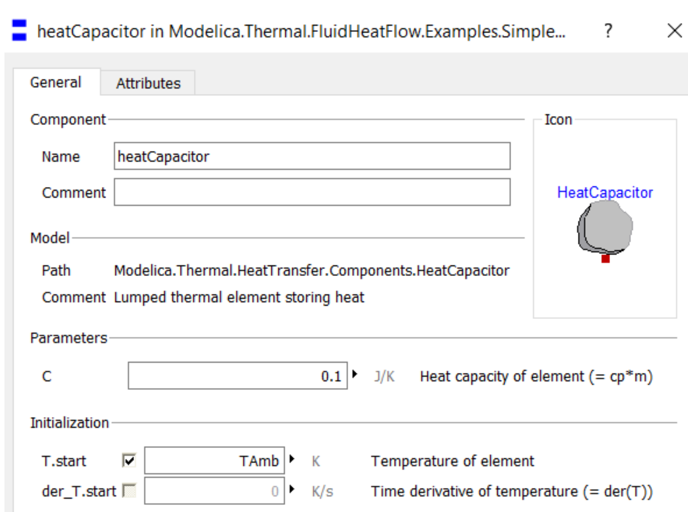



pipe in Modelica.Thermal.FluidHeatFlow.Examples.SimpleCooling
General
Attributes
Component
Name pipe
Comment
Icon
Pipe
Model
Path Modelica.Thermal.FluidHeatFlow.Components.Pipe
Comment Pipe with optional heat exchange
Parameters
medium medium Medium in the component
m 0.1 kg Mass of medium
T0 TAmb K Initial temperature of medium
T0fixed Initial temperature guess value or fixed
tapT 1 Defines temperature of heatPort between inlet and outlet temperature
useHeatPort = true, if HeatPort is enabled
h_g 0 m Geodetic height (height difference from flowPort_a to flowPort_b)
g Modelica.Constants.g_n m/s² Gravitation
Initialization
V_flow.start 0 m³/s Volume flow a->b
Simple friction
V_flowLaminar 0.1 m³/s Laminar volume flow
dpLaminar 0.1 Pa Laminar pressure drop
V_flowNominal 1 m³/s Nominal volume flow
dpNominal 1 Pa Nominal pressure drop
frictionLoss 0 Part of friction losses fed to medium


thermalConductance in Modelica.Thermal.FluidHeatFlow.Examples.Simpl...
General
Attributes
Component
Name thermalConductance
Comment
Icon
Constant
k=
Model
Path Modelica.Blocks.Sources.Constant
Comment Generate constant signal of type Real
Parameters
k 1 Constant output value
y
k
time


heatFlow in Modelica.Thermal.FluidHeatFlow.Examples.SimpleCooling
General
Attributes
Component
Name heatFlow
Comment
Model
Path Modelica.Blocks.Sources.Constant
Comment Generate constant signal of type Real
Icon
Constant
k=
Parameters
k 10
Constant output value
y
k
time


prescribedHeatFlow in Modelica.Thermal.FluidHeatFlow.Examples.Si...
General
Attributes
Component
Name prescribedHeatFlow
Comment
Model
Path Modelica.Thermal.HeatTransfer.Sources.PrescribedHeatFlow
Comment Prescribed heat flow boundary condition
Icon
PrescribedH...
Parameters
T_ref 20 °C Reference temperature
alpha 0 1/K Temperature coefficient of heat flow rate


heatCapacitor in Modelica.Thermal.FluidHeatFlow.Examples.Simple...
General
Attributes
Component
Name heatCapacitor
Comment
Model
Path Modelica.Thermal.HeatTransfer.Components.HeatCapacitor
Comment Lumped thermal element storing heat
Icon
HeatCapacitor
Parameters
C 0.1 J/K Heat capacity of element (= cp*m)
Initialization
T.start TAmb K Temperature of element
der_T.start 0 K/s Time derivative of temperature (= der(T))

RESULTS

Temperature at the outlet of the pipe is increasing with time when inlet temperature is constant.

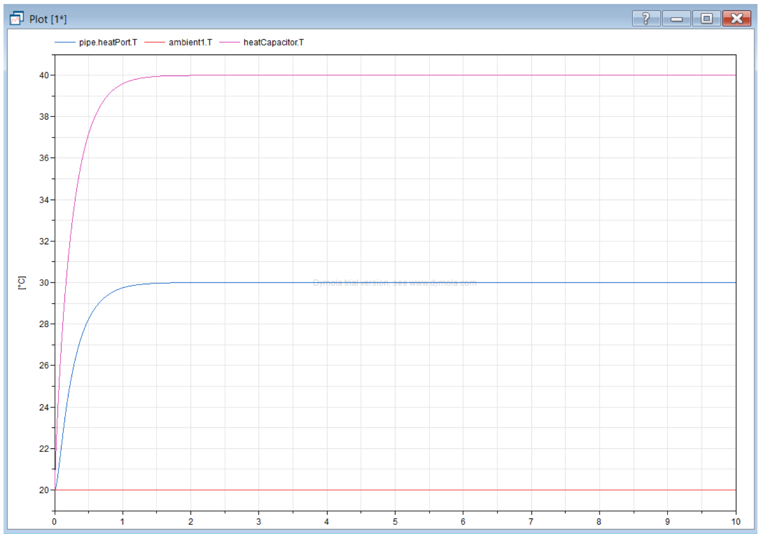


Figure 28:Temperatures of ambient vs. pipe vs. heat capacitor

As we can see from above graph pipe temperature is the mean average of ambient pressure and heat capacitor temperatures.

,

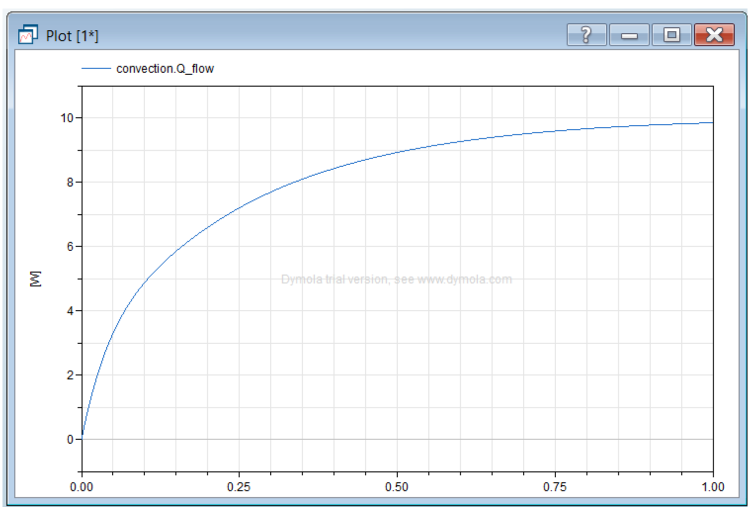


Figure 29: Heat flow rate of convection

4. BUILDING THE ENGINE COOLING MODEL

The model we built is basically a harmony of two examples in the Modelon library, CoolingLoop_ExternalTempControl from cooling library and TestCell_NaturallyAspiratedSI from engine dynamics library. In the combination process there is a lot of parameters and parts that needs to be changed.

This diagram in the Figure 30 shows how the model we built works simply.

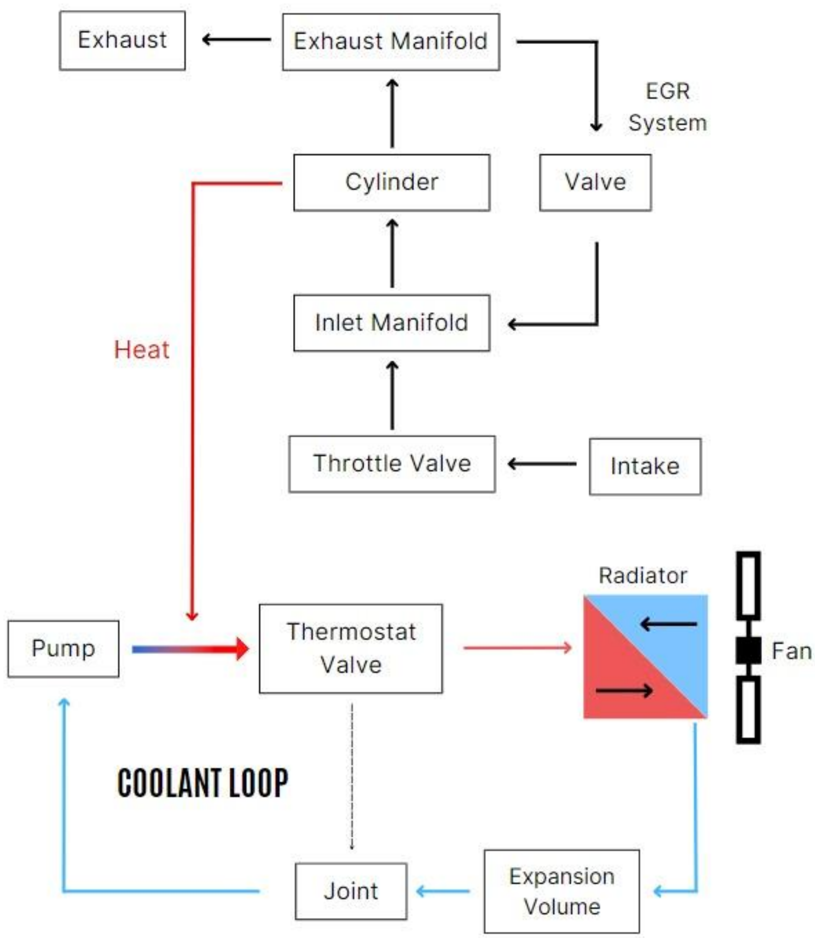


Figure 30: Simple Diagram of the System

After some work we built the model in the Figure 31. For the parts we used look at the **Error! Reference source not found.**.

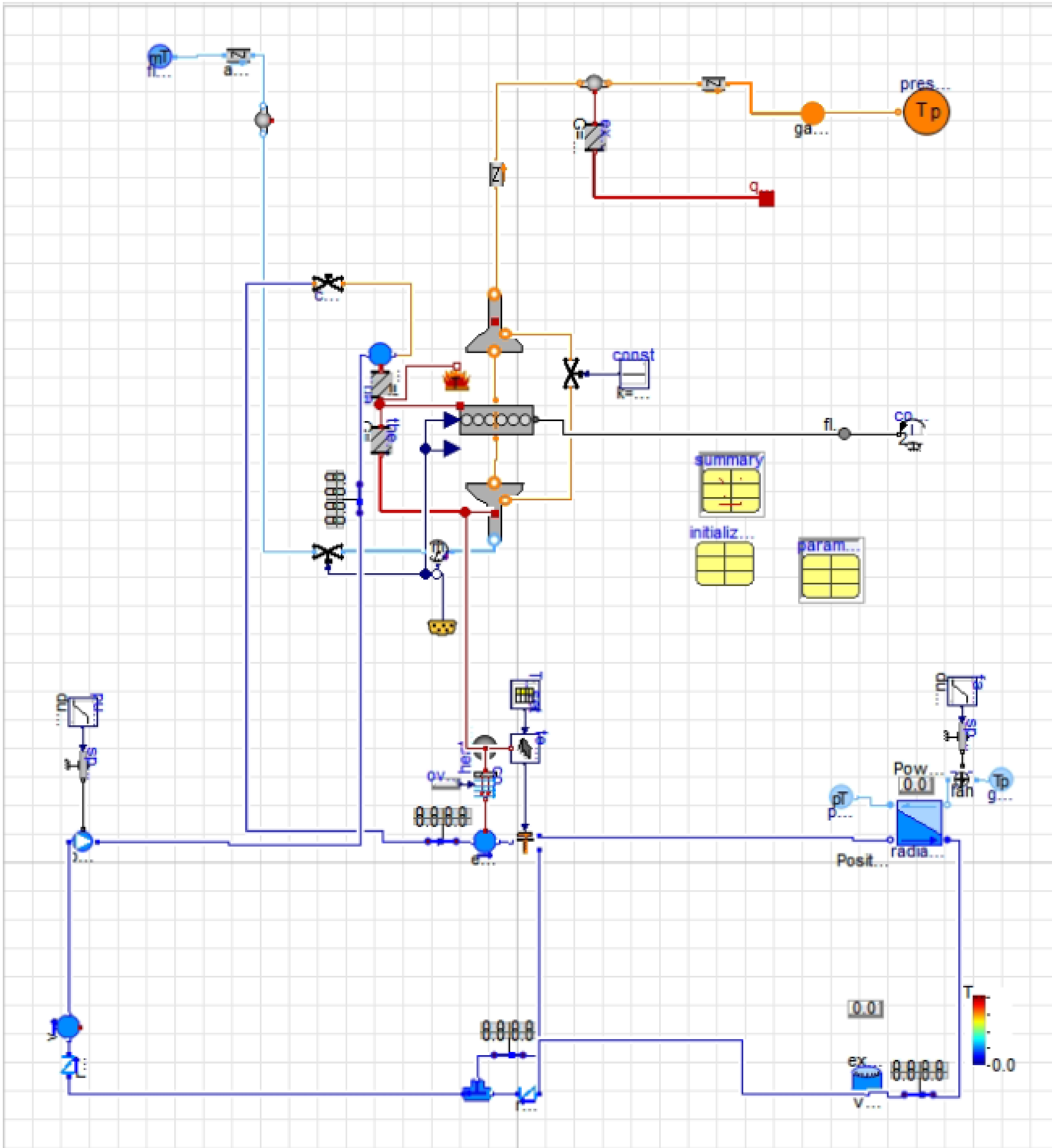


Figure 31: Engine Cooling Model in Dymola

There is a close look up for the cooling loop. Both pump and fan is working with a ramp signal input. So as the engine works we can control their speeds either increase or decrease. The control unit works with the table for checking the temperature and control the thermostat valve to open or close.

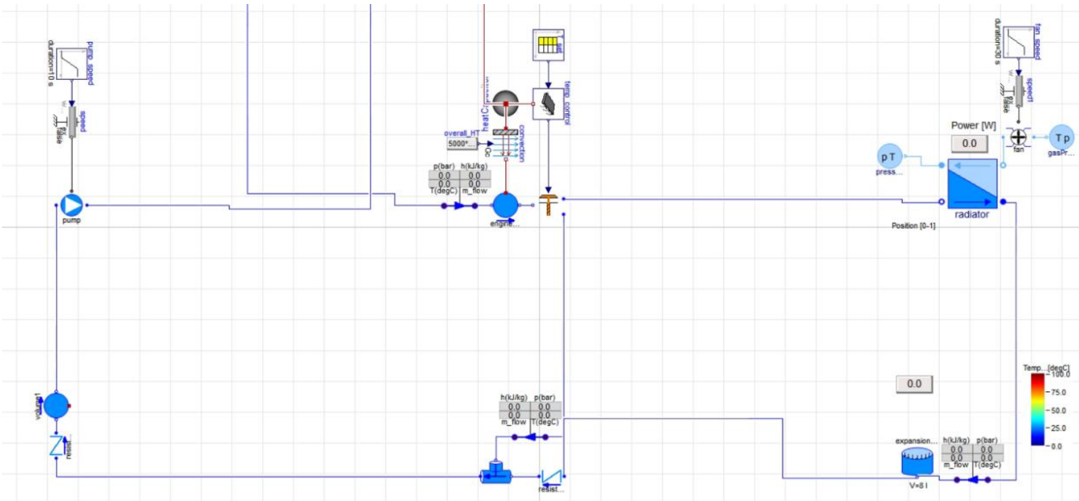


Figure 32: Close up to Cooling Loop

For engine we used the simple SI cylinder and controlled it with constant speed input not with the control unit (Engine Bus). For our model we used an external temperature source for simulating the combustion in the cylinder and give the system a constant temperature to heat the coolant.

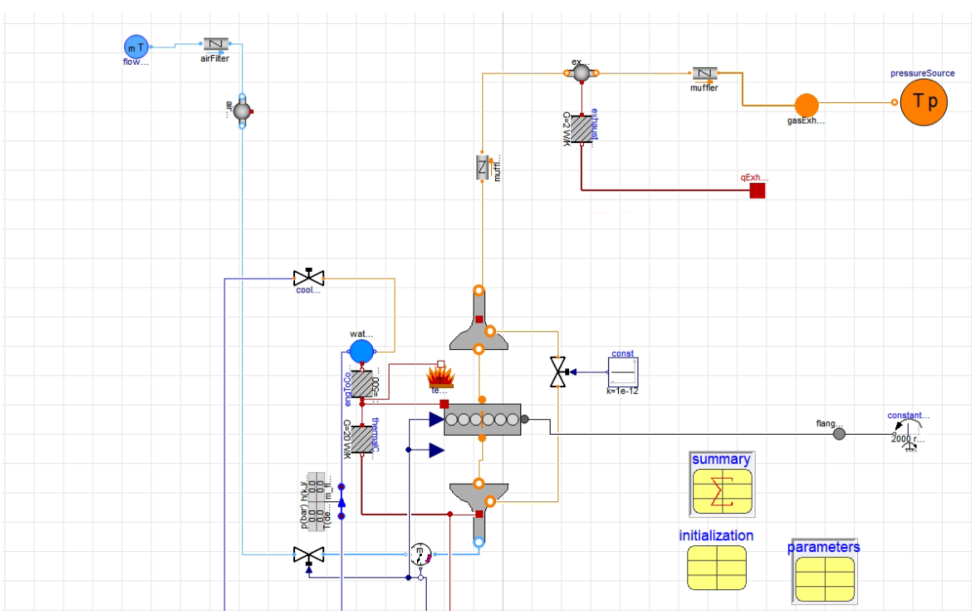


Figure 33: Close up to the engine

General Specifications

<u>Four Stroke Engine</u>

Numbers number of cylinder : 4

Compression ratio : 10

Displacement volume : $1.4 \times 10^{-3}\ m^3$

<u>Initialization</u>

Initial cylinder wall temperature: 300°C

Initial air filter temperature : 25°C

Initial inlet manifold pressure : 1.013 bar

Initial inlet manifold temperature : 25°C

Initial exhaust manifold pressure : 1.1 bar

Initial exhaust manifold temperature : 300°C

Initial engine coolant temperature : 0°C

Ambient Temperature : 0°C

Thermostat valve open/close temperature: 90 °C

<u>Radiator</u>

Liquid Side

Length : 0.42 m

Flow cross section area : 0.001615 m^2

Gas side

Length : 0.03 m

Flow cross section area : 0.012 m^2

5. TEST CONDITIONS

We tested our model in four conditions to see the reaction of the model to different working conditions. With optimal normal working conditions and two extreme conditions.

Normal Conditions at 2000 rpm

Pump and fan speed can see from the below graphs. For temperature source we used 300°C.

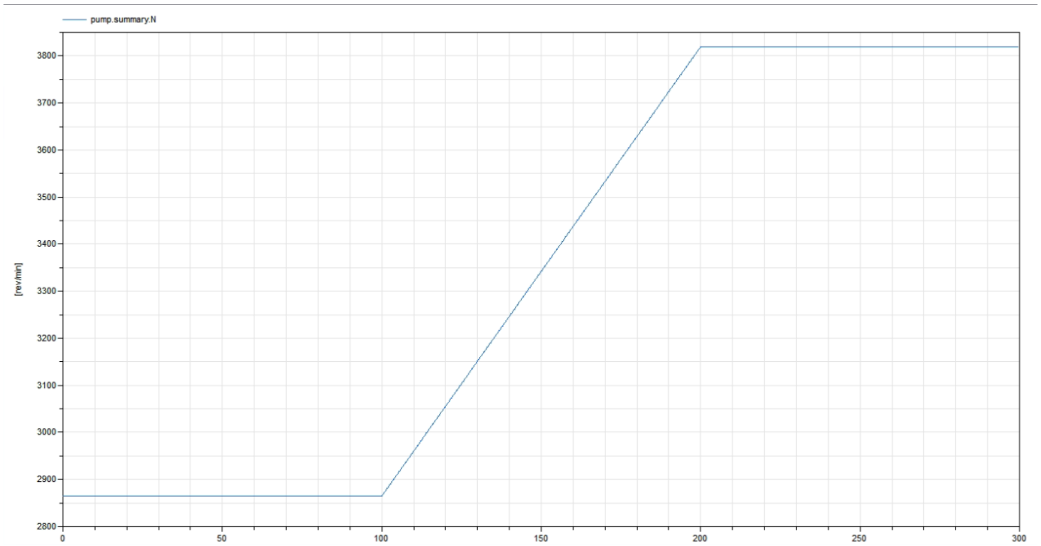


Figure 34: Pump speed (rpm) vs time(s) graph

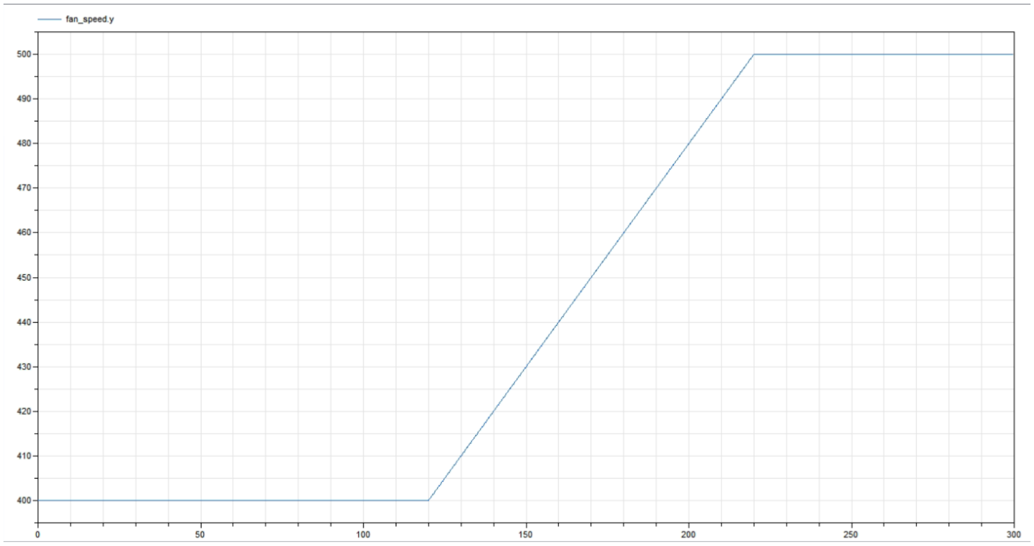


Figure 35: Fan speed (rad/s) to time(s) graph

Over Cooled Conditions at 2000 rpm

Pump and fan speed can see from the below graphs. In this experiment we used constant pump for reaching the over cooled state. For temperature source we used 300°C.

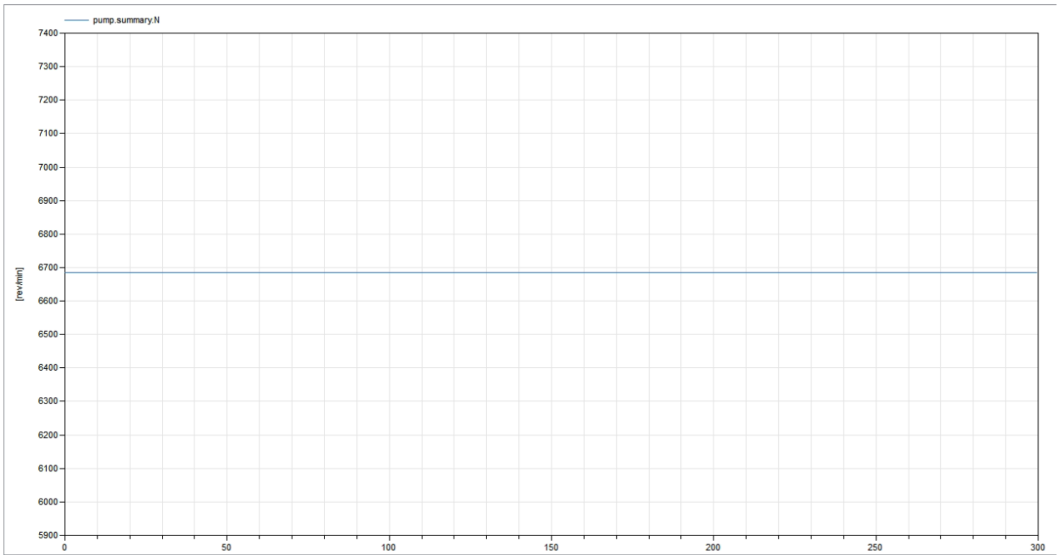


Figure 36:Pump speed (rpm) vs time(s) graph

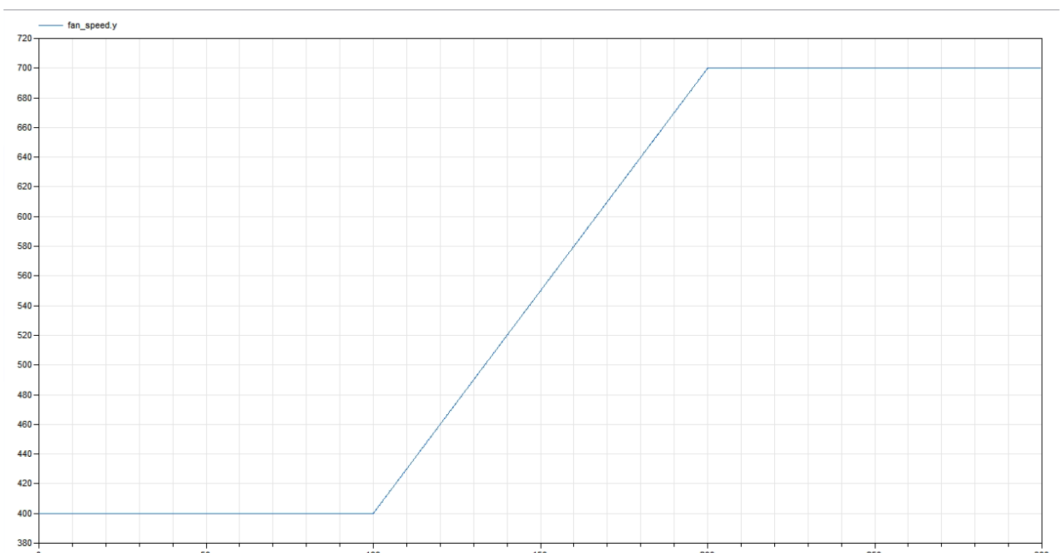


Figure 37: Fan speed (rad/s) to time(s) graph

Normal Conditions at 4000 rpm

Pump and fan speed can see from the below graphs. For temperature source we used 400°C.

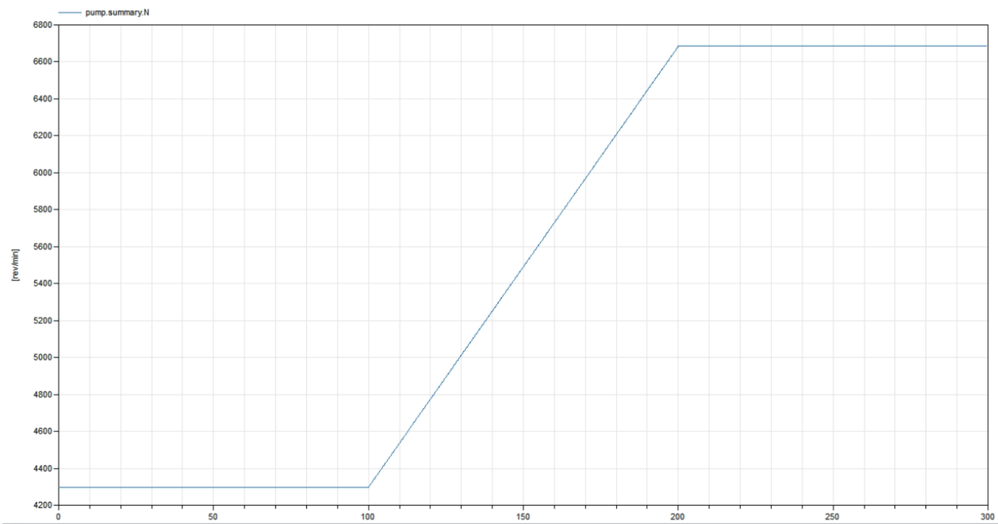


Figure 38: Pump speed (rpm) vs time(s) graph

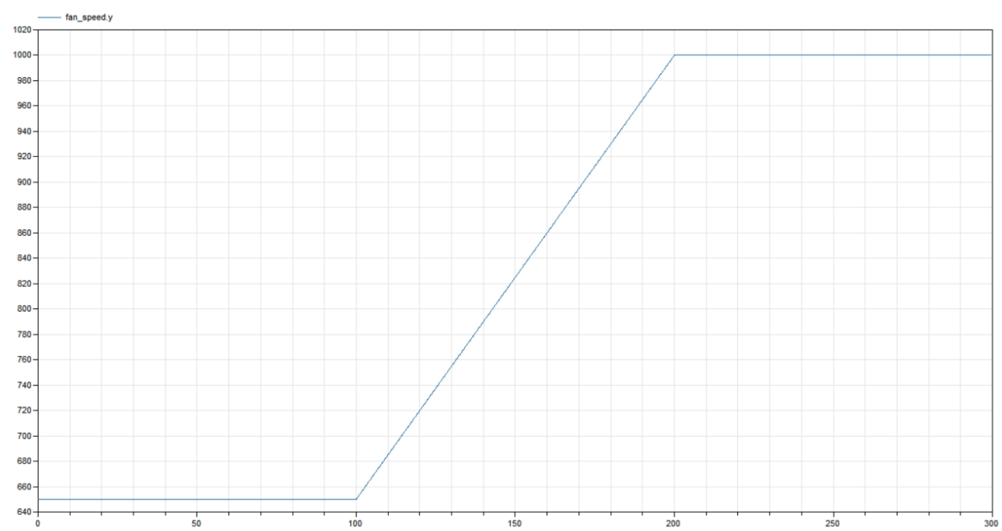


Figure 39: Fan speed (rad/s) to time(s) graph

Super Heated Conditions at 4000 rpm

Pump and fan speed can see from the below graphs. For temperature source we used 400°C

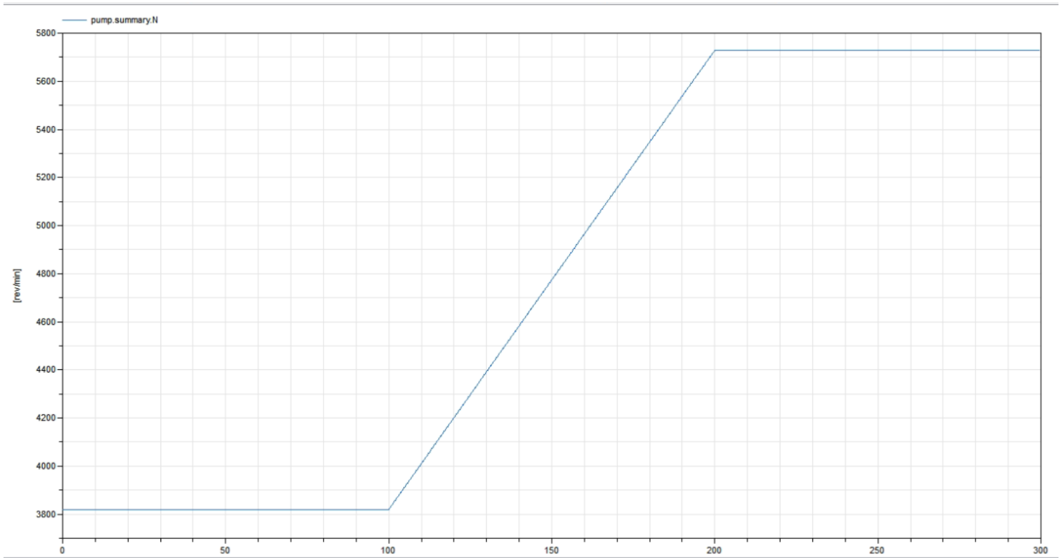


Figure 40: Pump speed (rpm) vs time(s) graph

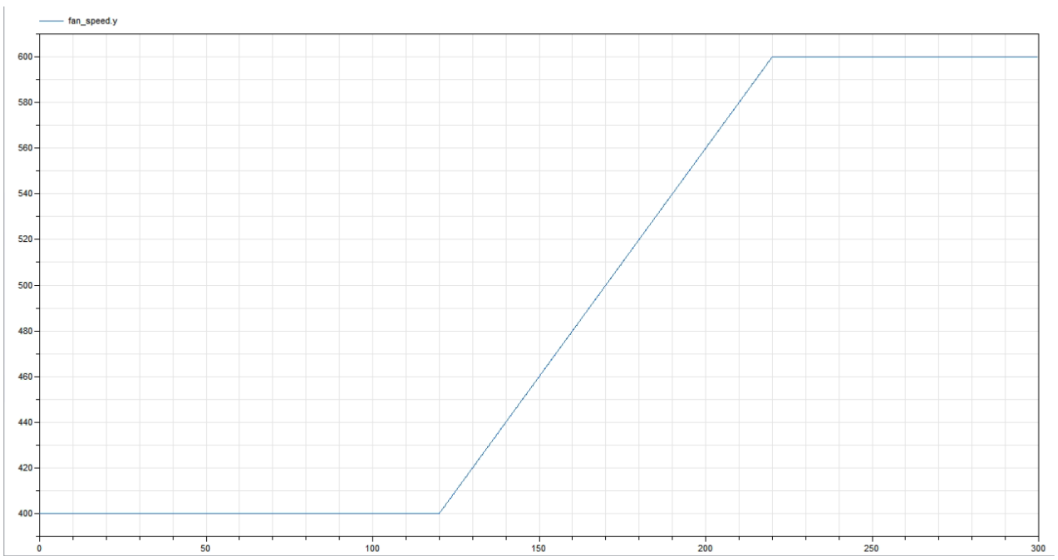


Figure 41: Fan speed (rad/s) to time(s) graph

6. RESULTS

TEST SIMULATIONS

2000 rpm Engine with Normal Conditions

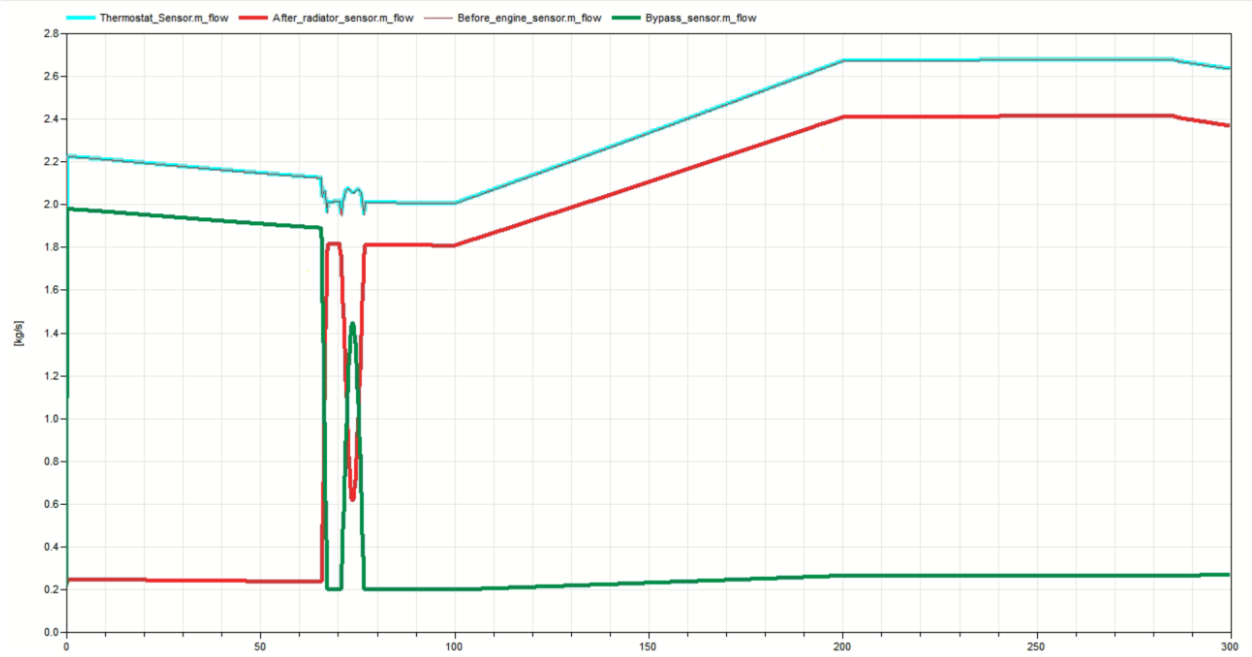


Figure 42: 2000 rpm Normal Conditions m_flow graph

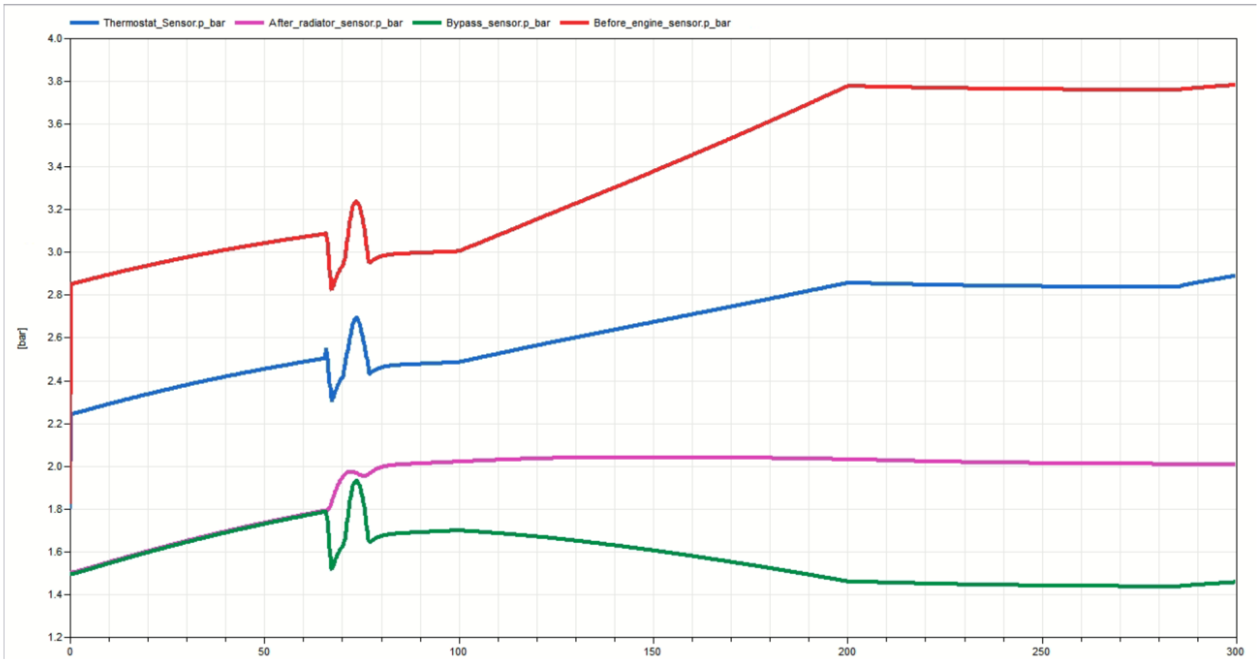


Figure 43: 2000 rpm Normal Conditions pressure graph

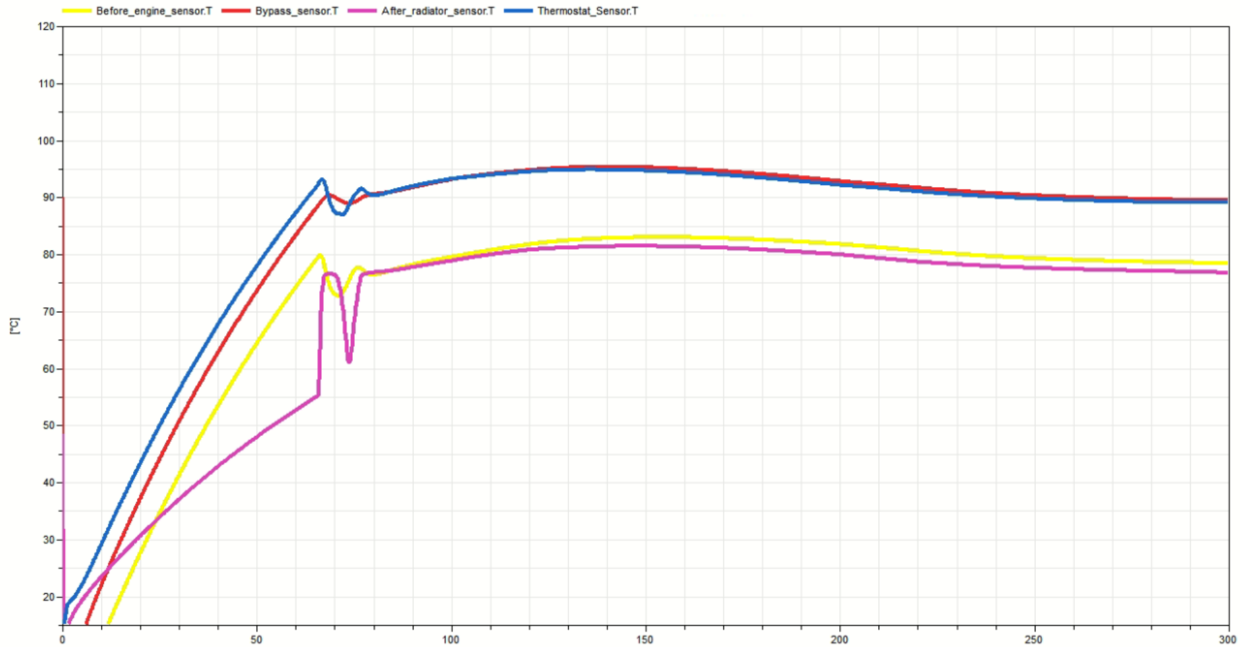


Figure 44: 2000 rpm Normal Conditions temperature graph

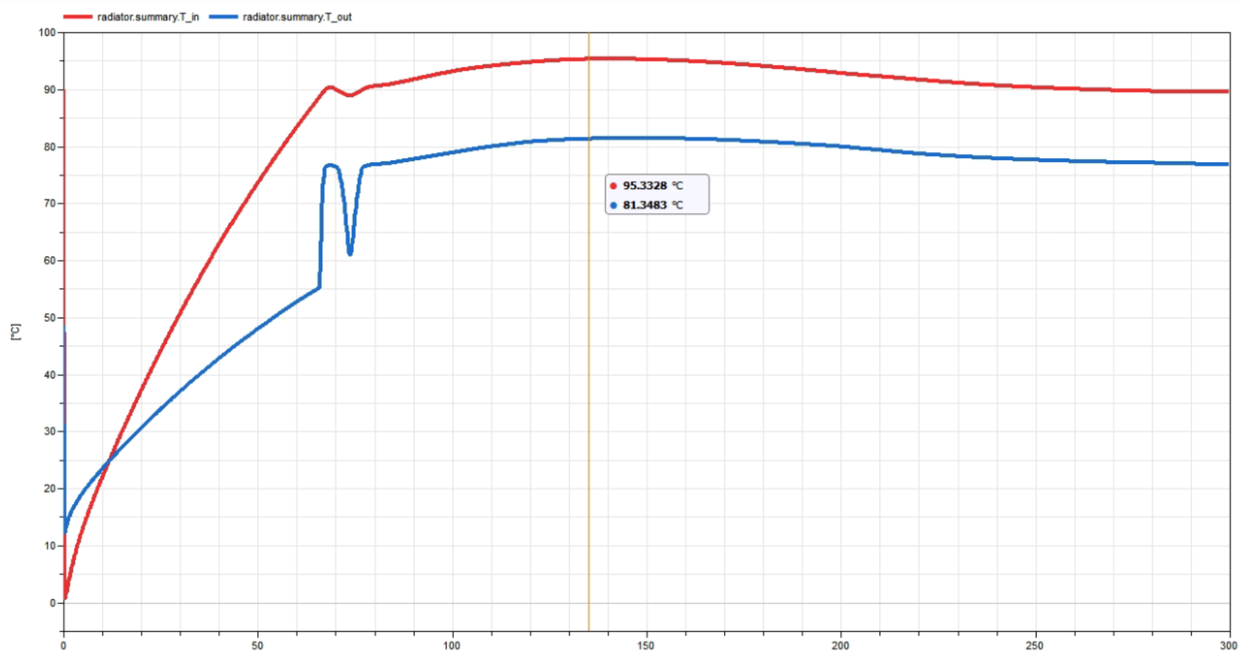


Figure 45: 2000 rpm Normal Conditions radiator T_in and T_out graph

2000 rpm Engine with Overcooling

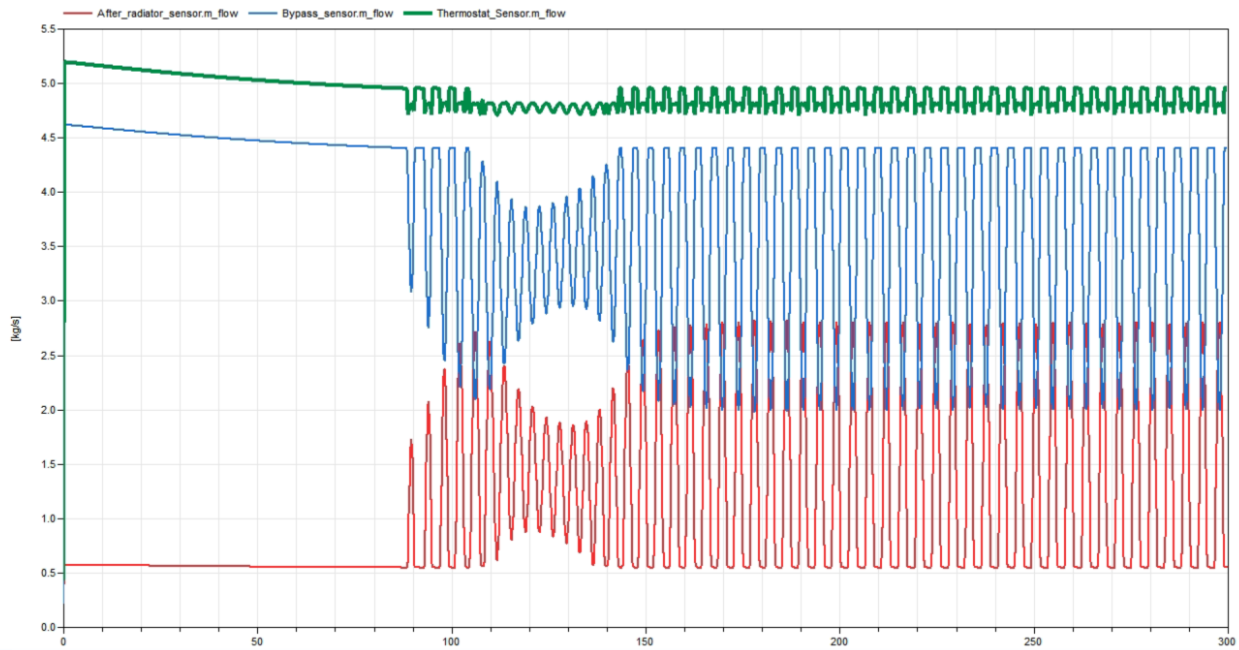


Figure 46: 2000 rpm Overcooling m_flow graph

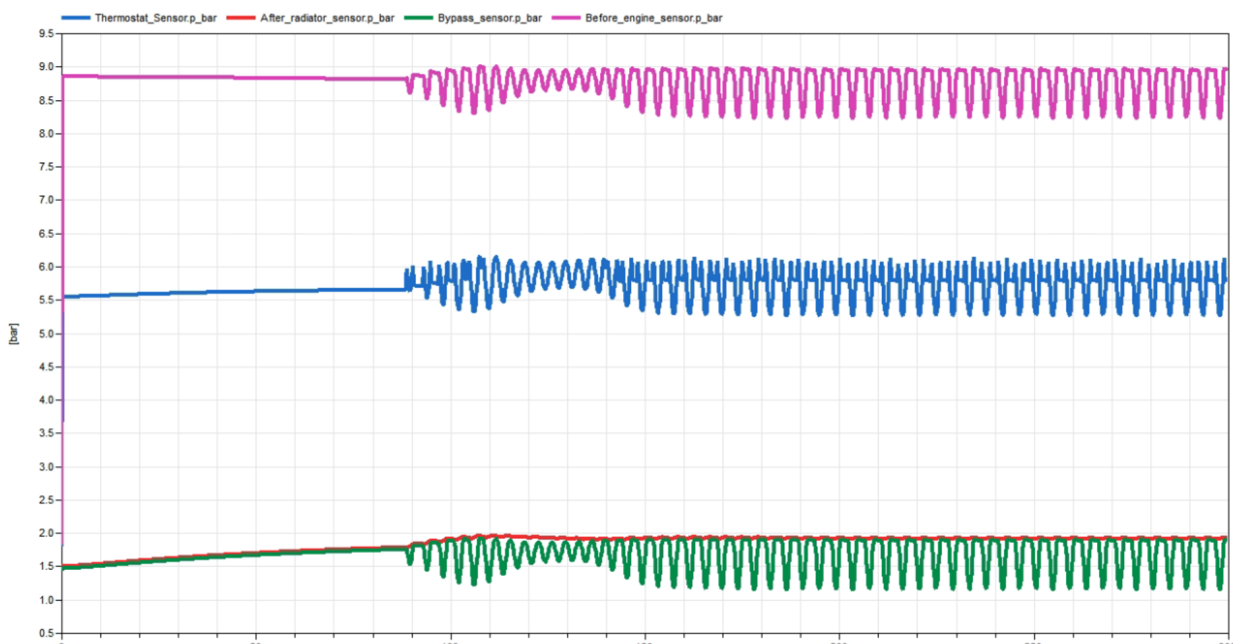


Figure 47: 2000 rpm Overcooling pressure graph

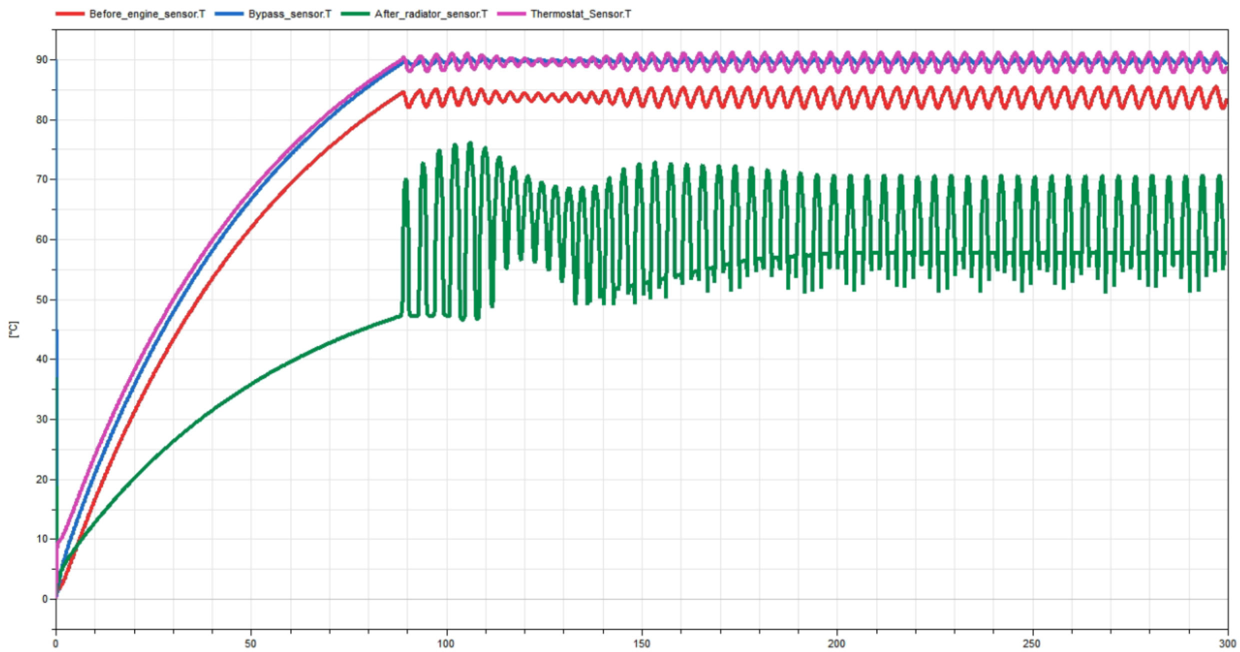


Figure 48: 2000 rpm Overcooling temperature graph

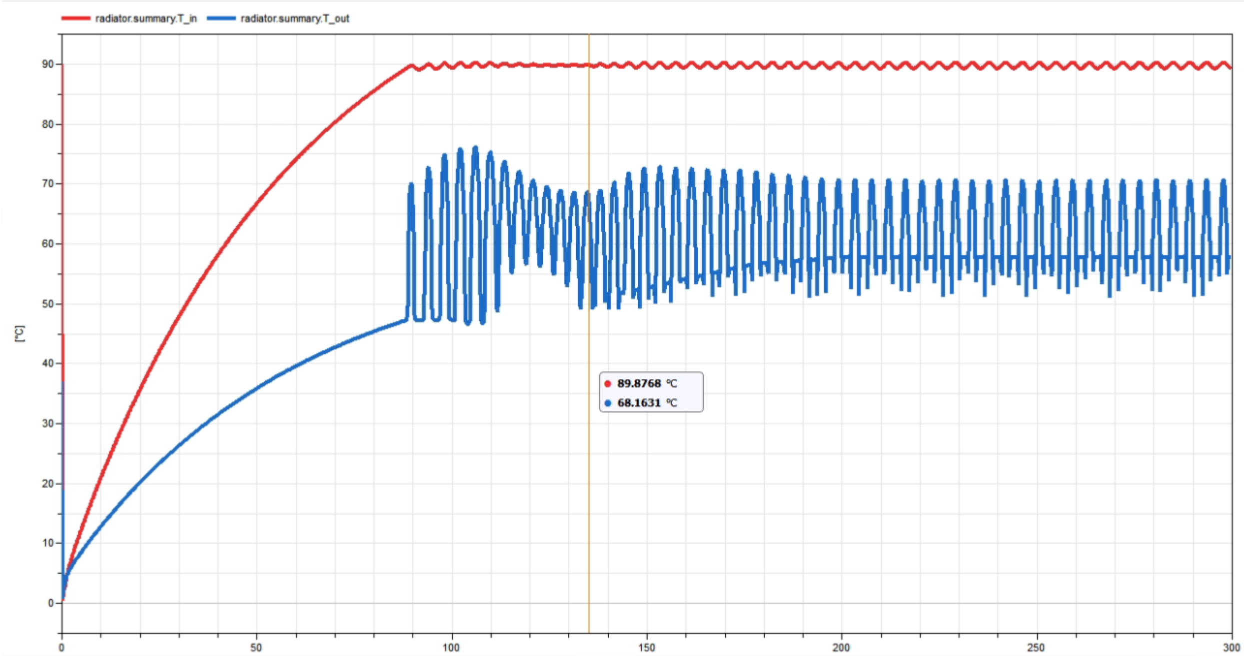


Figure 49: 2000 rpm Overcooling radiator T_in and T_out graph

4000 rpm Engine with Normal Conditions

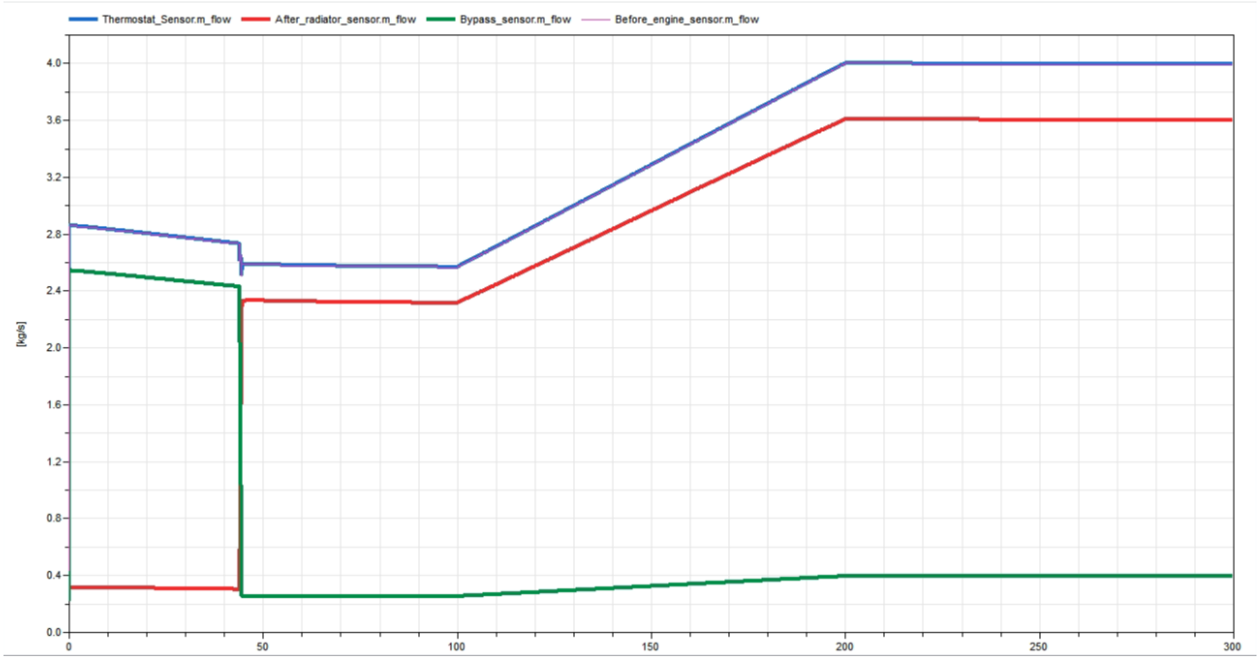


Figure 50: 4000 rpm Normal Conditions m_flow graph

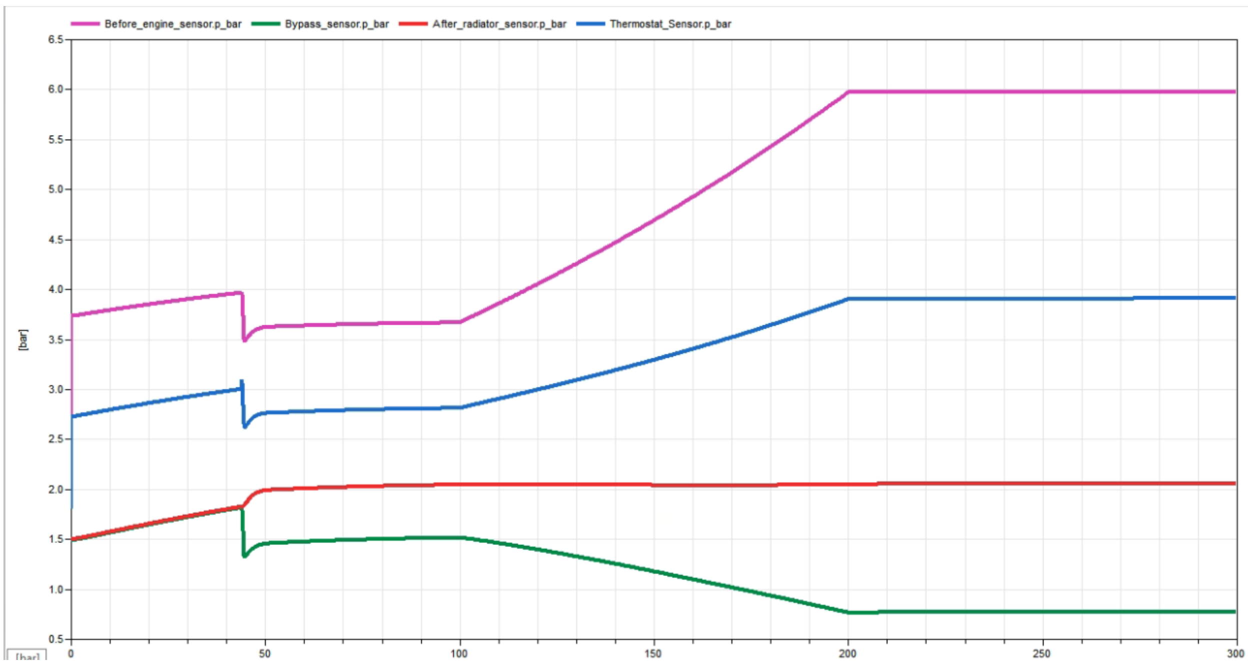


Figure 51: 4000 rpm Normal Conditions pressure graph

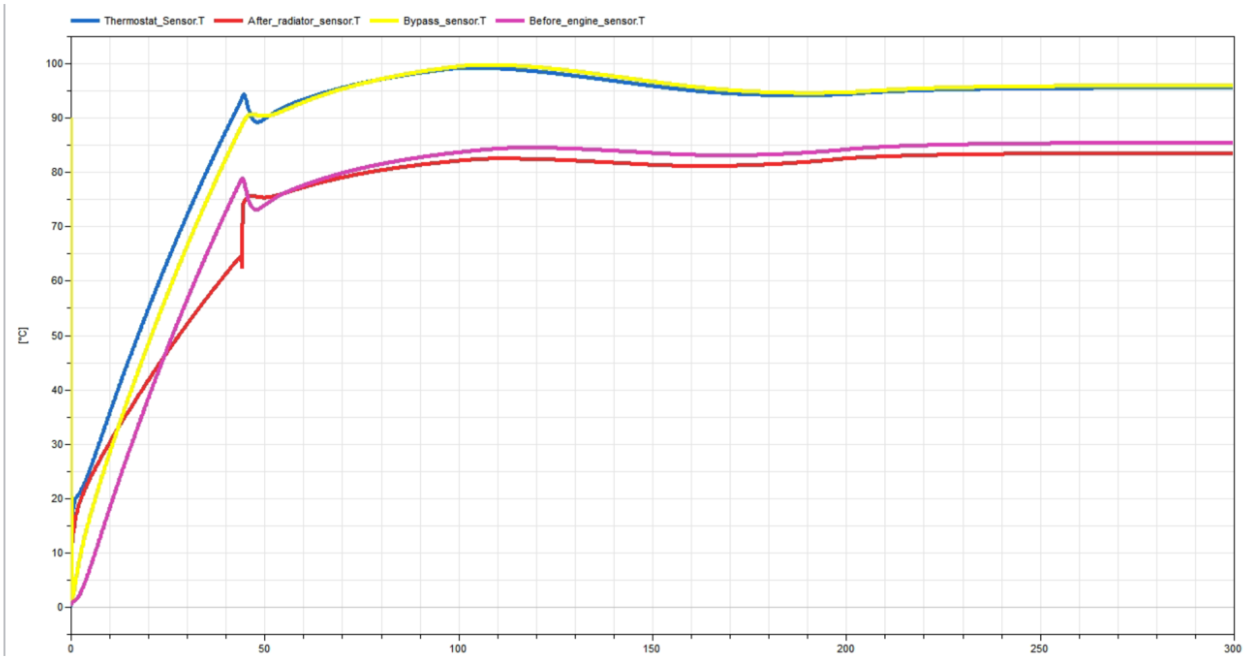


Figure 52: 4000 rpm Normal Conditions temperature graph

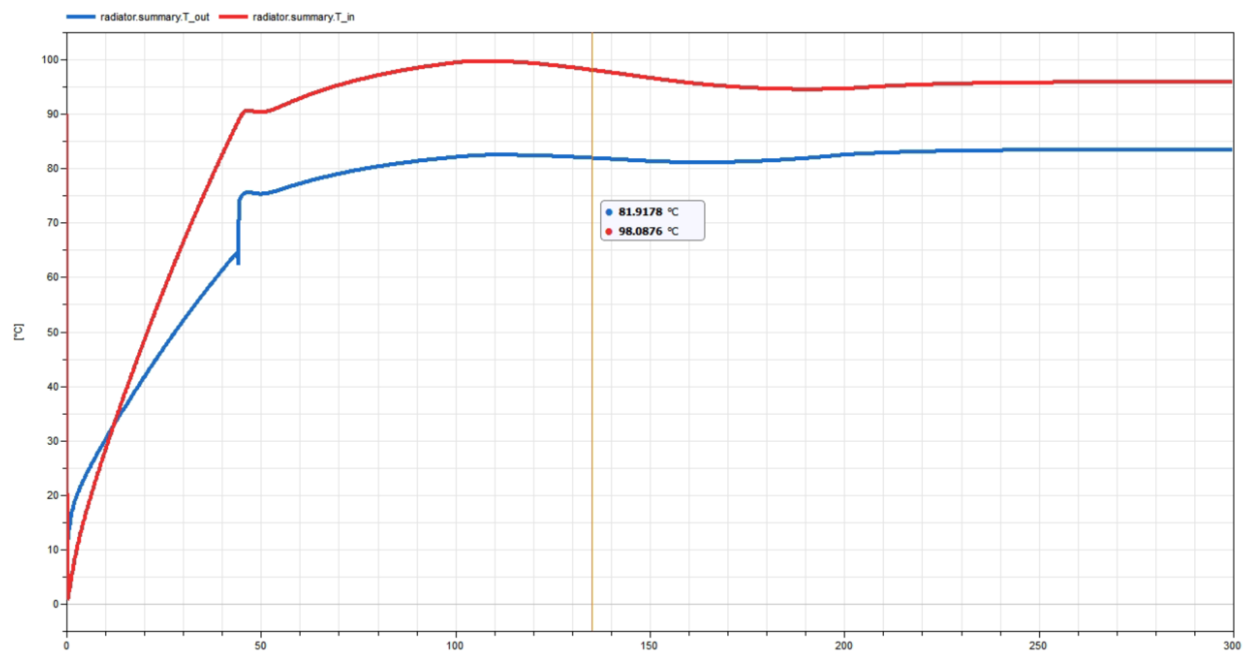


Figure 53: 4000 rpm Normal Conditions radiator T_in and T_out graph

4000 rpm Engine with Overheating

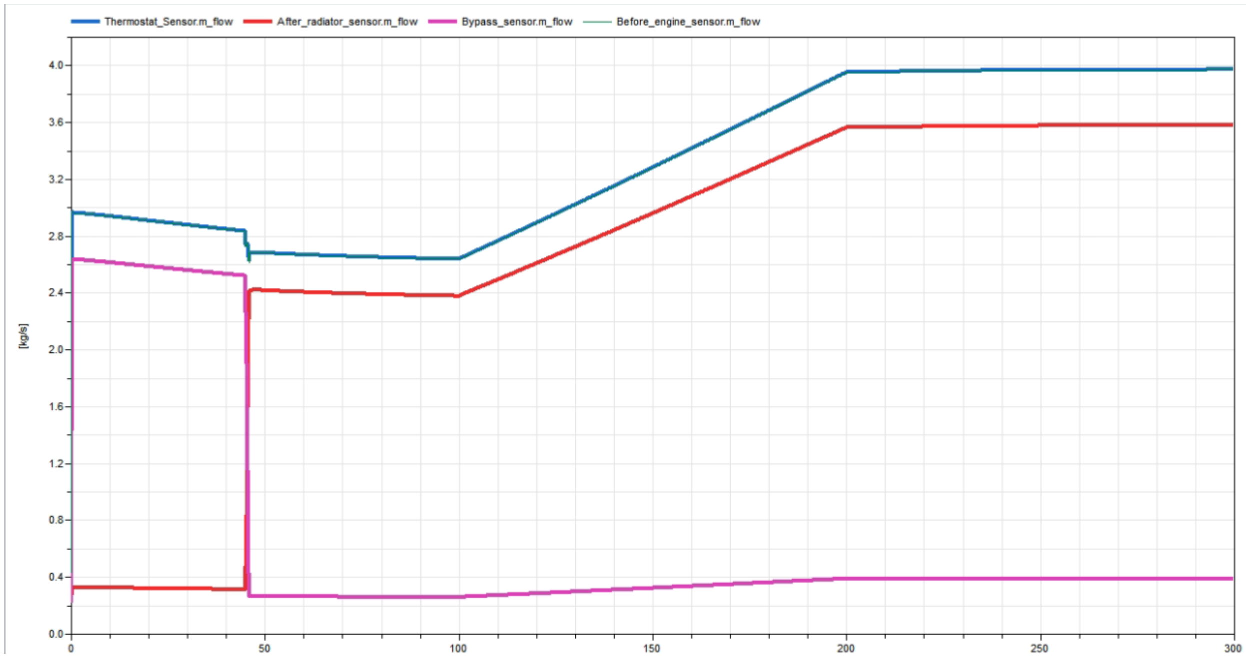


Figure 54: 4000 rpm Overheating m_flow graph

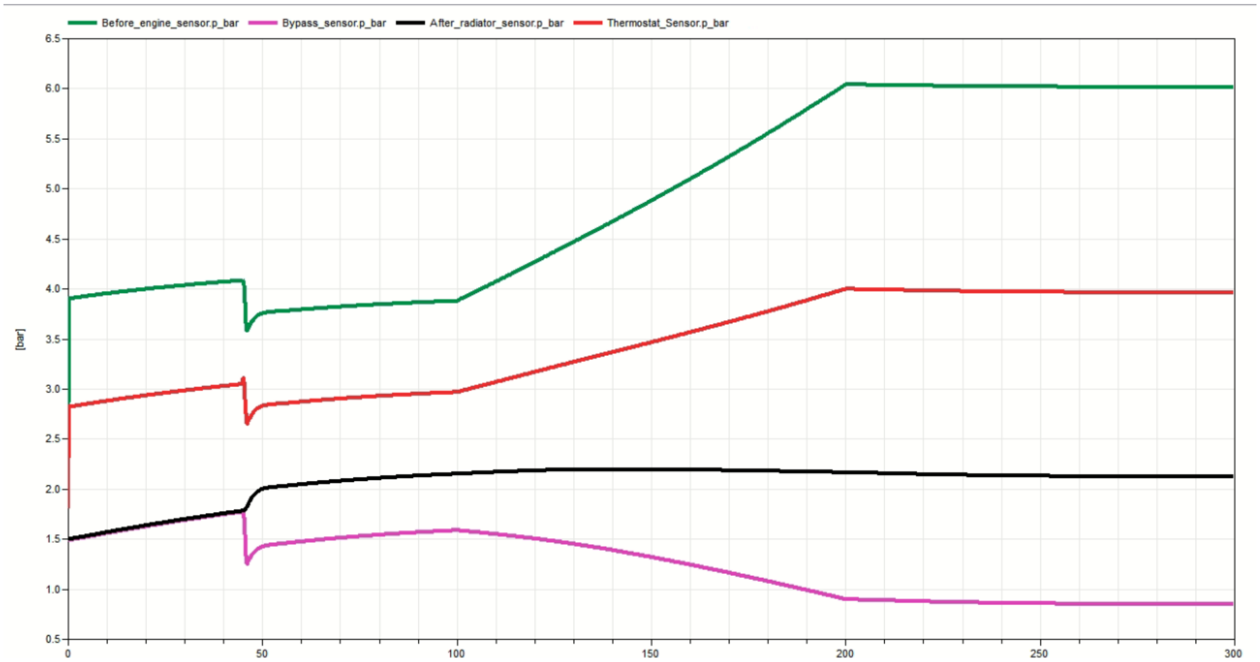


Figure 55: 4000 rpm Overheating pressure graph

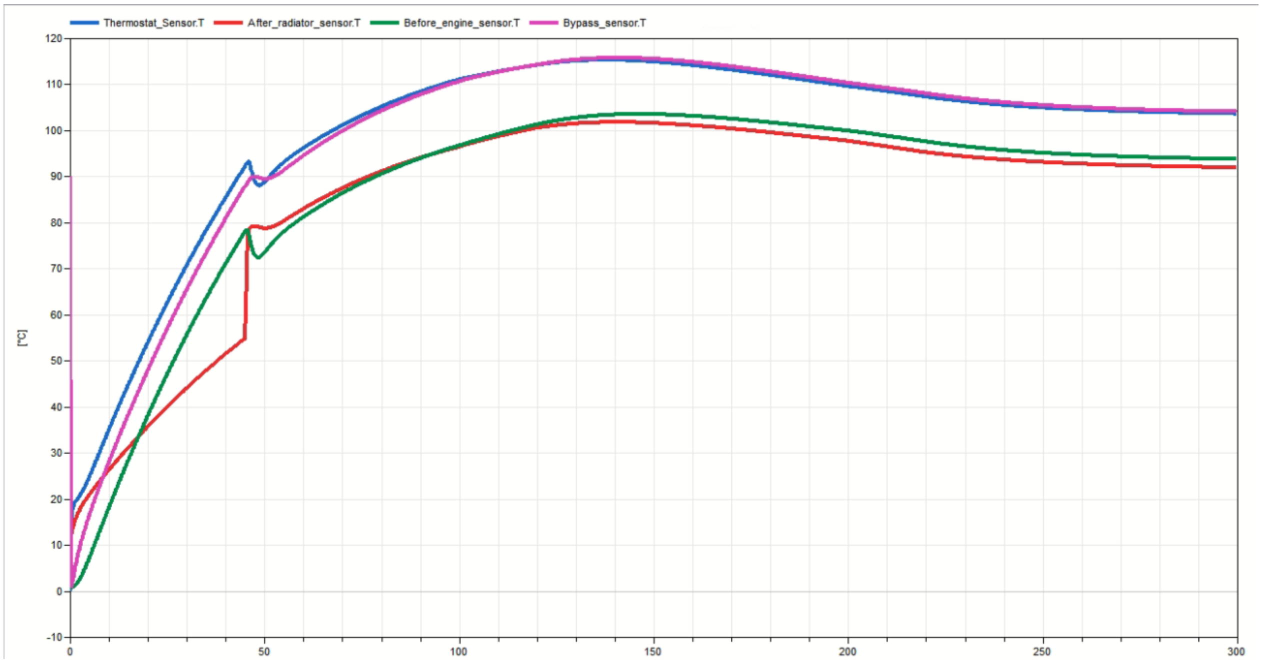


Figure 56: 4000 rpm Overheating temperature graph

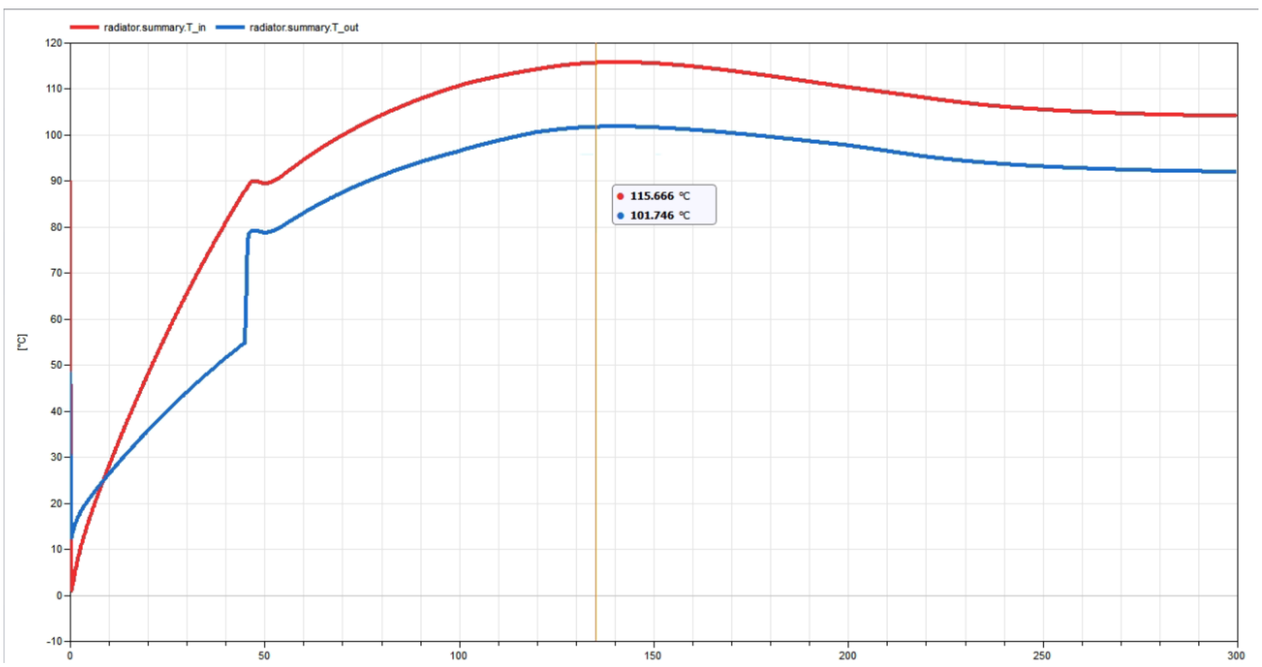


Figure 57: 4000 rpm Overheating radiator T_in and T_out graph

7. DISCUSSION

When we examine the simulations one by one, we see that the models operating under normal conditions give the desired results. The system starts to work at initial values. In order for the initially cold engine to warm up quickly, the thermostat remains closed and sends the coolant back to the engine block through the bypass channel. The coolant that reaches sufficient heat is directed to the radiator. We can examine the movement of the thermostat in the M flow graph.

While the m_{flow} passing through the bypass sensor is higher at the beginning, we see that the m_{flow} passing through the radiator sensor increases after a while. From here, we can understand that the thermostat is constantly open and directs the coolant to the radiator. Although they work in different parameters, we see that the graphics are similar. That is, the cooling system creates the cycle that will allow the engine to operate at the optimal temperature, even though there are different values when the correct parameters are given.

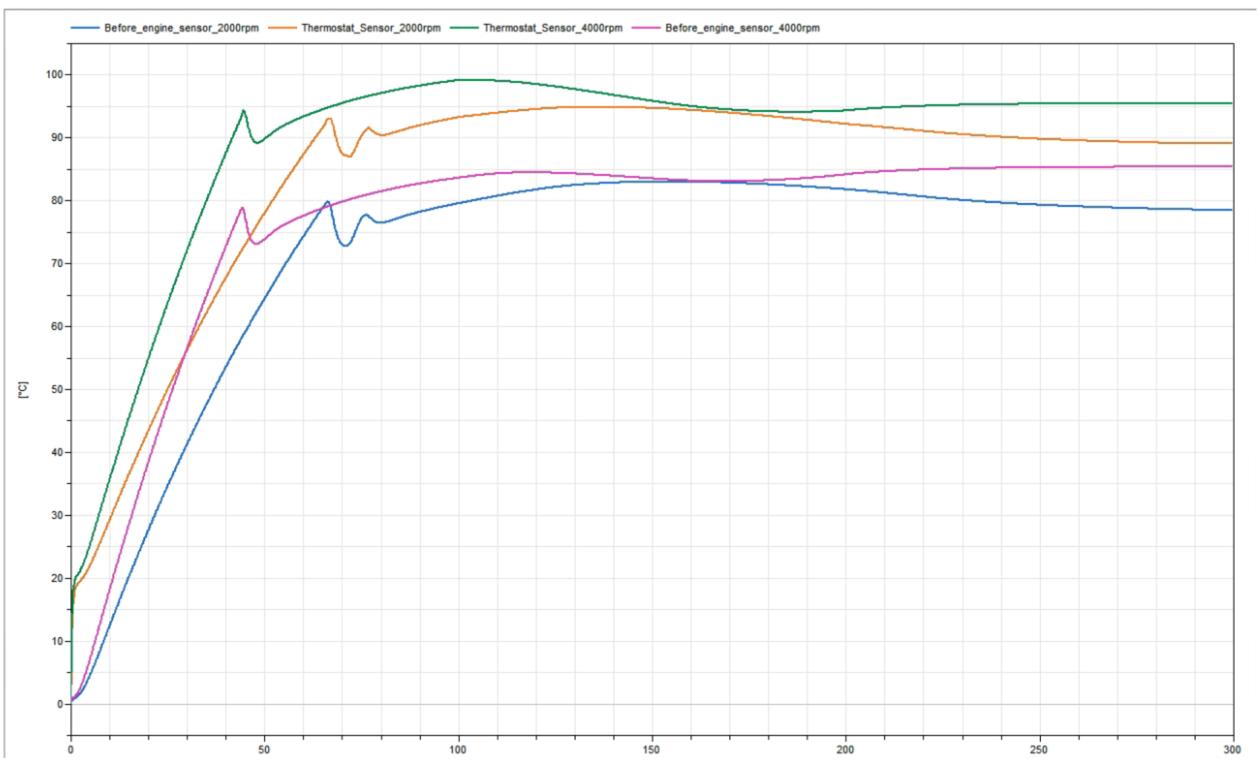


Figure 58: 2000 rpm Engine with Normal Conditions - 4000 rpm Engine with Normal Conditions

We wanted to examine how the system reacts in difficult situations after normal conditions work properly. For this, we created two fail scenarios to examine how the system responds in overcooling and overheating situations.

In the overcooling scenario, we connected the pump to constant speed and accelerated it even more. We also adjusted the fan and radiator to cool the coolant more. We increased the fan's operating speed and the radiator's heat transfer area. As a result of these, the system started to reduce the coolant to very low temperatures very quickly. The low temperature coolant was entering the engine block again and then going back to the radiator. The thermostat was constantly opened and closed between the bypass and main valves to maintain the optimal temperature. We see the movement of the thermostat in the m_{flow} graph. As a result of such a move, the thermostat loses its durability and shortens its life. We see that the air entering and leaving the radiator is constantly increasing and decreasing. As a result, using a very powerful fan increased the intake airflow, causing the coolant to cool more, and the powerful pump sent more coolant to the engine block, lowering the engine's optimal temperature.

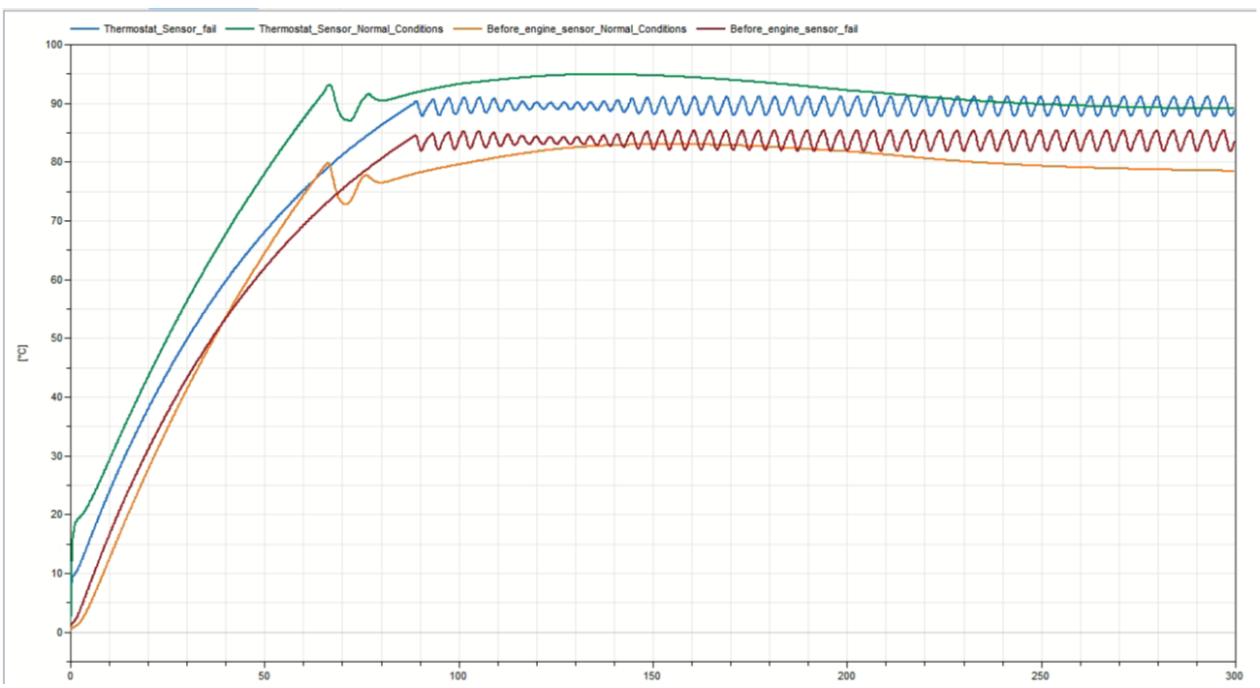


Figure 59: 2000 rpm Engine with Normal Conditions – 2000 rpm Engine with Overcooling

In the overheating scenario, we wanted to examine the inability of the fan, assuming that the system operates at high rpm and the engine overheats after a while. Despite the engine running at 4000 rpm, the fan performance decreased as time progressed and its causing the radiator to not cool the coolant enough. The coolant circulating in the system starts to damage the system parts. Cracks occur in the parts and their lifetime is shortened.

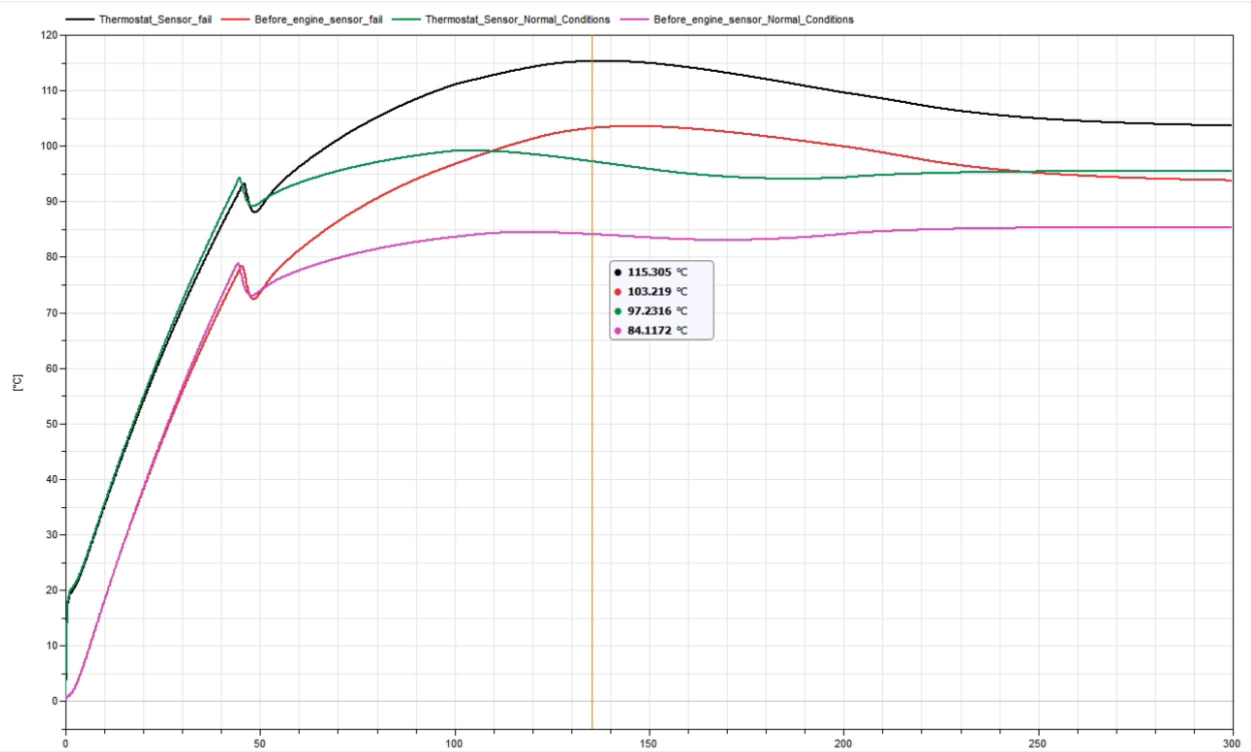


Figure 60: 4000 rpm Engine with Normal Conditions - 4000 rpm Engine with Overheating

8. CONCLUSION

In conclusion, the study demonstrates that a liquid engine cooling system can be effectively modeled in Dymola and can works perfectly when the optimal parameters are given in the model.

The experiments conducted under various conditions shows the importance of selecting the appropriate pump, radiator, and fan for achieving an optimally running system. Over or under engineering these components can cause problems on the system and damage the engine. Also as we can see from the Figure 46 thermostat valve is constantly working opposite to it’s intended design.

Another critical aspect is the selection of the coolant. Initially, the coolant used in the model was not good based on the simulation results. After a literature search, 40% Ethylene Glycol- water mixture found to be the most suitable coolant for the system. After choosing the right coolant, the desired values were achieved.

By carefully considering these factors and ensuring the cooling system is properly designed and calibrated, it is possible to maintain the engine's optimal temperature and prevent overheating or inadequate cooling.

As a result, in this study, we learned the modeling of the vehicle engine and cooling system in detail. We have seen the working principle of the system, the purpose for which the components in the system are used one by one, the parameters necessary for the optimal operation of the engine and how the changes in these parameters affect the functioning of the system.

9. REFERENCES

[1].Willard W. Pulkrabek, 'Engineering Fundamentals of the Internal Combustion Engine', First Edition, 1997.

[2]. Edward Ekstedt, 'Modeling and Simulation of the Cooling System in Heavy Duty Trucks', Department of Automatic Control, Lund University 2016.

[3]. Peter Jonsson, 'Physical modeling of a heavy-duty engine for test-cycle simulations in Modelica', Department of Automatic Control, Lund University 2018.

[4].Italo Longo, 'Advanced Engine Cooling Systems for Vehicle Application', 2015.

[5].Jaume Corvera Ripoll, 'Engine Cooling & Model Verification', Department of Automatic Control, Lund Institute of Technology May 2006.

[6].Sinan Kula, Emre Bulut, Esad Altay, Osman Sümer, Ferruh Öztürk, 'Smart cooling design using dual loop cooling to increase engine efficiency and decrease fuel emissions with artificial intelligence', 2022.

[7] https://en.wikipedia.org/wiki/Dymola

[8] Johan Dahl, Daniel Andersson, 'Gas Exchange and Exhaust Condition Modeling of a Diesel Engine using the Engine Dynamics Library', Gotherburg, Sweden, 2012.

[9] Liquid Cooling Library User's Guide. Version 1.5. Modelon. 2016.