UNIVERSITÀ DEGLI STUDI DI PADOVA
Dipartimento di Ingegneria Industriale
Corso di Laurea in Ingegneria Energetica

&

E.ON Energy Research Center | ERC
Institute for Energy Efficient Buildings and Indoor Climate | EBC

Tesi di laurea magistrale

Modeling of HVAC system for coupled plant and buildings simulation under Modelica

Caterina Toninato
Matriculation number: 1077614

Supervisor
Dott. Michele De Carli
Co-supervisors
Dipl.-Phys.Azem Abdul
Univ.-Prof. Dr.-Ing. Dirk Müller

Academic year 2015/16

Acknowledgments

Travel is the only thing you buy that makes you richer. This period in Aachen has been one of the best experience in my life. Working here on my thesis would not have been possible without the help, guidance and support of the people who contributed to realize it.
First of all I would like to thank Prof. Dirk Müller for giving me the opportunity to do my thesis with the E.on Energy Research Center.
Thanks to my supervisor Abdul Azem, my guidance for all the project period, for your patience and availability.
Thanks to my italian supervisor Prof. Michele De Carli, without which this experience would not have been possible.
Thanks to all the people of the E.on EBC and E.on ACS: I have learnt a lot in your Institute.
Thanks to all the people, who helped me, not only the one who helped me to find a solution to some work problems, computer problems or documents research problems, but also and especially to that friends that support me, advise me and make me feel like home.
Thanks to Sushruth, for the english part, the corrections, the tips, the time and for the suggestions you gave to me.
Thanks to Siyu, Edo and Gian who had the role of family for me during my thesis period and to all the people I had the pleasure to know in Aachen.
I cannot forget all the friends in Italy and all over the world since each one of them have helped me in my growth with their valuable advise.
Last but not least, thanks to my family, for your support in my whole life, for believing in me and for giving me the possibility to do my thesis at RWTH.

Abstract

The primary mission of the study of a Heating, Ventilation, Air Conditioning system (HVAC) is to develop a comprehensive understanding about how a sustainable energy supply can be achieved. According to the vision of this energy research center, this aim can be realized by focusing on reducing the energy consumption and using more environmentally friendly energy sources. This includes promoting energy-saving measures, as well as improving the efficiency of existing systems and increasingly utilizing renewable energy resources.

HVAC stands for "heating, ventilation, air conditioning", three functions often combined into one system in today's buildings. Through a series of ducts the system is able to distribute warm or cool air to the building interior room. HVAC is basically a system for controlling the environment (temperature, humidity, air flow, and air filtering). Its goal is to provide thermal comfort and acceptable indoor quality air. The traditional system is composed of different elements: humidifier, heat exchanger, filters, fans, air washer and sound absorber.

Air handling units are necessary to provide thermal comfort and indoor air quality. In order to increase the efficiency of existing and future air handling units and thus contribute to a sustainable energy supply, it is fundamental to simulate air handling units with dynamic models.

This Thesis deals with creating a Modelica based model library to compute and optimize complete air handling units in annual simulations. Modelica is a language for modeling a physical system and the equations implemented are used for simulating the dynamic behavior of complex and heterogeneous systems. The aim of the thesis is to extend the model library to a steam humidifier, evaporative humidifier, a plate crossflow heat exchanger and a rotary heat exchanger. All these models are fundamental in order to simulate the Air Handling Units system.

Nowadays, model's simulation is fundamental for reaching a complete knowledge of the system and a full developed technology. Simulation is a powerful tool for saving money and time. In other words, the aim of the thesis is to create simple but efficient models which allow a fast but accurate simulation of the system.

Contents

Glossary

Symbol and Unit

Symbol	Description	Unit
A	Area	m^2
c_p	specific heat capacity for constant pressure	$J/(kg \cdot K)$
C	heat capacity	W/kg
H	enthalpy	J
$\dot{H}$	enthalpy flow	J/s
E	exergy	J
e	specific exergy	J/kg
$\dot{m}$	mass flow	kg/s
p	pressure	Pa
$\dot{Q}$	heat flow	W
R	specific gas constant	$J/(kg \cdot K)$
S	entropy	J/K
$\dot{S}$	entropy flow	W/K
T	temperature	K
t	time	s
U_T	thermal transmittance	$W/(m^2 \cdot K)$
α	heat transfer coefficient	$W/(m^2 \cdot K)$
V	volume	m^3
$\dot{V}$	volumetric flow	m^3/s
$\dot{W}$	Power	W
x	water load per kg dry air	g/kg

Greek Symbols

Symbol	**Description**	**Unit**
η_C	Carnot efficiency	—
Φ	thermal power	W
ϱ	mass density	kg/m^3
σ	temperature spread	K
ϑ	temperature	°C
$\Delta\vartheta$	temperature difference	K

Indices and Abbreviations

Symbol	**Description**
0	ambient dead state
A	ambience
e	external
In	input
LabVIEW	programming language and development environment of National Instruments, Inc.
s	surface
Q	in reference to a heat flow
T	temperature
Δ t	time step of Δ t
HTF	heat transfer fluid

List of Figures

List of Tables

Structure of the Thesis

The thesis is structured as follows:

The first chapter, Basics and state of art, introduces the state of art of the technology and the software used for create the model of the humidifier and the heat exchanger.

The second chapter, Description of the Models, explains in detail every model that has been created. There will be a short description of the component and how it works, and then it will be explained more in depth what work was done with Modelica.

The third chapter, System Simulation and Results, contains the simulation of the every system. Every model will be simulated; then the results will be collected and analyzed. Simulating a system before actually building it is always a good way to save money; this is the main reason for the project actually.

The fourth chapter, Summary and Conclusions, deals with the final conclusion of the work.

1 Basics and State of Art

1.1 HVAC System

The HVAC System or Air Handling Unit is a system for reaching the required air comfort standards. The parameters modified by the unit are temperature, humidity, velocity and pureness. This system is used for heating, cooling, ventilating and conditioning. The Air Handling Units can be used in domestic environment (e.g. house-hotels-offices air conditioning), industrial spaces and public structures (hospitals, swimming-pools, sports centres). Air Handling Unit is a system composed by different elements:

- ▷ Filter
- ▷ Primary heat exchanger for heating
- ▷ Heat exchanger for cooling
- ▷ Secondary heat exchanger
- ▷ Fans
- ▷ Humidifier

Filters are essential for cleaning the air and removing the polluting within it. After filters heat exchangers are used. Depending on the season, the configuration and position of the heat exchangers is different. The primary heat exchanger is used only during winter season and its goal is to increase the temperature of the air. The humidifier is necessary to increase the value of humidity of the air that comes out from the primary heat exchanger. The cooling heat exchanger is used only during summer and its aim is to cool and dehumidify the air. Secondary heat exchanger is used for both seasons. The next component in the system is the fan. Fans are used to provide air volume flow. It is important to control the velocity of the air injected in a room in order to avoid problems of noise [1].

1.1.1 Working of HVAC System

In summer season the warm and humid air is picked in and passed through a filter. The filter purifies the air from the dust and small particles. The first heat exchanger is not turned on and the air overlaps it and reaches the cooling exchanger where the air is cooled and dehumidified. After that the air goes to the fan. In winter time the process differs. The cold and dry air is picked in and a filter purifies it from polluted particles. Then the air has to pass through the first heat exchanger where it is heated. After this process the air is too dry for being injected in the room. It has to be conditioning therefore a humidifier increases the value of humidity. Now the humidified air is not warm enough, therefore it is heated by the last heat exchanger. At this point the air is well conditioned for being injected in the room.

1.2 Humidifier

Humidity is water vapor contained in the air. The relative humidity is the percentage of water contained in air to a certain temperature (e.g. 50% R. U. at 20 °C) compare to the maximum quantity of water vapour can be contain at the same temperature (100% U.R. at 20 °C). Usually it is necessary to humidify the air during winter season when the outside air is cold and dry. So the air is heating till 20°C (comfort condition inside a building) but his value of relative humidity will be around 15% R.U. This value is out of comfort range and the air will be too “dry” for the people occupying the building. Also during cooling there will be a problem linked with the quantity of water vapour contained in the air. For this reason, nowadays, humidity is becoming an important parameter that has to be controlled and regulated. Depending on the situation, different kinds of humidifier can be used.

1.2.1 Steam Humidifier

Steam humidifiers can use different sources of energy for producing steam. Depending on the costs of production and the local availability of the sources, electricity or gas can be used for obtaining steam. Three main technologies exist: with electrodes, with electrical resistances or with gas [2]. The first option is the cheapest but due to the sedimentations of the minerals presented inside the water; electrodes have to be change few times per year. The second technology is the humidification with electrical resistances. A constant control of the level of water is necessary; otherwise the overheating of the resistances can damage the system. The last solution is the one using gas. Basically a boiler is used for heating the water and producing steam. This option allows to work in case of heavy duty, when the quantity of air to be humidify is high. Usually for applications that do not require a big amount of steam, the first two solutions are preferred.

1.2.2 Evaporative Humidifier

The evaporative humidifier use forced-air heat to provide pure vapour humidity. Water is used instead of steam; this is a benefit in terms of energy efficiency. Hence evaporative humidifier consumes less electricity than the steam humidifier. This technology can be used also in summer season for cooling the air [2]

1.3 Heat Exchanger

The operating principle of air-to-air heat exchangers (or air-to-air heat recovery systems) consists of exchange temperature between the exhaust and supply airflows inside the Air Handling Units. This way the thermal energy contained in the exhaust air is recovered for the supply air and thus the load of energy required is reduced [3] There are several types of air-to-air heat exchangers with mean recovery efficiencies from 50% up to 90%, depending on the type of heat exchanger and the outdoor and indoor air conditions (temperatures and humidity). From all types of air-to-air heat exchangers the most used in Air Handling Units applications are: Plate air-to-air heat exchanger and Rotary heat exchanger.

1.3.1 Plate Heat Exchanger

There are two types of plate heat exchangers: Counter Flow Plate Heat Exchangers and Cross Flow Plate Heat Exchangers. As the name itself suggests the design of the cross flow plate heat exchanger forces the exhaust airflow and the supply airflow to crosses between them. If comparing with the counter flow heat exchanger, where the exhaust airflow and the supply airflow have an opposite path (counter), the cross flow heat exchanger for the same conditions have lower efficiency. Opposite airflows can maximize the heat energy exchange. The lower efficiency, of the cross flow heat exchanger, is compensate with the possibility to work with a wider range of airflows (due to his structural design can be larger in size). In the case of counter flow heat exchangers with the right conditions, indoor and outdoor, manufacturers claim 85%-90% efficiency. This value can reach to around 70% for the cross flow heat exchangers [3]. For industrial applications the pressure loss will play an important role. The standard material used to produce the plates is aluminium, but they can also be manufactured in plastic and aluminium with epoxy coating for special application like processes with high temperatures corrosive atmospheres and swimming pools. The model that has been studied in this Thesis is the cross-flow type.

1.3.2 Rotary Heat Exchanger

The rotary heat exchanger is one of the most efficient air-to-air heat exchangers with the ability to handle large airflows. The core of this heat exchanger is the rotor, or wheel, made out of aluminium foils. Half of the rotor is in contact with the exhaust airflow while the other half contacts the supply airflow. The wheel is driven by means of a motor and a belt, rotating the wheel and transport the heat recovered in the half wheel from the exhaust airflow to the supply airflow. The efficiency claimed by manufacturers for a rotary heat exchanger can reach up to 90 %. The size of the rotors can go from small diameters size (domestic applications, 0.5 m) to huge diameters for industrial applications (5 meters).

There are several types of rotors, for different applications, that are able to recover temperature and/or humidity. Depending the coating applied to the wheel they can be used also to drying and humidification. Wheels used to dry the air are also called as desiccant wheels. Usually this kind of heat exchanger is not used in applications that demands zero contamination between the exhaust airflow and the supply airflow [3]. The type of rotary heat exchanger studied in this Thesis project is able to recover temperature, but not humidity. In other words it is a “dry model”.

1.4 Software and Tools used

The program, used for developing the models present in the Thesis, is Dymola. Dymola is a modelling and simulation environmental based on the open Modelica modeling language.

1.4.1 Modelica

Modelica is a language for modeling a physical system (mechanical, electrical, hydraulic and control system) [4]. Equations are used for modeling the dynamic behavior of complex and heterogeneous systems. From a user’s point of view, models are described by schematics, also called object diagrams, few examples of the connector used are shown in the figure below [Fig 1.1]. A schematic consists of connected components, like a resistor, a pump or a hydraulic cylinder. Every component is provided with “connectors” (often also called “ports”) that describe the interaction possibilities.

Every type of connector is compatible only with itself, because it is defined by parameters belonging only to this type of field. By drawing connection lines between connectors a physical system or block diagram model is constructed. Internally a component is defined by another schematic level which is also defined by an equation based description of the model in Modelica syntax [5]. Basically, all Modelica language elements are mapped to differential, algebraic and discrete equations. There are no language elements to describe directly partial differential equations, although some types of discretized partial differential equations can be reasonably defined. Modelica is also provided with a powerful ´Library‘ where different components belonging to different fields are available.

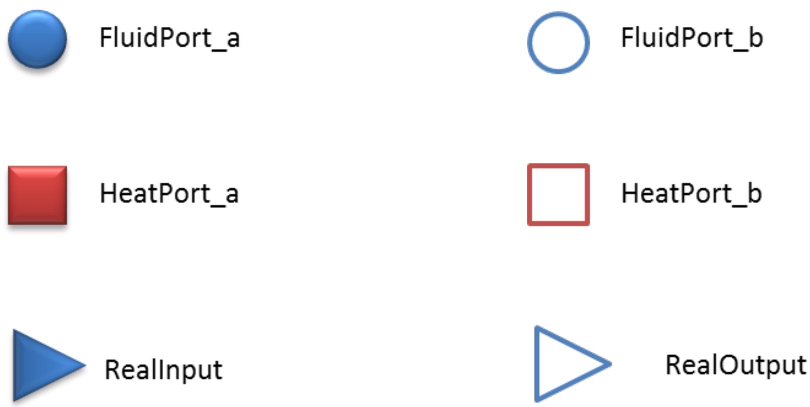


Figure 1.1: Examples of some connectors

1.4.2 Dymola

Dymola is a commercial simulation environment based on the open Modelica modeling language. Hybrid models are developed thanks to this powerful tool, and a complete data evaluation is available. Dymola is language quite intuitive, the Libraries and the Graphic Editor make easier modeling objects [4]. The libraries are provided with components equal to physical components, and the User just need to drag it in the modeling window and start to create his model. A big advantage Dymola has is the symbolic manipulation which avoids the user from converting equations in blocks. Dymola has two kinds of views: main window and library window [Fig 1.2] .

The Main window also operates in two different modes: Modeling view and Simulation view [6]. The modeling mode of the main window is used to compose models and model components. It is the big space on the right side [Fig 1.3]. The Simulation mode, as the same word says, is used to simulate the model, plot results and animate the behavior. The Simulation mode also has a scripting sub-window for automation of experimentation and performing calculations [Fig 1.4].

A simple example that explains how it is possible to create a model is shown in [Fig 1.5]. Firstly it is necessary to define a connector that will be the input/output of the model. Secondly it is necessary to define the parameters of the model and his internal variables. Then equations (algebraic or differential), that describe how the model should work, are written after the word “equation” [6] [7] .

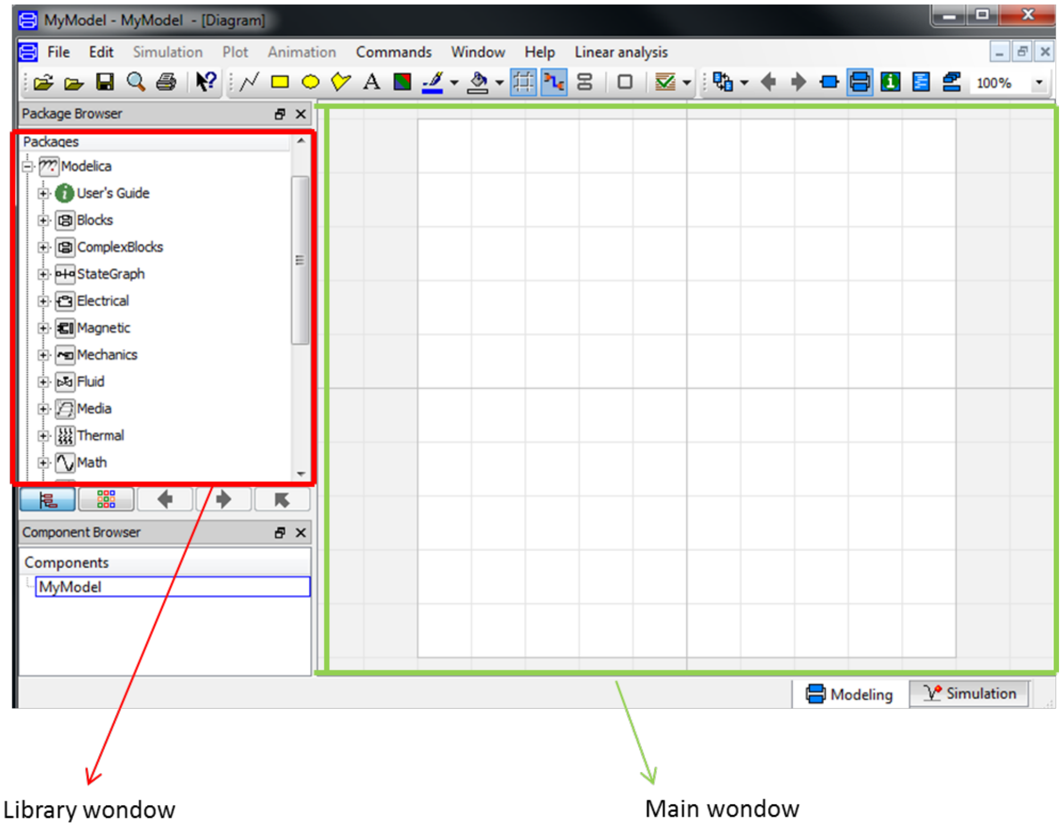
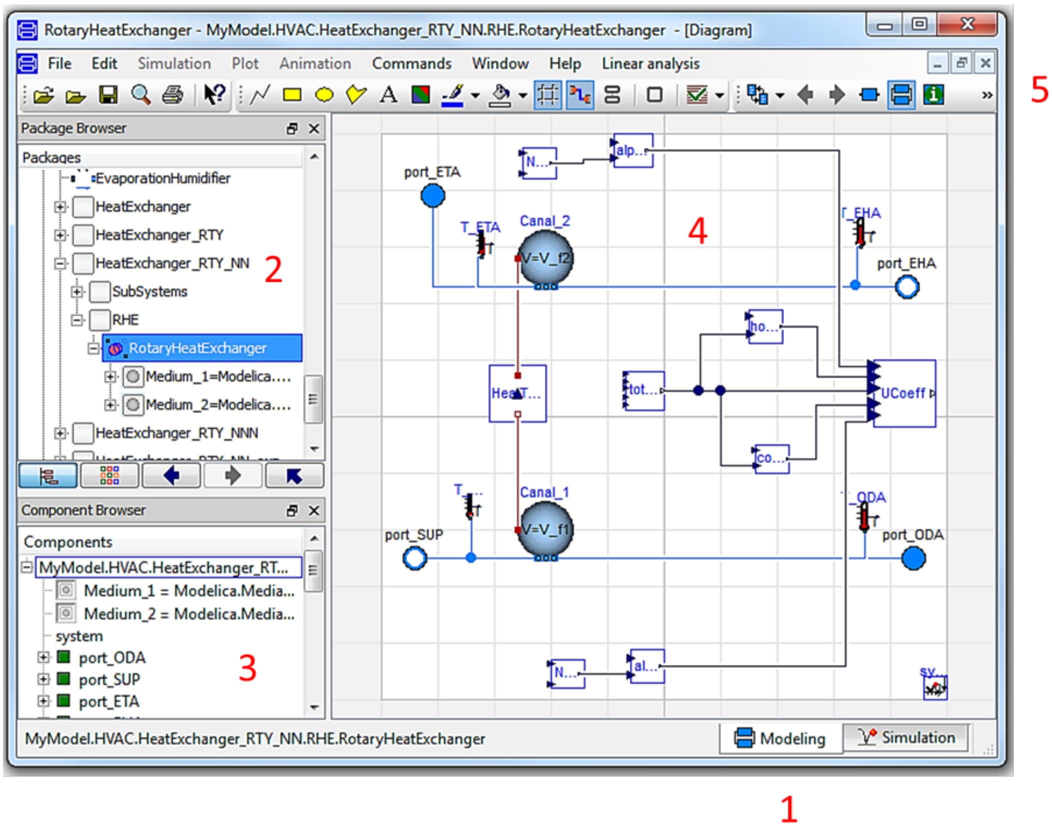
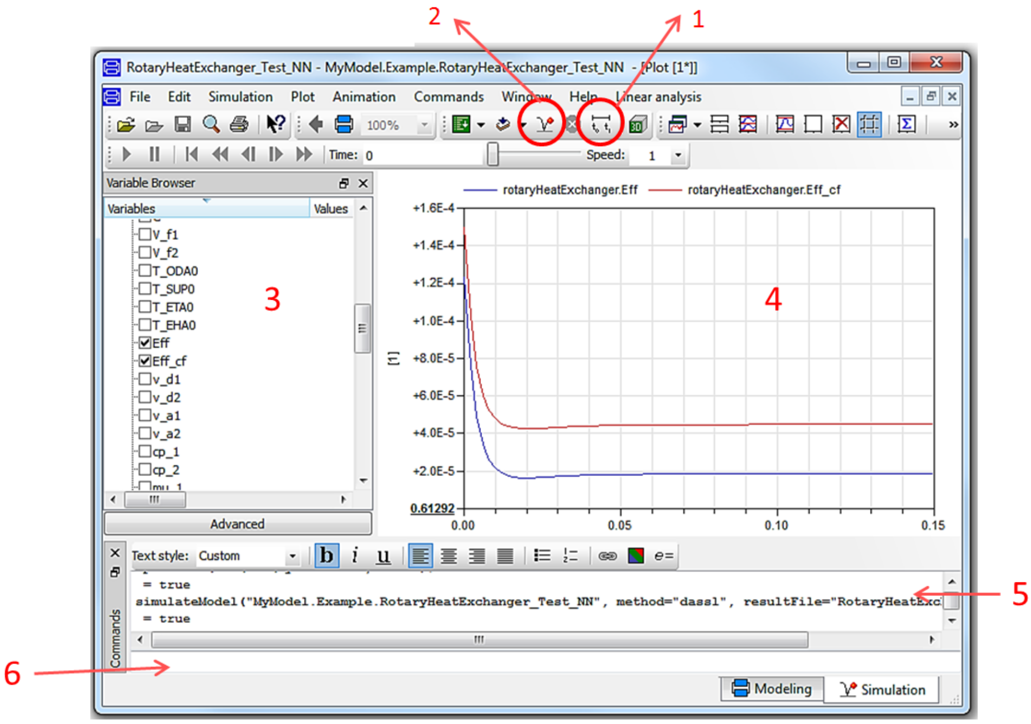


Figure 1.2: Modellica window



1. Modeling
2. Package view
3. Components
4. Model representation
5. Tool bar

Figure 1.3: Example modelling view



1. Simulation step
2. Run simulation
3. Variable Browser
4. Plot-Window
5. Log-Window
6. Command line

Figure 1.4: Example simulation view

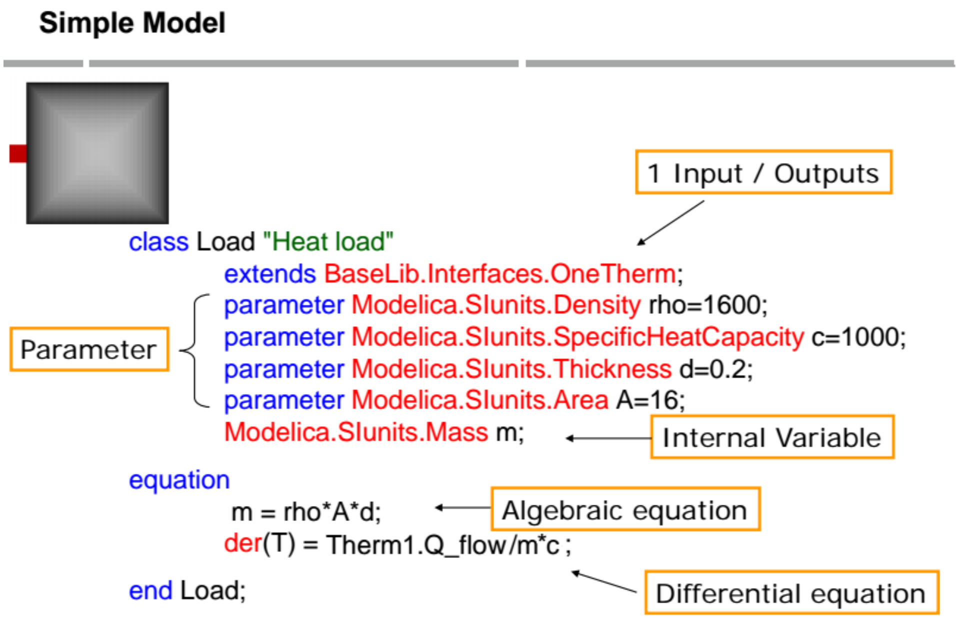


Figure 1.5: Example model code

2 Description of construction of Simulation Models

2.1 Steam Humidifier

Humidifying air with steam means that water is added to air as steam that has been produced beforehand. The air flow and the steam mixed at the atmospheric pressure (the temperature is around 110-120°C; it is important to remember that in this range of temperature the value of the enthalpy is 2675.6 - 2691 kJ/kg). It was considered a case of adiabatic mixing. The creation of the model starts with the declaration of the input/output of the system and with the all parameters involved in the transformation. The model can be schematically represented like show in [Fig 2.1].

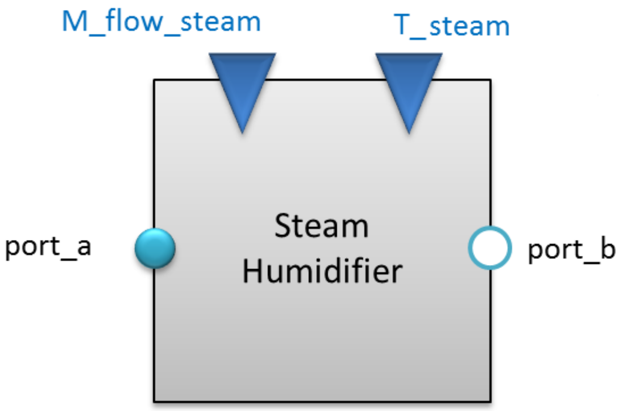


Figure 2.1: Scheme steam humidifier

Where *M flow steam* and *T steam* are the Input of the model, *port a* is also an input and *port b* is the output.

The quantity of the air and steam mass flow rate, enthalpy and mass fraction are included in the definition of *Fluid port* [8].The law mass and energy balance on either fluids permit to formulate the equations below:
Air mass balance equation is:

$$\dot{m}_1 = \dot{m}_2 \tag{2.1}$$

And the water mass balance equation for the open system show in the [Fig 2.2] gives:

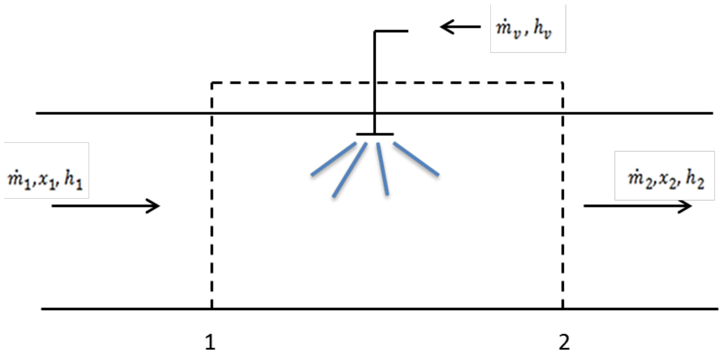


Figure 2.2: Simple scheme of the humidification transformation

$$\dot{m}_1 x_1 = \dot{m}_2 x_2 - \dot{m}_v \tag{2.2}$$

It means that the quantity of steam is

$$\dot{m}_v = \dot{m}_1 (x_2 - x_1) \tag{2.3}$$

The Energy balance equation is given by:

$$\dot{m}_1 h_1 = \dot{m}_2 h_2 - \dot{m}_v h_v \tag{2.4}$$

Where $\dot{m}_1$,$\dot{m}_2$ are the inlet and outlet air mass flow rates respectively, $\dot{m}_v$ is the steam mass flow rate, x_1,x_2 represent the absolute humidity, while h_1,h_2 are the enthalpies.
From all the equations above it is possible to demonstrate that the slope of the transformation in the ASHRAE diagram is equal to the value of steam enthalpy:

$$M = \frac{h_2 - h_1}{x_2 - x_1} = h_v = 2675.6 \frac{kJ}{kg_v} \tag{2.5}$$

The temperature value of the steam is about 110-120 °C (it means $h_v = 2,6 - 2,7 \frac{kJ}{kg_v}$) the transformation is almost a horizontal line [Fig 2.3]. Refers again to the h-x diagram, it important to clarify that the slope of the adiabatic humidification with steam is a little bit higher than the straight line at T=0°C (where $h_v = 2500 \frac{kJ}{kg_v}$). This seems quite reasonable because after the injection of the steam the air flow is not only humidified but also heated [5]. Anyways this temperature difference is quite small (1-3 °C) and it can be neglected, so it is possible to consider the transformation as isothermal [9].

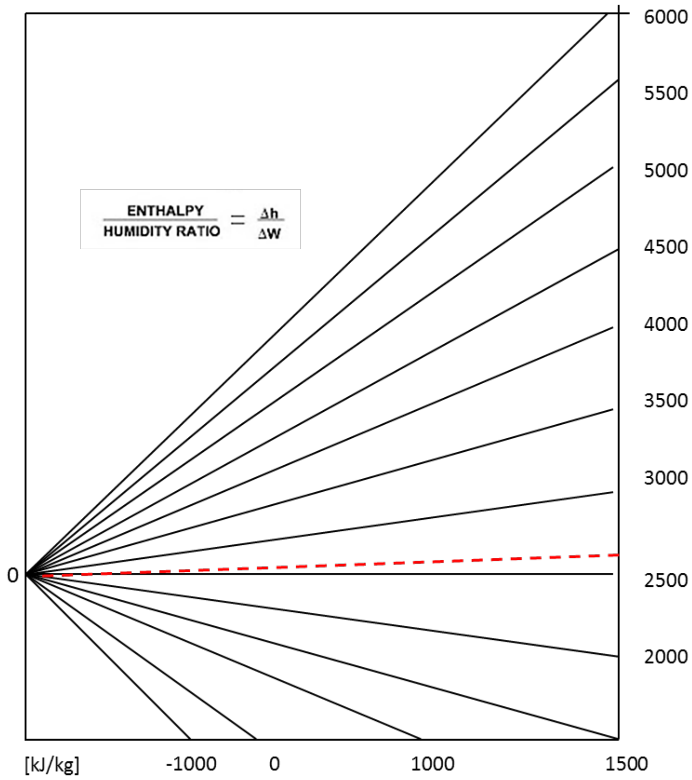


Figure 2.3: Enthalpy - Humidity ratio

The heat flow rate exchanged by the injected steam, and the one responsible for the small increasing of air temperature, is given by the following equation:

$$Q_{flow} = \dot{m}_v h_v \tag{2.6}$$

2.2 Evaporative Humidifier

Evaporative occurs if water has not yet reached the boiling temperature at the current pressure while the partial pressure of water vapor at the phase interface, i.e., the water surface, is greater than the partial pressure of water vapor in the ambient air. Humidification performance can be calculated according to the equations proposed in the steam humidifier paragraph. Due to simplification only h_v and $\dot{m}_v$ are replaced by h_w and $\dot{m}_w$.
A general logic diagram about the model is shown in [Fig 2.4].

According to the mass and energy balance laws, it is possible to formulate the following equations:
Air mass balance

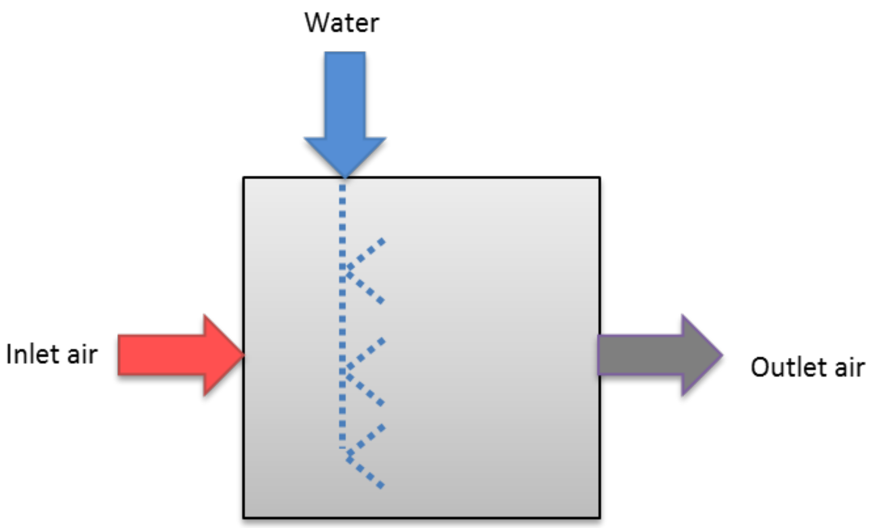


Figure 2.4: Scheme evaporative humidifier

$$\dot{m}_1 = \dot{m}_2 \tag{2.7}$$

Water mass balance

$$\dot{m}_1 x_1 = \dot{m}_2 x_2 - \dot{m}_w \tag{2.8}$$

Energy balance

$$\dot{m}_1 h_1 = \dot{m}_2 h_2 - \dot{m}_w h_w \tag{2.9}$$

Where $\dot{m}_1$,$\dot{m}_2$ are the inlet and outlet air mass flow rates respectively, $\dot{m}_w$ is the water mass flow rate,x_1,x_2 are the specific humidity, while h_1, h_2,h_w are the enthalpy.
From all the equations above the value of steam enthalpy is:

$$M = \frac{h_2 - h_1}{x_2 - x_1} = h_w = 50.4 \frac{kJ}{kg_v} \tag{2.10}$$

In contrast to steam injection, humidification with water results in a considerable cooling of the air stream as the air has to provide the vaporization heat of water. This means that the transformation is not an isothermal transformation but it is isenthalpic [Fig 2.6].

Inside the evaporative humidifier, the humidification water is atomized by nozzles while the air to be humidified is moved in or against the flow direction. If humidification is the primary task, the air washer is operated adiabatically. The humidification performance of an adiabatic air washer can be characterized with the humidifying efficiency:

$$\eta = \frac{x_2 - x_1}{x_k - x_1} \tag{2.11}$$

This describes the ratio of the real to the maximal achievable change in air humidity.
In the present equation x_1,x_2 are the specific humidity of the inlet and outlet air mass flow rate respectively, while x_k is the achievable maximum humidity and it can be obtained by extending the fog isotherm (line of constant wet bulb temperature) running through the entry state of the air [Fig 2.5] [5] .

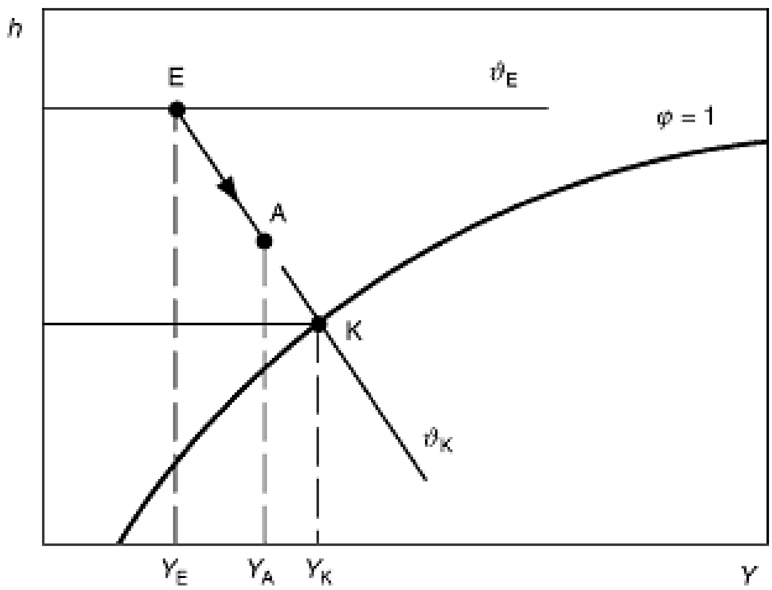


Figure 2.5: Letter E stands for the conditions of the inlet air, A for the conditions of the outlet air and K is for the condition of saturation; Y stands for specific humidity and ϑ stands for temperature.

The humidifying efficiency is influenced by many parameters. The most important are the droplet-size distribution of the spray and the resulting overall droplet surface. A number used to quantify these effects is K_y. The quantity K_y contains the droplet surface area relating to the water flow rate and can be quantify by the following equation:

$$K_y = -(\frac{\dot{m}_w}{\dot{m}_1})\log(1-\eta) \tag{2.12}$$

Where $\dot{m}_w$ is the water mass flow rate injected,$\dot{m}_1$ is the air mass flow rate and is the efficiency of the humidifier.

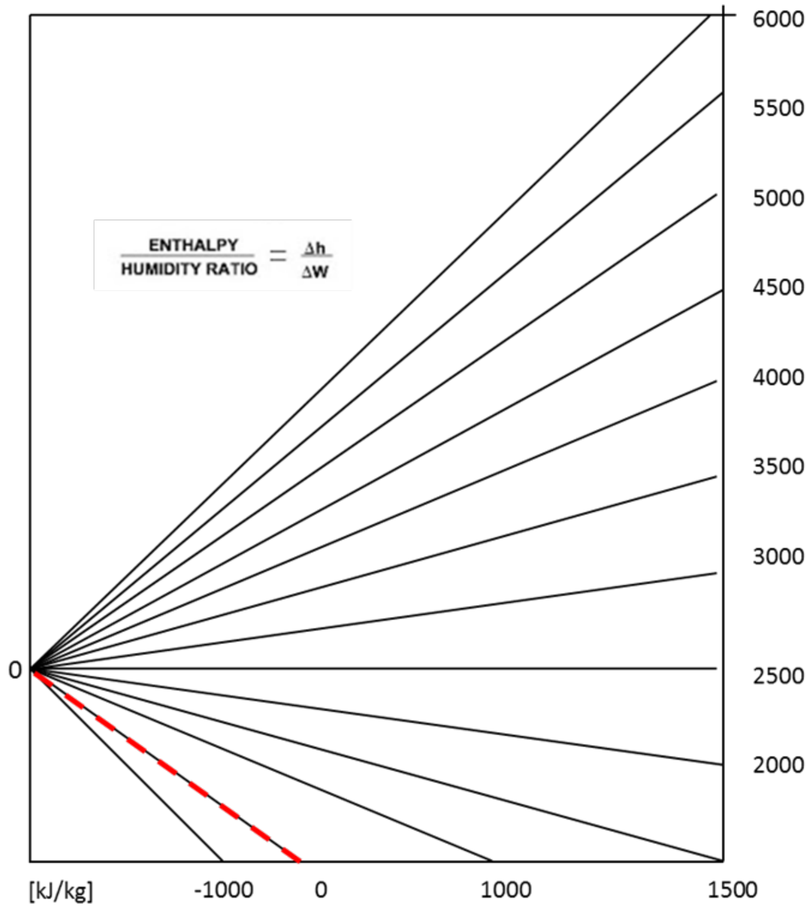


Figure 2.6: Enthalpy/Humidity ratio.

2.3 Cross-flow Air-to-Air Plate Heat Exchanger

Plate Heat Exchangers are available in many configurations, materials, sizes and air flow patterns. They are manufactured in modular form, arranged to handle any airflow, effectiveness and pressure drop requirement. They are simple devices with no moving parts. The casing of the unit is compartmented to form narrow passages carrying, alternatively, supply and extract air streams. The construction of the plates permits a large surface area to be packed into a compact space. Heat is transferred by conduction and convection through the separating plates. The plates are normally constructed from aluminum or stainless steel [1].

The goal of heat exchanger design is to relate the inlet and outlet temperatures, the overall heat transfer coefficient, and the geometry of the heat exchanger, to the rate of heat transfer between the two fluids.

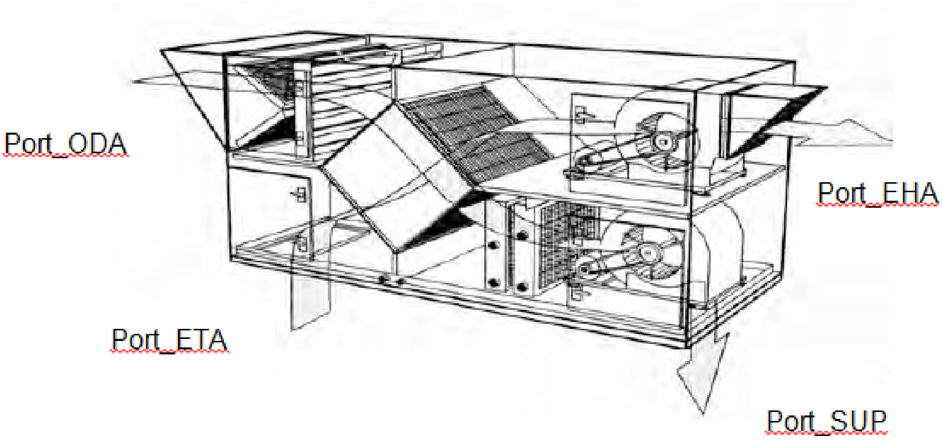


Figure 2.7: Scheme of the Cross-flow Air-to-Air Plate Heat Exchanger [10] .

Different Subsystems are created and connected together in a simple but complete code for realizing the total heat exchanger model.
The real plate heat exchanger model, as shown in [Fig 2.7, has four port accesses corresponding to the four real air accesses. A general and schematic idea of the model created in Modelica Modeling window it is represented in Fig.4.9:

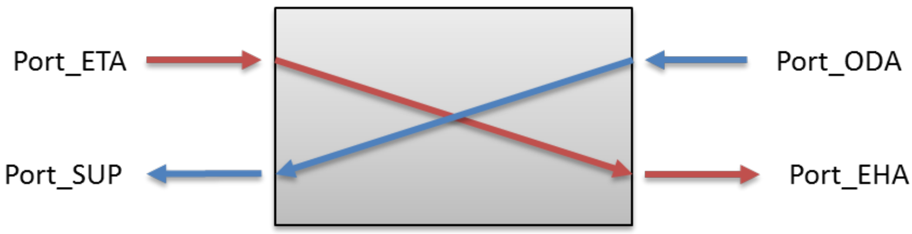


Figure 2.8: Scheme Cross-flow Air-to-Air plate heat exchanger .

where:

- ▹ $Port_{ODA}$ is for the outdoor air,
- ▹ $Port_{SUP}$ is for the supply air,
- ▹ $Port_{ETA}$is for the extract air,
- ▹ $Port_{EHA}$ is for the exhaust air.

As explain before, the plate heat exchanger model is composed by different Subsystems, each of them is fundamental for the good working of the total System. The first important Subsystem component that will be explain is the component called "HeatTrans" [Fig 2.9]. This component is re-

sponsible for the transfer of the heat exchanging between the two fluids.
The enthalpy law balance on either fluids permits to formulate the following equations:

$$\dot{Q}_c = \dot{m}_c(h_{c2} - h_{c1}) \tag{2.13}$$

for the cold air stream flow, and

$$\dot{Q}_h = \dot{m}_h(h_{h1} - h_{h2}) \tag{2.14}$$

for the hot air stream flow.
For constant specific heats with no change of phase, it is also allowed to write:

$$\dot{Q}_c = (\dot{m}c_p)_c(T_{c2} - T_{c1}) \tag{2.15}$$

And

$$\dot{Q}_h = (\dot{m}c_p)_h(T_{h1} - T_{h2}) \tag{2.16}$$

Now from the law of energy conservation it is known that $\dot{Q}_c = \dot{Q}_h = Q\dot{_{flow}}$ and that it is possible to formulate a relation between the heat transfer Q and the overall heat transfer coefficient U, to the mean difference temperature $_m$ by means of:

$$Q\dot{_{flow}} = UA\Delta T_m \tag{2.17}$$

where A is the total surface area for the heat exchange that U is based upon. The exchange of this $Q\dot{_{flow}}$ is permitted through the HeatTrans Subsystem with two heat port connectors ($port_1$ and $port_2$) [Fig 2.9].

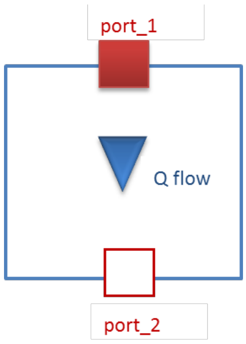


Figure 2.9: Scheme of heat transfer submodel.

As explain in the equation before, the $Q\dot{_{flow}}$ is a function of the total surface area A, the overall heat transfer coefficient U and the mean difference temperature ΔT_m.

The total surface area is calculating inside the model considering the geometry of the heat exchanger. A table shows and explains the geometrical parameters defined and used for obtain the total surface area A.

Table 2.1: Geometry parameters heat exchanger

Geometry Parameters

Type	Name	Description
Thickness	s	Plate thickness [m]
Thickness	p_s	Plate spacing [m]
Thickness	d_h	Hydraulic diameter [m]
Lenght	A	Length of the heat exchanger [m]
Lenght	B	Height of the heat exchanger [m]
Lenght	C	Depth of the heat exchanger [m]
Lenght	D	Plate length [m]
Lenght	e	Plate width [m]
Lenght	L_{sheet}	Sheet length [m]
Lenght	F_p	Sheet height [m]
Lenght	L_{flow}	Mean flow length [m]
Real	f	Extension factor of the plate surface 0<= f <=1

So, the total surface area is:

$$A_{Hxsurface} = A_{pl} n_{tot} \tag{2.18}$$

Where A_{pl} is the plate surface and n_{tot} is the total plate number.

$$A_{pl} = [(A-e)F_p + eB + \frac{(A-e)(B-F_p)}{2}](1+f) \tag{2.19}$$

$$n_{tot} = \frac{C}{p_s} \tag{2.20}$$

According to the equation (2.17) the mean temperature difference ΔT_m has to be define. This temperature depends on the flow arrangement of the heat exchanger and on the degree and the direction of mixing within the two fluid streams.

$$\Delta T_m = \Theta(T_{h1} - T_{c1}) \tag{2.21}$$

where Θ is the dimensionless mean temperature difference and it is defined as:

$$\Theta = \frac{P}{NTU} \tag{2.22}$$

Where P is the dimensionless temperature change and NTU is the number of transfer unit. P is also the thermal efficiency of the heat exchanger and is a function of NTU and R, where R is the ratio of the thermal power capacity.

$$R = \frac{C_{min}}{C_{max}} \tag{2.23}$$

And

$$NTU = \frac{AU}{C_{min}} \tag{2.24}$$

For $R \neq 1$

$$P = \frac{1 - exp[F(R-1)NTU]}{1 - Rexp[F(R-1)NTU]} \tag{2.25}$$

And for R=1

$$P = \frac{NTU * F}{1 + NTU * F} \tag{2.26}$$

Where F is the correction factor to the mean temperature difference and is given by the following equation:

$$F = \frac{1}{(1 + aR^{db}NTU^{b})^{c}} \tag{2.27}$$

The coefficient *a,b,c,d* where determined through least square fits for numerous flow arrangements, based on the data calculated for the design charts. The value of *a,b,c,d* are given in table 1 [11].
The last parameter that has to be explaining is the overall heat transfer coefficient. It is the most complex one, because it is depending on different variables. Actually the overall heat transfer coefficient can be defined in terms of individual thermal resistances of the system and a Subsystem is creating for obtaining as output the right value of U. Combining all the individual thermal resistances in series gives:

$$U = \frac{1}{\frac{1}{\alpha_1} + \frac{s}{\lambda_p} + \frac{1}{\alpha_2}} \tag{2.28}$$

where α is the heat transfer coefficient, λ_p is the thermal conductivity of the plate and s is the plate thickness. This formula is implemented in a Subsystem called "UCoeff", where the inputs are the two heat transfer coefficients α_1 and α_2, and the output is the overall heat transfer coefficient U.

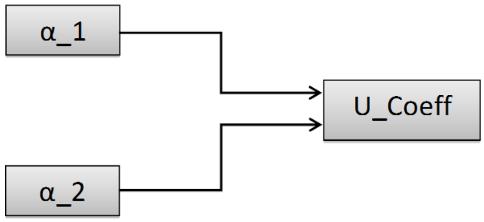


Figure 2.10: Scheme correlation alpha-value with the overall heat transfer coefficient.

Two second Subsystems, one for each fluid and called $Alpha_m$, are creating for obtaining the value of the heat transfer coefficient for both fluids, and the implemented equation is:

$$\alpha_m = \frac{Nu_m \lambda_m}{l} \tag{2.29}$$

Where λ_m is the thermal conductivity of the m fluid, and Nu_m is the corresponding Nusselt Number. The inputs give to the Subsystem are the thermal conductivity and Nusselt number.

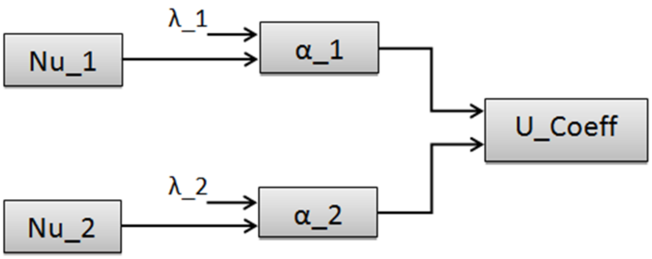


Figure 2.11: Correlation Nusselt number and Heat transfer coefficient.

As Nusselt number is a function of Reynolds number, Prandtl number and the geometry of the plate heat exchanger, another Subsystem is created. It is called $Nusselt_m$ and his function is to calculate the value of the Nusselt number according to the following equations:

$$Nu_m = 0.023 Re^{0.8} Pr^{0.33} \tag{2.30}$$

$$Nu_m = 7.55 + \frac{0.024 \left[Pr Re \left(\frac{d_h}{l} \right) \right]}{1 + 0.00358 Pr^{0.81} \left[Re \left(\frac{d_h}{l} \right) \right]^{0.64}} \tag{2.31}$$

The equation (2.30) is for laminar flow and the equation (2.31) is refered to a turbolent flow.

Where *Re* is Reynolds number, *Pr* is Prandtl number, d_h is the hydraulic diameter and l is the flow length. Re, Pr ,d_h, l are all inputs give to $Nusselt_m$.
As conclusion of this chain of operations, the equations for Reynolds number and Prandtl number are implemented inside the model with the following equations:

$$Re = \frac{\rho u d_h}{\mu} \tag{2.32}$$

$$Pr = \frac{c_p \mu}{\lambda} \tag{2.33}$$

Where ρ is the density, u is the velocity of the fluid, μ is the viscosity. The final representation of a part of the model for finding the overall heat transfer coefficient is shown in [Fig 2.12].

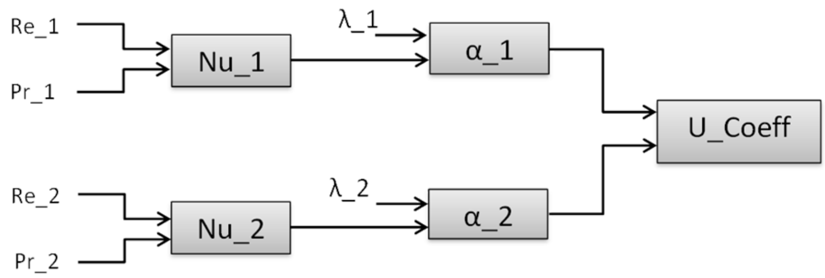


Figure 2.12: Reynold number correlation with the U-value.

The last part of the code is about the pressure drop. Fluids need to be pumped through the heat exchanger. The fluid pumping power is proportional to the fluid pressure drop, which is associated with fluid pressure and others pressure drop contributions along the fluid flow path. For this reason it is important to evaluate the value of the pressure drop.
Pressure drop in heat exchangers is an important consideration during the design stage. Pressure drop is affected by a number of factors, namely the type of flow (laminar or turbulent) and the passage geometry. First, a fluid experiences an entrance loss as it enters the heat exchanger core due to a sudden reduction in flow area, then the core itself contributes a loss due to friction and other internal losses, and finally as the fluid exits the core it experiences a loss due to a sudden expansion. All these effects are discussed below. Before some major assumptions for pressure drop analysis are made [12] :

- ▹ Flow is steady and isothermal,

- ▷ Fluid density is dependent on the local temperature only,
- ▷ The pressure in the fluid is independent of direction,
- ▷ Body forces are causes only by gravity,
- ▷ There are no energy sinks or sources along the a streamline,
- ▷ The friction factor is considered as constant with the passage flow length.

The total pressure drop on one side of the heat exchanger is:

$$\Delta p = \Delta p_{1-2} + \Delta p_{2-3} + \Delta p_{3-4} \tag{2.34}$$

Here the subscripts 1, 2, 3, and 4 stands for upstream, passage entrance, passage exit, and for downstream, respectively, as shown in [Fig 2.13]. The Δp_{1-2} is the pressure drop at the core entrance due

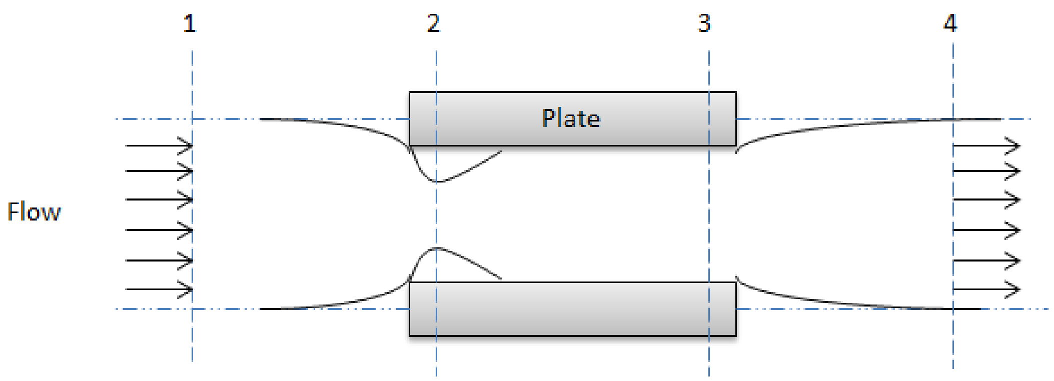


Figure 2.13: Scheme of drop pressure.

to a sudden reduction of area, Δp_{2-3} is the pressure drop within the core and is the largest contribution to the total pressure drop (around 90 % of the total delta p), and Δp_{3-4} the pressure rise at the core exit.

Entrance Loss

The core entrance pressure drop consists of two contributions: the pressure drop due to the flow area change, and the pressure losses associated with the free expansion of that follow sudden contraction. To evaluate the core entrance losses, it will be assumed that the temperature change at the entrance is small and the fluid velocity is small. Thus the fluid is treated as incompressible. Combing the two effects, the total pressure drop Δp_{1-2} is:

$$\Delta p_{1-2} = \frac{1}{2}\frac{G^2}{\rho_i g_c}\left(1-\rho_i^2+K_c\right) \tag{2.35}$$

where is the passage contraction radio and $G = \dfrac{\dot{m}}{A_{o,2}}$ is the mass flux of the fluid. In general:

$$\sigma = \frac{minimumflowarea}{frontalarea} = \frac{A_{o,2}}{A_{o,1}} \tag{2.36}$$

Core Loss

The pressure drop within the core consists of two contributions: the pressure loss caused by fluid friction, that is about the 90% of the total pressure loss, and the pressure change due to the momentum rate change in the core.

$$\Delta p_{2-3} = \delta p_{momentumchange} + \delta p_{friction} = \frac{1}{2}\frac{G^2}{\rho_i g_c}\left[\frac{4fL}{D_h}\left(\frac{1}{\rho}\right)+2\left(\frac{\rho_i}{\rho_e}-1\right)\right] \tag{2.37}$$

Where f is the fanning friction factor [9] .

$$f = \frac{\tau_w}{\dfrac{G^2}{\rho_i g_c}} \tag{2.38}$$

Where τ_w is the wall shear stress (is due to the flow kinetic energy per unit volume).

$$f = \frac{64}{Re} = \frac{64\mu}{\rho u D_h} \tag{2.39}$$

$$f = \frac{0.3164}{Re^{0.25}} \tag{2.40}$$

Exit Loss

Finally, as the flow exits the core, the fluid may pass through a sudden expansion. Application of Bernoulli's equation with mass conservation results in

$$\Delta p_{3-4} = \frac{1}{2}\frac{G^2}{\rho_i g_c}\left[-\left(1-\rho_i^2+K_e\right)\frac{\rho_i}{\rho_e}\right] \tag{2.41}$$

Total Pressure Loss

So combing together all the contributions to the total pressure drop the result is this:

$$\Delta p = \frac{1}{2}\frac{G^2}{\rho_i g_c}\left[\left(1-\rho_i^2+K_c\right)+\frac{4fL}{D_h}\left(\frac{1}{\rho}\right)+2\left(\frac{\rho_i}{\rho_e}-1\right)-\left(1-\rho_i^2+K_e\right)\frac{\rho_i}{\rho_e}\right] \tag{2.42}$$

But considering that the main influence to the total pressure drop is given by the pressure drop due to the friction, it is possible to assume that the total pressure drop is equal to the pressure drop due to the friction. The final equation implemented in the model is the equation above:

$$\Delta p = \frac{1}{2}\frac{G^2}{\rho_i g_c}\frac{4fL}{D_h}\left(\frac{1}{\rho}\right) \tag{2.43}$$

The final result of the model appears as the figure below shows: After the creation of the model,

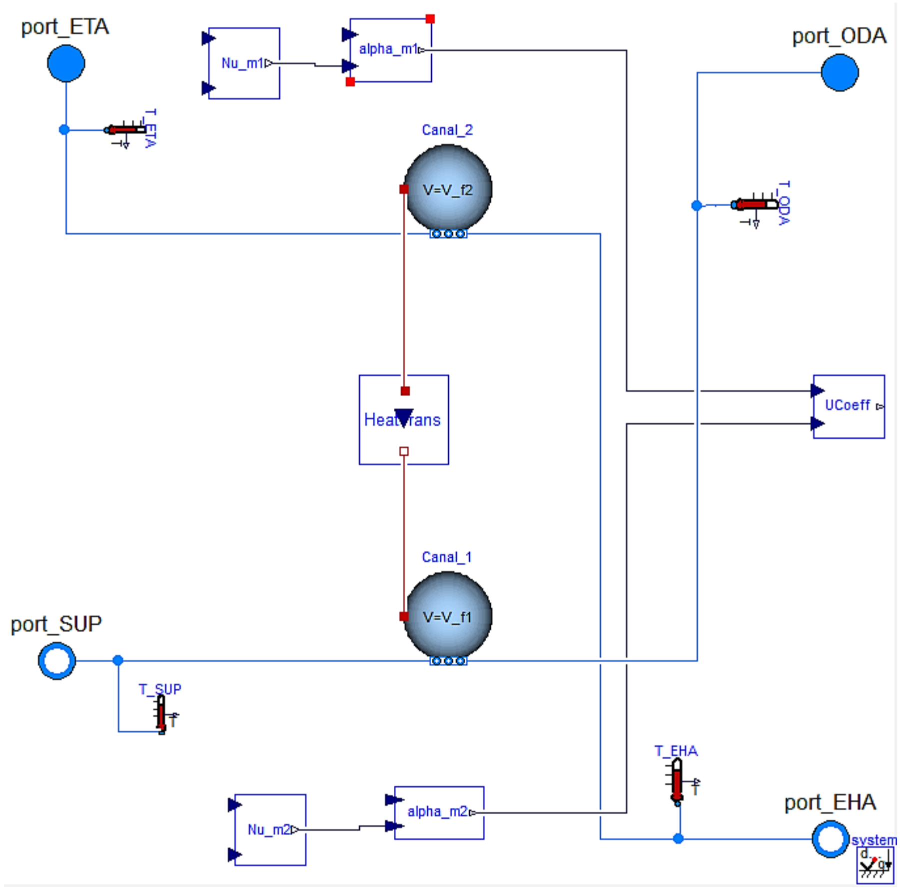


Figure 2.14: Scheme of the Plate heat exchanger model.

it is necessary to build a Simulation Test which is the final model that will be simulated. In this part, the geometry of the heat exchanger, the initial conditions of the inlet air and the final comfort conditions of the supply air are considered as the known variables of the system; while the total heat exchanged has to be discovered.

2.4 Rotary Heat Exchanger

Rotary heat exchangers are regenerators with rotating heat accumulators. The accumulator is a heat-storing mass which is usually made by a porous material and which function is to allow the heat transfer from one fluid to another. Different kind of porous materials are available in the current market and some examples are illustrated in [Fig 2.16]; the geometry used in the model created and simulated in this thesis is the one shown in [Fig 2.17]. The geometry of the matrix material is directly connected with the rotary heat exchanger´s performances. If regenerators are well- design they can recover 60% to 80% of the energy that would otherwise be needed to heat or cool one of the two fluids involved. An example of the real rotary heat exchanger is shown in [Fig 2.15], in the illustration it is possible to understand how the tool operates. Two different gases are alternately routed through these storage mass in opposite directions to obtain a counter-flow movement. The packing material temporary absorbs a certain amount of heat from the hot fluid before releasing it to the cold fluid. The heat transfer is considered as a quasi-steady-state process, it takes place firstly between the hot flow and the storage mass and, secondly, between the storage mass and the cold fluid.

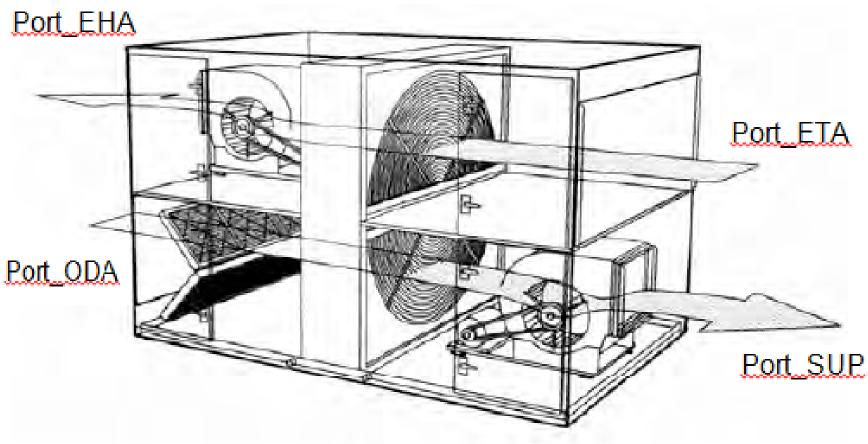


Figure 2.15: Scheme of a Rotary Heat Exchanger [10] .

In case of constant conditions of the fresh air and constant speed of rotation, the local temperature of the supply air is constant with the time. Rotary Regenerators are nowadays used in a wide field of applications, especially in heat recovery industrial equipment as well as in process technology.
A schematic illustration of a rotary heat exchanger gives a first idea of the structure of the model.

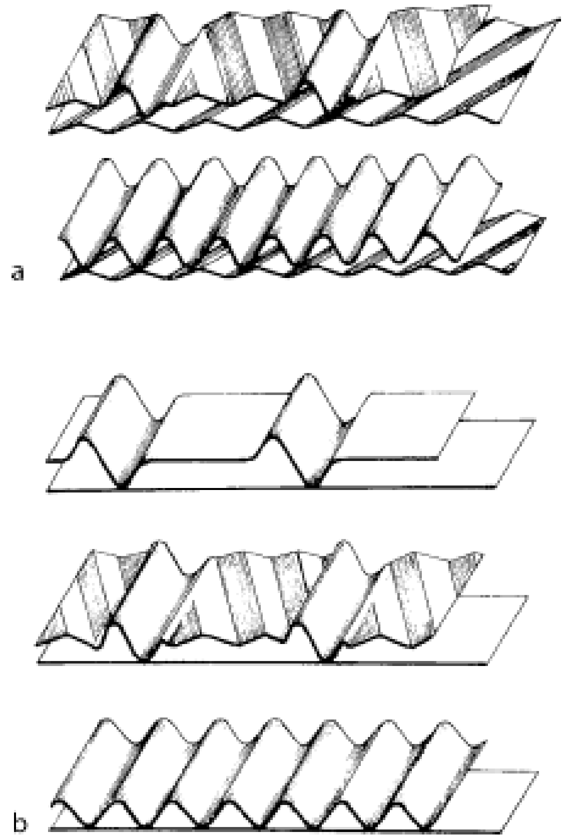


Figure 2.16: Metallic plate profiles for regenerators for power plant applications: (a) open profiles for high heat transfer and (b) closed profiles for good cleaning efficiency [10] .

As explain before there are four air accesses which correspond in the model to:

- ▹ $Port_{ODA}$ is for the outdoor air,
- ▹ $Port_{SUP}$ is for the supply air,
- ▹ $Port_{ETA}$ is for the extract air,
- ▹ $Port_{EHA}$ is for the exhaust air.

The model created is a dry model, there is no condensation or humidity transfer. A “HeatTrans” Subsystem is created to allow the heat transfer and the two ´heat ports´ connectors make it possible. The “HeatTrans” Subsystem is connected with two ClosedVolume, one for each air flow stream, and they are the schematic representation of the two ducts involved in the system. Each ClosedVolume is provide with two fluid ports and one medium model. The two ClosedVolume fluid ports are respectively connected with the entrance and the exit of one air flow. The connections created

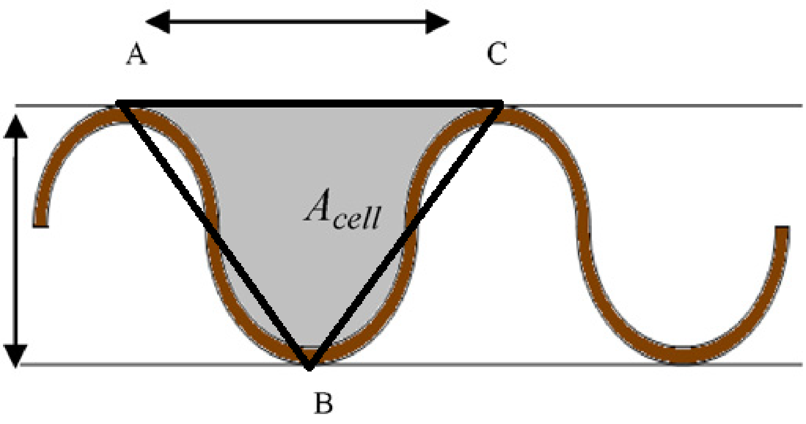


Figure 2.17: The geometry of the regenerator matrix [10] .

between the components represent real physically connections and they allow the transfer of mass from one side to another.
As introduced before the HeatTrans subsystem allow the heat transfer and the equation implemented in the model comes from energy balance

$$Q_{flow} = UA(T_{ODA} - T_{ETA}) \tag{2.44}$$

Where U is the overall heat transfer coefficient and A is the total matrix surface area for the heat exchange that U is based upon.
The total surface area is calculating inside the model considering the geometry of the heat exchanger [13]. A table shows and explains the geometrical parameters defined and used for obtain the total matrix surface area A.

Table 2.2: Geometry parameters of the rotary heat exchanger

Geometry Parameters

Type	**Name**	**Description**
Diameter	D	Disk diameter [m]
Real	α	Disk sector angle [deg]
Lenght	h_{cell}	Height of the cell [m]
Lenght	H	Height of the rotary heat exchanger [m]
Lenght	d	Hub diameter [m]
Real	B	Percentage of seal face coverage
Real	β	Packing density [m2/m3]
Real	σ	Porosity

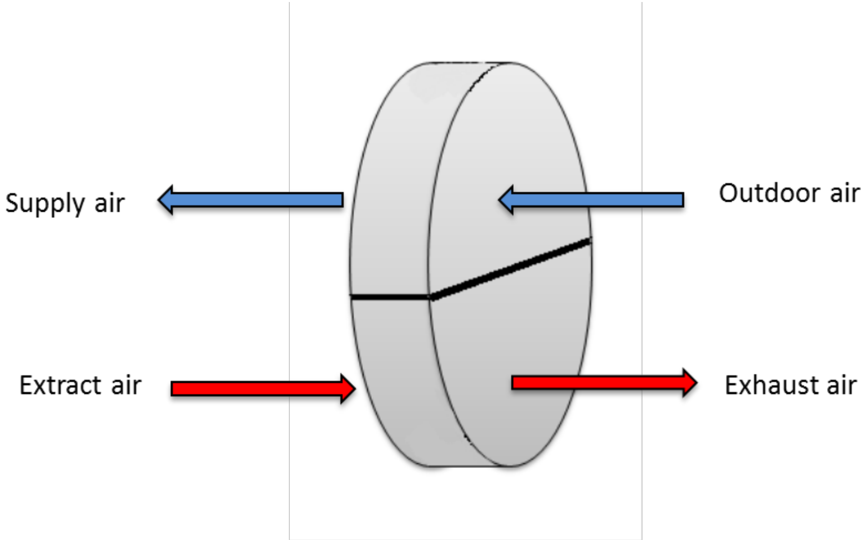


Figure 2.18: Scheme of the rotary heat exchanger.

Due to the geometrical parameters store in [Tab 2.2], the total matrix surface area is [13] :

$$A = \frac{\pi}{4} H\beta(D^2 - d^2)(1 - B) \tag{2.45}$$

While the total surface area A_t is given by:

$$A_t = \frac{\pi}{4}(D^2 - d^2)(1 - B) \tag{2.46}$$

The overall heat transfer coefficient has been calculating with a sequence of blocks connected between them [14] . The formula implemented in the model is:

$$UA = \frac{1}{\frac{1}{\alpha_1 A_h} + \frac{1}{\alpha_2 A_c}} \tag{2.47}$$

where α_1, α_2 are the heat transfer coefficients and A_h, A_c are the hot- and cold-side heat surface areas. The hot- and cold-side heat surface areas are proportional to the respective sectors angles, so that:

$$A_h = \frac{\alpha}{360°} A \tag{2.48}$$

$$A_c = \frac{360° - \alpha}{360°} A \tag{2.49}$$

In this case, it is not necessary to use that equations because the value of alpha fixed in the model is equal to 180°, and it will be easier just divide the total matrix surface area per two. Anyways, the

equations implemented make possible to use the model also for different value of the disk sector angle.
Others parameters to take into account are the heat transfer coefficients α_1, α_2. The equations for both are:

$$\alpha_m = \frac{Nu_m \lambda_m}{l} \tag{2.50}$$

Where λ_m is the thermal conductivity of the fluid, Nu_m is the corresponding Nusselt Number and l is the hydraulic diameter. The inputs give to the Subsystem are the thermal conductivity and Nusselt number.
Nusselt number is a function of Reynolds number, Prandtl number and the geometry of the heat exchanger. Equations are implemented in the model for laminar flow and turbulent flow, so

$$Nu_m = 0.023 Re^{0.8} Pr^{0.33} \tag{2.51}$$

$$Nu_m = 7.55 + \frac{0.024 \left[Pr Re \left(\frac{d_h}{l} \right) \right]}{1 + 0.00358 Pr^{0.81} \left[Re \left(\frac{d_h}{l} \right) \right]^{0.64}} \tag{2.52}$$

Where Re is Reynolds number, Pr is Prandtl number, d_h is the hydraulic diameter and H is the flow length. Re, Pr ,d_h, l are all inputs. Moreover if Re is lower then 2300 Nusselt number is given by the equation [Eq (2.51)], while if Reynolds is higher then 3000 the equation used is [Eq (2.52)].
As conclusion of this chain of operations, the equations for Reynolds number and Prandtl number are implemented inside the model with the following equations:

$$Re = \frac{\rho u d_h}{\mu} \tag{2.53}$$

$$Pr = \frac{c_p \mu}{\lambda} \tag{2.54}$$

Where ρ is the density, u is the velocity of the fluid, μ is the viscosity.
Another important parameter is the efficiency of the heat exchanger. The equation, used to quantify how well the model is working, is:

$$\epsilon = \epsilon_{cf} \left[1 - \frac{1}{9 \left(C_r^* \right)^2} \right] \tag{2.55}$$

$$\epsilon_{cf} = \frac{1 - exp[(C-1)NTU]}{1 - C exp[(C-1)NTU]} \tag{2.56}$$

$$C = \frac{C_{min}}{C_{max}} \tag{2.57}$$

$$NTU = \frac{AU}{C_{min}} \tag{2.58}$$

where $_{cf}$ is the counterflow recuperator effectiveness, C is the ratio of the thermal power, NTU is the number of transfer units, C_r^* is the matrix wall heat capacity rate. All these equations are implemented inside the model with an if-clause.

$$C_r^* = \frac{M_w C_w n}{C_{min}} \tag{2.59}$$

Where M_w is the matrix mass, C_w is the specific heat capacity, and n is the rotor speed.

$$M_w = \frac{\pi}{4} H\rho(D^2 - d^2)(1-\sigma) \tag{2.60}$$

At the end the complete model appears like show in [Fig 2.19].

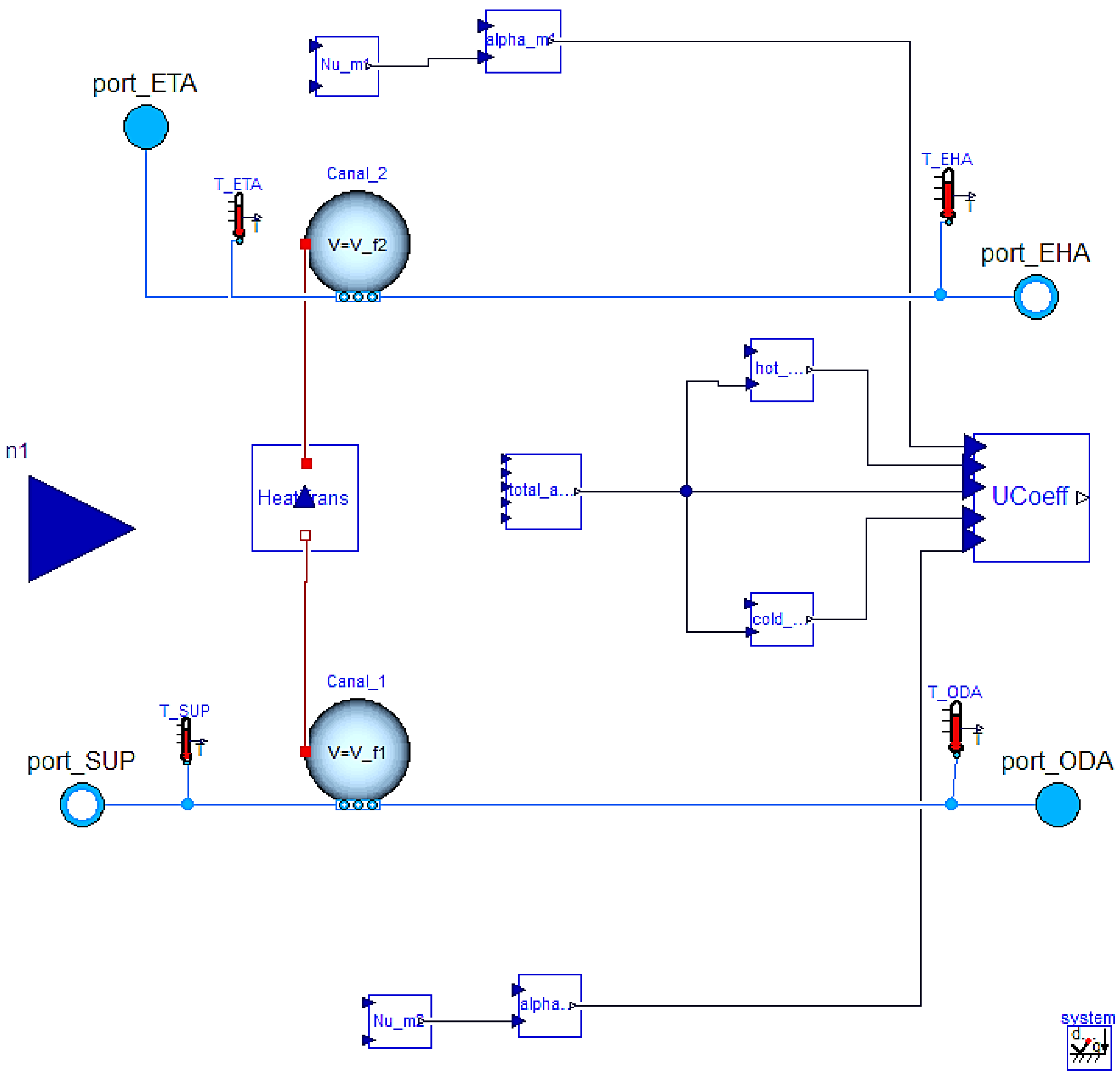


Figure 2.19: Rotary heat exchanger model.

3 Simulations and Results

3.1 Steam Humidifier

The simulation test of the steam humidifier appears like shown in [Fig. 3.1]. The source "inlet air" contains the information regarding the incoming air. A sine function is chosen as the variation of the steam mass flow rate and his frequency is set to 1/3600 Hz. The temperature of the steam is fixed to 100°C with a constant function. Then the values of the specific humidity and the temperature of the outlet air are measured with “massFraction” and “thermometer”.

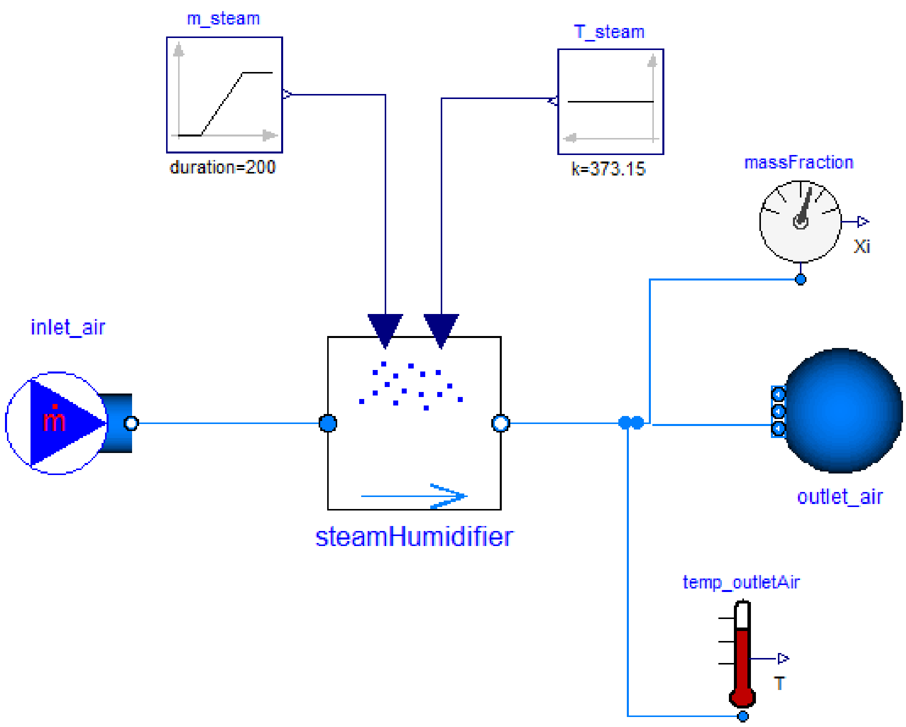


Figure 3.1: Steam Humidifier Simulation Test

For the steam humidifier simulation test inlet air conditions have been set to a value of mass flow air 5.44 kg/s, temperature 15°C, specific humidity 1g/kg. The value of the mass flow steam is the parameter that changes. Firstly it is set to a value of 0.018 kg/s, then to 0.037 kg/s and after to 0.074 kg/s. The diagrams below show how the specific humidity, the enthalpy and the temperature of the

outlet air change if the steam mass flow changes.
As it is possible to see in [Fig. 3.2], if the value of steam mass flow increases, the specific humidity also increases his value.

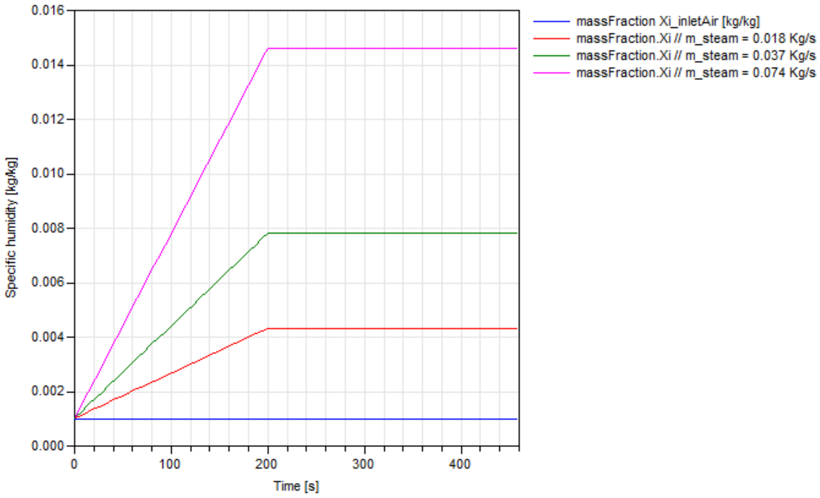


Figure 3.2: Specific humidity for a value of steam mass flow rate of 0.018 kg/s, 0.037 kg/s and 0.074 kg/s.

Similar evaluations can be made with enthalpy parameter [Fig. 3.3]. The value of the enthalpy increase if the amount of steam added to the air flow is higher till reach the saturation curve.

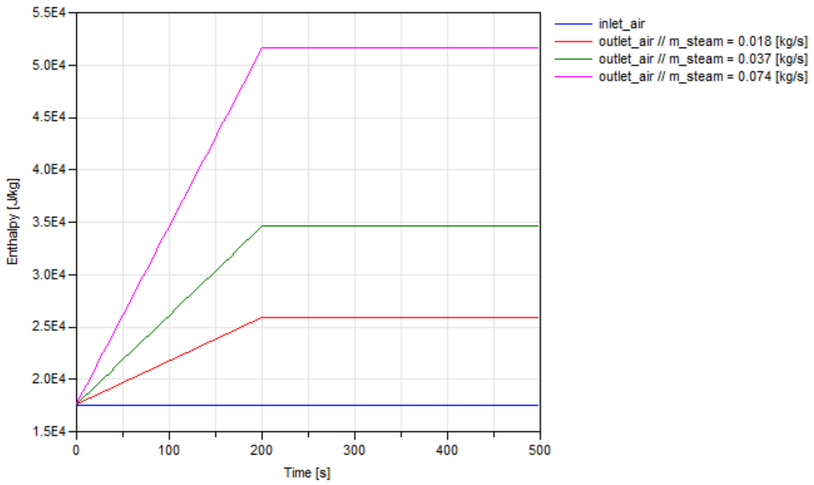


Figure 3.3: Enthalpy for a value of steam mass flow rate of 0.018 kg/s, 0.037 kg/s and 0.074 kg/s.

If the humidification process reaches the conditions of the saturation curve, then the value of temperature increases more than before. What happen is that in saturation condition the air cannot absorb more heat and the hot steam ejected makes the temperature increase [Fig. 3.4].

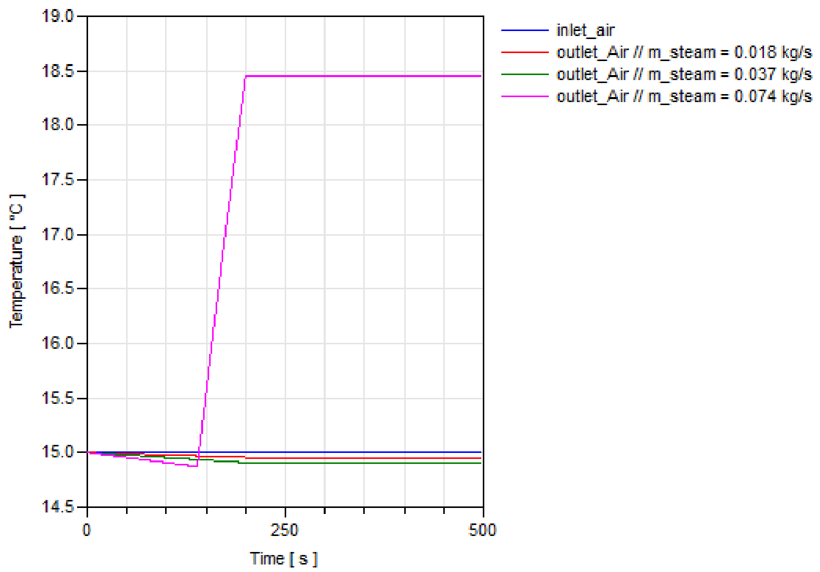


Figure 3.4: Temperatures for a value of steam mass flow rate of 0.018 kg/s, 0.037 kg/s and 0.074 kg/s..

In [Fig. 3.5] it is explained how the transformation operates and it is evident why there is a big increment in the temperature after reaching the saturation curve.

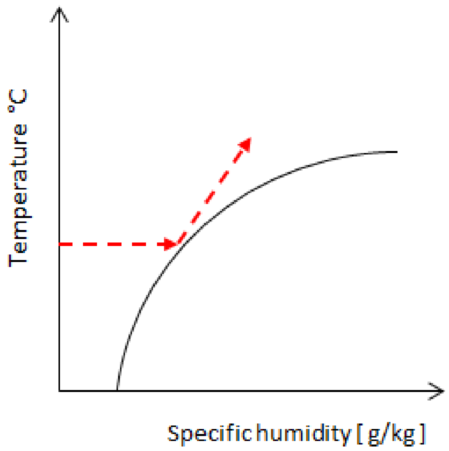


Figure 3.5: Scheme of the air transformation in a steam humidifier.

3.2 Evaporative humidifier

The simulation test of the evaporative humidifier appears like shown in [Fig 3.6]. The configuration of the test is the same that has been created for the steam humidifier. Inputs for the inlet air flow are the same as the previews test (temperature, mass flow rate, specific humidity), but different values for the water mass flow rate and his temperature are set.

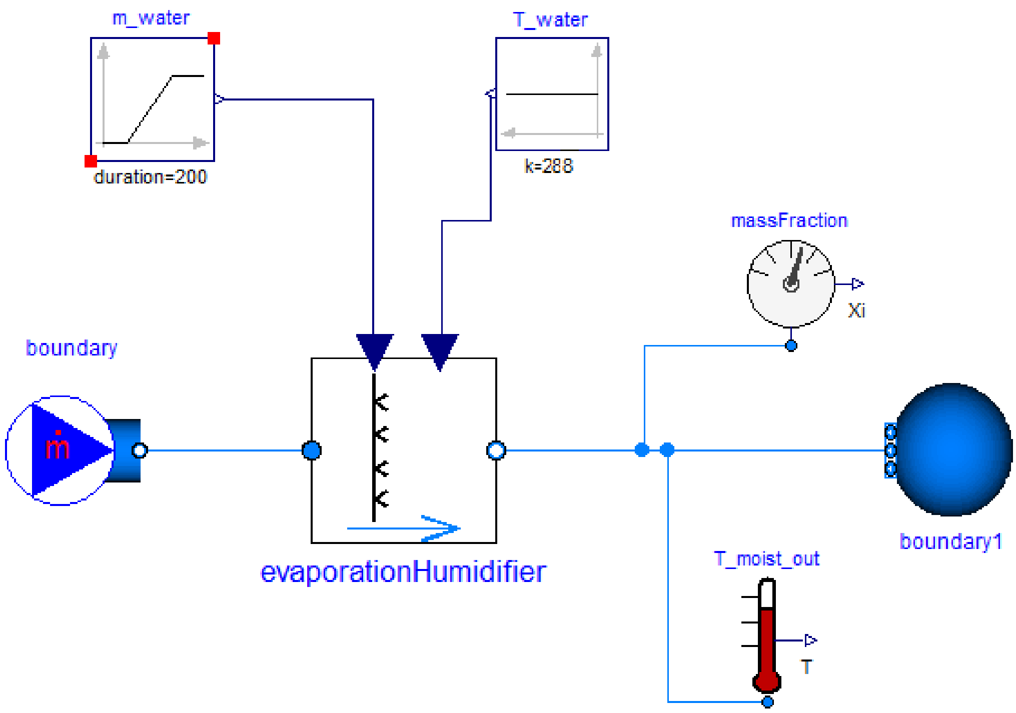


Figure 3.6: Scheme of evaporative humidifier modelica.

The inlet air conditions have been set to a value of mass flow air 1 kg/s, temperature 20°C, specific humidity 3 g/kg. The model has been simulated for three different values of the water mass flow: 0,002 kg/s; 0,004 kg/s and 0,008 kg/s, at 15°C. In the figures below it is possible to see how the specific humidity, the enthalpy and the temperature of the outlet air flow change.

The behavior of the temperature curve it is interesting. It is possible to notice that the curve stops decreasing his temperature at the value of 9.5°C. This is because the air has reached the saturation curve. Then what happen is that, as the temperature of the water injected is higher than the outdoor air´s temperature, the supply air starts to increase a bit his temperature [Fig 3.10].

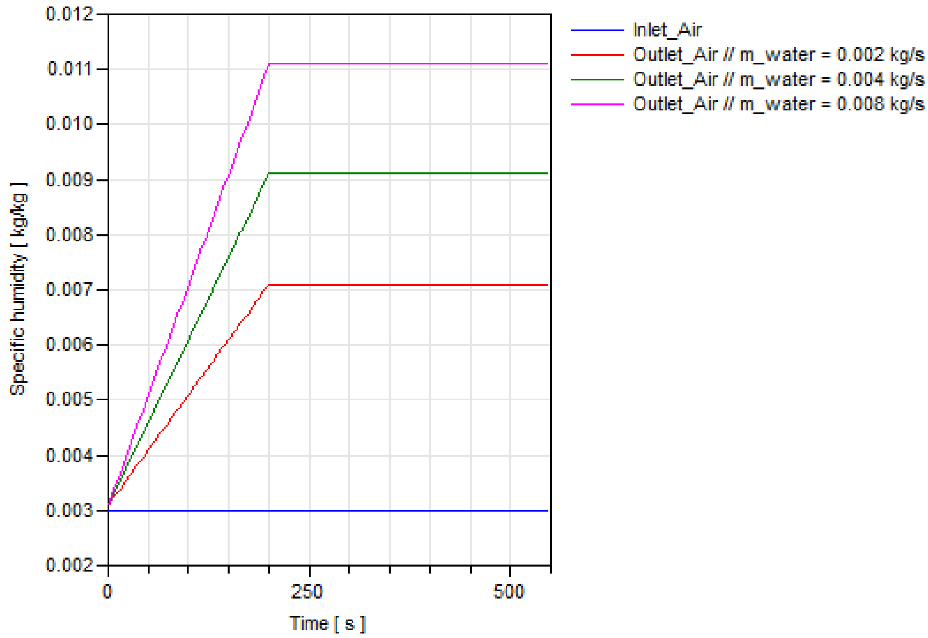


Figure 3.7: Specific humidity for a water mass flow value of 0,002 kg/s; 0,004 kg/s and 0,008 kg/s.

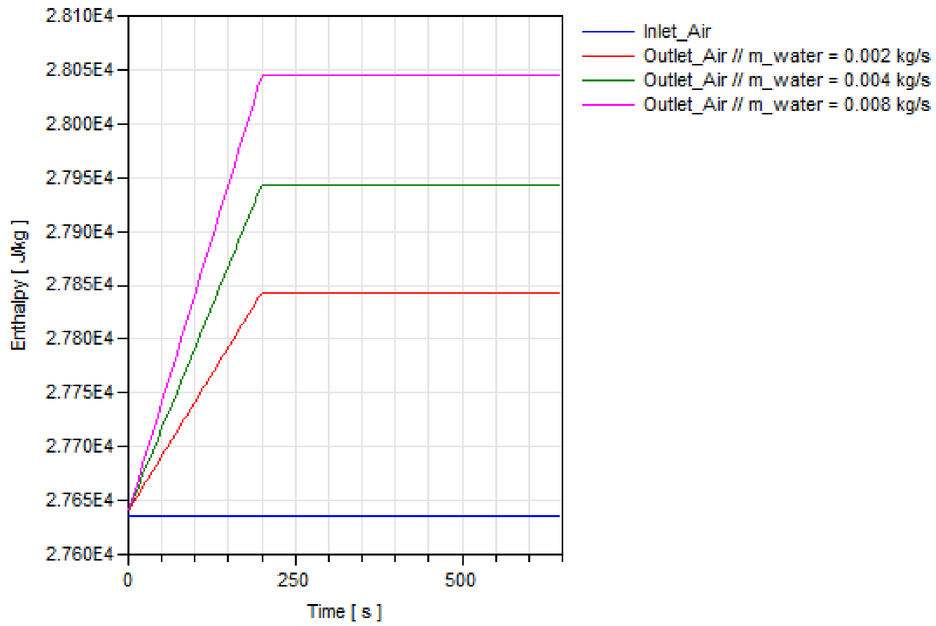


Figure 3.8: Enthalpy of the outlet air for a water mass flow value of 0,002 kg/s; 0,004 kg/s and 0,008 kg/s.

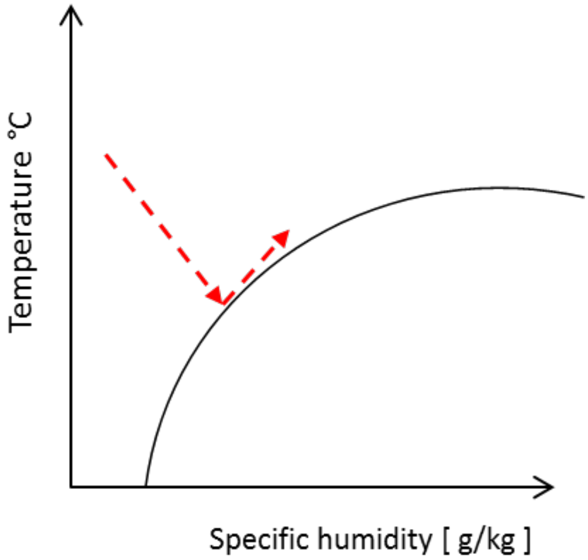


Figure 3.9: Scheme of the air transformation in a evaporative humidifier.

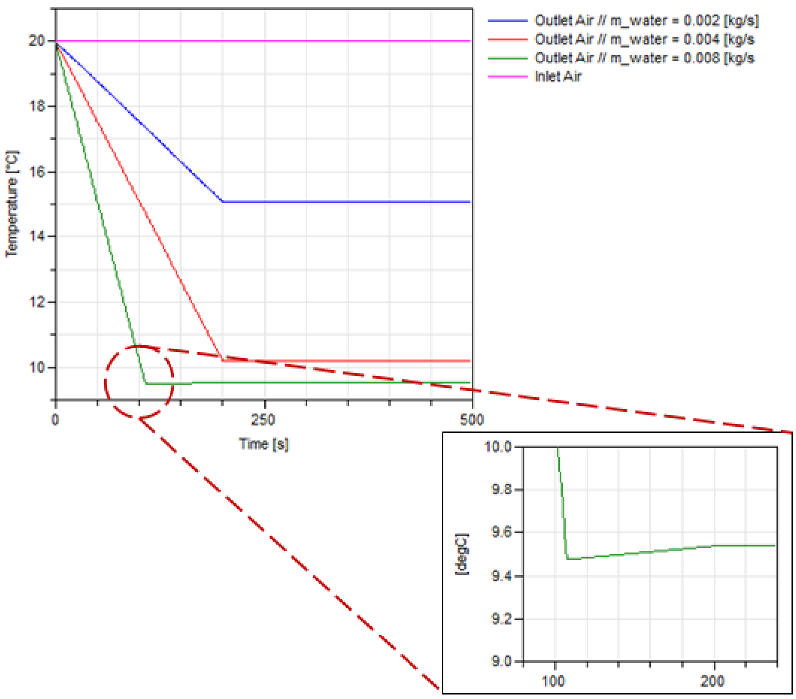


Figure 3.10: Temperatures of the outlet air for a water mass flow value of 0,002 kg/s; 0,004 kg/s; 0,008 kg/s.

3.3 Cross-flow Air-to-Air Plate Heat Exchanger

The simulation test for the plate air to air heat exchanger is shown in the figure below [Fig 3.11]. It is visible the presence of two sources: one for the fresh air and one for the extract air. The conditions of both sources are fixed; it means that both temperature and mass flow rate have been set. Geometry of the plate heat exchanger is also fixed. The model that has been created is a dry-air-model, it means that there is a simplification in the model: the work fluid used is dry air and only sensible recovery takes place.

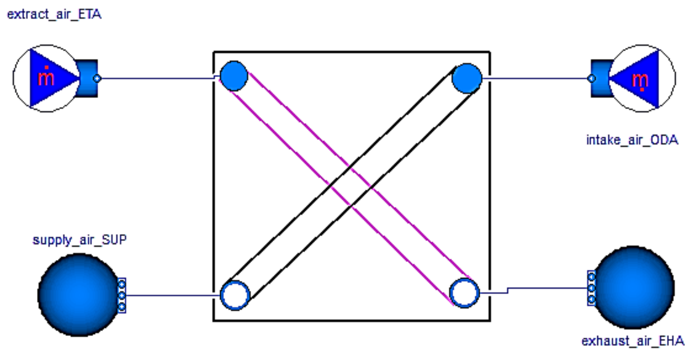


Figure 3.11: Scheme of plate heat exchanger.

The air mass flow rates take the value of 0.2 kg/s the first test, 0.5 kg/s for second and 0.8kg/s the last one. The temperature of the inlet air is set to 5°C, while the extract air is at 26°C for all tests (according to DIN EN 308) The simulation result of this first example is represented in [Fig 3.12], [Fig 3.13], [Fig 3.14].
With increasing the amount of air mass flow the gap of temperature, between the supply air and the exhaust air, become smaller. In other words, the hot flow is less cool down if the air mass flow rate is bigger and this is because the mass has increased more than the heat exchanged has increased. This means that there is more air to cooling down or eating up with the same amount of heat available more or less. Actually this is even clearer if the rate between the heat exchanged and the air mass flow rate is analyzed.

Table 3.1: Variations in $Q_{exc}/\dot{m_{air}}[kWs/kg]$

$\dot{m_{air}}$	Q_{exc}	$Q_{exc}/\dot{m_{air}}[kWs/kg]$
0.2	3.236	16.18
0.5	7.253	14
0.8	10.330	12.9

In 3.1 it is possible to see that the rate heat exchanged and the air mass flow rate decrease if the air mass flow rate increases, and this means that the heat exchanged is increasing less than the air mass flow rate is doing. That is why the temperatures profiles of the exhaust and extract air are like [Fig 3.12], [Fig 3.13], [Fig 3.14].

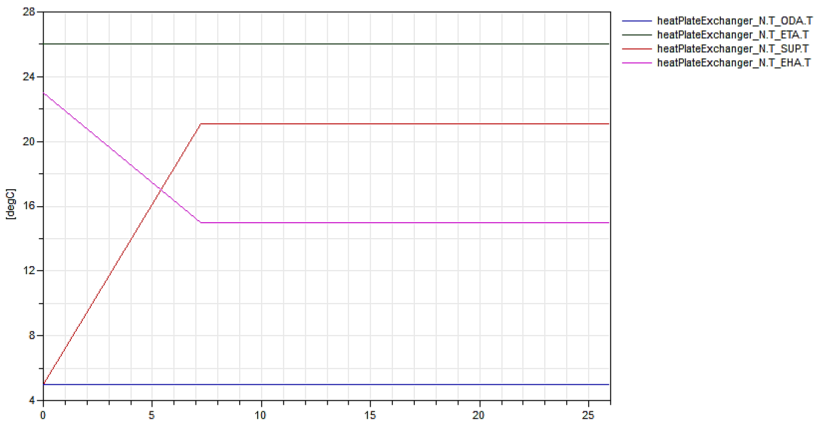


Figure 3.12: Temperatures the inlet, supply, exhaust and extract air with 0.2 kg/s of mass flow rate.

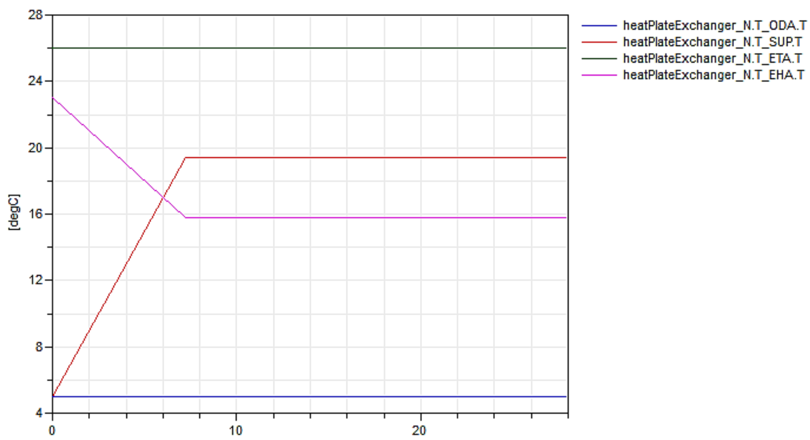


Figure 3.13: Temperatures the inlet, supply, exhaust and extract air with 0.5 kg/s of mass flow rate.

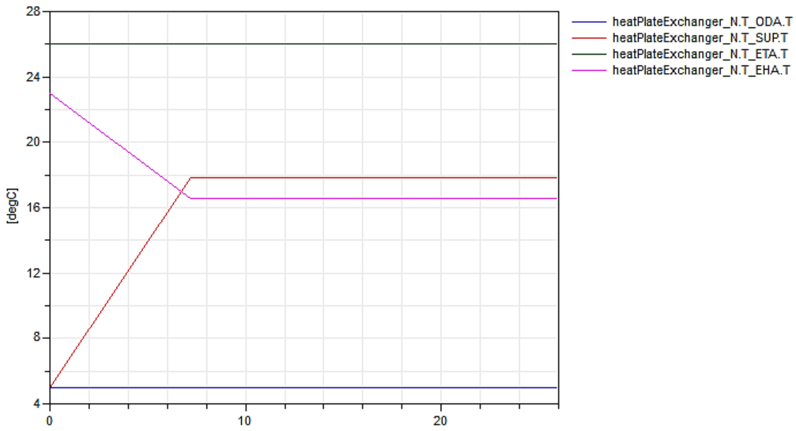


Figure 3.14: Temperatures the inlet, supply, exhaust and extract air with 0.8 kg/s of mass flow rate.

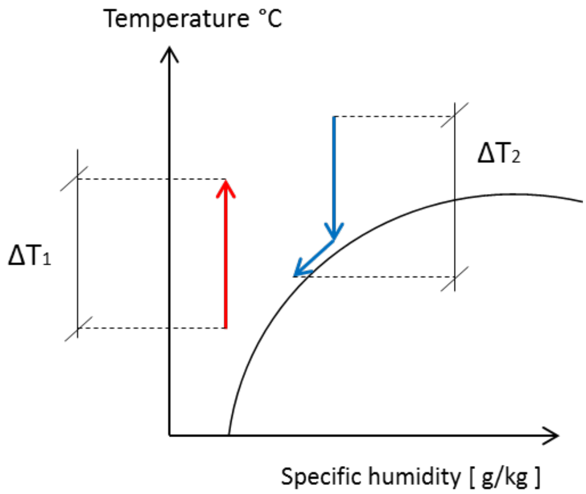


Figure 3.15: Transformation of the two fluids in Mollier diagram $_{12}$.

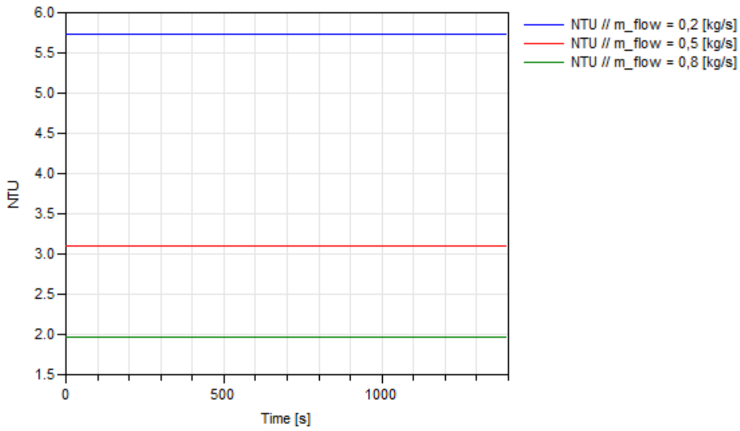


Figure 3.16: NTU number for different values of air mass flow rate.

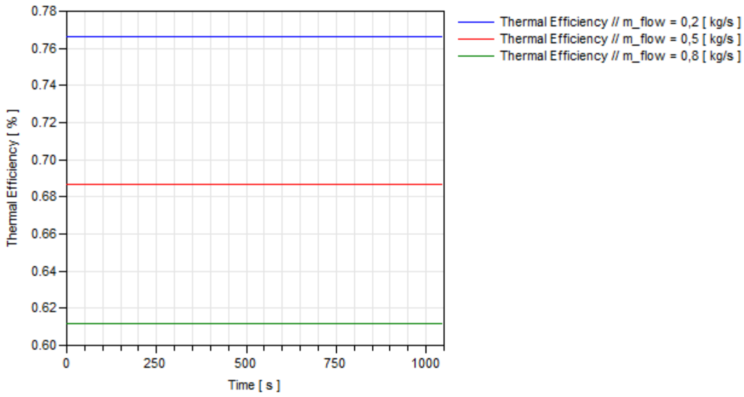


Figure 3.17: Thermal Efficiency for a value of mass flow rate of 0.2 kg/s, 0.5 kg/s and 0.8 kg/s.

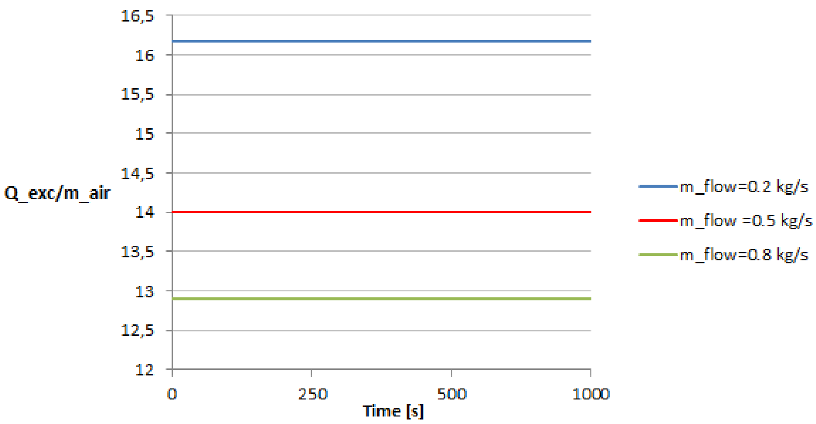


Figure 3.18: Heat flow exchanged rate for a value of mass flow rate of 0.2 kg/s, 0.5 kg/s and 0.8 kg/s .

3.4 Rotary Heat Exchanger

The simulation test of the steam humidifier appears like shown in [Fig 3.19]. The source $intake_{air}$ and $extract_{air}$ contain the information's regarding the air coming from outside and the air extracts from the room. It is chosen to vary the speed of the rotor as a sine function, and this is what the "ramp" input means. Then temperature measurements are used for every air flow.Geometry of the plate heat exchanger is also fixed. The model that has been created is a dry-air-model, it means that there is a simplification in the model: the work fluid used is dry air and only sensible recovery takes place.

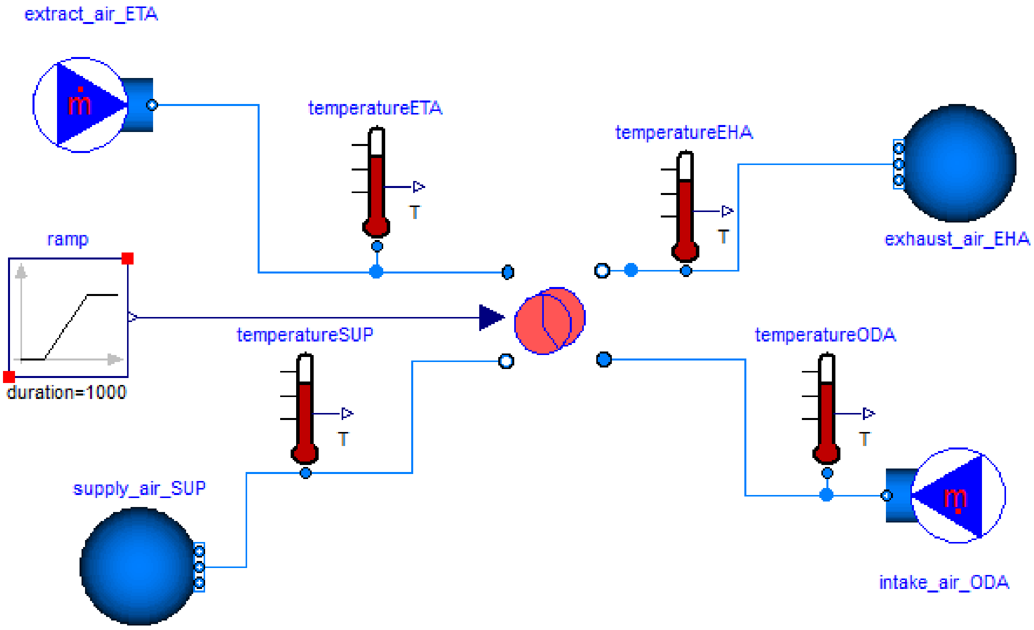


Figure 3.19: Simulation view rotary heat exchanger.

The value used for the air mass flow rate is 0,2 kg/s, 0,5 kg/s and 0,8 kg/s with a rotor speed that varies in the range of 50 – 300 rpm through a ramp sign. The efficiency result for the three different values of air mass flow rate is shown below. As shown in Fig 5.18 if the air mass flow rate changes or/and the rotor speed changes, the value of efficiency changes too. It is interested to notice that higher is the value of the mass and higher is the variation of the efficiency for the same variation of rotor speed. There is an increase in the value of the efficiency if the rotor speed increase. Plus if the mass increases, the efficiency decreases.

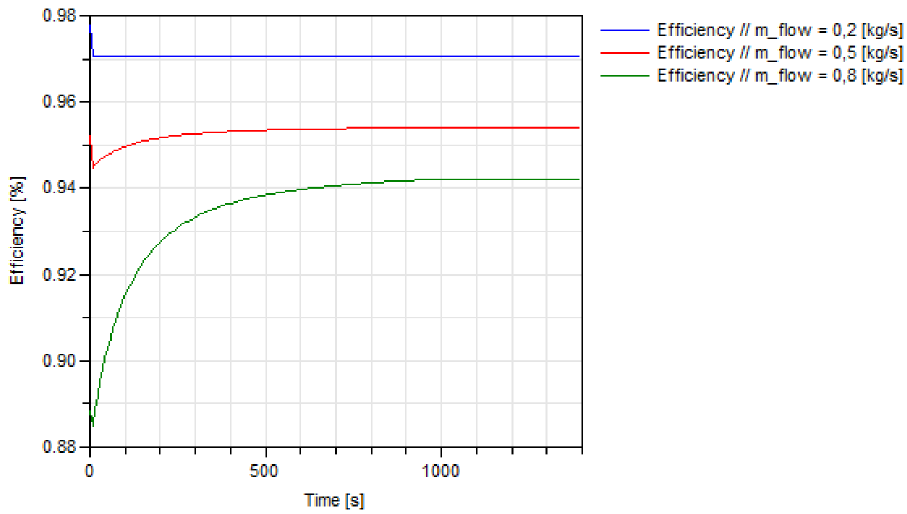


Figure 3.20: Efficiency Values for air mass flow rate of 0,2 kg/s, 0,5 kg/s and 0,8 kg/s with a rotor speed that varies in the range 50 – 300 rpm through a ramp sign.

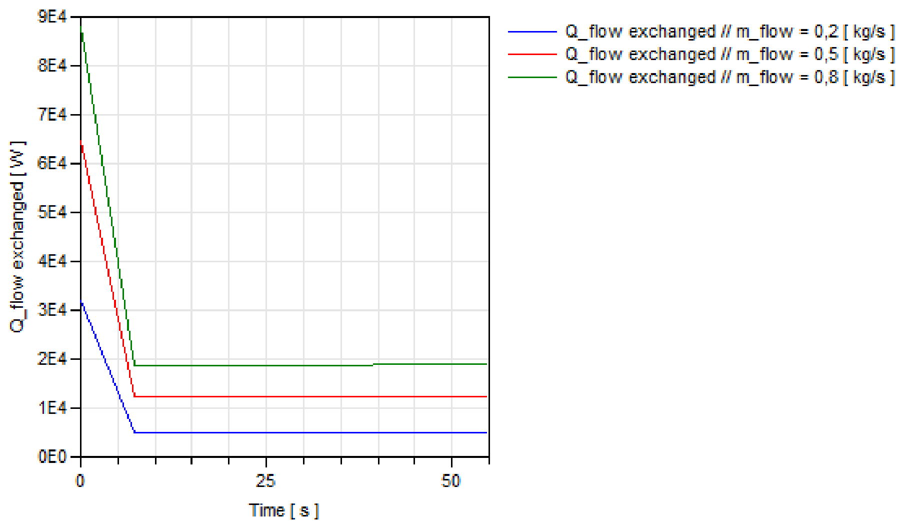


Figure 3.21: Heat flow exchanged for air mass flow rate of 0,2 kg/s, 0,5 kg/s and 0,8 kg/s with a rotor speed that varies in the range 50 – 300 rpm through a ramp sign.

4 Summary and conclusions

The thesis summarizes the behavior of four different models (steam humidifier, evaporative humidifier, air-to-air cross flow plate heat exchanger and rotary heat exchanger) under specific chosen conditions. In order to do this, simulation tests have been created for each of these models. It has been shown that the models of both humidifiers are able to operate according to the Mollier-diagram.

In the case of the steam humidifier, it has been demonstrated that if too much steam is injected and mixed with the outdoor air, the air will reach the saturation state and it will increase its temperature, according to Mollier diagram. This high increase of temperature shows that the steam humidifier model is working properly, and since the aim of the humidifier is not to increase air temperature, this “deviation” in the transformation has to be avoided. The control of the injected steam can be done , once all the models are integrated and the entire system is provided with PID controllers.

For the evaporative humidifier the transformation is an isenthalpic transformation and also in that case if the water injected overcomes the limit value for that pressure, the saturation point can be reached. Hence if the value of relative humidity is about 100% and more water is added to the air, the transformation will start to change direction, as shown in Mollier diagram. It will not be an isenthalpic process anymore. Hence depending on the temperature of the water injected, this change of direction can increase or decrease the temperature of the supply air. Like the steam humidifier model, the way to control the quantity of water injected is through a PID controller.

It is not possible to compare these two different kinds of humidifier, because each of them has a different task. The aim of the thesis was to create a simple but efficient and correct model that simulates the behavior of the two humidifiers separately. One of the main aspects for choosing a humidifier is actually the cost of the energy. This means it should also interesting to consider how the steam for the humidifier is produced, if it would be produced by using electrical resistances, gas or waste steam of some industrial process. It would be also important to know the country where the machine is used and the price of electricity in that country, the environment where the Air Handling Unit is located. In order to get accurate results from the simulation model, it is essential to incorporate these factors during the calculation. This would require further more research and study, that it would be interesting to do in the future.

Concerning both heat exchangers, it is interesting to underline the different values of efficiency obtained with the similar variation of air mass flow rate. The rotary heat exchanger presents a higher value of efficiency than the cross-flow plate exchanger. So, in terms of efficiency, the behavior of the rotary heat exchanger is better. Now, comparing the results about the heat flow exchanged, also in which case, the rotary heat exchanger presents higher values of heat exchanged than the plate heat exchanger. These results and analysis leads to the conclusion that the rotary heat exchanger is more efficient and allows better exchange of heat than the plate heat exchanger. However the study doesn´t take into account the difference in production costs, installation space and smell. In fact, a big disadvantage, that the rotary heat exchanger has, is that it can transport smell from one side to the other. So at the end, these factors can make a difference on the final choice.

A possibility for future studies would be the simulation of a complete Air Handling Unit for the duration of one or more years. This can be done generating a simulation test where all the models of an Air Handling Unit are connected together and where a system of control, like PID, is also used.

Bibliography

[1] Tecnoblu rappresentanze.

[2] Carel industries s.r.l, high efficiency solutions, 2015.

[3] *AHU Magazine, Air-to-air Heat exchangers for handling Units: Fundamentals and Types.*

[4] Michael Tiller. *Introduction to physical modeling with Modelica*, volume 615. Springer Science & Business Media, 2012.

[5] *Dofware.*

[6] *Introduction Dymola / Modelica*, jul 2015. Institute for Efficiency Buildings and Indoor Climate.

[7] H Elmqvist, D Brück, and M Otter. Dymola. dynamic modeling laboratory. user's manual. version 4.1 b. dynasim ab, lund, sweden, 2001.

[8] Manfred Zeller and Ulrich Busweiler. M8 humidifying and drying of air. In *VDI Heat Atlas*, pages 1343–1362. Springer, 2010.

[9] Peter Stephan, S Kabelac, M Kind, H Martin, D Mewes, and K Schaber. Vdi heat atlas, 2010.

[10] Liga Ozolina. Energy management system in industry. experience in latvia. 2011.

[11] Wilfried Roetzel and Bernhard Spang. C1 thermal design of heat exchangers. In *VDI Heat Atlas*, pages 31–66. Springer, 2010.

[12] Ramesh K Shah and Dusan P Sekulic. *Fundamentals of heat exchanger design*. John Wiley & Sons, 2003.

[13] Sepehr Sanaye, Saeid Jafari, and Hadi Ghaebi. Optimum operational conditions of a rotary regenerator using genetic algorithm. *Energy and Buildings*, 40(9):1637–1642, 2008.

[14] Ramesh K Shah and Dušan P Sekulić. Thermal design theory for regenerators. *Fundamentals of Heat Exchanger Design*, pages 308–377, 1981.

Appendix

A appendix

A.1 Steam Humidifier Code

A.2 Evaporative Humidifier Code

A.3 Plate Heat Exchanger Code

A.4 Rotary Heat Exchanger Code

Listing A.1: Code Steam Humidifier.

```
model SteamHumidifier

extends Modelica.Fluid.Interfaces.PartialTwoPort;
    outer Modelica.Fluid.System baseParameters
    "SysModelica.Fluid.Systemtem properties";

  Medium.EnthalpyFlowRate port_a_H_flow; // W
  Medium.EnthalpyFlowRate port_b_H_flow; // W

  Modelica.SIunits.HeatFlowRate Q_steam;
  Modelica.SIunits.Power P_boiler;

  Modelica.Blocks.Interfaces.RealInput m_flow_steam annotation (Placement(
        transformation(
        extent={{-20,-20},{20,20}},
        rotation=-90,
        origin={-34,106})));  // Kg/s
  Modelica.Blocks.Interfaces.RealInput T_steam annotation (Placement(
        transformation(
        extent={{-20,-20},{20,20}},
        rotation=-90,
        origin={32,106})));

  constant Modelica.SIunits.SpecificEnthalpy Deltah_vo = 2500*10^3; // J/kg
  constant Modelica.SIunits.SpecificHeatCapacity cp_steam = 1.86; // J/(kg.K)

  Integer nXi = Medium.nXi;

equation

  0 = port_a.p - port_b.p;
  port_a.m_flow + port_b.m_flow = 0;
  port_a_H_flow = port_a.m_flow * actualStream(port_a.h_outflow)
    "Enthalpy flow at port a";
  port_b_H_flow = port_b.m_flow * actualStream(port_b.h_outflow)
    "Enthalpy flow at port b";

port_b.Xi_outflow=inStream(port_a.Xi_outflow)
                             +m_flow_steam/port_a.m_flow*ones(nXi);
    port_a.Xi_outflow = zeros(nXi);

  m_flow_steam*(Deltah_vo + cp_steam * T_steam) = P_boiler;

  m_flow_steam * Deltah_vo = Q_steam;

  port_b.h_outflow = inStream(port_a.h_outflow)
                          + Deltah_vo*(m_flow_steam / port_a.m_flow);
   port_a.h_outflow = 20*10^3;

  port_a.C_outflow = inStream(port_b.C_outflow);
  port_b.C_outflow = inStream(port_a.C_outflow);

end SteamHumidifier;
```

Listing A.2: Code Evaporative Humidifier.

```
model EvaporationHumidifier

extends Modelica.Fluid.Interfaces.PartialTwoPort;
    outer Modelica.Fluid.System baseParameters
    "SysModelica.Fluid.Systemtem properties";

  parameter Modelica.SIunits.Efficiency Efficiency=0.8;

  Medium.EnthalpyFlowRate port_a_H_flow; // W
  Medium.EnthalpyFlowRate port_b_H_flow; // W

  Modelica.SIunits.HeatFlowRate Q_steam;
  Modelica.SIunits.Power P_boiler;
  Real Ky;

  Modelica.Blocks.Interfaces.RealInput m_flow_steam // Kg/s
  Modelica.Blocks.Interfaces.RealInput T_steam
  constant Modelica.SIunits.SpecificEnthalpy Deltah_vo = 50.4*10^3; // J/kg
  constant Modelica.SIunits.SpecificHeatCapacity cp_steam = 1.86; // J/(kg.K)

  Integer nXi = Medium.nXi;

equation

  0 = port_a.p - port_b.p;
  port_a.m_flow + port_b.m_flow = 0;
  port_a_H_flow = port_a.m_flow * actualStream(port_a.h_outflow)
    "Enthalpy flow at port a";
  port_b_H_flow = port_b.m_flow * actualStream(port_b.h_outflow)
    "Enthalpy flow at port b";

    port_b.Xi_outflow = inStream(port_a.Xi_outflow) + m_flow_steam / port_a.m_flow * ones(nXi);

  m_flow_steam*(Deltah_vo + cp_steam * T_steam) = P_boiler;

  m_flow_steam * Deltah_vo = Q_steam;

  port_b.h_outflow = inStream(port_a.h_outflow) + Deltah_vo*(m_flow_steam / port_a.m_flow);
   port_a.h_outflow = 20*10^3;

  port_a.C_outflow = inStream(port_b.C_outflow);
  port_b.C_outflow = inStream(port_a.C_outflow);

 // Efficiency = (1 - exp(Ky*(m_flow_steam / port_a.m_flow)));

 Ky = -(m_flow_steam / port_a.m_flow)*log( 1-Efficiency);

end EvaporationHumidifier;
```

Listing A.3: Code Plate Heat Exchange.

```
model HeatPlateExchanger_N

 inner Modelica.Fluid.System system
   annotation (Placement(transformation(extent={{92,-100},{100,-92}})));
  import SI = Modelica.SIunits;
  import Modelica.Math;
protected
  parameter SI.MassFlowRate m_flow_min=1 " Smallest Mass flow rate in modell";
                                        //1e-3
    //*******************************************************************// declaration of the fluids

public
  replaceable package Medium_1 =
      Modelica.Media.Interfaces.PartialMedium "outside air"
                  annotation (choicesAllMatching=true);
  replaceable package Medium_2 =
      Modelica.Media.Interfaces.PartialMedium "Indoor air"
                 annotation (choicesAllMatching=true);

  parameter Real Flowstreaming=2 annotation (choices(
      choice=1 "Pur coflow",
      choice=2 "Pur cross flow",
      choice=3 "Pur counter flow",
      choice=4 "Cross counterflow"));
  //************************************************************// Geometrical properties of the heat Exchanger

  /// ************************************************************************** Geometric parameters
  parameter SI.Thickness s "Plate thickness"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Thickness p_s "Plate spacing"
    annotation (Dialog(tab="Geometry"));

  parameter SI.Thickness d_h=2*(p_s - s) "Hydraulic diameter"
    annotation (Dialog(tab="Geometry"));

  parameter SI.Length A "Lenght of the heat exchanger"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Length B "Height of the heat exchanger"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Length C " Depth of the heat exchanger"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Length D " Length of the plate"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Length L_sheet "Sheet length"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Length e "Plate width" annotation (Dialog(tab="Geometry"));
  parameter SI.Length L_flow=SubSystems.GeometryData.FlowLength(
          A,
          B,
          e,
          D,
          F_p) "Mean flow length" annotation (Dialog(tab="Geometry"));

  parameter SI.Length F_p "Sheet heigth"
    annotation (Dialog(tab="Geometry"));
  parameter Real f=0.45 " Extension factor of the plate surface 0<= f <=1"
    annotation (Dialog(tab="Geometry"));

    //*******************************************************************************************
protected
  parameter Integer n_total=SubSystems.GeometryData.PlateNumber(p_s, C)
    "Number of plate in the exchanger" annotation (Dialog(tab="Geometry"));

  parameter Integer n1_gap=SubSystems.GeometryData.GapNumber(n_total)
    "Number of gap on the side 1" annotation (Dialog(tab="Geometry"));
```

```
  parameter Integer n2_gap=if (n_total - n1_gap*2) == 0 then n1_gap else
      n1_gap - 1 "Number of gap on the side 2"
    annotation (Dialog(tab="Geometry"));

  parameter SI.Area A_HxC=SubSystems.GeometryData.HxCrossArea(
          D,
          C,
          L_sheet) " Cross area of the heat exchanger"
    annotation (Dialog(tab="Geometry", group="Area"));

  parameter SI.Area A1_DuC=SubSystems.GeometryData.DuctCrossArea(
          s,
          p_s,
          D,
          L_sheet,
          C,
          n1_gap,
          f) " Cross sectional area of the duct on side 1"
    annotation (Dialog(tab="Geometry", group="Area"));

  parameter SI.Area A2_DuC=SubSystems.GeometryData.DuctCrossArea(
          s,
          p_s,
          D,
          L_sheet,
          C,
          n2_gap,
          f) " Cross sectional area of the duct on side 2"
    annotation (Dialog(tab="Geometry", group="Area"));

  parameter SI.Area A_pl=SubSystems.GeometryData.PlateSurface(
          A,
          e,
          B,
          F_p,
          f) "Surface of a plate"
    annotation (Dialog(tab="Geometry", group="Area"));

  parameter SI.Area A_HX=SubSystems.GeometryData.ExchangeSurface(A_pl,
      n_total) "Heat exchange surface"
    annotation (Dialog(tab="Geometry", group="Area"));
 // *****************//

  parameter SI.Volume V_f1=SubSystems.GeometryData.FluidVolume(
          A_pl,
          n1_gap,
          (p_s - s)) "Volume of the fluid in the duct 1"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Volume V_f2=SubSystems.GeometryData.FluidVolume(
          A_pl,
          n2_gap,
          (p_s - s)) "Volume of the fluid in the duct 2"
    annotation (Dialog(tab="Geometry"));

  //********************************************************************* Physically parameters
public
  parameter SI.Temperature T_ODA0 "Ouside air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));
  parameter SI.Temperature T_SUP0 "Indoor air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));
                               //=T_ODA0 - (T_ODA0 - T_ETA0)*0.1
```

```
 parameter SI.Temperature T_ETA0 "Outlet air temperature at the time t=0 s"
   annotation (Dialog(tab="Initialization"));
  parameter SI.Temperature T_EHA0 "Exhaust air temperature at the time t=0 s"
   annotation (Dialog(tab="Initialization"));
                              //=T_ETA0 + (T_ODA0 - T_ETA0)*0.1
  parameter SI.Density rho_p=2700 "Density of the Lamella, Alu =2700"
   annotation (Dialog(tab="Wall Plate"));
  parameter SI.SpecificHeatCapacityAtConstantPressure cp_p=897
   " spec. heat capacity of the Lamella, Alu=897"
   annotation (Dialog(tab="Wall Plate"));
  parameter SI.ThermalConductivity lambda_P=237
   " Thermal conductivity of the plate, Alu-plate=237 W/(m.K)"
   annotation (Dialog(tab="Wall Plate"));

//*********************************************************************************///loss of pressure

 Real dp_1( unit="Pa") "loss of pressure canal1";
 Real dp_2( unit="Pa") "loss of pressure canal2";

// Modelica.SIunits.Efficiency Eff "Efficiency";

//*****************************************************************************************************

 // The variables

  SI.Velocity v_d1 "Velocity of the Air inside the duct 1";
  SI.Velocity v_d2 "Velocity of the Air inside the duct 2";

  SI.Velocity v_a1 " Approach velocity in the outsidestream cross section";
  SI.Velocity v_a2 " Approach velocity in the Inletstream cross section";

  /// Temperatures
  SI.TemperatureDifference dTm(start=((T_ODA0 - T_EHA0) + (T_SUP0 -
      T_ETA0))/2, nominal=((T_ODA0 - T_EHA0) + (T_SUP0 - T_ETA0))/2)
   "Cross flow logarithmic mean temperatur difference";

  /// Specific heat capacity of the fluids.
  SI.SpecificHeatCapacity cp_1=Medium_1.specificHeatCapacityCp(
     MeanState_1);
  SI.SpecificHeatCapacity cp_2=Medium_2.specificHeatCapacityCp(
     MeanState_2);

  Real mu_1=Medium_1.dynamicViscosity(MeanState_1);
  Real mu_2=Medium_1.dynamicViscosity(MeanState_2);

  Real rho_1=Medium_1.density(MeanState_1)
   "Density of the medium 1 at the mean state";
  Real rho_2=Medium_2.density(MeanState_2)
   "Density of the medium 2 at the mean state";

  /// Dimensionless temperature change accords to VDI-Wrmeatlas.
  Real P;

  Real NTU;

  Real tsi " Heat grad transfer";

  /// ** Thermal power capacity ***
  Real W_1=(port_ODA.m_flow)*(Medium_1.specificHeatCapacityCp(medium_ODA.state)
      + Medium_1.specificHeatCapacityCp(medium_SUP.state))/2;
```

```
Real W_2=(port_ETA.m_flow)*(Medium_2.specificHeatCapacityCp(medium_ETA.state)
      + Medium_2.specificHeatCapacityCp(medium_EHA.state))/2;

  /// Ratio of the thermal power capacity
  Real R=W_1/W_2;

  Real phi_ETA "Degree of efficiency of the outletstream";
  Real phi_ODA "Degreee of efficiency of the inletstream";

  // *************** Medium Properties in Ports ***************
  Medium_1.BaseProperties medium_ODA(preferredMediumStates=true, T(start=
        T_ODA0, nominal=T_ODA0)) "Medium properties in port_ODA ";
  Medium_1.BaseProperties medium_SUP(preferredMediumStates=true, T(start=
        T_SUP0, nominal=T_SUP0)) "Medium properties in port_SUP ";
  Medium_2.BaseProperties medium_ETA(preferredMediumStates=true, T(start=
        T_ETA0, nominal=T_ETA0)) "Medium properties in port_ETA ";
  Medium_2.BaseProperties medium_EHA(preferredMediumStates=true, T(start=
        T_EHA0, nominal=T_EHA0)) "Medium properties in port_EHA ";

  /// *** Declaration of the Meanstate of the both Fluid ***
  Medium_1.ThermodynamicState MeanState_1
    " Thermodynamic properties of fluid 1 at the mean temperature";
  Medium_2.ThermodynamicState MeanState_2
    " thermodynamic properties of fluid 2 at the mean temperature";
      //// Sub-classes and Ports

  // ******************* Exchangers port **************************************************************
 Modelica.Fluid.Interfaces.FluidPort_a port_ODA(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.SimpleAir) "Outside air"
                annotation (Placement(transformation(extent={{78,72},{
            98,92}}), iconTransformation(extent={{72,70},{94,92}})));
 Modelica.Fluid.Interfaces.FluidPort_b port_SUP(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.SimpleAir) "Inlet air"
              annotation (Placement(transformation(extent={{-98,-60},
            {-78,-40}}),iconTransformation(extent={{-96,-92},{-74,-70}})));
  Modelica.Fluid.Interfaces.FluidPort_a port_ETA(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.SimpleAir) "Exit air"
             annotation (Placement(transformation(extent={{-96,74},{-76,94}}),
                 iconTransformation(extent={{-96,72},{-76,94}})));
  Modelica.Fluid.Interfaces.FluidPort_b port_EHA(
                m_flow(
      start=m_flow_min,
      nominal=m_flow_min,
      min=1), redeclare package Medium = Modelica.Media.Air.SimpleAir)
    "Exhaust air " annotation (Placement(transformation(extent={{76,-100},
            {96,-80}}), iconTransformation(extent={{70,-90},{92,-68}})));
  //*************** Temperature Sensores **************************************************************

  Modelica.Fluid.Sensors.Temperature T_ODA(
                T(start=T_ODA0, nominal=T_ODA0), redeclare package Medium =
        Modelica.Media.Air.SimpleAir) "Outside air ( Aussenluft)"
                          annotation (Placement(transformation(
        extent={{-6.5,-5},{6.5,5}},
        rotation=270,
        origin={69,52.5})));
```

```
Modelica.Fluid.Sensors.Temperature T_SUP(
                T(start=T_SUP0, nominal=T_SUP0), redeclare package Medium =
        Modelica.Media.Air.SimpleAir) "Inlet air (Zuluft)"
    annotation (Placement(transformation(
        extent={{-4.5,-4},{4.5,4}},
        rotation=0,
        origin={-64.5,-62})));

  Modelica.Fluid.Sensors.Temperature T_ETA(
                T(start=T_ETA0, nominal=T_ETA0), redeclare package Medium =
        Modelica.Media.Air.SimpleAir) " Exit air (Abluft) "
    annotation (Placement(transformation(
        extent={{-5,-4.5},{5,4.5}},
        rotation=270,
        origin={-72.5,69})));

  Modelica.Fluid.Sensors.Temperature T_EHA(
                T(start=T_EHA0, nominal=T_EHA0), redeclare package Medium =
        Modelica.Media.Air.SimpleAir) "Exhaust air( Fortluft)"
    annotation (Placement(transformation(extent={{45,-82},{58,-72}})));
  /// **** components to represent the heat exchanger room ********************************************

  Modelica.Fluid.Vessels.ClosedVolume Canal_2(
    use_HeatTransfer=true,
    use_T_start=true,
    V=V_f2,
    T_start=T_ETA0,
    redeclare model HeatTransfer =
      Modelica.Fluid.Vessels.BaseClasses.HeatTransfer.IdealHeatTransfer,
    portsData={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
        diameter=d_h),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
        diameter=d_h)},
    nPorts=2,
    use_portsData=false,
    redeclare package Medium = Modelica.Media.Air.SimpleAir)
    annotation (Placement(transformation(extent={{-10,46},{10,66}})));

  Modelica.Fluid.Vessels.ClosedVolume Canal_1(
    use_HeatTransfer=true,
    use_T_start=true,
    V=V_f1,
    nPorts=2,
    T_start=T_ODA0,
    redeclare model HeatTransfer =
      Modelica.Fluid.Vessels.BaseClasses.HeatTransfer.IdealHeatTransfer,
    portsData={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
        diameter=d_h),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
        diameter=d_h)},
    use_portsData=false,
    redeclare package Medium = Modelica.Media.Air.SimpleAir)
    annotation (Placement(transformation(extent={{-10,-50},{10,-30}})));

  // **** U-Value; Nu-Number; alpha-Value ************************************************************
  SubSystems.HeatTransfer HeatTrans
    annotation (Placement(transformation(extent={{-20,-6},{0,14}})));

  SubSystems.ThermalOperatingData.HeatTransferCoefficient_U UCoeff(s=s)
    annotation (Placement(transformation(extent={{82,0},{98,14}})));

  SubSystems.ThermalOperatingData.NusseltNumber.NussMean_LevelColumn
    Nu_m1(d_h=d_h, l=L_flow)
```

```
"Mean Nu-number in a level coloumn on the streaming side 1"
    annotation (Placement(transformation(extent={{-54,76},{-38,92}})));
  SubSystems.ThermalOperatingData.CoefficientsofHeatTransfer_alpha
    alpha_m1(l=d_h)
    annotation (Placement(transformation(extent={{-22,80},{-4,94}})));

  SubSystems.ThermalOperatingData.NusseltNumber.NussMean_LevelColumn
    Nu_m2(l=L_flow, d_h=d_h)
    "Mean Nu-number in a level coloumn on the streaming side 2"
    annotation (Placement(transformation(extent={{-48,-96},{-32,-80}})));
  SubSystems.ThermalOperatingData.CoefficientsofHeatTransfer_alpha
    alpha_m2(l=d_h)
    annotation (Placement(transformation(extent={{-12,-90},{8,-78}})));

equation
  // ***** evaluation of the approach velocity v_i *****************************************************
  v_d1 = (port_ODA.m_flow)/rho_1/A1_DuC;
  v_d2 = (port_ETA.m_flow)/rho_2/A2_DuC;

  v_a1 = (port_ODA.m_flow)/rho_1/A_HxC;
  v_a2 = (port_ETA.m_flow)/rho_2/A_HxC;

  // ************ evaluation of the reynoldnumber *****************************************************
  Nu_m1.Re = (d_h*v_d1*rho_1)/mu_1;
  Nu_m2.Re = (d_h*v_d2*rho_2)/mu_2;

  // **** Variables assign to the sub-classes
  Nu_m1.Pr = Medium_1.prandtlNumber(MeanState_1);
  Nu_m2.Pr = Medium_2.prandtlNumber(MeanState_2);

  alpha_m1.lambda_m = Medium_1.thermalConductivity(MeanState_1);
  alpha_m2.lambda_m = Medium_2.thermalConductivity(MeanState_2);
  // ************ Properties in the ports ******************

  medium_ODA.p = port_ODA.p;
   medium_ODA.h = actualStream(port_ODA.h_outflow);
//   port_ODA.h_outflow=10000;
  medium_ODA.Xi = actualStream(port_ODA.Xi_outflow);
  medium_SUP.p = port_SUP.p-dp_1;
  medium_SUP.h = actualStream(port_SUP.h_outflow);
  medium_SUP.Xi = actualStream(port_SUP.Xi_outflow);

  medium_ETA.p = port_ETA.p;
   medium_ETA.h = actualStream(port_ETA.h_outflow);
//   port_ETA.h_outflow =50000;
  medium_ETA.Xi = actualStream(port_ETA.Xi_outflow);

  medium_EHA.p = port_EHA.p-dp_2;
  medium_EHA.h = actualStream(port_EHA.h_outflow);
  medium_EHA.Xi = actualStream(port_EHA.Xi_outflow);

  // ************** Definition of the Means states *****************************************************
  MeanState_1 = Medium_1.setState_pTX(
        (port_ODA.p + port_SUP.p)/2,
        (T_ODA.T + T_SUP.T)/2,
        (inStream(port_ODA.Xi_outflow) + inStream(port_SUP.Xi_outflow))
     /2);

  MeanState_2 = Medium_2.setState_pTX(
        (port_ETA.p + port_EHA.p)/2,
        (T_ETA.T + T_EHA.T)/2,
```

```
 (inStream(port_ETA.Xi_outflow) + inStream(port_EHA.Xi_outflow))
     /2);
  // *** Operating characteristics ****
  NTU*W_1 = (UCoeff.U*A_HX);

  (P,) = SubSystems.ThermalOperatingData.P_Valu(
        Flowstreaming,
        NTU,
        R);

  tsi = P/NTU;

  // **** The mean temperatur difference******************************************************************
   dTm = tsi*(T_ODA.T - T_ETA.T);

  // **** Efficiencies factor
  phi_ETA = abs((T_EHA.T - T_ETA.T)/(T_ETA.T - T_ODA.T))*100;
  phi_ODA = abs((T_SUP.T - T_ODA.T)/(T_ETA.T - T_ODA.T))*100;

  // **** Evaluation of the Energy flow******************************************************************
  HeatTrans.Q_flow = (UCoeff.U*A_HX*dTm);

  // **** Evaluation of loss of pressure*****************************************************************

  if Nu_m1.Re<2500 then
    dp_1 = (256*D)/(rho_1*d_h*Nu_m1.Re);
  else
    dp_1 = 0.3164*4*D/((rho_1*d_h)*((Nu_m1.Re)^0.25));
  end if;

   if Nu_m2.Re<2500 then
     dp_2 = (256*D)/(rho_2*d_h*Nu_m2.Re);
  else
    dp_2 = 0.3164*4*D/((rho_2*d_h)*((Nu_m2.Re)^0.25));
   end if;

//    Eff = (HeatTrans.Q_flow/(W_1*(T_ODA.T-T_ETA.T)))*100;

  // Connections ****************************************************************************

  connect(Canal_2.ports[1], port_ETA)
  connect(Canal_2.ports[2], port_EHA)
  connect(T_EHA.port, port_EHA)
  connect(alpha_m1.Nu_m, Nu_m1.Nu_m)
  connect(alpha_m1.alpha_m, UCoeff.alpha_m1)
  connect(alpha_m2.alpha_m, UCoeff.alpha_m2)
  connect(Nu_m2.Nu_m, alpha_m2.Nu_m)
  connect(Canal_2.heatPort, HeatTrans.port_1)
  connect(HeatTrans.port_2, Canal_1.heatPort)
  connect(Canal_1.ports[1], port_SUP)
  connect(T_SUP.port, port_SUP)
  connect(Canal_1.ports[2], port_ODA)
  connect(T_ETA.port, port_ETA)
  connect(T_ODA.port, port_ODA)

end HeatPlateExchanger_N;
```

Listing A.4: Code Rotary Heat Exchange.

```
model RotaryHeatExchanger

 inner Modelica.Fluid.System system
    annotation (Placement(transformation(extent={{92,-100},{100,-92}})));
  import SI = Modelica.SIunits;
  import Modelica.Math;
protected
  parameter SI.MassFlowRate m_flow_min=1 " Smallest Mass flow rate in modell";
                                        //1e-3
    //*********************************************************************// declaration of the fluids

public
  replaceable package Medium_1 =
      Modelica.Media.Interfaces.PartialMedium "outside air"
                 annotation (choicesAllMatching=true);
  replaceable package Medium_2 =
      Modelica.Media.Interfaces.PartialMedium "Indoor air"
                annotation (choicesAllMatching=true);

  parameter Real Flowstreaming=2 annotation (choices(
      choice=1 "Pur coflow",
      choice=2 "Pur cross flow",
      choice=3 "Pur counter flow",
      choice=4 "Cross counterflow"));
  //*****************************************************// Geometrical properties of the heat Exchanger

  /// *********************** Geometric parameters

  parameter SI.Diameter D "disk diameter";

  parameter Real alpha(unit="deg") "disk sector angle";

  parameter SI.Length h_cell "Height of the cell";

  parameter SI.Thickness d_h=2/3*h_cell "Hydraulic diameter 1"
    annotation (Dialog(tab="Geometry"));
      parameter SI.Length H "Height of the rotary heat exchanger";

  parameter SI.Diameter d "hub diameter";

  parameter Real B "percentage of seal face coverage";

  SI.Area A "total matrix surface area";

  SI.Area A_t "total r.h.e surface area";

  parameter Real beta(unit="m2/m3") "packing density";

  parameter Real sigma "porosity";

  parameter SI.SpecificHeatCapacityAtConstantPressure cp_m
    " spec. heat capacity of the matrix";

  parameter SI.Density rho_m=7800 "Density of the Matrix";

Modelica.Blocks.Interfaces.RealInput n1 annotation (Placement(
        transformation(
        extent={{-20,-20},{20,20}},
        rotation=0,
        origin={-106,0})));   // rpm
```

```
  SI.Mass M_m "Matrix mass";

  Real C "dimensionless group";

  SI.HeatFlowRate Q_flowmax;
          D,
          d,
          alpha,
          H) "Volume of the fluid in the duct 1"
    annotation (Dialog(tab="Geometry"));

// *****************//

  parameter SI.Volume V_f1=SubSystems.GeometryData.FluidVolume2(
          D,
          d,
          alpha,
          H) "Volume of the fluid in the duct 1"
    annotation (Dialog(tab="Geometry"));
  parameter SI.Volume V_f2=SubSystems.GeometryData.FluidVolume1(
      D,
      d,
      alpha,
      H) "Volume of the fluid in the duct 2" annotation (Dialog(tab="Geometry"));

  //**************************************************************************************** Physically parameters
public
  parameter SI.Temperature T_ODA0 "Ouside air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));
  parameter SI.Temperature T_SUP0 "Indoor air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));

  parameter SI.Temperature T_ETA0 "Outlet air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));
  parameter SI.Temperature T_EHA0 "Exhaust air temperature at the time t=0 s"
    annotation (Dialog(tab="Initialization"));

//*********************************************************************************///loss of pressure

//   Real dp_1( unit="Pa") "loss of pressure canal1";
//
//   Real dp_2( unit="Pa") "loss of pressure canal2";

   Modelica.SIunits.Efficiency Eff "Efficiency";

   Modelica.SIunits.Efficiency Eff_cf "Ideal Efficiency";

//*****************************************************************************************************

 // The variables

   SI.Velocity v_d1 "Velocity of the Air inside the duct 1";
   SI.Velocity v_d2 "Velocity of the Air inside the duct 2";

   SI.Velocity v_a1 " Approach velocity in the outsidestream cross section";
   SI.Velocity v_a2 " Approach velocity in the Inletstream cross section";

  /// Specific heat capacity of the fluids.
  SI.SpecificHeatCapacity cp_1=Medium_1.specificHeatCapacityCp(
      MeanState_1);
  SI.SpecificHeatCapacity cp_2=Medium_2.specificHeatCapacityCp(
      MeanState_2);
```

```
  Real mu_1=Medium_1.dynamicViscosity(MeanState_1);
  Real mu_2=Medium_1.dynamicViscosity(MeanState_2);

  Real rho_1=Medium_1.density(MeanState_1)
    "Density of the medium 1 at the mean state";
  Real rho_2=Medium_2.density(MeanState_2)
    "Density of the medium 2 at the mean state";

  /// Dimensionless temperature change accords to VDI-Wrmeatlas.
//   Real P;

  Real NTU;

//   Real tsi " Heat grad transfer";

  /// ** Thermal power capacity ***
  Real C_1( unit="W/K")=(port_ODA.m_flow)*(Medium_1.specificHeatCapacityCp(medium_ODA.state)
      + Medium_1.specificHeatCapacityCp(medium_SUP.state))/2;
 Real C_2( unit="W/K")=(port_ETA.m_flow)*(Medium_2.specificHeatCapacityCp(medium_ETA.state)
      + Medium_2.specificHeatCapacityCp(medium_EHA.state))/2;

  Real C_r "matrix wall heat capacity rate";

  /// Ratio of the thermal power capacity
  Real R=C_1/C_2;

//   Real phi_ETA "Degree of efficiency of the outletstream";
//   Real phi_ODA "Degreee of efficiency of the inletstream";
  // *************** Medium Properties in Ports ***************
  Medium_1.BaseProperties medium_ODA(preferredMediumStates=true, T(start=
        T_ODA0, nominal=T_ODA0)) "Medium properties in port_ODA ";
  Medium_1.BaseProperties medium_SUP(preferredMediumStates=true, T(start=
        T_SUP0, nominal=T_SUP0)) "Medium properties in port_SUP ";
  Medium_2.BaseProperties medium_ETA(preferredMediumStates=true, T(start=
        T_ETA0, nominal=T_ETA0)) "Medium properties in port_ETA ";
  Medium_2.BaseProperties medium_EHA(preferredMediumStates=true, T(start=
        T_EHA0, nominal=T_EHA0)) "Medium properties in port_EHA ";

  /// *** Declaration of the Meanstate of the both Fluid ***
  Medium_1.ThermodynamicState MeanState_1
    " Thermodynamic properties of fluid 1 at the mean temperature";
  Medium_2.ThermodynamicState MeanState_2
    " thermodynamic properties of fluid 2 at the mean temperature";
      //// Sub-classes and Ports

  // ****************** Exchangers port **************************************************************
 Modelica.Fluid.Interfaces.FluidPort_a port_ODA(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.MoistAir) "Outside air"
                annotation (Placement(transformation(extent={{80,-60},{100,-40}}),
                    iconTransformation(extent={{72,-82},{94,-60}})));
 Modelica.Fluid.Interfaces.FluidPort_b port_SUP(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.MoistAir) "Inlet air"
              annotation (Placement(transformation(extent={{-104,-60},{-84,-40}}),
                      iconTransformation(extent={{-92,-84},{-70,-62}})));
  Modelica.Fluid.Interfaces.FluidPort_a port_ETA(
                m_flow(start=m_flow_min, nominal=m_flow_min), redeclare
      package Medium = Modelica.Media.Air.MoistAir) "Exit air"
             annotation (Placement(transformation(extent={{-90,68},{-70,88}}),
                 iconTransformation(extent={{-90,66},{-70,88}})));
```

```
Modelica.Fluid.Interfaces.FluidPort_b port_EHA(
              m_flow(
    start=m_flow_min,
    nominal=m_flow_min,
    min=1), redeclare package Medium = Modelica.Media.Air.MoistAir)
  "Exhaust air " annotation (Placement(transformation(extent={{78,36},{98,56}}),
                    iconTransformation(extent={{68,68},{90,90}})));
//*************** Temperature Sensores ***************************************************************

Modelica.Fluid.Sensors.Temperature T_ODA(
              T(start=T_ODA0, nominal=T_ODA0), redeclare package Medium =
      Modelica.Media.Air.MoistAir) "Outside air ( Aussenluft)"
                          annotation (Placement(transformation(
      extent={{-6.5,-5},{6.5,5}},
      rotation=0,
      origin={69,-35.5})));

Modelica.Fluid.Sensors.Temperature T_SUP(
              T(start=T_SUP0, nominal=T_SUP0), redeclare package Medium =
      Modelica.Media.Air.MoistAir) "Inlet air (Zuluft)"
  annotation (Placement(transformation(
      extent={{-4.5,-4},{4.5,4}},
      rotation=0,
      origin={-74.5,-38})));
Modelica.Fluid.Sensors.Temperature T_ETA(
              T(start=T_ETA0, nominal=T_ETA0), redeclare package Medium =
      Modelica.Media.Air.MoistAir) " Exit air (Abluft) "
  annotation (Placement(transformation(
      extent={{-5,-4.5},{5,4.5}},
      rotation=0,
      origin={-62.5,61})));

Modelica.Fluid.Sensors.Temperature T_EHA(
              T(start=T_EHA0, nominal=T_EHA0), redeclare package Medium =
      Modelica.Media.Air.MoistAir) "Exhaust air( Fortluft)"
  annotation (Placement(transformation(extent={{65,60},{78,70}})));
/// **** components to represent the heat exchanger room ********************************************

Modelica.Fluid.Vessels.ClosedVolume Canal_2(
  use_HeatTransfer=true,
  use_T_start=true,
  V=V_f2,
  T_start=T_ETA0,
  redeclare model HeatTransfer =
    Modelica.Fluid.Vessels.BaseClasses.HeatTransfer.IdealHeatTransfer,
  portsData={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
      diameter=d_h),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
      diameter=d_h)},
  use_portsData=false,
  nPorts=3,
  redeclare package Medium = Modelica.Media.Air.MoistAir)
  annotation (Placement(transformation(extent={{-50,46},{-30,66}})));

Modelica.Fluid.Vessels.ClosedVolume Canal_1(
  use_HeatTransfer=true,
  use_T_start=true,
  V=V_f1,
  T_start=T_ODA0,
  redeclare model HeatTransfer =
    Modelica.Fluid.Vessels.BaseClasses.HeatTransfer.IdealHeatTransfer,
  portsData={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
```

```
      diameter=d_h),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
      diameter=d_h)},
    use_portsData=false,
    nPorts=2,
    redeclare package Medium = Modelica.Media.Air.MoistAir)
    annotation (Placement(transformation(extent={{-52,-50},{-32,-
  // **** U-Value; Nu-Number; alpha-Value ************************************************************
  SubSystems.HeatTransfer HeatTrans
    annotation (Placement(transformation(extent={{-62,-2},{-42,18}})));

  SubSystems.ThermalOperatingData.NusseltNumber.NussMean_LevelColumn
    Nu_m1(d_h=d_h, l=H)
    "Mean Nu-number in a level coloumn on the streaming side 1"
    annotation (Placement(transformation(extent={{-50,84},{-38,95}})));
  SubSystems.ThermalOperatingData.CoefficientsofHeatTransfer_alpha
    alpha_m1(l=d_h)
    annotation (Placement(transformation(extent={{-18,88},{-4,100}})));

  SubSystems.ThermalOperatingData.NusseltNumber.NussMean_LevelColumn
    Nu_m2(l=H, d_h=d_h)
    "Mean Nu-number in a level coloumn on the streaming side 2"
    annotation (Placement(transformation(extent={{-40,-96},{-28,-86}})));
  SubSystems.ThermalOperatingData.CoefficientsofHeatTransfer_alpha
    alpha_m2(l=d_h)
    annotation (Placement(transformation(extent={{-12,-94},{0,-82}})));
  SubSystems.ThermalOperatingData.HeatTransferCoefficient_U UCoeff
    annotation (Placement(transformation(extent={{74,-4},{96,20}})));
  SubSystems.ThermalOperatingData.Hot_Side_surface_area hot_Side_surface_area(
      alpha=alpha)
    annotation (Placement(transformation(extent={{26,26},{38,38}})));
  SubSystems.ThermalOperatingData.Cold_Side_surface_area cold_Side_surface_area(
      alpha=alpha)
    annotation (Placement(transformation(extent={{32,-20},{44,-10}})));
  SubSystems.ThermalOperatingData.Total_area total_area(
    D=D,
    d=d,
    H=H,
    beta=beta,
    B=B) annotation (Placement(transformation(extent={{-14,2},{0,16}})));
equation
  // ***** evaluation of the approach velocity v_i ******************************************************
   v_d1 = (2*port_ODA.m_flow)/rho_1/(A_t/2);
   v_d2 = (2*port_ETA.m_flow)/rho_2/(A_t/2);

   v_a1 = (2*port_ODA.m_flow)/rho_1/A_t;
   v_a2 = (2*port_ETA.m_flow)/rho_2/A_t;

  // ************ evaluation of the reynoldnumber ****************************************************
  Nu_m1.Re = (d_h*v_d1*rho_1)/mu_1;
  Nu_m2.Re = (d_h*v_d2*rho_2)/mu_2;

  // **** Variables assign to the sub-classes
  Nu_m1.Pr = Medium_1.prandtlNumber(MeanState_1);
  Nu_m2.Pr = Medium_2.prandtlNumber(MeanState_2);

  alpha_m1.lambda_m = Medium_1.thermalConductivity(MeanState_1);
  alpha_m2.lambda_m = Medium_2.thermalConductivity(MeanState_2);
  // ************ Properties in the ports ******************

  medium_ODA.p = port_ODA.p;
  medium_ODA.h = actualStream(port_ODA.h_outflow);
```

```
 medium_ODA.Xi = actualStream(port_ODA.Xi_outflow);

  medium_SUP.p = port_SUP.p;
  medium_SUP.h = actualStream(port_SUP.h_outflow);
  medium_SUP.Xi = actualStream(port_SUP.Xi_outflow);

  medium_ETA.p = port_ETA.p;
  medium_ETA.h = actualStream(port_ETA.h_outflow);
  medium_ETA.Xi = actualStream(port_ETA.Xi_outflow);

  medium_EHA.p = port_EHA.p;
  medium_EHA.h = actualStream(port_EHA.h_outflow);
 medium_EHA.Xi = actualStream(port_EHA.Xi_outflow);
// ************** Definition of the Means states ******************************************************
  MeanState_1 = Medium_1.setState_pTX(
        (port_ODA.p + port_SUP.p)/2,
        (T_ODA.T + T_SUP.T)/2,
        (inStream(port_ODA.Xi_outflow) + inStream(port_SUP.Xi_outflow))
     /2);

  MeanState_2 = Medium_2.setState_pTX(
        (port_ETA.p + port_EHA.p)/2,
        (T_ETA.T + T_EHA.T)/2,
        (inStream(port_ETA.Xi_outflow) + inStream(port_EHA.Xi_outflow))
     /2);
  // *** Operating characteristics ****
     if C_1 > C_2 then
    C_r = M_m*cp_m*n1/60/C_2; // C_r = M_m*cp_m*n1*n/n/60/C_2;
    C = C_2/C_1;
    NTU = (UCoeff.U*A)/C_2;
    Eff_cf = (1-exp(-NTU*(1-C)))/(1-C*exp(-NTU*(1-C)));
    Eff = Eff_cf*(1-1/(9*(C_r/C_2)^1.93));
   else
    C_r = M_m*cp_m*n1/60/C_1; //C_r = M_m*cp_m*n1*n/n/60/C_1;
    C = C_1/C_2;
    NTU = (UCoeff.U*A)/C_1;
    Eff_cf = (1-exp(-NTU*(1-C)))/(1-C*exp(-NTU*(1-C)));
    Eff = Eff_cf*(1-1/(9*(C_r/C_1)^1.93));
  end if;
// **** total matrix surface area***********************************************************************
  A_t = Modelica.Constants.pi/4*(D^2-d^2)*(1-B);

  A = Modelica.Constants.pi/4*(D^2-d^2)*H*beta*(1-B);

  M_m = Modelica.Constants.pi/4*(D^2-d^2)*H*rho_m*(1-sigma);

  Q_flowmax = (UCoeff.U*A*(T_ODA.T-T_ETA.T));

  HeatTrans.Q_flow = Q_flowmax*Eff;

  // Connections **************************************************************************************

  connect(Canal_2.heatPort, HeatTrans.port_1) ;
  connect(HeatTrans.port_2, Canal_1.heatPort) ;
  connect(port_SUP, port_SUP);
  connect(Canal_1.ports[1], port_SUP);
  connect(T_SUP.port, port_SUP) ;
  connect(port_ETA, Canal_2.ports[1]) ;
  connect(T_ETA.port, Canal_2.ports[2]) ;
  connect(T_EHA.port, port_EHA) ;
  connect(Nu_m2.Nu_m, alpha_m2.Nu_m) ;
```

```
connect(Nu_m1.Nu_m, alpha_m1.Nu_m) ;
  connect(alpha_m2.alpha_m, UCoeff.alpha_m2) ;
  connect(alpha_m1.alpha_m, UCoeff.alpha_m1) ;
  connect(hot_Side_surface_area.A_h, UCoeff.A_h) ;
  connect(cold_Side_surface_area.A_c, UCoeff.A_c) ;
  connect(total_area.A, hot_Side_surface_area.A) ;
  connect(total_area.A, cold_Side_surface_area.A) ;
  connect(total_area.A, UCoeff.A) ;
  connect(T_ODA.port, port_ODA) ;
  connect(Canal_1.ports[2], port_ODA) ;
  connect(Canal_2.ports[3], port_EHA);
  annotation (Diagram(coordinateSystem(preserveAspectRatio=false, extent={{-100,
          -100},{100,100}}), graphics), Icon(coordinateSystem(extent={{-100,
          -100},{100,100}},
                      preserveAspectRatio=false), graphics={Ellipse(
        extent={{50,48},{-46,-50}},
        lineColor={0,0,255},
        fillColor={255,85,85},
        fillPattern=FillPattern.Solid),                Ellipse(
        extent={{28,36},{-68,-62}},
        lineColor={0,0,255},
        fillColor={255,85,85},
        fillPattern=FillPattern.Solid),
      Line(
        points={{-22,36},{-22,-16},{6,-54}},
        color={0,0,255},
        smooth=Smooth.None)}));
end RotaryHeatExchanger;
```

Declaration of Originality

I hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources. This thesis has not been submitted as exam work in neither the same nor a similar form. I agree that this thesis may be stored in the institutes library and database. This work may also be copied for internal use.

Aachen,Padova July 5, 2016

Caterina Toninato