ISSN 0280–5316
ISRN LUTFD2/TFRT--5584--SE

Modelling and Control of a Plate Heat Exchanger in Steam Applications

Ola Löfgren
Patrik Svensson

Department of Automatic Control
Lund Institute of Technology
October 1997

Department of Automatic Control
Lund Institute of Technology
Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
October 1997

Document Number
ISRN LUTFD2/TFRT--5584--SE

Author(s)
Ola Löfgren and Patrik Svensson

Supervisor
Jonas Eborn, Tore Hägglund
and Stefan Burg (Alfa Laval Thermal AB)

Sponsoring organisation

Title and subtitle
Modelling and Control of a Plate Heat Exchanger in Steam Applications

Abstract

The plate heat exchanger (PHE) has many advantages compared to other heat exchangers. The PHE has low space requirements due to the compact construction. It provides a high efficiency and it can be used in a broad range of applications.The industry has shown an interest for the PHE due to the many advantages and this is the reason for this thesis. In the industry a common heat source is steam which is used in various processes. For this application traditionally other heat exchangers than the PHE has been used, but with the PHE a low cost solution can be provided.
The main part of this thesis consists of the modelling in Omola of a PHE with steam as a heating media. The model is validated and is applied to the control problem.
The control problem is approached in two ways, one is to tune a PID from experimental data and the second is to do the control design based on simulated data.
The thesis shows that it is possible to use the simulation model to tune controllers.

Key words
Plate heat exchanger modelling, Steam applications, PID-tuning

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
94

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, S-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@uub2.lu.se

Contents

Preface 5
Acknowledgments 5

1. Introduction 6
1.1 Background 6
1.2 Work Strategy 6
1.3 Thesis Outline 7

2. Model 8
2.1 About Omola/OmSim 8
2.2 K2 Model Database 9
2.3 Plate Heat Exchanger Model 9
2.4 Plate Heat Exchanger Model with three Water Compartments 14
2.5 Steam Trap Model 14
2.6 Valve Models 16
2.7 Simulation Model 18

3. Manual 21
3.1 Simulation Setup 21

4. Validation 24
4.1 Test Rig Setup 24
4.2 Validation Experiments 26
4.3 Differences and Error Sources 29

5. Controller Tuning 30
5.1 About the Process 30
5.2 Tuning based on Measurements 31
5.3 Test of Control Parameters designed from Experimental Data . 34
5.4 Tuning based on Simulation 35
5.5 Test of Control Parameters designed from Simulated Data . . . 37

6. Conclusions 39
6.1 Model 39
6.2 Tuning Methods 39
6.3 Future Work 39

7. References 41

A. Thermodynamics and Heat Exchanger Theory 42
A.1 Mass and Energy Balance 42
A.2 Heat Exchanger with 1-phase Fluids 43
A.3 A Simple Model of a Plate Heat Exchanger (one plate) 44
A.4 Some Key Coefficients 46
A.5 Physical Properties of the Plate Heat Exchanger 48
A.6 Steam as a Media 50
A.7 Heat Transfer from Steam to Water 53
A.8 Float operated Steam Traps and Intermittent Pumps 54

B. Conceptions and Definitions 56
B.1 Units and their Meaning 56

C. Class Tree Structure 58

D. Omola Code 59
D.1 VVXTest 59

D.2 PlateHeatExchangerFM 62
D.3 PlateHeatExchanger3ZFM 66
D.4 SteamTrapFM 72
D.5 PlateSteamCompartmentFM 73
D.6 PlateWaterCompartmentFM 75
D.7 PlateCompartmentIC 77
D.8 PlateFlashCompartmentFM 78
D.9 PlateTrapCompartmentIC 79
D.10 PlateTrapCompartmentFM 80
D.11 PlateTrapHeatResistorIC 83
D.12 PlateTrapHeatResistorFM 83
D.13 PlateHeatResistorIC 84
D.14 PlateHeatResistorFM 85
D.15 SteamValveFM 86
D.16 SteamValveCFM 87
D.17 KE73kvs100FM 87
D.18 KE31kvs25FM 88
D.19 KE73kvs63FM 89
D.20 PlateSteamFlowResistorFM 90
D.21 PlateFlowResistorFM 91
D.22 PlateCondFlowResistorFM 92
D.23 PlateFlashFlowResistorFM 93
D.24 WaterValveCFM 93
D.25 PlateHeatTransferInTC 94
D.26 PlateHeatRateInTC 94

Preface

This report presents the Master thesis "Modelling and Control of a Plate Heat Exchanger in Steam Applications". It contains modelling of a plate heat exchanger in Omola and control strategies of the same. The model is made with a steam supply on the primary side and water on the secondary side. The Master thesis is made for Alfa Laval Thermal AB in Lund in cooperation with Department of Automatic Control at Lund Institute of Technology. The goal is to achieve an effective and generally applicable control strategy for a steam-water Plate Heat Exchanger applications.

Acknowledgments

We would like to show our gratitude to Stefan Burg and Rolf Ekelund at Alfa Laval Thermal AB for the good support in our work and for the lectures concerning theory of thermodynamics and plate heat exchangers which resulted in Appendix A.

We would also like to thank Jonas Eborn and Tore Hägglund at the Department of Automatic Control for their support and guidance throughout the work. Jonas Eborn has been an injection of knowledge and a good help in the work with Omola/OmSim.

To the rest of the staff at Department of Automatic Control we would like to say thank you all for the nice response and helping hands in our work and smiling faces at the coffee breaks.

1. Introduction

1.1 Background

Alfa Laval Thermal has a long history of manufacture and development of heat exchangers. One of the main products is the plate heat exchanger of which Alfa Laval Thermal is the world leading producer. The plate heat exchanger (PHE) is used in many applications, e.g. in the dairy and chemical industry. Originally the PHE was designed for the dairy industry, where the sanitary demands early were set to a high standard. The industry had the need for an easy to clean heat exchanger and this was provided by the simple construction of the PHE which is easy to disassemble.

The PHE has many advantages compared to other heat exchangers. The PHE has low space requirements due to the compact construction. It provides a high efficiency and it can be used in a broad range of applications. The price is low due to the simple construction. Alfa Laval offers a full range of choices in plate materials and gasket materials. The usual plate material is stainless steel but for hostile duties plates are available in exotic materials. The plates can also be brazed together, making the gasket unnecessary. This kind of PHE is used for lower pressure applications. Also a combination of brazed and ordinary gasket plates are available.

A disadvantage with the PHE is that it has limits concerning pressure and is therefore not usable for high pressure applications. In those cases other heat exchangers are preferable, e.g. tubular heat exchangers.

The industry has shown an interest for the PHE due to the many advantages and this is the reason for this thesis. In the industry a common heat source is steam, which is used in various processes. For these applications traditionally other heat exchangers than the PHE has been used, but with the PHE a low cost solution can be provided.

1.2 Work Strategy

Since the PID-controller is the most common controller in industry it is used in the control strategies in this thesis. The main goal of this thesis is to develop a general method of tuning the PID by means of the heat exchanger physical data. The quantity to be controlled is the temperature of, in this case, the water flowing out of the secondary side of the PHE. The thesis will focus on one control strategy, the control of the steam flow into the primary side of the PHE.

The major part of this work has been to develop a physical model of the PHE in Omola/OmSim [1, 2]. The advantage of using a simulation model in the first steps when investigating the behaviour of the system for different control parameters, is the lower time requirements. It would be very time consuming to test all our ideas on a real application.

Alfa Laval has provided a test rig with possibility to try out the control strategy. This rig will be used for validation of the Omola model and as a final step in the choice of control parameters.

1.3 Thesis Outline

This thesis is divided into 6 chapters. Chapter 2 contains a introduction to Omola/OmSim. Also the simulation model of the plate heat exchanger is presented. A model of a float operated steam trap and some valves are also defined in this chapter. The components are then combined together into a simulation model.

A short manual to the simulation model is presented in Chapter 3. This is a complement to the Omola/OmSim tutorial [1]. The parameters of interest are also explained shortly in this chapter.

In Chapter 4 the validation of the model is conducted.

Chapter 5 treats the aspect of nonlinearities of the open loop process that is important to take into consideration when the control design of the system is made. The chapter also treats two control design strategies. The first one is to make a control design from experimental data of a PHE system. The second is to use the simulation model to conduct the design.

Chapter 6 gives concluding remarks and suggests further work to be done as a continuation of this thesis.

2. Model

The plate heat exchanger (PHE) this thesis will focus on is a brazed plate heat exchanger. The physical properties of this PHE is the base for the model made in Omola. In cases when the PHE is used as a water to water heat exchanger, Alfa Laval has good knowledge of the behaviour of the unit concerning e.g. the pressure drop on each side of the unit and the heat transfer from one side to the other. The properties are measured and identified in these cases. Steam however complicates the situation. It is very compressible and when it is condensing the pressure can drop rapidly. Steam has dynamic properties which make calculations even more difficult. During transients it can be subcooled or superheated. To make a simulation model of the pressure dynamics on the primary side is therefore a challenge of its own.

The thermodynamical theory behind the models can be found in Appendix A. The foundation of the models are energy and mass flow balances. These balances are implemented in the classes of the model database K2 [3].

2.1 About Omola/OmSim

Omola (Object oriented Modelling Language) is developed by Department of Automatic Control at Lund Institute of Technology. It provides a tool for modelling dynamical continuous and/or discrete systems in a structured way. It supports the building of model data bases. Variables can be given units according to the SI-system. It has all the advantages of object oriented programming with single inheritance. This means that one could for example divide an object into an interface class instance and one class instance containing the relations within the object. Omola is a good way of describing dynamical models even if they are not intended for simulation, since the structure is easy to follow. Omola-files intended to be run in OmSim have the file extension `.om`.

OmSim is a simulator environment in which Omola models can be graphically combined (*Med-window*) using graphical terminals. The fundamental principles for this kind of modelling can be found in [1]. The simulator (*Simulator-window*) gives access to all variables and parameters etc. Parameters can be altered by the user during simulation, if allowed by the Omola-code. Plots can be made of any quantity. OmSim does a unit check on all variables connected via the graphical terminals. A variety of numerical solvers for the simulation of the system can be chosen and the parameters, e.g. step length, of these can be altered. The simulator requires that some initial values are given to the dynamical states and that all parameters used are set to some values. This can be made already in the Omola-code by using the declaration `default`,assigning a default value, for parameters and `initial` for variables.

OmSim further provides a way to write macros creating a, by the programmer, specified simulator environment e.g. with variables and parameters set to values and plots connected to any of these. This is a fast way to initialize a system to a certain state. Macro-files of this kind are called *OCL-files (Omola command language-files)* and have the extension `.ocl`. In the OmSim main window under the menu `File` the OCL-files can be selected (`RunOCL`).

2.2 K2 Model Database

We have chosen to use the thermo dynamical data base K2 [3] in Omola which is a result of a research project at the Department of Automatic Control, in which Jonas Eborn has been involved. This data base includes a set of primitives which can be used to model thermodynamic systems, such as compartments, flow resistors and pumps. Also more complicated units can be modeled, e.g. heat exchangers and boilers. In our modelling we have used some of the primitives as they are and altered some to suit our specific application. All classes with names beginning with `Plate` are new or redefined.

The class names in the database can end with one of the letter combinations given by Table 2.1, indicating the function of the class and appearance in the class tree.

suffix	*description*
`FM`	indicate that the model is a *Full Model* i.e. can be used as a stand alone building block.
`IC`	is a class describing the communication between the instance and other instances. IC stands for *Interface Class*
`TC`	ends all classes defining a *Terminal Connection.*

Table 2.1 Class name suffices

2.3 Plate Heat Exchanger Model

The model is divided into three main parts, the primary side and the secondary side of the heat exchanger and the heat transfer between them as a third part (see Figure 2.1). The interaction part is called a *heat resistor* and it describes the heat transfer from one side to the other. It takes into consideration the heat resistance in the fluids and in the wall which separates them. The model of the primary and secondary side describes the media that flow in each side and the physical properties of the PHE, such as volumes and flow resistances.

Primary side

The primary side is the more complicated one, due to the complex steam dynamics. The volume is divided into two parts, one modelling the inlet where condensation has not yet started and one modelling the volume between the plates. In between the two compartments the entire pressure drop on the primary side is modeled, down to the outlet which is placed after the second compartment. In the second compartment all heat transfer on the primary side is taking place and thereby the condensation will also be modeled here. The extent of the heat flow is given by the heat flow resistor. The primary side consists of six instances (see Figure 2.2) which are described below.

InH is an instance of class `FlowInTC`. InH is the inlet terminal of the primary side of the PHE model. It contains the quantities mass flow (w), pressure (p), enthalpy (h) and a media description (M) of the phase (q) and the temperature (T) of the fluid. The terminal also gives the flow direction into the PHE.

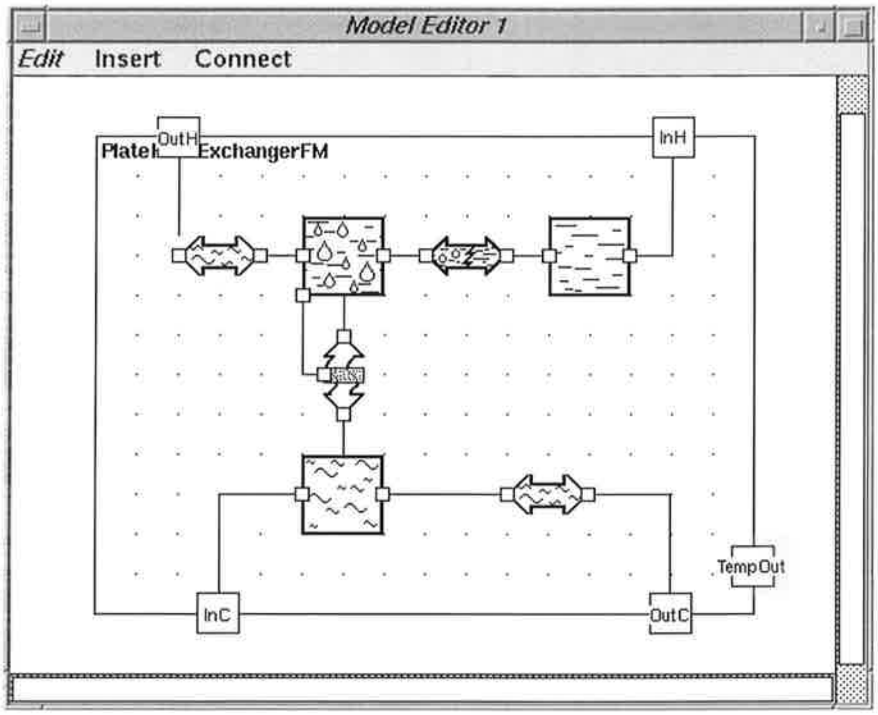


Figure 2.1 The plate heat exchanger model structure in Omola/OmSim.

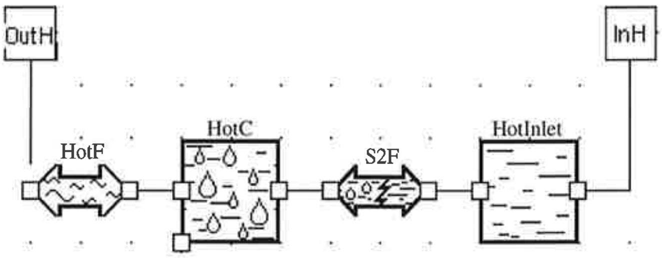


Figure 2.2 The primary side.

OutH is an instance of class `FlowOutTC`. OutH is the outlet terminal of the primary side of the PHE model. Its quantities is the same as for InH with the exception for the media description (M) which instead of temperature contains the height (z) above a reference level. The terminal also gives the flow direction out from the PHE.

HotInlet is an instance of class `PlateSteamCompartmentFM`. The class models the compartment with dynamic pressure and enthalpy balances. This instance correspond to the volume of the inlet. The terminals of the instance contains the same quantities as InH.

HotC is an instance of class `PlateFlashCompartmentFM`. In this instance the volume between the plates on the primary side is described. It contains aspects of heat flow, condensation and pressure dynamics. The condensation alters the effective heat transfer area as the level of condensate in the PHE can change. The condensate contributes very little to the heat transferred to the secondary side. The heat flowing from the condensate is approximated to be zero.

S2F is an instance of class `PlateSteamFlowResistorFM`. This is where the pressure drop takes place over the PHE. The steam saturation temperature (depends on the pressure) in HotInlet is fed forward into HotC. This temperature is used when calculating the *logarithmic mean temperature difference* ($LMTD$ or ΔT_m)(see Appendix equation (A.18)).

HotF is an instance of class `PlateFlashFlowResistorFM`. HotF models the flow resistance after the PHE.

The quantities of `InH` is connected to `HotInlet` via the in-terminal on this instance. `InH` must be preceded by a flow resistance in order to get a specified mass flow into the PHE-model. As mentioned before `HotInlet` is modelling the inlet on the PHE before the plates. The fluid is in steam phase in this compartment. The temperature t_{in1} in the calculation of LMTD (A.18) is the saturation temperature of the fluid in `HotInlet`. Since this temperature is connected to the heat flow resistor `HeatF` it has to be fed forward trough `S2F`.

`S2F` in the model correspond to the pressure drop over the plates and the inlet of the primary side. This introduces changes in the steam saturation temperature, enthalpy and absolute pressure.

`HotC` is connected to `HeatF` via two terminals. One of these terminals is of invisible type, i.e. the terminal is not visible in the graphical editor. This terminal specifies the heat flow and temperature out of `HotC`. The terminal has a positive direction into the compartment, i.e. heat flowing into the compartment is a positive quantity. The second terminal is containing the temperature fed forward from `HotInlet` and the effective heat transfer area. The fluid in `HotC` is a mix of condensate and steam called *flash*. In the model the condensate and the steam are not treated as a homogeneous mix, but as two separate medias. To determine the mixture flowing out at the outlet of the primary side a function has been made depending on the condensate level. As the level of condensate in the PHE-model is above 20% of the height of the PHE only water will flow out of the compartment. At levels below 20% a mix begins to flow. The transition is soft from water to the full homogeneous mix in the compartment. The mix flowing out of `HotC` is determined by the enthalpy of the fluid at the flow out terminal in the compartment model. To make the model evacuate the condensate as the PHE does when the pressure in it is above atmosphere pressure (minus the water column in the pipes succeeding the PHE), the value of the level variable also is weighted with a smooth function. The level is repressed when the pressure is high, increasing the effective heat transfer area.

Finally `HotF` specifies the final pressure drop on the primary side corresponding to the outlet. If there is a column of water due to a difference in height, this is included in the pressure drop calculations. `HotF` is connected to `OutH` and this ends the primary side of the PHE. To avoid algebraic loops objects connected to `OutH` should not be of the same kind as a flow resistor.

Secondary side

This part of the model describes the water side of the PHE. The modelling gets less complicated on this side since water is almost incompressible. The dynamic behaviour of water as a fluid is well known and has been thoroughly investigated. The secondary side consists of four instances (see Figure 2.3) which are described below.

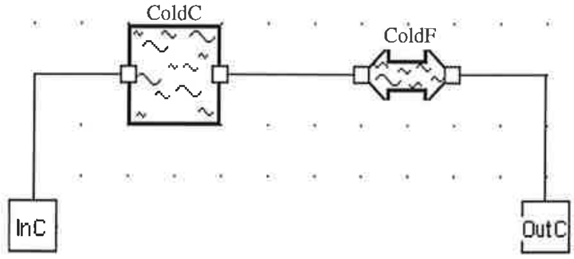


Figure 2.3 The secondary side.

InC is an instance of class `FlowInTC`. InC is the inlet terminal of the secondary side of the PHE model. It contains the quantities mass flow (w), pressure (p), enthalpy (h) and a media description (M) of the phase (q) and the height (z) above a reference level. The terminal also gives the flow direction into the PHE.

OutC is an instance of class `FlowOutTC`. OutC is the outlet terminal of the secondary side of the PHE model. Its quantities is the same as for InC. The terminal also gives the flow direction out from the PHE.

ColdC is an instance of class `WaterCompartmentFM`. ColdC is the heat transfer component of the secondary side. It represents the entire volume of the cold part of the PHE, including the inlet and outlet.

ColdF is an instance of class `PlateFlowResistorFM`. The pressure drop on the secondary side is modeled in this component.

The quantities of `InC` is connected to `ColdC` via the in-terminal on this instance. `InC` must be preceded by a flow resistance in order to get a specified mass flow into the PHE-model. `ColdC` is connected to `HeatF` via one terminal. The terminal is of invisible type and contains the temperatures of the flows at the inlet and outlet of the compartment, and the heat flow. As for `HotC` the heat flow has the positive direction into the compartment. The out-terminal of `ColdC` is connected to the flow resistor `ColdF`. The flow resistor models the entire pressure drop of the secondary side, according to the characteristics given by Alfa Laval Thermal for the PHE used. The flow resistor is then connected to `OutC`.

All terminals on the secondary side, except the heat flow terminal, contains the quantities mass flow (w), pressure (p), enthalpy (h), a media description (M) of the phase (q) and the height (z) above a reference level and finally a flow orientation.

Heat Flow Resistor

HeatF is an instance of class `PlateHeatResistorFM`. It represents the walls between the two sides of the PHE. In this instance the heat flow is calculated according to equation (A.17). The calculation takes into consideration the heat transfer area, wall mean temperature (affects the λ-value), the thickness of the walls and mass flow trough the secondary side (affects the α_{water}-value). The k-value is a measure of the heat conductance and is calculated according to equation (2.1).

$$\frac{1}{k} = \frac{1}{\alpha_{steam}} + \frac{1}{\alpha_{water}} + \frac{\delta}{\lambda} \tag{2.1}$$

The *heat transfer coefficient* for steam, α_{steam}, is modeled according to equation (2.2) and describes the heat conductivity of the steam and the film of condensate on the wall in the primary side of the PHE [7].

$$\alpha_{steam} = 1.06 \left[\lambda_c^3 \rho_c^2 g \frac{r_s}{\eta_c H \cdot DT}\right]^{0.25} \cdot \left(\frac{H}{d_H}\right)^{0.134} \tag{2.2}$$

where H is the plate height of the PHE, d_H is the *hydraulic diameter* (see Appendix A.4.), r_s is the *latent heat* and $DT = T_{steam_{in}} - T_{wall}$.In the last equation T_{wall} denotes the mean temperature of the wall.

In equation (2.1) α_{water} is the heat transfer coefficient for water on the secondary side of the PHE. This quantity is modeled according to equation (2.3)

$$\alpha_{water} = Nu_w \frac{\lambda_w}{d_H} \tag{2.3}$$

where Nu_w is given by equation (2.4), Re_w and Pr_w are given by equations (2.5) and (2.6), d_H is the hydraulic diameter and λ_w is the *heat conduction coefficient* for water, which can be looked up in tables.

$$Nu_w = C_{Re} \cdot Re_w^{n_{Re}} \cdot Pr_w^{n_{Pr}} \tag{2.4}$$

The constants C_{Re}, n_{Re} and n_{Pr} have to be calculated from measurements on the specific PHE used.

$$Re_w = \frac{m_w}{n_w} \cdot \frac{d_H}{\eta_w \cdot A} \tag{2.5}$$

In equation (2.5) m_w denotes the mass flow through all the channels on the secondary side of the PHE, n_w is the number of channels on the secondary side, η_w is the *viscosity* for water and A is the channel inlet area.

$$Pr_w = c_{pw} \frac{\eta_w}{\lambda_w} \tag{2.6}$$

The variable c_{pw} denotes the *specific heat capacity* for water.

How to connect the PHE

In Figure 2.1 the four terminal connections `InH`, `OutH`, `InC`, `OutC` are shown. The In-terminals require a flow given by the preceding components, e.g. a valve or an end flow terminal. The media flowing into `InH` must be steam and the media flowing into `InC` have to be water. The Out-terminals need a pressure given by the succeeding components, and the media out of the terminals is water. `TempOut` is a measurement terminal that contains the temperature of the media flowing out of the secondary side of the PHE.

2.4 Plate Heat Exchanger Model with three Water Compartments

When precision calculations are made on the heat transfer in a PHE, the decreasing temperature on the primary side and the increasing temperature on the secondary side has to be taken into consideration. In fact, the heat flow is described by a partial differential equation. As OmSim does not support such equations, it is necessary to discretize these. The heat flow in the first model of the PHE acts on the entire volume of the secondary side. This can result in an inertia of the temperature rise. To overcome this, and make a more correct heat transfer description, a model is made with three water compartments describing the secondary side. The compartment on the primary side involved in the heat transfer, `PlateFlashCompartmentFM`, is divided into three zones, each interacting via a heat resistor to one of the three water compartments (see Figure 2.4). The equations for the heat flow is the same as for the original PHE model. The difference is that the temperatures in and out differs for each zone respectively water compartment. On the primary side the media gets cooler the further down the temperature is measured due to the pressure drop introduced by the heat flow. On the secondary side the temperature rises when moving upwards (the flows are countercurrent, see Appendix A).

In the model the pressure drop is described as linear from the inlet of the second compartment down to the condensate level surface (this pressure is calculated as the pressure at the outlet of the compartment minus the pressure introduced by the water column). The temperatures are then given by steam table calculations for saturated steam. Each zone transfers heat to the secondary side via one third of the PHE area. If the level of condensate rises above the top of a zone, the heat transfer area for the zone will be zero and no heat will be exchanged. The level will now affect the heat transfer area in the zone above in the same way. The only change on the secondary side regarding the pressure dynamics is that the three water flow resistors each have a third of the original pressure drop given by the data sheets.

When the model was used in simulation the performance of the simulator program was reduced and the gain in calculation accuracy was not in proportion with the time required to conduct a simulation. Because of this the model is not used further in this thesis.

2.5 Steam Trap Model

A steam trap is used to prevent the steam in a system to escape. The function of a steam trap is further explained in Appendix A.

The model consists of five components. These are described below and can be found in Figure 2.5.

TrapIn is an instance of class `FlowInTC`. This is the inlet terminal to the steam trap. It contains the quantities mass flow (w), pressure (p), enthalpy (h) and a media description (M) of the phase (q) and the height (z) above a reference level. The terminal also gives the flow direction into the trap. It is necessary to have a flow given by the preceding components into this terminal.

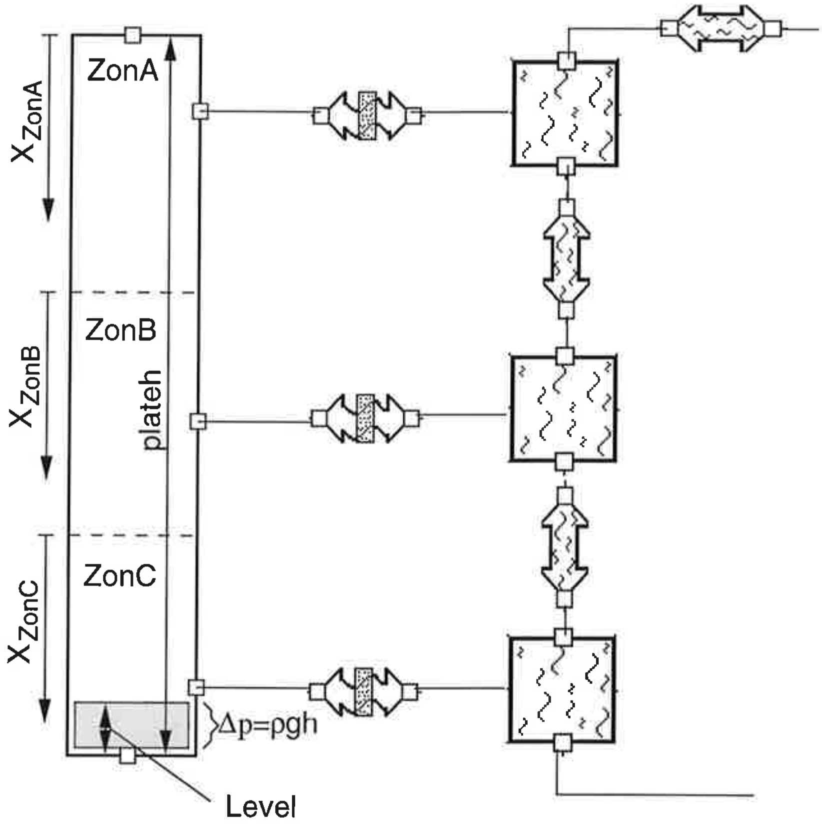


Figure 2.4 The PHE model with three water compartments on the second side and the flash-compartment on the primary side divided into three zones.

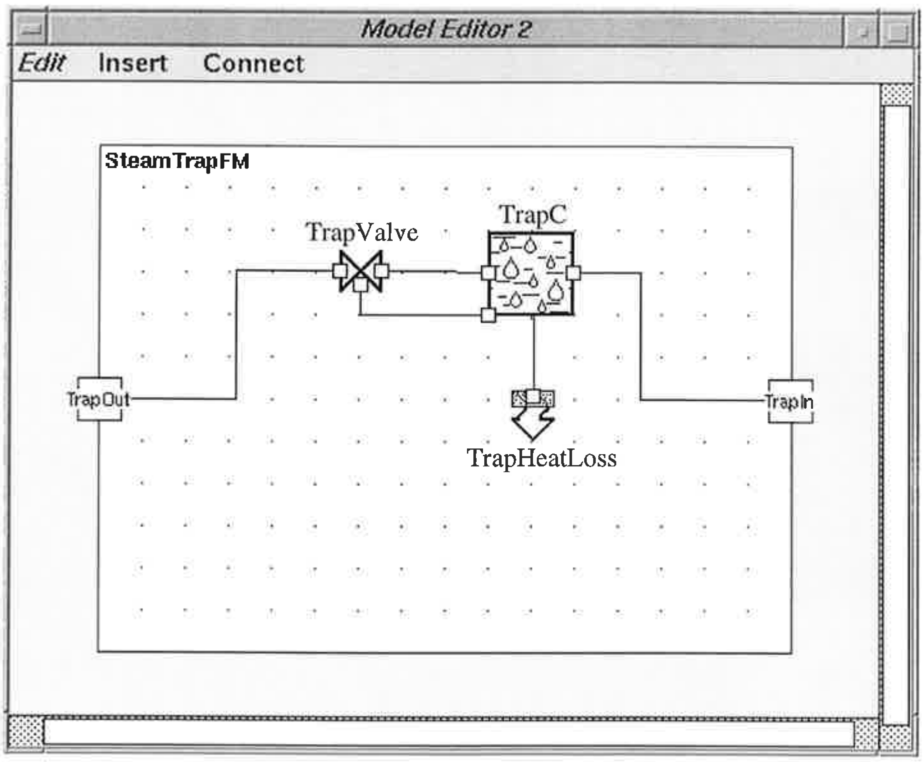


Figure 2.5 The Steam Trap Model.

TrapOut is an instance of class `FlowOutTC`. Its quantities is the same as for TrapIn. The terminal also gives the flow direction out from the steam trap.

TrapC is an instance of class `PlateTrapCompartmentFM`. It represents the entire volume of the steam trap.

TrapValve is an instance of class `WaterValveCFM`. This component models the float operated valve in the steam trap. It is controlled by the level of condensate in the trap compartment.

TrapHeatLoss is an instance of class `PlateTrapHeatResistorFM`. The instance describes the heat transfer to the surrounding air.

The compartment in the steam trap model contains flash. The level in the steam trap controls the opening of the valve succeeding the compartment. The valve is closed until the level has reached half the height of the trap. The valve then opens linearly from 0 % up to 100 % as the float mechanism reaches maximum height when the trap is full. If the pressure after the valve is greater than the pressure in the compartment, a steam trap will work in an erroneous way as air can flow into the system. When this occurs, the valve model sets the flow to zero, i.e. it only allows one flow direction. This is done because of the difficulties of modelling a mix of steam, water and air, where the air works as an insulator in the heat transfer process. One method to overcome the problem is to include a condensate pump before the inlet of the steam trap, as this prevents the outer atmosphere to interfere with the system, i.e. it works as an atmosphere insulator. The trap condenses some of the steam flowing into it, due to the heat transfer with the air surrounding the trap.

2.6 Valve Models

The simulation model contains two types of valves, water valves and steam valves. Both these valves have an nonlinear behaviors, witch will be explained further in this Section.

Several different valve models have been made to make it easier for the user to change valve in the simulation model. A valve name ending with `CFM`, means that it is controllable i.e. it is possible to connect a (PID) controller to this valve. Table 2.2 shows the valves that have been made.

Two key numbers describe the valve characteristics in data tables, the K_{vs}-value and the rangeability (R). If the rangeability is known a third key number, K_{vr}, can be calculated with the relation

$$R = \frac{K_{vs}}{K_{vr}}$$

The valves in this thesis have a logarithmic behaviour, as equation (2.9) describe. Figure 2.6 shows the relation between the K_v-value, which describes the opening area of the valve, and the valve opening position h for the `Spirax Sarco KE31-kvs25` valve. All valves modeled have the rangeability, R = 50:1, this means that the valve can be controlled down to $\frac{1}{50}$ of its maximum flow with preserved valve characteristics. Figure 2.7 shows the relation between K_{vo}, K_{vr} and the valve leakage at low flows. One can see that K_{vo} and K_{vr} almost have the same value, this approximation is made in the Omola code for all valves. K_{vo} is the lowest theoretical flow through the valve.

Valve Type	*Description*
`WaterValveCFM`	Simple controllable water valve.
`SteamValveFM`	Simple steam valve.
`SteamValveCFM`	As above but controllable.
`KE31kvs25FM`	Controllable steam valve with $K_{vs} = 25$. Spirax Sarco KE31.
`KE73kvs63FM`	Controllable steam valve with $K_{vs} = 63$. Spirax Sarco KE73.
`KE73kvs100FM`	Controllable steam valve with $K_{vs} = 100$. Spirax Sarco KE73.

Table 2.2 Valve Models

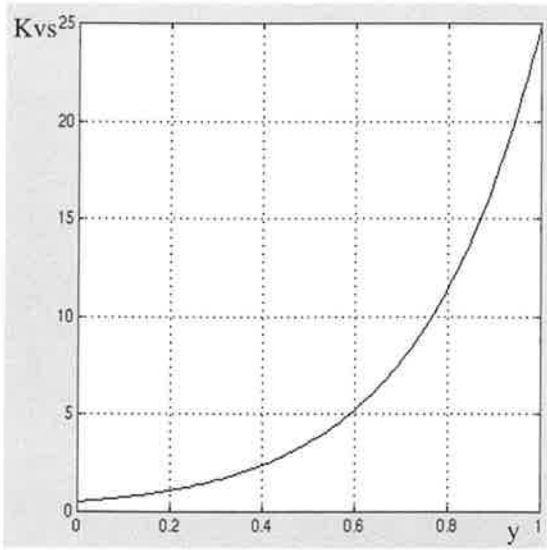


Figure 2.6 Valve characteristics for the `Spirax Sarco KE31-kvs25`

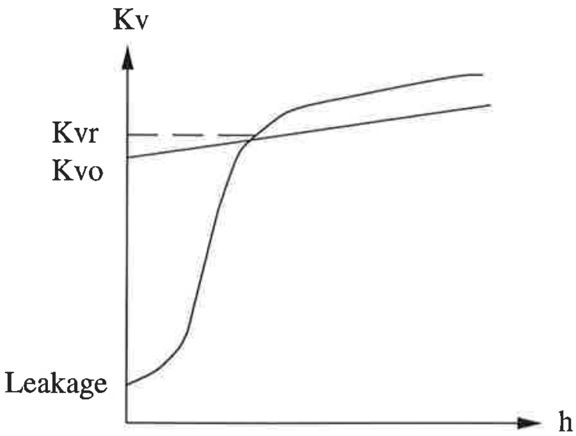


Figure 2.7 The connection between K_{vo}, K_{vr} and the valves leakage.

Valve equations

When a valve is connected to a system there will be a pressure drop over the valve. This pressure drop is proportional to the flow in square.

$$\dot{m} = A_v \cdot \sqrt{\Delta p \cdot \rho} \tag{2.7}$$

$$K_v = A_v \cdot 3.6 \cdot 10^4 \tag{2.8}$$

$$K_v = K_{vo} \cdot e^{(\ln(\frac{K_{vs}}{K_{vo}}) \cdot y)} \tag{2.9}$$

When the equations 2.7 to 2.9 are combined together the resulting equation for the flow will be:

$$\dot{m} = \frac{K_{vo} \cdot e^{(\ln(\frac{K_{vs}}{K_{vo}}) \cdot y)} \cdot \sqrt{\Delta p \cdot \rho}}{3.6 \cdot 10^4} \tag{2.10}$$

where Δp is the pressure drop over the valve, y is the valve opening in percentage, the density for the media is given by ρ, K_v denotes the valve coefficient, A_v is the valve coefficient given in SI-units and this transformation is seen in equation 2.8.

Connecting valves in series and parallel

When two or several valves are connected together the resulting valve coefficient is calculated in different ways. Equation 2.11 is valid if the valves are in series and equation 2.12 if they are in parallel structure.

$$K_v = \sum_{i=1}^{n} K_{v_i} \tag{2.11}$$

$$\frac{1}{K_v^2} = \sum_{i=1}^{n} \frac{1}{K_{v_i}^2} \tag{2.12}$$

In the equations n is the number of valves. More details about valves is given by [6].

2.7 Simulation Model

In this Section the components, previously described, are combined together into a complete simulation model. Figure 2.8 shows how the submodels are connected graphically. The components in the simulation model are

Plateheatexchangerfm4 is an instance of class `PlateHeatExchangerFM`. The class is described in Section 2.3 .

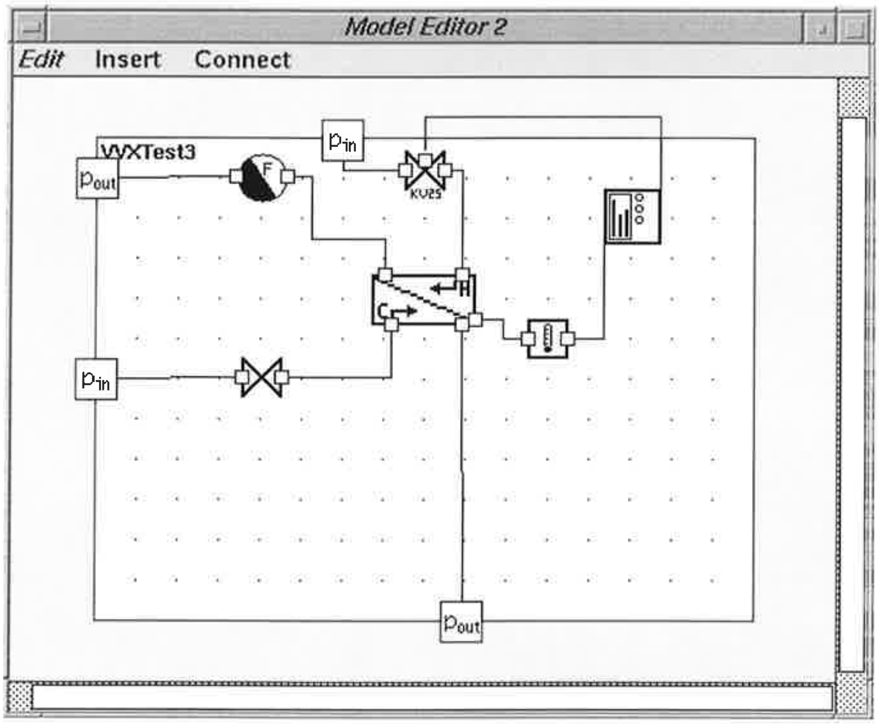


Figure 2.8 The complete simulation model.

SteamTrapFM1 is an instance of class `SteamTrapFM` explained in Section 2.5 .

PID is an instance of class `PIDControllerFM`. This class is predefined in ControlSystemLib. The PID-controller is used to control the steam flow, and thereby the temperature of the flow out on the secondary side of the PHE.

TempSensorFM5 is an instance of class `TempSensorFM`. The definition of this class is found in K2SensorLib in K2. The function of this component is to simulate the time constant of a temperature sensor.

KE31kvs25FM1 is an instance of class `KE31kvs25FM`. The class describes a controllable steam valve. The control signal to the valve is in the range 0 - 1.

WaterValveFM2 is an instance of class `WaterValveFM`. The class definition can be found in the K2 library K2FlowUnitLib.

The simulation model is connected to four end-terminals, which all can be found in the K2 library K2EndTerminalLib. The in-terminal to the steam valve is connected to an end-terminal instance of class `SteamFlowPinTC`. The water valve is connected to an end-terminal instance of class `WaterFlowPinTC`. The two remaining end-terminals are instances of class `WaterFlowPoutTC`. After the pressure and enthalpy have reached the reference values, the end-terminals have a static behaviour.

About the PID Controller

The PID Controller algorithm used in the simulation system PID is described by equation (2.13). It has a setpoint weighting (b) on the proportional part of the control signal, a limited derivative gain and it is equipped with integrator anti-windup. In all it is similar to the PID:s used in the industry. For further theory confer [4] and [5].

$$U = K(bY_{sp} - Y + \frac{1}{sT_i}E + \frac{sT_d}{1 + \frac{sT_d}{N}}E) \tag{2.13}$$

where

$$E = Y_{sp} - Y$$

3. Manual

This chapter will treat the usage of the simulation model. It will describe how to start the simulation, what the simulation macro will show, which parameters that are interesting to change and which variables that are plotted. To get information about how to use OmSim, confer [1].

3.1 Simulation Setup

To make it easier for the user an OCL-file have been made. To start the OCL-macro open the `File`-menu in OmSim main window, select `RunOCL`. Select PHESim in the dialog. PHESim starts a simulation environment to the system described in Section 2.7. It gives the system an initial state and sets all necessary parameters. As a user interface the macro will open seven windows (see Figure 3.1) displaying the simulator, three plots, the parameters and the dynamic states of the system. The last window contains all variables of the system in hierarchically order. After the PHESim has been run, the system is ready for simulation.

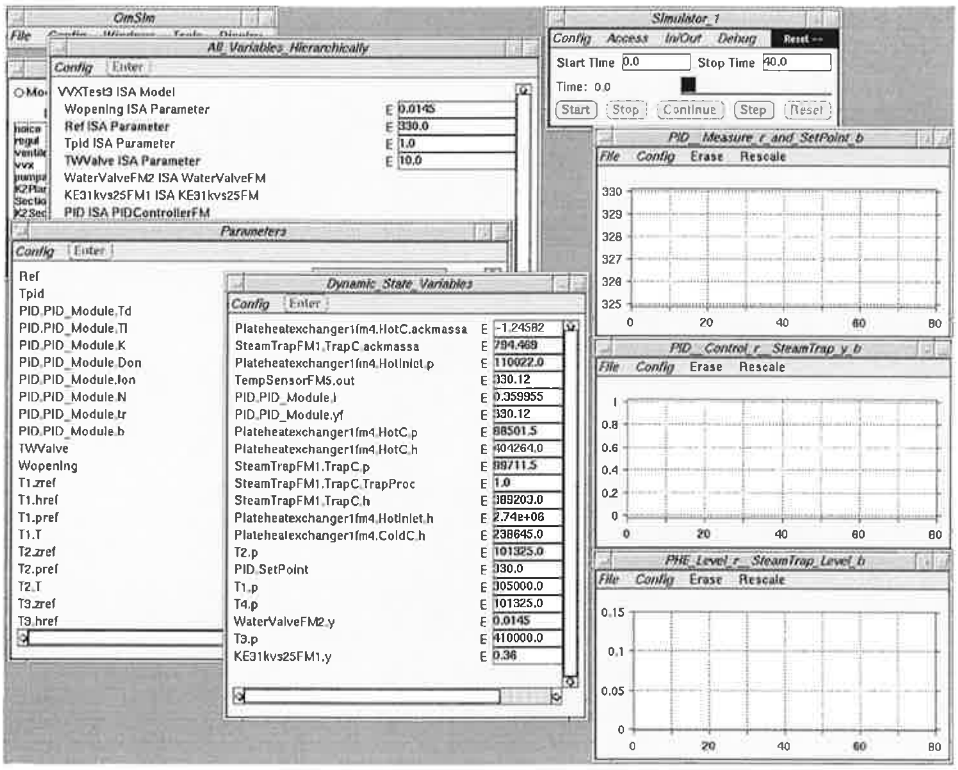


Figure 3.1 The plate heat exchanger simulator system generated by the file PHESim.ocl.

Plots

The upper plot shows the setpoint of the PID controller and the measured value of the temperature on the outlet of the secondary side. The plot in the middle displays the PID control signal, the position of the steam valve and the steam trap valve position. The lower plot shows the condensate levels in the PHE and the steam trap. These have been theoretically derived from the enthalpy and pressure in each of the compartments.

Parameters

The system includes a large set of parameters, defining the main part of the physical quantities of the system, e.g. the compartment volumes, heat transfer areas , rangeability of the steam valve etc. The parameters also define quantities that does not affect the physical properties of the model. These parameters describe the conditions under which the model is to be run, e.g. the setpoint of the PID controller or the water valve opening position. To manage to sort out the parameters of interest when simulating the PHE, Table 3.1 below can be helpful. It contains those parameters that needs to be changed in order to get different kinds of controllers, loads, enthalpies etc. into the system.

Parameter	*Description*
`Ref`	is the PID setpoint reference value. The setpoint value is set by a function with a time constant, in order to limit the derivatives when the reference value is changed.
`Tpid`	is the time constant to the PID setpoint value function.
`PID.PID_Module.Td`	is the derivative time of the PID controller.
`PID.PID_Module.Ti`	is the integral time of the PID controller.
`PID.PID_Module.K`	is the PID controller gain.
`PID.PID_Module.Don`	(Don=0.0) = The derivative part of the PID is off. (Don=1.0) = The derivative part of the PID is on.
`PID.PID_Module.Ion`	(Ion=0.0) = The integrating part of the PID is off. (Ion=1.0) = The integrating part of the PID is on.
`PID.PID_Module.N`	is a factor limiting the derivative gain (see equation (2.13) and [4])
`PID.PID_Module.tr`	is the tracking time constant (used for integrator anti-windup)
`PID.PID_Module.b`	is a setpoint weighting constant for the proportional part of the PID controller algorithm.(see equation (2.13) and [4])
`Wopening`	is the reference value of the water valve position value.
`TWValve`	is the time constant for the water valve position value, used in order to limit the derivatives when the valve position is changed.

Parameter	*Description*
T1.href	is the enthalpy reference value of the steam flowing out of the end-terminal and into the steam valve.
T1.pref	is the pressure reference value at the end-terminal.
T1.T	is the time constant for the enthalpy and pressure values, used in order to limit the derivatives when the values are changed.
T3.href	is the enthalpy reference value of the water flowing out of the end-terminal and into the water valve.
T3.pref	is the pressure reference value at the end-terminal.
T3.T	is the time constant for the enthalpy and pressure values, used in order to limit the derivatives when the values are changed.
SteamTrapFM1.TrapHeatLoss.K	is the value of the k-value for the heat transfer function (see equation(A.17) in appendix A) describing the heat flow from the media in the steam trap to the air outside.
SteamTrapFM1.TrapC.TimeTrap	is the time constant for the trap valve position value, used in order to limit the derivatives when the valve position is changed.
SteamTrapFM1.TrapC.trapheight	is the height of the trap compartment.
TempSensorFM5.T	is the time constant of the temperature sensor.
PID.Limiter.Umax	is the maximum value of the PID control signal.
PID.Limiter.Umin	is the minimum value of the PID control signal.
PID.AutoMan.UMan	is the PID control signal, when the PID is operated in manual mode.
PID.AutoMan.Manual	(Manual=0.0) = The PID controller operates in automatic mode.(Manual=1.0) = The PID controller operates in manual mode.

Table 3.1 Some of the parameters in the simulation model.

4. Validation

To validate the model some comparisons are made between experimental data on the Plate Heat Exchanger and the simulation model. Since the PHE was not connected to a condensate pump, it has to have an operating point with a pressure at the outlet above atmosphere pressure (minus the water column in the pipes succeeding the PHE). Otherwise the PHE will build up condensate and possibly start to stall. This would introduce a mechanical exhaustion on the PHE, eventually leading to a malfunction of the unit. This means that the valve position have to operate in the upper region (e.g. 50% - 100%). The validation is used to verify the model and work as a basis for the tuning of some of the constants, e.g. the valve loss factor function on the steam valve, to make the simulation model work more properly.

4.1 Test Rig Setup

Alfa Laval Thermal AB supplied with a test rig for experimental investigation of the PHE. Figure 4.1 shows the test rig experimental setup.

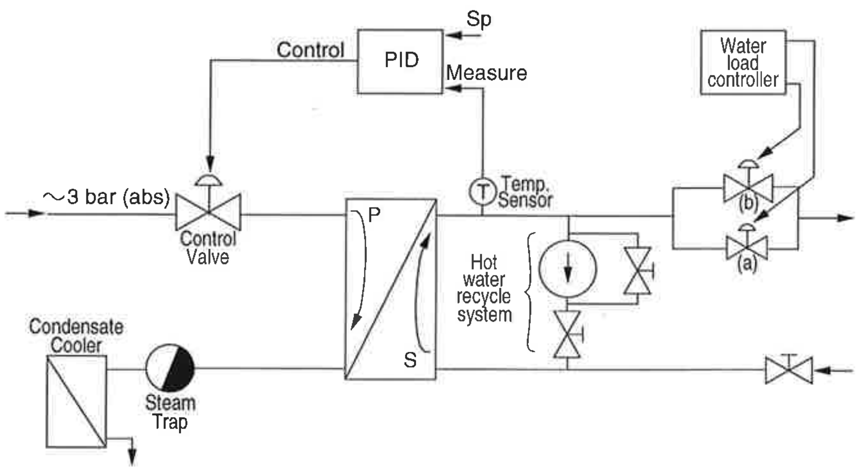


Figure 4.1 The plate heat exchanger system, with the control loop from the outlet temperature on the secondary side to the valve position at the inlet on the primary side.

The flow into the primary side consists of superheated steam at a pressure of about 3.0 bar (abs.). The steam is superheated to prevent mechanical damage to the PHE. The steam flowing in the system has a high velocity and if it was saturated, it would contain small drops of water, with a high kinetic energy. The steam flow is set by a controllable steam valve (`Spirax Sarco KE31 kvs=25.0 DN40`). The PID-controller acts on this valve, controlling the valve position. The quantity measured by the PID is the temperature of the water at the outlet of the secondary side. The outlet on the primary side is connected to a steam trap (float-mechanism, see Appendix A), succeeded by a condensate cooler.

The inlet on the secondary side is connected to the water supply. The tap water can be mixed with recirculated water from the outlet, allowing a rise in the inlet temperature. The recirculation flow is controlled by the pump, the valve parallel to it and the succeeding valve. The circulation pump was not used, due to lack of time. The flow trough the secondary side is controlled by the two valves (see Figure 4.1 (a) and (b)) before the drain and the recirculation system.

Controller

The controller installed in the test rig is a `Spirax Sarco TMC`. The advantage with this unit is that it is prepared for temperature control. The controller has a built in electric drive for a variety of $pt100\Omega$-sensors, depending on what range that is required. The controller also supports other kind of input signals, e.g. 4-20 mA and 0-5 V. The input has a resolution of 30000 counts. The output however has a low resolution of 100 counts (0% - 100%) and this can be a source of distortion. Figure 4.2 is an example of this, taken from a closed loop experiment on the test rig. The low resolution makes the control signal from the PID alternate between two values, introducing a small oscillation on the outlet temperature of the secondary side. It is important to be aware of this if the closed loop system is intended to operate at a static level. The oscillations also results in wear of the valve and actuator.

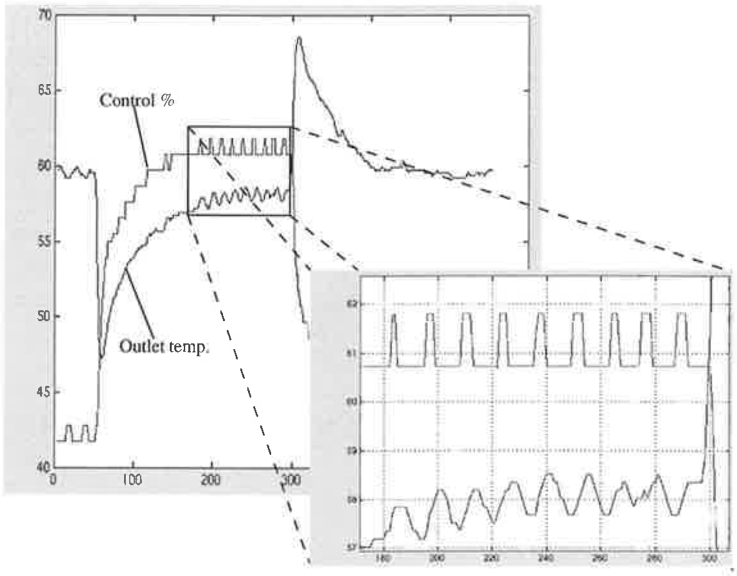


Figure 4.2 Oscillation on the outlet temperature of the secondary side introduced by the control signal, due to the low output resolution.

The `Spirax Sarco TMC` has three parameters defining the control law P_b, T_i and T_d. P_b is an alternative approach to describe the controller gain denoted K in equation (2.13). The idea is to represent both the input and the output in values from 0% up to 100%. The controller gain is calculated as

$$K_{\%} = \frac{100\%}{P_b\%} \tag{4.1}$$

In the test rig case the temperature sensor signal spans from $-99.9°C$ to $400°C$ representing 0% respectively 100%. The control signal to the actuator spans from 4 mA to 20 mA. Figure 4.3 shows the rescale of the input and output.

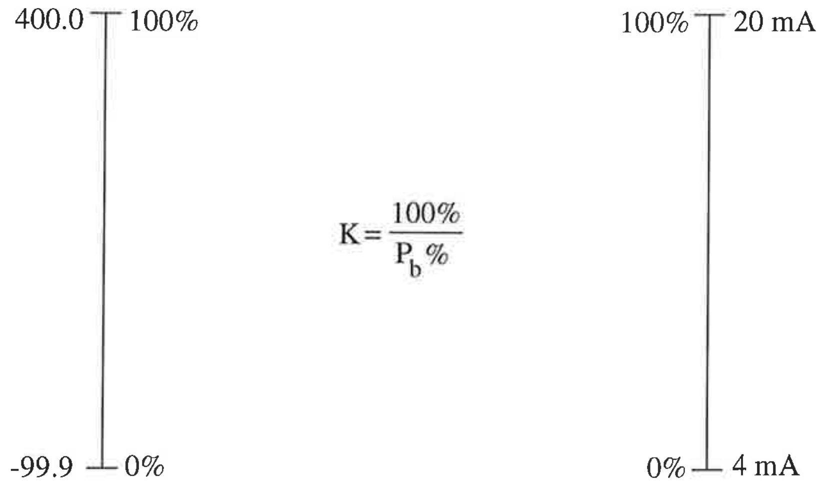


Figure 4.3 Rescale of input signal and output signal

The transform of the controller gain from $K_{\%}$ to a K-value with input signals presented in $^\circ C$ to an output signal represented in the span 0 - 1 is given by equation (4.2)

$$K = \frac{100\%}{P_b\%} \cdot \frac{1}{\frac{400-(-99.9)^\circ C}{100\%} \cdot 100\%} = \frac{0.2}{P_b} \frac{1}{^\circ C} \tag{4.2}$$

4.2 Validation Experiments

The validation experiments are made on the open system, i.e. no control loop active. This is made by setting the PID-controller in manual mode. The PID-controller can now be used to set the valve position value. Then a fixed load is decided, i.e. the flow and the temperature in to the secondary side.

In validation experiment 1 and 2 the operating point is selected to a region where the process works properly, with suitable pressures on the primary side. When the process is stable, steps are made and the response of the open system is recorded by a data logger.

In validation experiment 3 the heat transferred as a function of the valve position was investigated.

When the data necessary has been collected, a file compatible with OmSim is made. The file contains relative time, valve position value and the recorded temperature. The valve position data is fed into the simulation model and the temperature output of the model is compared with the recorded data of the PHE. Below the validation experiments are presented.

Validation experiment 1

The first open loop experiment was made with a medium load. The flow on the secondary side was set to 0.4 l/s and the water temperature into the PHE was 14$^\circ C$. The valve position was set to 50%. When the system was stable the valve position was altered to 60%, 70%, 60% and 50%. In between the system was allowed to settle. Figure 4.4 shows the recorded outlet temperature of the secondary side, the simulated outlet temperature of the same side and the

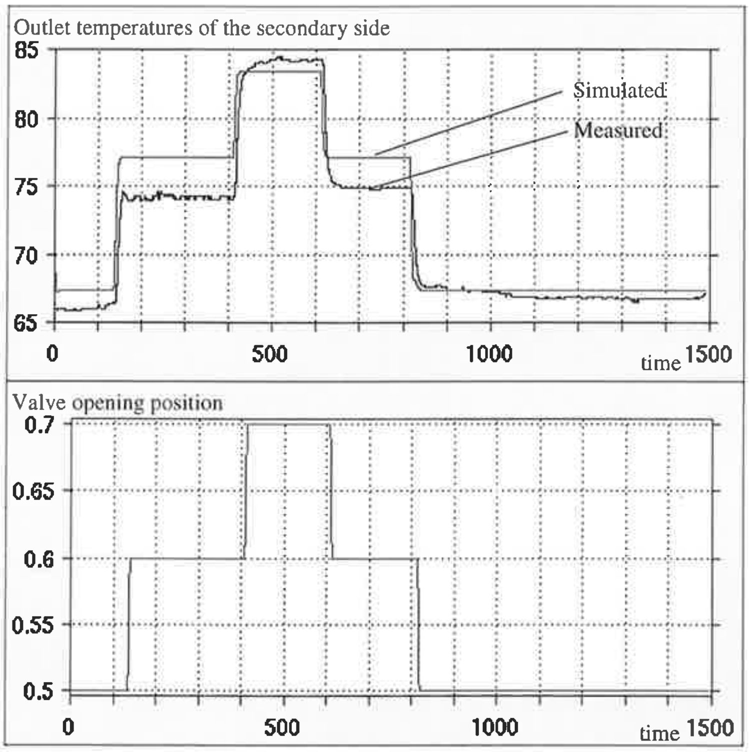


Figure 4.4 Upper plot: The outlet temperature of the secondary side as a function of time, measured and simulated. Lower plot: The valve position value (measured) as a function of time. Water flow on the secondary side is $m_w = 0.40$ l/s

recorded valve position, which was used as the valve position signal in the simulation. As the logged temperature is measured further away than the one connected to the PID, a time delay and a small temperature loss are added to the recordings. In Figure 4.4 this time delay has been removed. As can be seen the model differs from the recorded data with a maximum of $3°C$ in the final model.

Validation experiment 2

This experiment was made on a somewhat higher load with a water flow $m_w = 0.60$ l/s. The temperature into the secondary side was $14°C$. The valve position was set in the sequence 60%, 70%, 80%, 70% and 60%. The valve position was altered when the system was stable. As can bee seen in Figure 4.5 the simulation model follows the recorded data in a satisfactory way. The maximum difference is, as for validation experiment 1, $3°C$. Observe the nonlinearity in temperature rise between the steps 60% to 70% and 70% to 80%.

Validation experiment 3

The load in the experiment was held constant at $m_w = 0.60\ l/s$. The valve position was changed in small steps from 50% to 100%. Figure 4.6 shows that the model follows the experimental data very well up to a valve opening of 60%. For larger valve openings the model starts to deviate from the logged data with a maximum of $\approx 8\ kW$.

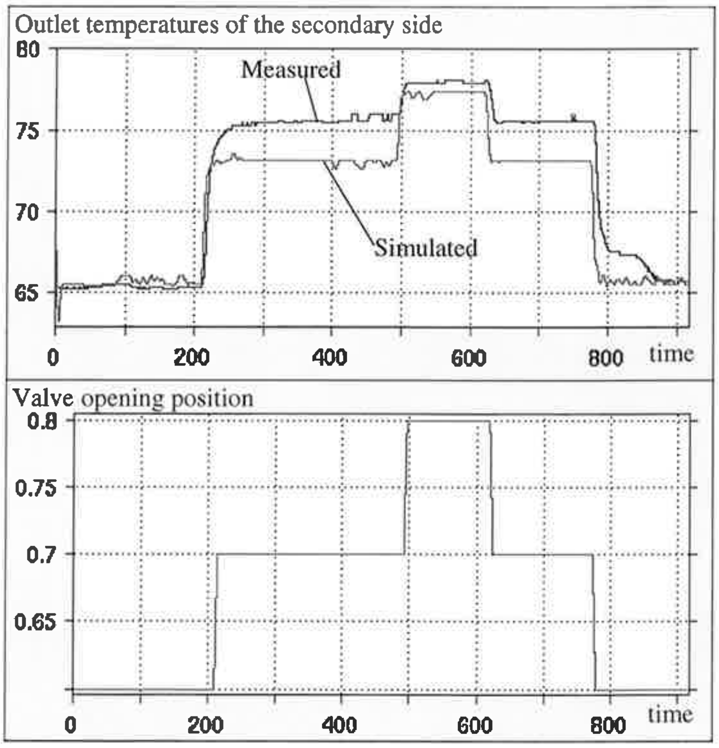


Figure 4.5 Upper plot: The outlet temperature of the secondary side as a function of time, measured and simulated. Lower plot: The valve position value (measured) as a function of time. Water flow on the secondary side is $m_w = 0.60$ l/s

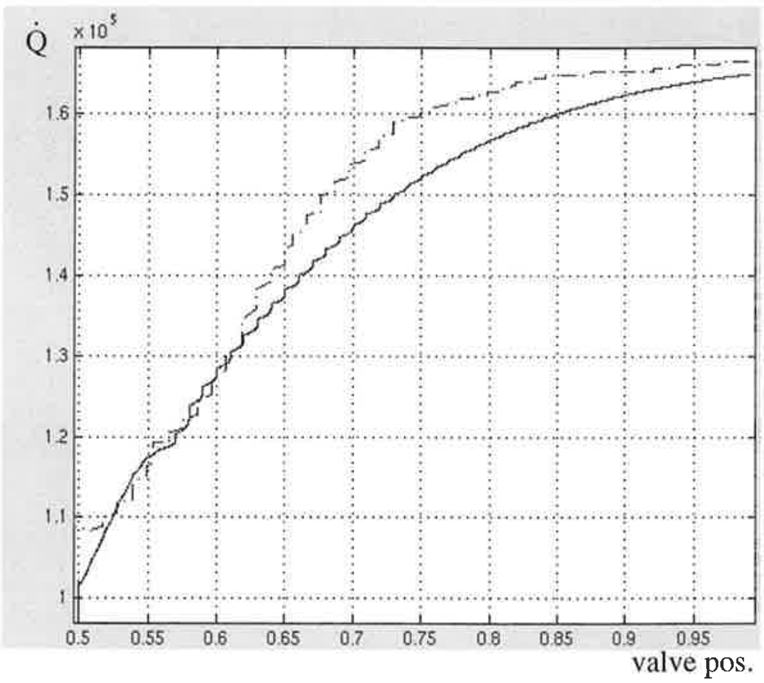


Figure 4.6 The heat transferred as a function of the valve position value (measured). Dash-dotted line is the measured data and solid line is the simulated. Water flow on the secondary side is $m_w = 0.60$ l/s.

4.3 Differences and Error Sources

The differences between the simulation model and the test rig PHE can be explained by many factors, that each and every one contributes to the model deviations. For example, the simulation model works under ideal conditions without any disturbances or noise. The test rig has fluctuations in the steam and water supplies. The test rig data logger can be one source of disturbance, due to the sampling of the measured quantities. The sensors and gauges on the test rig also add uncertainties to the logged data, due to noise and calibration errors.

Mechanical changes and wear can be yet another factor to take into consideration, e.g. calcification of the PHE and steam trap leakage. If a steam trap leaks and allows air to enter the steam system, this will have an insulating effect on the heat transfer, reducing the heat flow.

The model is built on physical relations. The description of the heat transport in the PHE can be done with a variety of equations. Any of these equations will introduce an error, since they are derived theoretically or empirically. Also, the simulation will introduce numerical errors.

5. Controller Tuning

The heat transferred depends on several quantities, e.g. the pressure drop over the steam valve, ΔP, and temperature at the inlet of the secondary side. The quantities also have dependencies on each other. Equation (5.1) describes the heat flow,$\dot{Q}$, through the PHE. For example, if the steam valve position, y, is increased, more steam flows into the primary side and the pressure drop over the steam valve decrease. If the pressure drop over the steam valve decrease the flow trough the valve decreases. The steam flowing into the PHE will transfer heat to the secondary side. As the steam condensates, the pressure inside the PHE will decrease, and thereby the pressure drop over the valve increases. Naturally the system will reach stationarity after a while, but it is difficult to predict where this point will be. It is obvious that some of the quantities in equation (5.1) can be difficult to measure, e.g. the entalphy, h_{out}, of the flow out of the primary side. This flow can be a mix of water and steam (flash). To design control parameters from these premises is very difficult and the result from the design in the closed loop system is most uncertain. An experimental approach to the control design is preferable.

In this Section two methods are presented to get the parameters to the controller i.e. K, T_i, T_d and b. The first method is based on measurements on the real process and analysis of the result, in the second method the analysis is based on the simulation model result. One way to get the parameters is to use some kind of step response method. The Ziegler-Nichols step response method and the $\kappa\tau$-method will be briefly explained. Also some examples will be given. To learn more about the methods confer [4].

$$\dot{Q} = (h'' - h_{out}) \frac{k_{v0}\sqrt{\Delta P \rho_s \frac{2}{z_v}}}{3.6 \cdot 10^4} e^{(ln(\frac{k_{vs}}{k_{v0}})y)} \tag{5.1}$$

5.1 About the Process

To get an idea about the linearity of the open loop system the valve position value of the simulation model was swept from 50% to 100% and back. The heat transferred vs. valve position, outlet temperature of the secondary side vs. time and valve position vs. time are given by Figure 5.1.

The plots in Figure 5.1 shows the nonlinearity in the process gain. The nonlinearity of the open loop system, due to the plate heat exchanger, is somewhat reduced by the valve since this is logarithmic, i.e. the valve gain increases with an increasing valve position (see Figure 2.6). It is important to make allowances for the nonlinearity of the open system when the control design is made, since the process gain is different from one point of operation to another. When an operating point for the PHE is decided, the region close to this point can be considered as linear. However, if it is intended to have the PHE working in a span of different operating points, it is necessary to introduce some compensation for the nonlinearity, e.g. by using gain scheduling in the controller. Many PID-controllers support this function. Another aspect of the nonlinearity of the open loop system is that the maximum outlet temperature of the secondary side of the PHE varies, depending on the load. This in turn affects the process gain and for that reason it is important to include the load

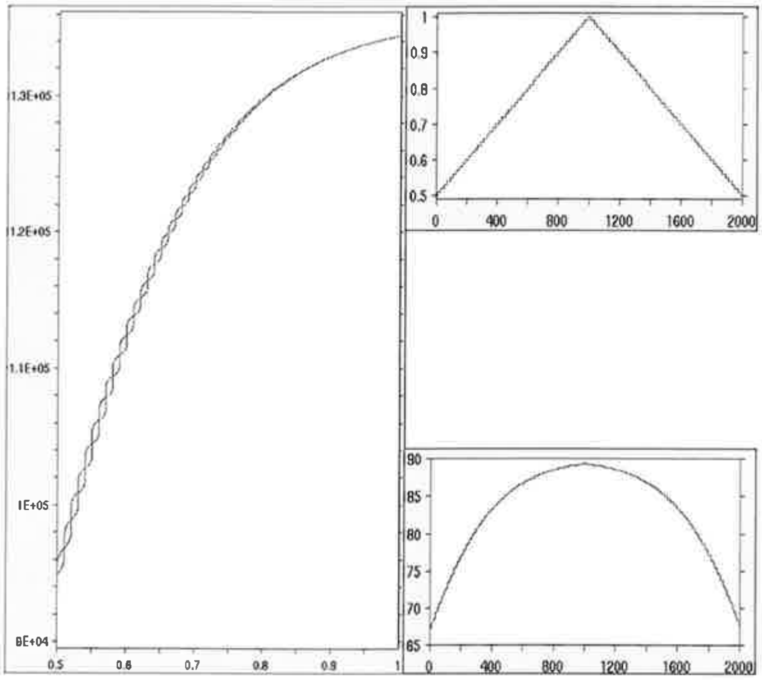


Figure 5.1 Left plot: Transferred heat ($\dot{Q}$) vs. valve position. Right upper plot: Valve position vs. time. Right lower plot: The temperature at the outlet of the secondary side vs. time. Inlet water temperature is $14°C$

in the operating point. The steam pressure flowing into the system has to be approximately constant for the same reasons. It is obvious that many variables affect the operating point and thereby the control design. By designing a robust controller deviations from the operating point can be accepted within some limits without greater loss of control performance.

5.2 Tuning based on Measurements

To design a controller from measured data the user of the PHE has to make a step response analysis of the process. This is made by first setting the process in a stable mode i.e. an operating point somewhere in the working range. When the temperature on the secondary side is stable, change the valve position 10% and wait until the temperature is stable again. A typical result is shown in Figure 5.2.

The first thing to do after this is to determine where the slope of the step response has its maximum. From this point draw a tangent that crosses the x-axis down to the y-axis, see Figure 5.3. This figure shows the relation between the input and output versus time i.e. $\frac{\Delta y}{\Delta u}$ vs t. This is done because now the static gain (K), dead time (L) and the time constant (T) can be seen directly.

Ziegler-Nichols step response method

In Ziegler-Nichols method only a and L is of interest. These parameters will give the control parameters according to Table 5.1.

In this table T_p is an estimate of the period of the closed-loop system. If the step signal in to the valve is not an ideal step but a slower ramp or filtered

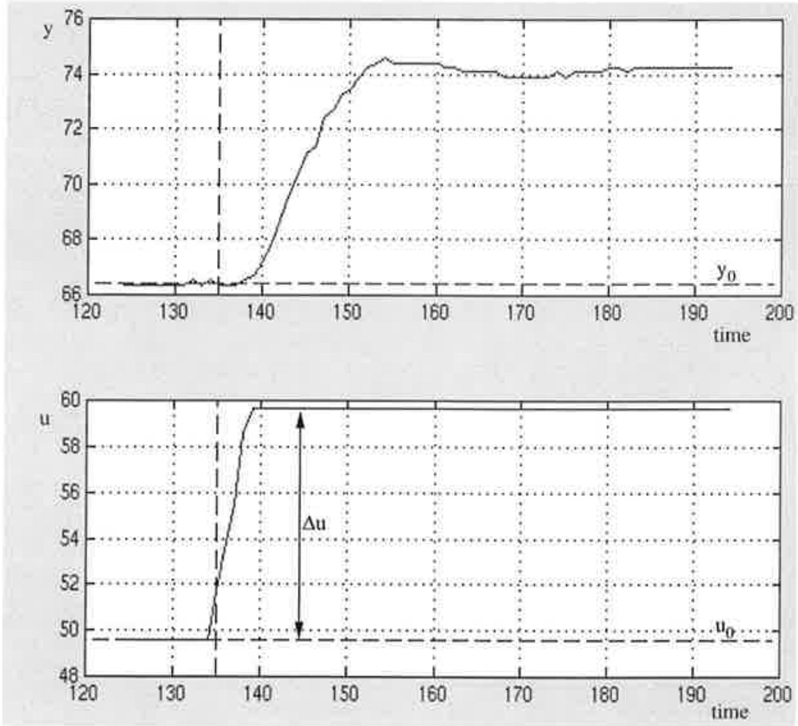


Figure 5.2 A measured step response.

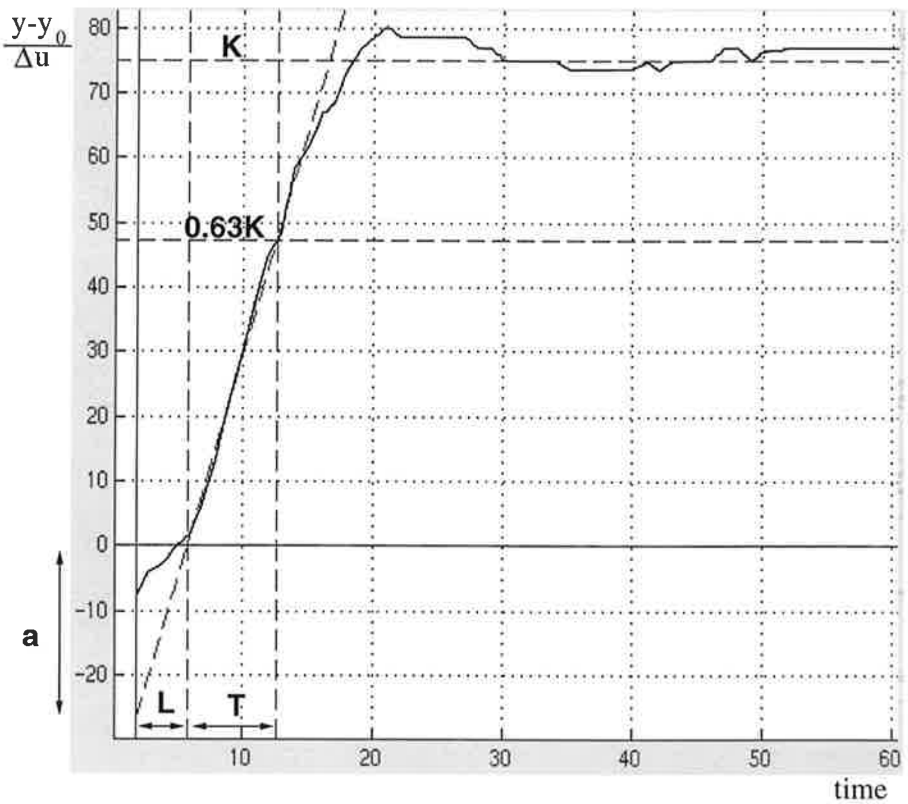


Figure 5.3 Normalized step response

Controller	K	T_i	T_d	T_p
P	$\frac{1}{a}$			$4L$
PI	$\frac{0.9}{a}$	$3L$		$5.7L$
PID	$\frac{1.2}{a}$	$2L$	$\frac{L}{2}$	$3.4L$

Table 5.1 PID controller parameters obtained from Ziegler-Nichols step response method

with a time constant, this will make the process seem slower and lead to slower control parameters, i.e. the system gets more robust.

EXAMPLE Ziegler-Nichols step response method

Figure 5.3 shows a normalized step response taken from the test rig used in this master thesis. To use Ziegler-Nichols method, a and L need a value. A look on the figure gives $a \approx 26.7$ and $L \approx 3.8$. Table 5.2 shows the control parameters.

Controller	K	T_i	T_d	T_p
P	0.037			15.2
PI	0.034	11.4		21.7
PID	0.045	7.6	1.9	12.9

Table 5.2 Control parameters given by Ziegler-Nichols method

#

$\kappa\tau$-method

The setpoint weighting b, the controller gain K, the integration time T_i and derivative time T_d are the design parameters of the $\kappa\tau$-method. The method uses the process apparent dead time L, the apparent lag T and a shown in Figure 5.3 as input parameters. It also requires a tuning parameter M_s (*maximum sensitivity*). The inverse of M_s is the distance from the critical point -1 to the Nyquist curve of G_pG_c (G_p = process transfer function, G_c = controller transfer function). The value of $1/M_s$ is a measure of the robustness of the PID-design. The controller parameters are calculated from the *relative dead time*, τ, which is a design parameter given by equation (5.2).

$$\tau = \frac{L}{L+T} \tag{5.2}$$

The controller parameters are derived from tuning diagrams which uses τ as an in parameter.

To learn more about the $\kappa\tau$-method confer [4]. One example is made for comparison with Ziegler-Nichols method.

EXAMPLE A step response method based on the $\kappa\tau$-method

Figure 5.3 is also used in this example. This method require the dead time ($L \approx 3.8$), time constant ($T \approx 6.7$) and $a \approx 26.7$. Table 5.3 shows the control parameters. The maximum sensitivity M_s is set to 1.4.

Controller	K	T_i	T_d	b
PI	0.0042	4.6		1.35
PID	0.018	6.7	1.7	0.62

Table 5.3 Control parameters given by $\kappa\tau$-method

#

5.3 Test of Control Parameters designed from Experimental Data

The parameters derived above was applied on the test rig process. Figure 5.4 shows the step response for the PID controller case. The setpoint was changed from $60°C$ to $70°C$ and back. The figure shows that the controller output and thereby the outlet temperature oscillates. This can result in unnecessary wear of the steam valve and actuator.

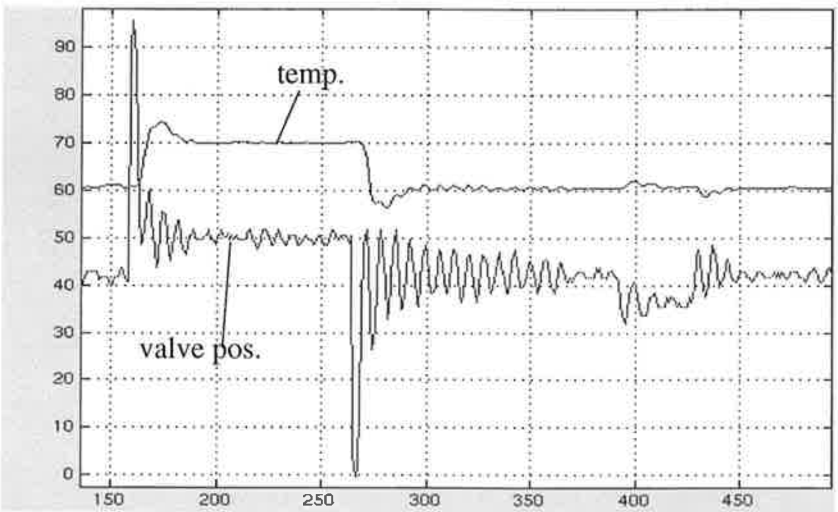


Figure 5.4 The step response of the test rig process with a Ziegler-Nichols PID control design. The temperature at the inlet on the secondary side is $14°C$ and $\dot{m}_w = 0.43l/s$.

Also a step response with the $\kappa\tau$-control design was made. Figure 5.5 shows the result. Note that the scales and load disturbances differ from the Ziegler-Nichols case (Figure 5.4). The PID-controller on the test rig does not provide setpoint weighting, i.e. $b = 1$ in equation (2.13) and thereby we can not use $b = 0.62$ as recommended by $\kappa\tau$. The controller shows very little oscillative behaviour and the response reaches the setpoint values in a smooth way. When comparing the two design methods it is obvious that the $\kappa\tau$-design gives a more robust controller.

Both the design methods give controllers that handle load disturbances rather fast. The Ziegler-Nichols design gives an oscillation in this case also. The test shows that the $\kappa\tau$-control design is preferable.

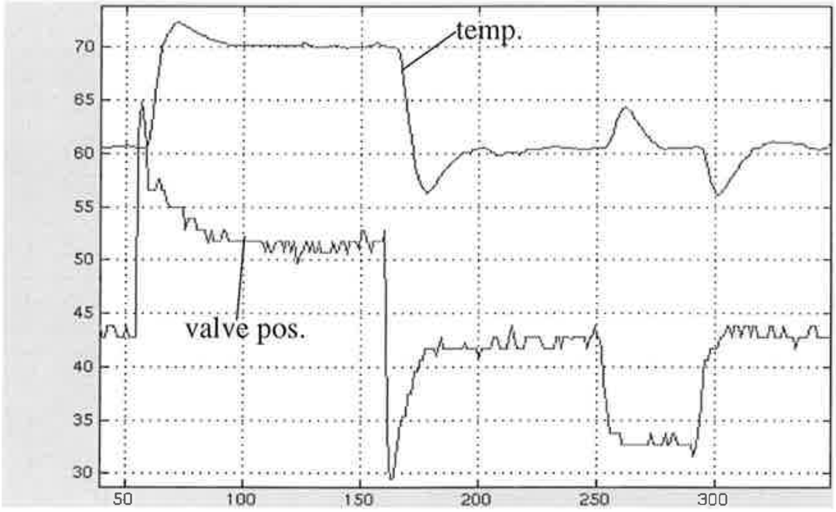


Figure 5.5 The step response of the test rig process with a $\kappa\tau$ PID control design. The temperature at the inlet on the secondary side is $14°C$ and $\dot{m}_w = 0.43l/s$. The set point weighting $b = 1$.

5.4 Tuning based on Simulation

To be able to use the simulation model for control parameter estimation, the user has to know all physical data, e.g. the pressure over the valves and the inlet temperature on the secondary side. Equation 5.3 shows the transfer from the valve position to the heat flow from the primary side to the secondary side. This equation is valid only if all steam condensates in the PHE.

$$\dot{Q} = (h'' - h_{out})\frac{k_{v0}\sqrt{\Delta P \rho_s \frac{2}{z_v}}}{3.6 \cdot 10^4} e^{(ln(\frac{k_{vs}}{k_{v0}})y)} \tag{5.3}$$

In this equation ΔP is the pressure drop over the steam valve, y is the valve position, z_v is a valve loss constant and ρ_s is the density for the steam i.e. $\rho_s = f(h,p)$, h'' and h_{out} is further explained in Appendix A. It can be seen that knowledge about the steam valve that is used is of importance. The analysis is the same as described in Section 5.1. Figure 5.6 shows a step response made by the simulation model when the valve changes from 50 - 60%. The mass flow on the primary side is 0.43 i.e. $\dot{m}_w = 0.43$.

Ziegler-Nichols step response method

Figure 5.7 shows a normalized step response based on the result from the simulation shown in Figure 5.6. Table 5.1 gives the control parameters.

Controller	K	T_i	T_d	T_p
P	0.045			4.8
PI	0.040	3.6		6.8
PID	0.054	2.4	0.6	4.1

Table 5.4 Control parameters given by Ziegler-Nichols method

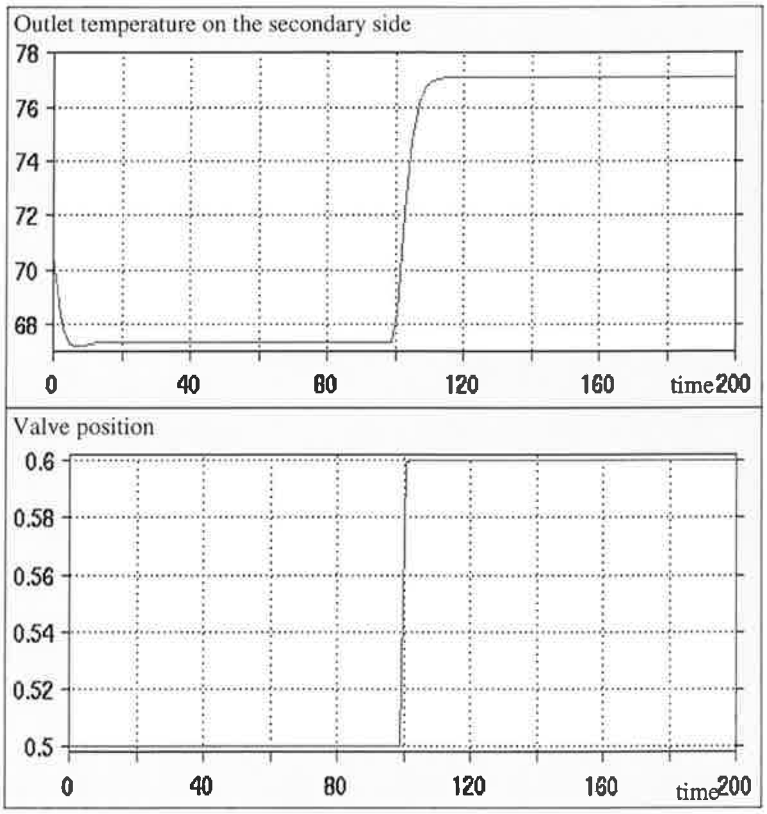


Figure 5.6 Step response made by the simulation model.

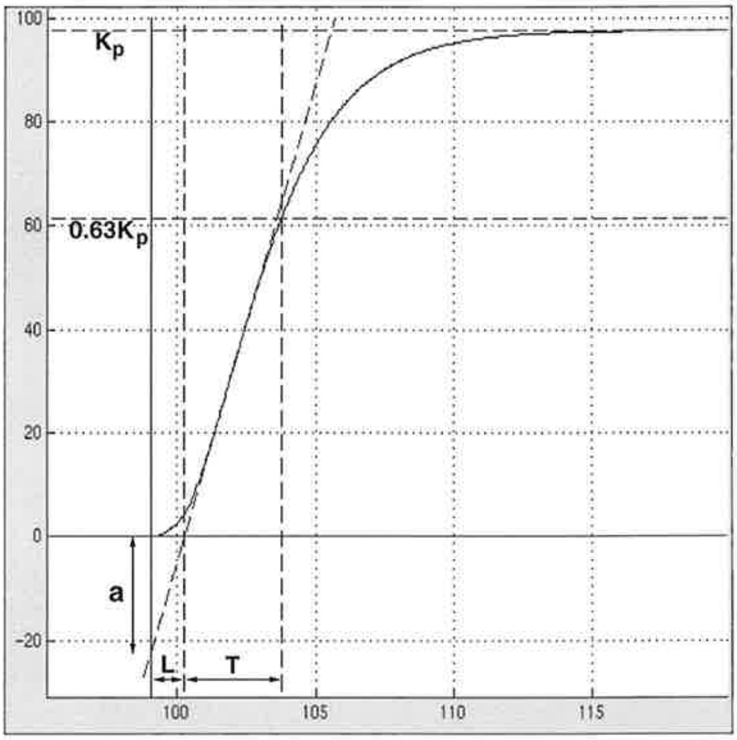


Figure 5.7 Normalized Step response.

EXAMPLE Ziegler-Nichols step response method based on the simulation model

Figure 5.7 gives the following parameters: $K_p \approx 97.6, L \approx 1.2, T \approx 3.5$ and $a \approx 22.4$. The control parameters are given by Table 5.4.

#

$\kappa\tau$-method

An example will be given in this section.

EXAMPLE A step response method based on the $\kappa\tau$-method

Figure 5.7 is also used in this example. The maximum sensitivity M_s is set to 1.4 and the control parameters are given by Table 5.5.

Controller	K	T_i	T_d	b
PI	0.0054	2.4		1.1
PID	0.021	3.0	0.7	0.5

Table 5.5 Control parameters given by $\kappa\tau$

#

5.5 Test of Control Parameters designed from Simulated Data

The parameters derived above were applied to the test rig process. Figure 5.8 shows the step response for the PID controller case. The setpoint was changed from $50°C$ to $60°C$ and back. The figure shows that the controller output and thereby the outlet temperature oscillates substantially. The bad design can be explained by the simulation model apparent dead time which is lower than it is for the test rig process. Since the dead time is a design parameter this will affect the tuned parameters substantially. The controller does not work as intended and the Ziegler-Nichols design method can be rejected when it is done on simulated data.

Also a step response on the $\kappa\tau$-control design was made. The setpoint was changed from $60°C$ to $70°C$ and back. Figure 5.9 shows the result. The controller shows some oscillative behaviour. The temperature reaches the setpoint value rather fast. The design gives a good idea of what parameters to choose, but some tuning has to be done to reduce the oscillations. In this case it would be appropriate to increase the T_i-value.

These experiments shows that it is possible to design controller parameters from simulated data with the $\kappa\tau$-method. Since Ziegler-Nichols tuning rules are less robust, the additional uncertainties introduced in the model makes it unsuitable for tuning the controller from simulated data.

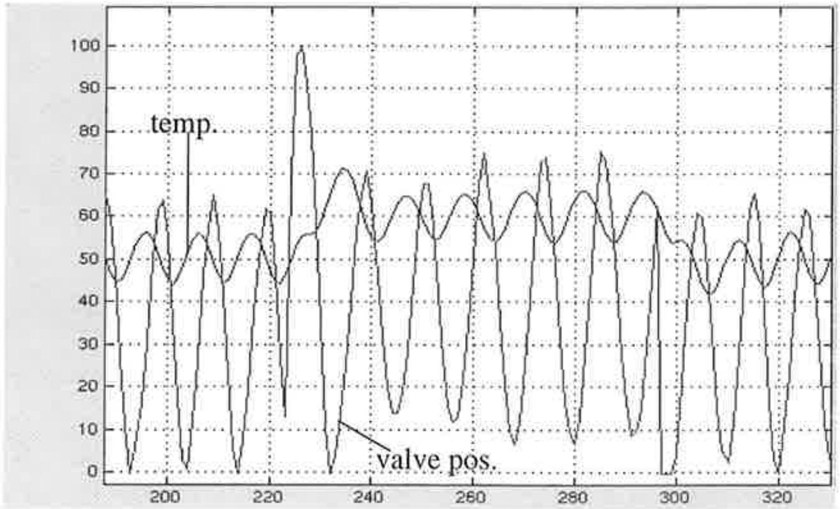


Figure 5.8 The step response of the test rig process with a Ziegler-Nichols PID control design from simulated data. The temperature at the inlet on the secondary side is $14°C$ and $\dot{m}_w = 0.43l/s$.

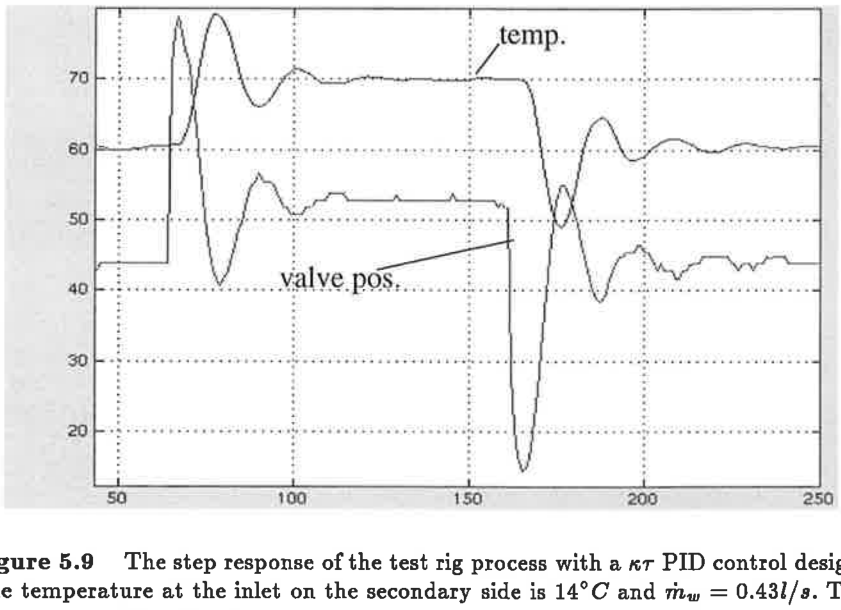


Figure 5.9 The step response of the test rig process with a $\kappa\tau$ PID control design. The temperature at the inlet on the secondary side is $14°C$ and $\dot{m}_w = 0.43l/s$. The set point weighting $b = 1$.

6. Conclusions

The thesis has presented a simulation model of a plate heat exchanger for steam heating applications, i.e. heating other medias with steam. The model was made in the modelling language Omola. Two PID-controller tuning methods have been presented. The two methods are based on step response analysis.

6.1 Model

The model is based on the laws of physics and is applied to the plate heat exchanger. The model supports the change of PHE properties allowing the user to simulate other PHE models. This is made by altering the parameters relevant in this matter in the `PlateHeatExchangerFM`-class.

It is important to have reasonable initial states as the simulation is started, or else the simulation will not always be executable. The OCL-file PHESim initialize all necessary state variables and parameters. PHESim sets the simulator in a *ready to run* state and presents a user interface showing plots, state variables and parameters.

A validation was made on the model. In Chapter 4 the model is compared with experimental data logged from the test rig. The simulation model showed good correspondence with the logged data. The maximum deviation from the data was less than 3 $°C$ in the tests conducted.

6.2 Tuning Methods

In Chapter 5 it was concluded that it is impossible to design a controller just by using physical properties of the PHE, since the dynamic behaviour of the process is not taken into consideration. Instead two methods were suggested, one based on experimental data and the other on simulated data. Both methods build on step response analysis. The controller designs used in this thesis are the Ziegler-Nichols-method and the $\kappa\tau$-method.

Both the design methods are applicable when designing the controller from experimental data. The Ziegler-Nichols method introduces some oscillations on the control signal and some further tuning is required. The $\kappa\tau$-method gives a more robust controller and is thereby to prefer.

Since the simulation model predicts a faster step response with a shorter dead time, the Ziegler-Nichols design method give control parameters that can not be used. The method can for that reason by rejected. The $\kappa\tau$-method gives a fair control design, but some additional tuning has to be made to restrain the somewhat oscillative behaviour.

6.3 Future Work

A direct continuation of the work in this thesis is to verify the simulation model for other plate heat exchangers. It is also of interest to increase the model libraries with e.g. condensate pumps. This allows modelling of other control strategies such as bypass control of the flows on the secondary side of the PHE. A natural continuation of this is to develop control designs for these.

Since the thesis has been focused on the building of the simulation model, the control design methods can be further investigated to suit the PHE application better. As they are presented in this thesis, they only give a good starting point.

7. References

[1] M. Andersson, "OmSim and Omola Tutorial and User's Manual - Version 3.4",Department of Automatic Control Lund Institute of Technology, (Available on Internet - www.control.lth.se/~ cace/manual.html), March 1995.

[2] M. Andersson, "Object-Oriented Modeling and Simulation of Hybrid Systems", Doctoral dissertation, Department of Automatic Control Lund Institute of Technology, ISSN 0280-5316, ISRN LUTFD2/TFRT–1043–SE, December 1994.

[3] B. Nilsson, J. Eborn, "K2 Model Database - Tutorial and Reference Manual", Technical report, Department of Automatic Control Lund Institute of Technology, ISSN 0280-5316, ISRN LUTFD2/TFRT–7528–SE, December 1994.

[4] K. Åström, T Hägglund, "PID Controllers: Theory, Design and Tuning",2nd edition, Instrument Society of America, ISBN 1-55617-516-7, 1995.

[5] T. Glad, L. Ljung, "Reglerteknik - Grundläggande teori", 2nd edition, Studentlitteratur AB, ISBN 91-44-17892-1, 1989, (Swedish)

[6] L. Håkansson, "Reglerteknik - Styrventiler", Tour & Andersson Control Division, 1987, (Swedish)

[7] L.K. Wang, Q.S. Yang, "Analysis of Steam Condensation Heat Transfer in Plate Heat Exchangers", Jiao Tong University, Shanghai, 1997.

A. Thermodynamics and Heat Exchanger Theory

This appendix is a compile of lectures given by Stefan Burg and Rolf Ekelund at Alfa Laval Thermal in Lund.

A.1 Mass and Energy Balance

Mass Balance

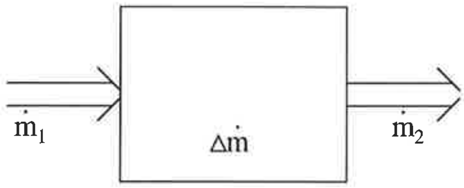


Figure A.1 Black Box with mass flow

The Law of Mass Conservation gives the equation (A.1). The mass flow in to the black box is denoted $\dot{m}_1$, the mass flow out of the black box is $\dot{m}_2$ and the accumulated mass flow $\Delta\dot{m}$.

The mass balance equation is

$$m_1 = m_2 + \Delta m \tag{A.1}$$

Energy Balance

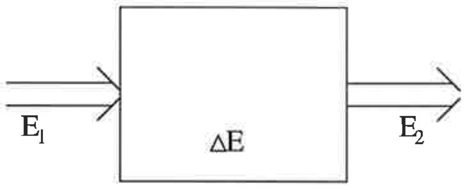


Figure A.2 Black Box with flow of energy

The Law of Energy Conservation gives the equation (A.4). Figure A.2 illustrates the equality where energy given to the black box is denoted E_1, from the black box E_2 and the energy accumulated in the black box ΔE.

The Energy balance equation (A.4) is given by equations (A.2) and (A.3).

$$\int \dot{E}_1 d\tau = \int dE + \int \dot{E}_2 d\tau \tag{A.2}$$

$$\dot{E} = \frac{dE}{d\tau} \tag{A.3}$$

$$E_1 = E_2 + \Delta E \tag{A.4}$$

For a stationary system

$$\Delta E = 0 \tag{A.5}$$

$$E_1 = E_2 \tag{A.6}$$

$$\dot{E}_1 = \dot{E}_2 \tag{A.7}$$

A.2 Heat Exchanger with 1-phase Fluids

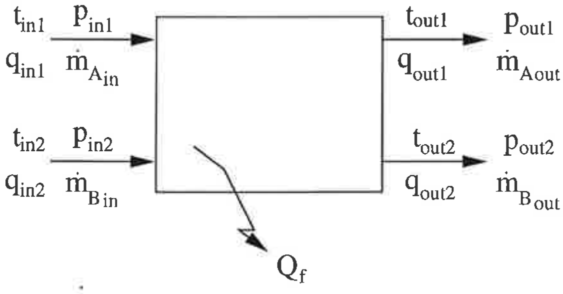


Figure A.3 The Heat Exchanger as a black box

Figure A.3 shows some important quantities acting on a Heat Exchanger. Temperature is denoted t, mass flow $\dot{m}$, pressure p and energy per mass unit q deriving from

$$\dot{Q} = \dot{m}q$$

where $\dot{Q}$ is the flow of energy. $\dot{Q}_f$ is the energy loss flow. $q = q(p, t)$ with independence between p and t. When the black box system is in steady state the following statements are valid

$$\left.\begin{array}{l} \dot{m}_{Ain} + \dot{m}_{Bin} = \dot{m}_{Aout} + \dot{m}_{Bout} \\ \dot{m}_{Ain} = \dot{m}_{Aout} \equiv \dot{m}_A \\ \dot{m}_{Bin} = \dot{m}_{Bout} \equiv \dot{m}_B \\ \dot{m}_A q_{in1} + \dot{m}_B q_{in2} = \dot{m}_A q_{out1} + \dot{m}_B q_{out2} \end{array}\right\} \Rightarrow$$

$$\Rightarrow \dot{m}_A(q_{in1} - q_{out1}) = -\dot{m}_B(q_{in2} - q_{out2}) \tag{A.8}$$

The equations are referring to a Membrane Heat Exchanger. In this case the fluids do not mix. The equation (A.8) can be rewritten as

$$\dot{m}_A \Delta q_1 = -\dot{m}_B \Delta q_2 \tag{A.9}$$

Now we do a simplification by introducing the concept of enthalpy. This is denoted H, i or h. In this Thesis we will use the latter one. There is a correlation between enthalpy and energy which we do not treat in this text, but the similarities in the equations is evident. For a one-phase fluid in constant pressure we have the following expression

$$\left(\frac{\partial h}{\partial t}\right)_{p=const.} = c_p$$

The constant c_p is the *specific heat capacity* and can be found in tables. When easing up on the stringency we can rewrite the expression as

$$\Delta h = c_p \Delta t \tag{A.10}$$

Now we rewrite equation (A.8) using entalphy

$$\dot{m}_A c_{pA}(t_{in1} - t_{out1}) = -\dot{m}_B c_{pB}(t_{in2} - t_{out2}) \tag{A.11}$$

A.3 A Simple Model of a Plate Heat Exchanger (one plate)

The flows in a Plate Heat Exchanger can be counter current or concurrent. The counter current flow gives a higher temperature rise than concurrent flow. The characteristics of the two cases is shown in Figure A.4. The advantage with the concurrent flow is that the plate has approximately constant temperature all over. This setup is used in high temperature applications where the mechanical construction has limits concerning the temperature. When counter current flow is used the temperature of the plate is low at one end and high at the other.

The heat transport trough the plate (Figure A.5) can be described with equation (A.12)

$$\dot{q}_L = \lambda \frac{dt}{dx} \tag{A.12}$$

where λ is the *heat conduction coefficient.* When integrating over the thickness of the plate we get

$$\dot{q}_L = \lambda \frac{(t_1 - t_2)}{\delta}$$

The heat transport from the fluid to the plate has a similar expression where α is the *heat transfer coefficient.* The primary side quantities is denoted

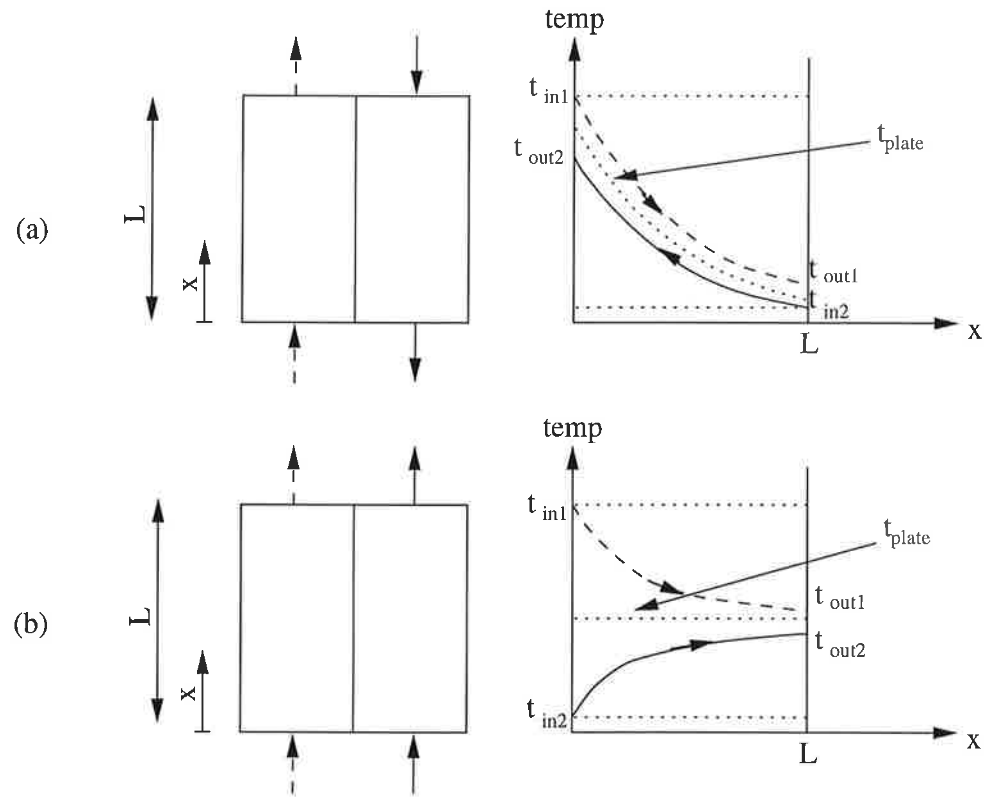


Figure A.4 A simple Plate Heat Exchanger Model with only one plate. (a) shows a counter current type and (b) shows a concurrent type. Dashed lines indicates the warm fluid.

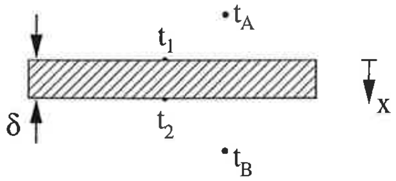


Figure A.5 A small part of the plate.

with the index A and the secondary side with B. The index w refers to a point on the plate (wall).

$$\dot{q}_A = \alpha_A(t_A - t_{wA}) \tag{A.13}$$

$$\dot{q}_B = -\alpha_B(t_B - t_{wB}) \tag{A.14}$$

The connection between temperature on the primary and secondary side of the Plate Heat Exchanger and the energy is given by

$$c_{pA}dt_A = c_{pB}dt_B + dq \tag{A.15}$$

The heat flow from the primary side to the secondary gives the relations

$$\dot{q}_A = \alpha_A(t_A - t_{wA}) = -\alpha_B(t_B - t_{wB}) = \lambda\frac{(t_{wA} - t_{wB})}{\delta} = \dot{q}_B \tag{A.16}$$

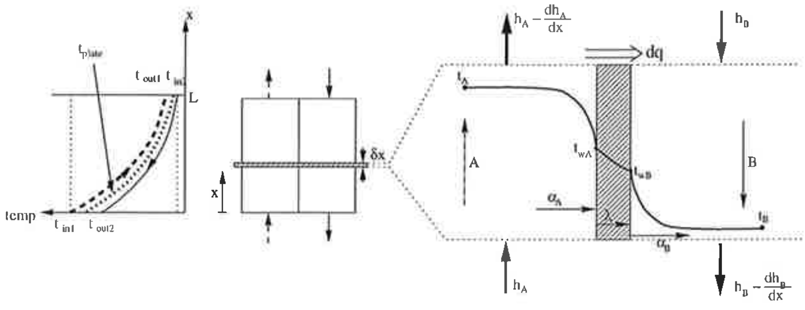


Figure A.6 The temperatures and heat exchanging of a small portion of the Plate Heat Exchanger.

The heat flow for the total Heat Exchanger plate area is given by the relations

$$\dot{Q} = kA\Delta t_m$$

$$\begin{aligned} \dot{Q}_A &= \alpha_A A(\bar{t}_A - \bar{t}_{wA}) = -\alpha_B A(\bar{t}_B - \bar{t}_{wB}) = \\ \lambda A &\tfrac{\bar{t}_{wA}-\bar{t}_{wB}}{\delta} = \dot{m}_A c_{pA}(t_{in1} - t_{out1}) = \\ -\dot{m}_B &c_{pB}(t_{in2} - t_{out2}) = kA\Delta t_m \end{aligned} \quad (A.17)$$

where $\bar{t}$ denotes *mean temperature.* Δt_m denotes the *formal mean temperature difference* and can defined in several ways. In this Thesis we will use the definition of *logarithmic mean temperature difference* (LMTD) given by

$$\Delta t_m = LMTD = \frac{\Delta t' - \Delta t''}{ln\left(\frac{\Delta t'}{\Delta t''}\right)} \quad (A.18)$$

where

$$\begin{cases} \Delta t' = t_{in1} - t_{out2} \\ \Delta t'' = t_{out1} - t_{in2} \end{cases}$$

A.4 Some Key Coefficients

When calculations are made on a Plate Heat Exchanger the *overall heat transfer coefficient k* (also known as the *thermal conductivity*) is frequently used and is given by equation (A.19). The quantity of *heat resistance* is the inversion of k. It takes into consideration the heat resistance in the fluids and the plate.

$$\frac{1}{k} = \frac{1}{\alpha_A} + \frac{1}{\alpha_B} + \frac{\delta}{\lambda} \quad (A.19)$$

Other interesting coefficients in thermodynamics are the Nusselt, Reynolds and Prandtl numbers. The Nusselt number is given by

$$Nu = \frac{\alpha_f d_H}{\lambda_f} \quad (A.20)$$

The *hydraulic diameter* d_H is

$$d_H = \frac{4A_c}{O}$$

where A_c is the channel area and O is the channel circumference.

EXAMPLE A.1

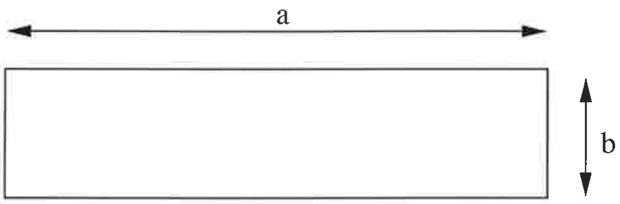


Figure A.7 A channel in the Plate Heat Exchanger

The hydraulic diameter of the channel in Figure A.7 is given by

$$d_H = \frac{4ab}{2a + 2b}$$

$$if \ \frac{b}{a} \to 0 \Rightarrow d_H = \frac{4b}{2(1 + \frac{b}{a})} \to 2b$$

#

The Reynolds number is given by

$$Re = \frac{wd_H}{\vartheta_f} \qquad (A.21)$$

where w denotes the flow velocity and ϑ_f the kinematic viscosity.
Finally the expression for the Prandtl number is

$$Pr = \frac{c_{pf}\vartheta_f\rho_f}{\lambda_f} \qquad (A.22)$$

where ρ_f is the density of the fluid.

The Nusselt number is not (!) dependent of what fluid the channel contains. However as can be seen in the definition (A.20) the dependency of the channel geometry is obvious. Finally there is a relationship between three coefficients given by (A.23) and visualized by Figure A.8

$$\frac{Nu}{f(Pr)} = cRe^z \qquad (A.23)$$

When the load of the Heat Exchanger is approximately constant the three constants can result in diagram shown in Figure A.9

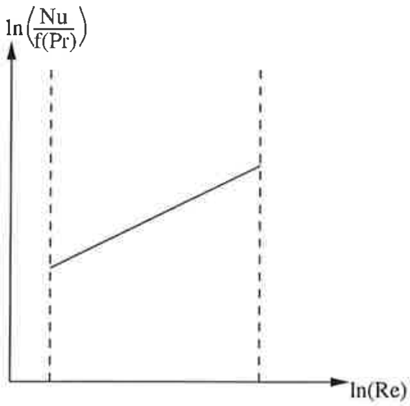


Figure A.8 The relationship between the Reynolds, Nusselt and Prandtl coefficients

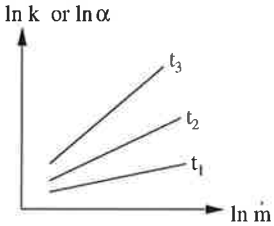


Figure A.9 ln k or ln α as a function of t and ln$\dot{m}$ when the load on the Plate Heat Exchanger is approximately constant

A.5 Physical Properties of the Plate Heat Exchanger

When building a model or making calculations on a Plate Heat Exchanger one has to take the connections to the unit into consideration. They will give some pressure drop which can not be neglected. In specifications all data of a Plate Heat Exchanger is related to the entire unit including the connections. The pressure drop over a tube is

$$\Delta p = 4F_p \frac{L}{d_H} \rho \frac{w^2}{2} \tag{A.24}$$

F_p is a constant without dimension and w denotes the velocity of the flow. Circular tubes like the one shown in Figure A.10 have the simplified expression (A.25)

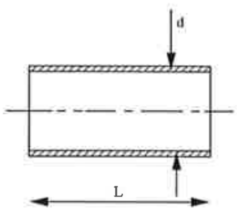


Figure A.10 A circular tube

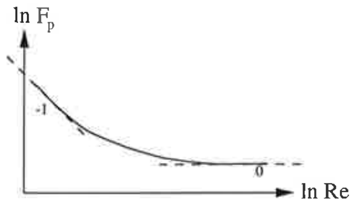


Figure A.11 $\ln F_p$ as a function of $\ln Re$

$$\left.\begin{array}{l} \frac{\dot{m}}{\rho} = wA \\ \frac{\pi d^2}{4} = A \end{array}\right\} \Rightarrow$$

$$\Rightarrow \Delta p = F_p \frac{\dot{m}^2}{\rho} \tag{A.25}$$

The constant F_p is measured as a function of the Reynolds number for each Heat Exchanger model. An example of such a measurement is shown Figure A.11. P in the following relations denote the pump energy

$$Q \sim k \sim \dot{m}^{\xi}\ , \quad 0,6 \leq \xi \leq 0,8$$

$$P \sim \dot{m}^2$$

The Reynolds number indicates how turbulent the flow is. A low Reynolds number is equivalent to laminar flow and a high number to turbulent flow. In Figure A.11 given by the expression (A.26) $x = -1$ when the flow is laminar and $x = 0$ when the flow is highly turbulent.

$$F_p = c_{\Delta p} Re^x \tag{A.26}$$

$$\Delta p = \Delta p_{connection\ in} + \Delta p_{channels} + \Delta p_{connection\ out} \tag{A.27}$$

The pressure drop over the out connection $\Delta p_{connection\ out}$ can be negative, i.e. a pressure rise. This happens if the area of the outlet is greater than the channel area.

The pressure drop over the Heat Exchanger unit is divided into three parts as equation (A.27) describes. The pressure drop over a plate channel in a Plate Heat Exchanger like the one in Figure A.12 is greater for a channel close to the inlet and outlet of the unit than for a channel further away form the connections. The equations beneath show this clearly for the plates on the primary side and they refer to the pressure drops shown in Figure A.12 (b). Notice that each side (primary and secondary) only use every second plate channel

$$\left\{\begin{array}{l} \Delta p_A = \Delta p_{ch11} + \Delta p_{p1} + \Delta p_{ch21} \\ \Delta p_A = \Delta p_{ch11} + \Delta p_{ch12} + \Delta p_{ch13} + \Delta p_{p3} + \Delta p_{ch23} + \Delta p_{ch22} + \Delta p_{ch21} \\ \Delta p_A = \Delta p_{ch11} + \ldots + \Delta p_{ch15} + \Delta p_{p5} + \Delta p_{ch25} + \ldots + \Delta p_{ch21} \end{array}\right.$$

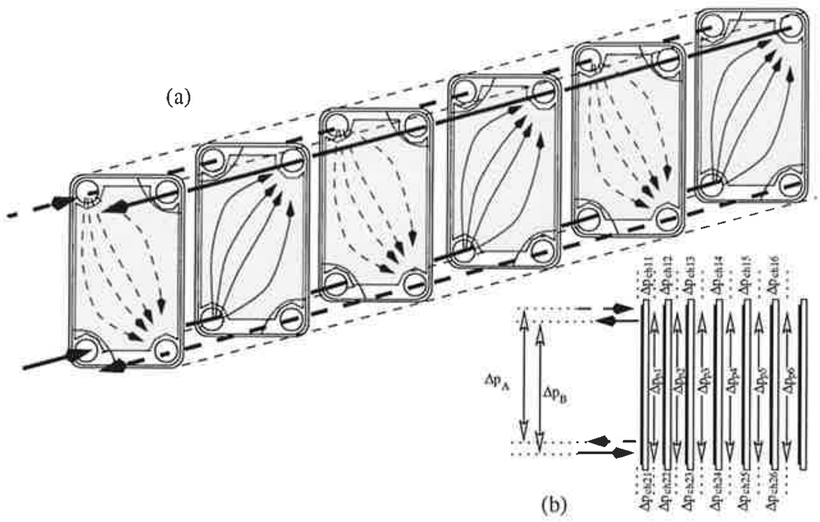


Figure A.12 (a) Schematic view over a Plate Heat Exchanger with counter current cross diagonal flows. (b) The pressure drops in a Plate Heat Exchanger.

A comparison of the first and the second equations show the differences in pressure drop over the plate channels.

$$\begin{aligned}
&\Delta p_{ch11} + \Delta p_{p1} + \Delta pch21 = \Delta p_A = \\
&= \Delta p_{ch11} + \Delta p_{ch12} + \Delta p_{ch13} + \Delta p_{p3} + \Delta p_{ch23} + \Delta p_{ch22} + \Delta p_{ch21} \\
&\Delta p_{p1} = \Delta p_{ch12} + \Delta p_{ch13} + \Delta p_{p3} + \Delta p_{ch23} + \Delta p_{ch22} \\
&\Delta p_{ch12} + \Delta p_{ch13} + \Delta p_{ch23} + \Delta p_{ch22} > 0 \;\Rightarrow \\
&\Rightarrow \;\Delta p_{p1} > \Delta p_3
\end{aligned}$$

A.6 Steam as a Media

The kind of steam used mostly in the industry, is saturated steam i.e. the steam is close to start condensation. In Heat Exchanger applications with steam as a heat source the heat used is the one coming from the phase transition from steam to water. In the transition and at constant pressure the temperature of the steam/condensate is constant e.g. for a pressure at 1 bar the temperature is 100^oC (see Figure A.13). One could be tempted to do the approximation that the temperature on the primary side of the Heat Exchanger is constant at the saturation temperature for the current pressure. However this is not true due to the pressure drop that arise when the steam condenses and the volume occupied by the condensate water is only a fraction of what the steam occupied ($V_{steam} \sim 1000 V_{water}$) (see Figure A.15 (a)). As Figure A.14 shows the temperature decreases as one move towards the outlet due to the pressure drop, since the saturation temperature decreases with the pressure (see Figure A.15 (b)).

Steam quality is a measure of mass relation between water and steam as they act together as a 2-phase fluid. The relation is given by the expression (A.28) where m_s is the amount of steam and m_w is the amount of water. Instead of the steam indexing m_s one sometimes in the literature find the denotation m'' (when in 2-phase region, i.e. saturated steam), $m_{å}$(Swedish)

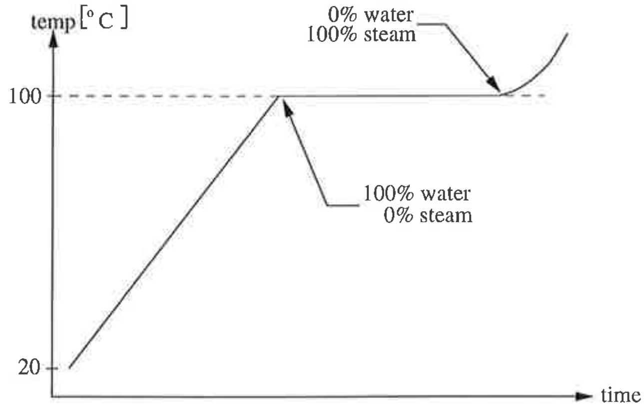


Figure A.13 The temperature as a function of time when heating water to steam with a constant heat supply at constant pressure of 1 atm.

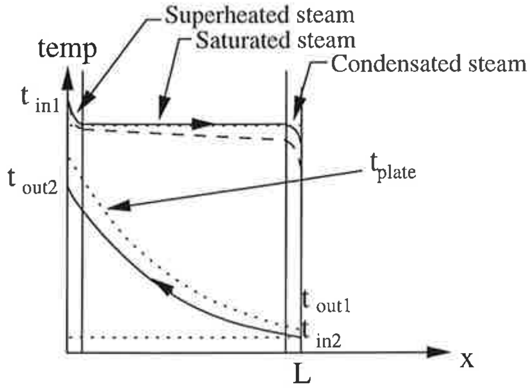


Figure A.14 Solid line shows the temperature for the approximation that the temperature is constant. Dashed line shows the actual temperature where the temperature decreases along the primary side due to the pressure drop in the Heat Exchanger.

and m_g (gas). For indexing water one can find the denotation m' (when in 2-phase region) and m_f (fluid).

$$x = \frac{m_s}{m_s + m_w} \tag{A.28}$$

The steam quality is used to calculate the specific volume (equation (A.29)) and enthalpy (equation (A.30)) for a mass unit of fluid in 2-phase.

$$v = v_s x + v_w(1 - x) \tag{A.29}$$

$$h = h_s x + h_w(1 - x) \tag{A.30}$$

where

$$\begin{cases} h_w = c_{pw} t_w \\ h_s = c_{ps} t_s \end{cases}$$

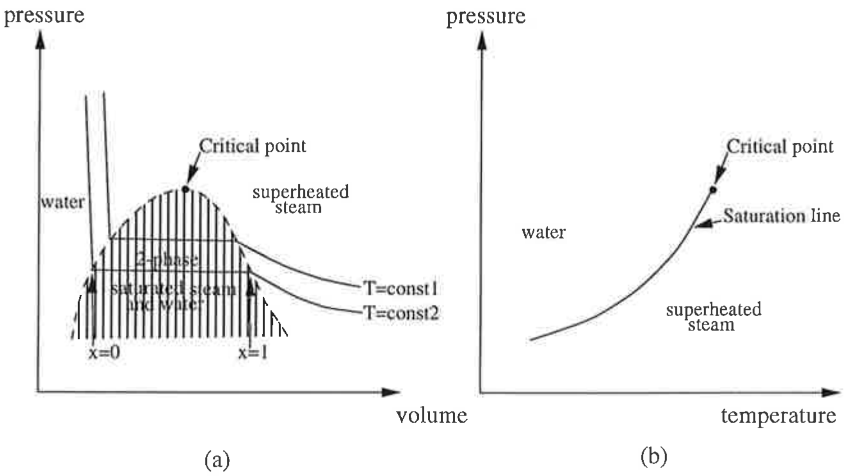


Figure A.15 Pressure as a function of volume (a) and as a function of temperature (b).

The enthalpy is a function of pressure and temperature. When the steam is saturated the steam pressure is a function of the temperature. We can write the following

$$h_s = h_s(t_s, p_s) \quad h_w = h_w(t_w, p_w)$$
$$p_{sat} = p(t_{sat})$$
$$h'' = h_s(t_{sat}, p(t_{sat})) = h_s(t_{sat})$$
$$h' = h_w(t_{sat}, p(t_{sat})) = h_w(t_{sat})$$

$$h = h''x + h'(1 - x)$$

A key number in Heat Exchanger context is NTU (=Number of Thermal Units), sometimes denoted θ. For a Plate Heat Exchanger in a 1-phase fluid application the NTU number is defined by equations (A.31) and (A.32) where the temperatures used are the temperatures shown in Figure A.4 (a) and (b). In Swedish literature the NTU number is sometimes referred to as *termisk längd = thermal length.*

$$NTU_1 = \theta_1 = \frac{t_{in1} - t_{out1}}{LMTD} = \frac{\Delta T_1}{LMTD} \tag{A.31}$$

$$NTU_2 = \theta_2 = \frac{t_{out2} - t_{in2}}{LMTD} = \frac{\Delta T_2}{LMTD} \tag{A.32}$$

To make the flow trough the Plate Heat Exchanger more turbulent the plates are corrugated in special patterns. The corrugation also gives better mechanical strength, allowing the use of thin materials for making the plates and thereby a reduction of the plate heat transfer resistance (δ/λ). It also increases the effective heat transfer area of the plate. One of the most common type is the *fish-bone* pattern (see Figure A.16). Interesting relations involving the NTU number are given below.

$$\dot{Q} = kA \cdot LMTD = \dot{m}c_p\Delta T \Rightarrow$$

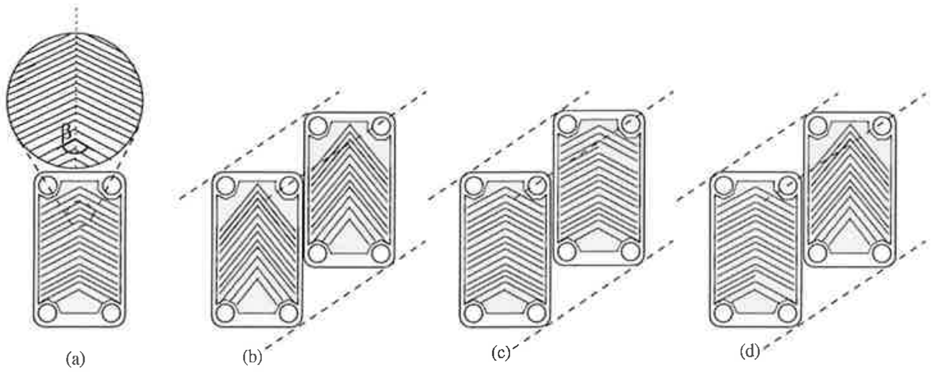


Figure A.16 (a) The plate with fishbone structure corrugation. The different combinations of plates with different top angle (β) on the fishbone structure are classified after the θ-value (NTU-value). (b) is a *low theta* (Lθ) combination, (c) is a *high theta* (Hθ) and (d) is a *medium theta* (Mθ).

$$\Rightarrow \frac{\Delta T}{LMTD} = \frac{kA}{\dot{m}c_p} = NTU = \theta$$

A.7 Heat Transfer from Steam to Water

In real steam applications the steam source supplies superheated steam. This means that the temperature of the steam at the given pressure is higher than the saturation temperature. The reason for using superheated steam is that no drops of water are present in the fluid. Since the velocity of the steam usually is high, the water drops would get a high kinetic energy due to the larger mass. It would be harmful to the components in the PHE system.

As the steam condenses into water a level of water will arise if the pressure at the outlet is greater than the water level surface. The effective heat transfer area for the saturated steam will then decrease.

The two aspects above implies that the heat transfer form the primary side to the secondary side is divided into three zones. The first zone (SC = *Steam Cooler*) is the heat transfer from the superheated steam to the water. In the second zone the saturated steam is the heat source. It is here the major part of the heat transfer is taking place. The third zone (DC = *Drain Cooler*) contains the condensate to water heat transfer. Figure A.17 shows the balance scheme for the temperatures and heat exchanges of the three zones in the PHE.

The following equations (A.33) (A.34) (A.35) give the energy balance for the PHE

$$c_p\dot{m}_w(t_o - t_x) = m_s(h_{s_{in}} - h'') \tag{A.33}$$

$$c_p\dot{m}_w(t_x - t_y) = m_s(h'' - h') \tag{A.34}$$

$$c_p\dot{m}_w(t_y - t_i) = m_s(h' - c_p t_{c_{out}}) \tag{A.35}$$

If the temperatures and pressures in and out of the PHE are known and the enthalpies for $h_{s_{in}}$,h'' and h' are looked up in tables, it is possible to calculate the temperatures t_x and t_y with the equations above. This makes it possible to divide the PHE into three separate zones.

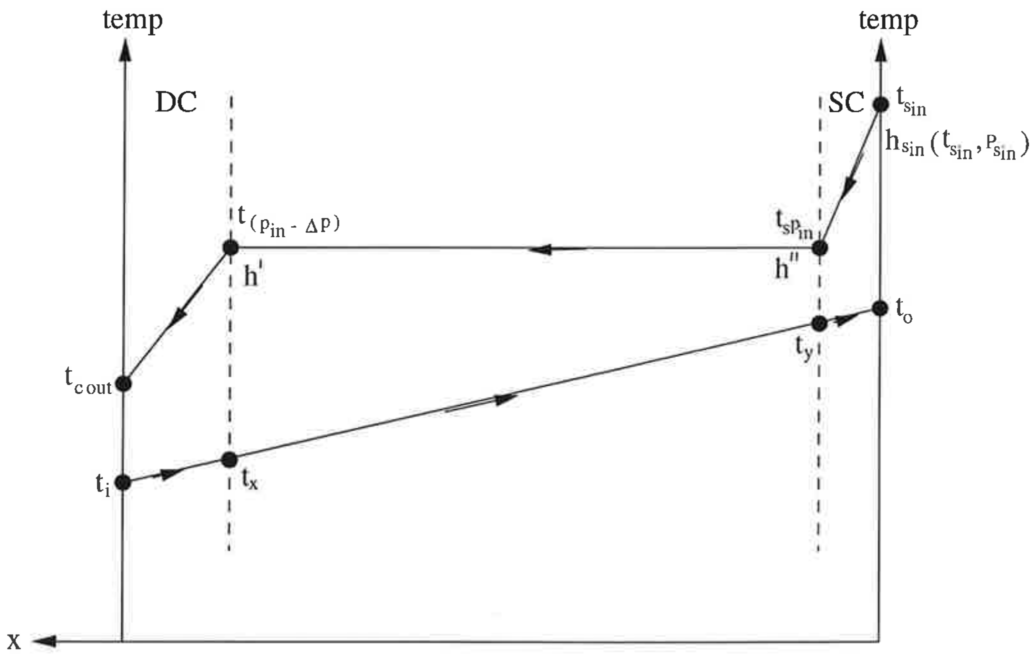
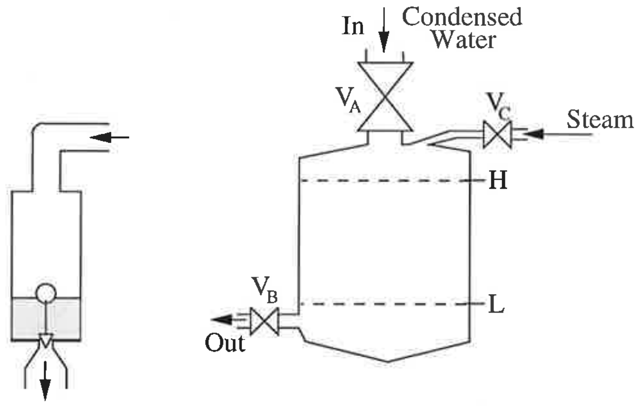


Figure A.17 Three zone heat transfer balance scheme.

A.8 Float operated Steam Traps and Intermittent Pumps



(a) (b) (c)

	V_A	V_B	V_C
H	closed	open	open
L	open	closed	closed

Figure A.18 (a) Float operated Steam Trap. (b) Intermittent Condensate Pump. (c) Table of the Intermittent Pump states.

The principle of the float operated Steam Trap is rather simple. The function is to stop the steam from leaving the system. This is made by having some amount of condense water between the inlet and the outlet of the trap as seen in Figure A.18 (a). When the level of condense water rise above some level in the trap the float mechanism opens the outlet, letting only water out. When the condensate water is below the minimum level of the float mechanism the trap close the outlet. The Intermittent Pump works in two states, one is filling the pump with condensed water and the other is to empty the tank. Figure A.18 (b) shows the principal of the pump. The difference is that the water is pushed out of the pump, by introducing a steam pressure above the condensation water surface. The valve operation is, like in the steam trap, controlled by a float mechanism. When the pump is emptied and the pump changes the operation mode from H to L (see table in Figure A.18 (c)) the steam in the pump quickly condensate, enabling a rapid pressure drop and thereby making the intermittent pump an effective low pressure atmosphere to normal atmosphere insulator.

B. Conceptions and Definitions

B.1 Units and their Meaning

In this theses we have many symbols and equations and instead of writing down all units we placed them in this appendix. Meanings marked with (∗) is explained further more on next page.

Symbol	Meaning	Unit
A	Total area	m^2
c	Specific heat capacity(∗)	$\frac{J}{kgK}$
k	Overall heat transfer coefficient(∗)	$\frac{W}{m^2K}$
α	Heat transfer coefficient(∗)	$\frac{W}{m^2K}$
λ	Thermal heat conductivity	$\frac{W}{mK}$
δ	Wall thickness(∗)	m
d_h	Hydraulic diameter	m
LMTD	Logarithm Mean Temperature Difference	^oC
Nu	Nusselt number(∗)	No unit
Pr	Prandtl number(∗)	No unit
P	Pressure	Pa
$\dot{q}$	Mass flow	$\frac{kg}{s}$
Re	Reynolds number(∗)	No unit
t	Temperature	^oC
Δt	Temperature difference(∗)	K
ρ	Density	$\frac{kg}{m^3}$
η	Dynamic viscosity	$Pa \cdot s$
		$(1cP = 1mPa \cdot s)$
$\dot{Q}$	Heat flow rate	W

Definitions

Specific heat capacity is the amount of heat in Joule required to increase the temperature of 1 kg for a specific medium with 1 K.

Overall heat transfer coefficient is the total heat transfer ability between two medias trough a separating wall.

Heat transfer coefficient is a measure of the heat conduction trough a boundary layer from a gas or a liquid to a solid body.

Wall thickness is the thickness for each wall in the PHD that separate the primary from the secondary side.

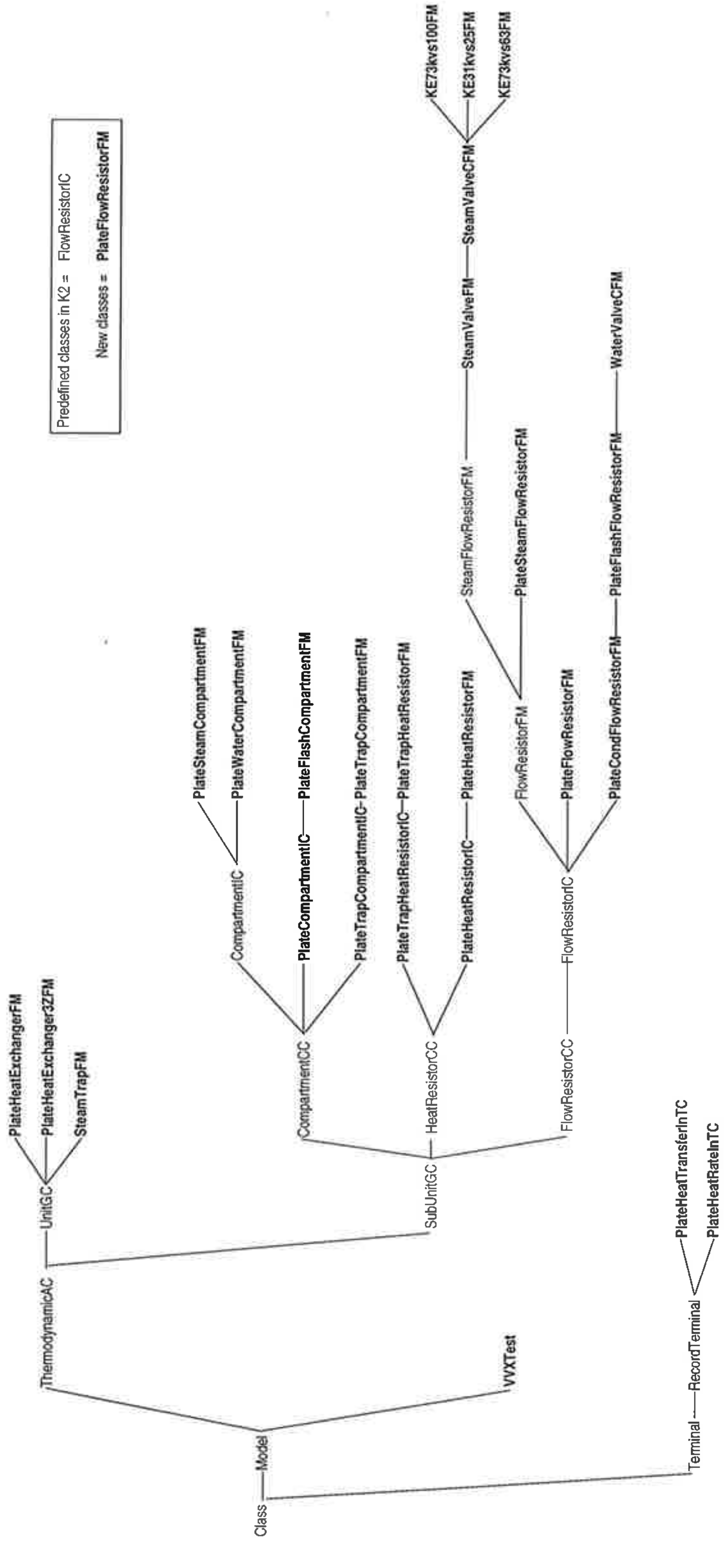
Nusselt number is the dimensionless temperature gradient at the surface and provides a measure of convective heat transfer occurring at the surface. The local Nusselt number is a function of the local Reynolds number and local Prandtl number: $Nu = f(Re, Pr)$, also see equation $A.20$.

Prandtl number provides a measure of the relative effectiveness of momentum and energy transport by diffusion in the velocity and thermal boundary layers, also see equation $A.22$.

Reynolds number further explained in equation $A.21$.

Temperature difference is the difference between the temperature for the two media. This temperature can vary for different places in the PHE. When calculating the amount of transferred heat ($\dot{Q}$) the LMTD is used, that is $\dot{Q}$=k·A·LMTD.

C. Class Tree Structure



D. Omola Code

D.1 VVXTest

```
VVXTest ISA Base::Model WITH
  %% A simulation model of a system with a plate heat exchanger
  %% Steam as warming media at the primary side.
  %% Water on the secondary side.
  %% A steamtrap at the outlet on the primary side.
  %% A steam valve controlled with a PID controller
  %% meassure value from a temperature sensor.
  %% A water valve controls the water flow.
  %%
  %% Made by Ola Löfgren and Patrik Svensson 1997-09

  Wopening ISA Parameter WITH default := 1.0; END;
    % The water valve opening position setpoint
  Ref ISA Parameter WITH default := 310.0; END;
    % The PID reference value setpoint
  Tpid ISA Parameter WITH default := 1.0; END;
    % The time constant to the PID reference value
  TWValve ISA Parameter WITH default := 10.0; END;
    % The time constant to the water valve opening position

icons:
  Graphic ISA super::Graphic;

submodels:
  WaterValveFM2 ISA K2FlowUnitLib::WaterValveFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 101.0;
      y_pos := 151.0;
    END;
    diameter := 0.05;
    Length := 0.1;
    TWValve*y' + y = Wopening;
    DeltaP.initial := 1000;
  END;
  KE31kvs25FM1 ISA ventiler::KE31kvs25FM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 275.0;
    END;
  END;
  PID ISA ControlSystemLib::PIDControllerFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 326.0;
      y_pos := 251.0;
```

```
    END;
  END;
  Plateheatexchangerfm4 ISA vvx::PlateHeatExchangerFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 200.0;
    END;
  END;
  TempSensorFM5 ISA K2SensorLib::TempSensorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 275.0;
      y_pos := 175.0;
    END;
  END;
  SteamTrapFM1 ISA vvx::SteamTrapFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 101.0;
      y_pos := 276.0;
    END;
  END;
terminals:
  T1 ISA K2EndTerminalLib::SteamFlowPinTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 150.0;
      y_pos := 300.0;
    END;
  END;
  T2 ISA K2EndTerminalLib::WaterFlowPoutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 1.0;
      y_pos := 276.0;
    END;
  END;
  T4 ISA K2EndTerminalLib::WaterFlowPoutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 225.0;
      y_pos := 0.0;
    END;
  END;
  T3 ISA K2EndTerminalLib::WaterFlowPinTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 1.0;
      y_pos := 151.0;
    END;
  END;
connections:
  C5 ISA Base::Connection WITH
    T3 AT WaterValveFM2.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [0.0, 150.0; 88.0, 150.0];
  END;
  C10 ISA Base::Connection WITH
    WaterValveFM2.Fout AT Plateheatexchangerfm4.InC;
```

```
    bpoints TYPE STATIC Matrix[3, 2] := [113.0, 151.0; 179.0, 151.0;
179.0, 184.0];
  END;
  C12 ISA Base::Connection WITH
    Plateheatexchangerfm4.OutC AT T4;
    bpoints TYPE STATIC Matrix[2, 2] := [223.0, 184.0; 224.0, 0.0];
  END;
  C13 ISA Base::Connection WITH
    KE31kvs25FM1.Fout AT Plateheatexchangerfm4.InH;
    bpoints TYPE STATIC Matrix[3, 2] := [211.0, 280.0; 223.0, 280.0;
223.0, 215.0];
  END;
  C15 ISA Base::Connection WITH
    T1 AT KE31kvs25FM1.Fin;
    bpoints TYPE STATIC Matrix[3, 2] := [150.0, 299.0; 150.0, 280.0;
187.0, 280.0];
  END;
  C18 ISA Base::Connection WITH
    KE31kvs25FM1.yc AT PID.Control;
    bpoints TYPE STATIC Matrix[4, 2] := [199.0, 285.0; 199.0, 313.0;
342.0, 313.0; 342.0, 250.0];
  END;
  C19 ISA Base::Connection WITH
    Plateheatexchangerfm4.TempOut AT TempSensorFM5.in;
    bpoints TYPE STATIC Matrix[4, 2] := [231.0, 186.0; 248.0, 186.0;
248.0, 174.0; 262.0, 174.0];
  END;
  C20 ISA Base::Connection WITH
    TempSensorFM5.out AT PID.Measure;
    bpoints TYPE STATIC Matrix[3, 2] := [286.0, 174.0; 308.0, 174.0;
308.0, 250.0];
  END;
  C1 ISA Base::Connection WITH
    T2 AT SteamTrapFM1.TrapOut;
    bpoints TYPE STATIC Matrix[2, 2] := [1.0, 275.0; 84.0, 275.0];
  END;
  C2 ISA Base::Connection WITH
    SteamTrapFM1.TrapIn AT Plateheatexchangerfm4.OutH;
    bpoints TYPE STATIC Matrix[5, 2] := [115.0, 275.0; 131.0, 275.0;
131.0, 237.0; 176.0, 237.0; 176.0, 215.0];
  END;

equations:
  Tpid*PID.SetPoint' + PID.SetPoint = Ref;

END;
```

D.2 PlateHeatExchangerFM

```
PlateHeatExchangerFM ISA K2ClassTreeLib::UnitGC WITH
  %% A model of a heat exchanger with steam on the primary
  %% side and water on the secondary.
  %%
  %% Assumptions: constant volume of the exchanger
  %%
  %%              no work interaction.
  %% Model Use:   given mass flow directions.
  %% States:      enthalpy (h) on primary side.
  %%              enthalpy (h) on secondary side.
  %% Medium:      Steam(Flash) and water.
  %% Model type:  full.
  %%
  %%
  %%
  %% Ola Löfgren & Patrik Svensson 1997
icons:
  Graphic ISA Base::Layout WITH bitmap TYPE String := "HeatExchanger"; END;

parameters:
  NbrOfWaterChan ISA Parameter WITH default := 11; END;
  NbrOfSteamChan ISA Parameter WITH default := 12; END;
  Primlength ISA Parameter WITH default := 0.25;END;
  Primdiameter ISA Parameter WITH default := 0.042;END;
    % The flow resistance between the plates is modelled
    % as a pipe, with diameter and length.
  sef ISA Parameter WITH default := 1.18; END;
    % The surface enlargement factor
  PlateArea ISA Parameter WITH default := 0.025; END;
    % The area of the plate uncorrugated
  delta ISA Parameter WITH default := 0.0004; END;
    % The thickness of the walls in the PHE
  Vprim ISA Parameter;
    % Total volume of the primary side
  Vsec ISA Parameter;
    % Total volume of the secondary side
  Vinlet ISA Parameter WITH default := 0.0003; END;
    % The volume of the inlet pipe of the prim. side.
  PlateHeight ISA Parameter WITH default := 0.25; END;
  Height ISA Parameter WITH default := 1.25; END;
    % The outlet pipe height over ref.level on the PHE
  HotFlength ISA Parameter WITH default := 2.0; END;
  diameter ISA Parameter WITH default := 0.03; END;
    % HotFlength and diameter determines the flow resistance
    % in the flow resistor HotF.
  NrPlates ISA Parameter WITH default := 24; END;
  ChanInletArea ISA Parameter WITH default := 0.0002; END;
  CRe ISA Parameter WITH default := 0.079; END;
  nRe ISA Parameter WITH default := 0.75; END;
  nPr ISA Parameter WITH default := 0.4; END;
```

```
  Vprim := 5e-05*NbrOfSteamChan;
  Vsec  := 5e-05*NbrOfWaterChan;

parameter_propagation:
  HotF.diameter := diameter;
  HotF.Length := HotFlength;
  S2F.length := Primlength;
  S2F.diameter := Primdiameter;
  HeatF.delta := delta;
  HeatF.NbrOfWaterChan := NbrOfWaterChan;
  HeatF.ChanInletArea := ChanInletArea;
  HeatF.CRe := CRe;
  HeatF.nRe := nRe;
  HeatF.nPr := nPr;
  HotC.plateh := PlateHeight;
  HotC.TotArea := sef*PlateArea;
  HotC.Height := Height;
  HotC.NrPlates := NrPlates;
  HotC.V := Vprim;
  ColdC.V := Vsec;
  ColdC.Height := Height;
  ColdC.plateh := PlateHeight;
  ColdF.NbrOfWaterChan := NbrOfWaterChan;
  HotInlet.V := Vinlet;

submodels:
  HotC ISA vvx::PlateFlashCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 150.0;
      y_pos := 225.0;
    END;
  END;
  HeatF ISA vvx::PlateHeatResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 150.0;
      y_pos := 150.0;
    END;
    dH := 4e-3;
    rhoK := HotC.M.Mout2.rhow;
    H := HotC.plateh;
  END;
  ColdF ISA vvx::PlateFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 275.0;
      y_pos := 75.0;
    END;
  END;
  ColdC ISA vvx::PlateWaterCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 150.0;
      y_pos := 75.0;
```

```
    END;
  END;
  HotF ISA PlateFlashFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 75.0;
      y_pos := 225.0;
    END;
  END;
  S2F ISA vvx::PlateSteamFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 225.0;
      y_pos := 225.0;
    END;
  END;
  HotInlet ISA PlateSteamCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 300.0;
      y_pos := 225.0;
    END;
  END;
terminals:
  OutH ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 50.0;
      y_pos := 300.0;
    END;
  END;
  InH ISA K2TerminalLib::FlowInTC WITH
    M ISA K2TerminalLib::SteamMediumTC; % Steam media
    Graphic ISA super::Graphic WITH
      x_pos := 351.0;
      y_pos := 301.0;
    END;
  END;
  InC ISA K2TerminalLib::FlowInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 75.0;
      y_pos := 0.0;
    END;
  END;
  OutC ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 350.0;
      y_pos := 0.0;
    END;
  END;
  TempOut ISA K2TerminalLib::TemperatureTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400.0;
      y_pos := 25.0;
    END;
  END;
```

```
connections:
  %% Non-graphical connections
  TempOut := ColdC.M.Tc;
    % The temperature at the outlet of secondary side
    % Use for temperature sensor connection

  %% Graphical connections
  C1 ISA Base::Connection WITH
    OutH AT HotF.Fout;
    bpoints TYPE STATIC Matrix[3, 2] := [50.0, 300.0; 50.0, 226.0;
50.0, 224.0];
  END;
  C7 ISA Base::Connection WITH
    InC AT ColdC.Fin;
    bpoints TYPE STATIC Matrix[3, 2] := [75.0, 0.0; 75.0, 74.0; 125.0,
74.0];
  END;
  C8 ISA Base::Connection WITH
    ColdC.Fout AT ColdF.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [174.0, 74.0; 250.0, 74.0];
  END;
  C10 ISA Base::Connection WITH
    HeatF.hout AT ColdC.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [149.0, 174.0; 149.0, 74.0];
  END;
  C11 ISA Base::Connection WITH
    HeatF.hin AT HotC.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [149.0, 125.0; 149.0, 224.0];
  END;

  C3 ISA Base::Connection WITH
    HotC.HTran AT HeatF.HTran;
    bpoints TYPE STATIC Matrix[3, 2] := [125.0, 200.0; 125.0, 149.0;
137.0, 149.0];
  END;
  C4 ISA Base::Connection WITH
    ColdF.Fout AT OutC;
    bpoints TYPE STATIC Matrix[3, 2] := [299.0, 74.0; 349.0, 74.0;
349.0, 0.0];
  END;
  C9 ISA Base::Connection WITH
    HotF.Fin AT HotC.Fout;
    bpoints TYPE STATIC Matrix[2, 2] := [99.0, 224.0; 174.0, 224.0];
  END;
  C13 ISA Base::Connection WITH
    HotC.Fin AT S2F.Fout;
    bpoints TYPE STATIC Matrix[2, 2] := [125.0, 224.0; 249.0, 224.0];
  END;
  C14 ISA Base::Connection WITH
    S2F.Fin AT HotInlet.Fout;
    bpoints TYPE STATIC Matrix[2, 2] := [200.0, 224.0; 324.0, 224.0];
  END;
```

```
  C2 ISA Base::Connection WITH
    HotInlet.Fin AT InH;
    bpoints TYPE STATIC Matrix[3, 2] := [275.0, 225.0; 350.0, 225.0;
350.0, 300.0];
  END;
END;
```

D.3 PlateHeatExchanger3ZFM

```
PlateHeatExchanger3ZFM ISA K2ClassTreeLib::UnitGC WITH
  %% A model of a heat exchanger with steam on the primary
  %% side and water on the secondary.
  %%
  %% The model is divided into three zones.
  %%
  %% Assumptions: constant volume of the exchanger
  %%
  %%              no work interaction.
  %% Model Use:   given mass flow directions.
  %% States:      enthalpy (h) on primary side.
  %%              enthalpy (h) on secondary side.
  %% Medium:      Steam(Flash) and water.
  %% Model type:  full.
  %%
  %%
  %%
  %% Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH bitmap TYPE String := "HeatExchanger"; END;

parameters:
  NbrOfWaterChan ISA Parameter WITH default := 11; END;
  NbrOfSteamChan ISA Parameter WITH default := 12; END;
  Primlength ISA Parameter WITH default := 0.25; END;
  Primdiameter ISA Parameter WITH default := 0.042; END;
  % The flow resistance between the plates is modelled
  % as a pipe, with diameter and length.
  sef ISA Parameter WITH default := 1.18; END;
  % The surface enlargement factor
  PlateArea ISA Parameter WITH default := 0.025; END;
  % The area of the plate uncorrugated
  delta ISA Parameter WITH default := 0.0004; END;
  % The thickness of the walls in the PHE
  Vprim ISA Parameter;
  % Total volume of the primary side
  Vsec ISA Parameter;
  % Total volume of the secondary side
  Vinlet ISA Parameter WITH default := 0.0003; END;
```

```
    % The volume of the inlet pipe of the prim. side.
    PlateHeight ISA Parameter WITH default := 0.25; END;
    Height ISA Parameter WITH default := 1.25; END;
    % The outlet pipe height over ref.level on the PHE
    HotFlength ISA Parameter WITH default := 2.0; END;
    diameter ISA Parameter WITH default := 0.03; END;
    % HotFlength and diameter determines the flow resistance
    % in the flow resistor HotF.
    NrPlates ISA Parameter WITH default := 24; END;
    Vprim := 5e-05*NbrOfSteamChan;
    Vsec := 5e-05*NbrOfWaterChan;

  parameter_propagation:
    HotF.diameter := diameter;
    HotF.Length := HotFlength;
    S2F.Length := Primlength;
    S2F.diameter := Primdiameter;
    HeatFZonA.delta := delta;
    HeatFZonA.NbrOfWaterChan := NbrOfWaterChan;
    HeatFZonB.delta := delta;
    HeatFZonB.NbrOfWaterChan := NbrOfWaterChan;
    HeatFZonC.delta := delta;
    HeatFZonC.NbrOfWaterChan := NbrOfWaterChan;
    HotC.plateh := PlateHeight;
    HotC.TotArea := sef*PlateArea;
    HotC.Height := Height;
    HotC.NrPlates := NrPlates;
    HotC.V := Vprim;
    ColdCZonA.V := Vsec;
    ColdFZonA.NbrOfWaterChan := NbrOfWaterChan;
    ColdCZonB.V := Vsec;
    ColdFZonB.NbrOfWaterChan := NbrOfWaterChan;
    ColdCZonC.V := Vsec;
    ColdFZonC.NbrOfWaterChan := NbrOfWaterChan;
    HotInlet.V := Vinlet;

  submodels:
    HotInlet ISA vvx::PlateSteamCompartmentFM WITH
      Graphic ISA super::Graphic WITH
        x_pos := 300.0;
        y_pos := 250.0;
      END;
    END;
    S2F ISA vvx::PlateSteamFlowResistorFM WITH
      Graphic ISA super::Graphic WITH
        x_pos := 226.0;
        y_pos := 251.0;
      END;
    END;
    HotF ISA vvx::PlateFlashFlowResistorFM WITH
      Graphic ISA super::Graphic WITH
        x_pos := 75.0;
```

```
      y_pos := 250.0;
    END;
  END;
  HotC ISA vvx::PlateFlashCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 151.0;
      y_pos := 251.0;
    END;
  END;
  ColdCZonC ISA vvx::PlateWaterCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 101.0;
      y_pos := 101.0;
    END;
  END;
  ColdCZonB ISA vvx::PlateWaterCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 100.0;
    END;
  END;
  ColdCZonA ISA vvx::PlateWaterCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 300.0;
      y_pos := 100.0;
    END;
  END;
  ColdFZonA ISA vvx::PlateFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 325.0;
      y_pos := 50.0;
    END;
  END;
  ColdFZonB ISA vvx::PlateFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 251.0;
      y_pos := 51.0;
    END;
  END;
  ColdFZonC ISA vvx::PlateFlowResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 150.0;
      y_pos := 50.0;
    END;
  END;
  HeatFZonC ISA vvx::PlateHeatResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 101.0;
      y_pos := 176.0;
    END;
    dH := 0.004;
    rhoK := HotC.M.Mout2.rhow;
```

```
    H := HotC.plateh;
  END;
  HeatFZonB ISA vvx::PlateHeatResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 175.0;
    END;
    dH := 0.004;
    rhoK := HotC.M.Mout2.rhow;
    H := HotC.plateh;
  END;
  HeatFZonA ISA vvx::PlateHeatResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 301.0;
      y_pos := 176.0;
    END;
    dH := 0.004;
    rhoK := HotC.M.Mout2.rhow;
    H := HotC.plateh;
  END;

terminals:
  InH ISA K2TerminalLib::FlowInTC WITH
    M ISA K2TerminalLib::SteamMediumTC;
    % Steam media
    Graphic ISA super::Graphic WITH
      x_pos := 350.0;
      y_pos := 300.0;
    END;
  END;
  OutH ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 51.0;
      y_pos := 301.0;
    END;
  END;
  OutC ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 350.0;
      y_pos := 0.0;
    END;
  END;
  InC ISA K2TerminalLib::FlowInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 76.0;
      y_pos := 1.0;
    END;
  END;
  TempOut ISA K2TerminalLib::TemperatureTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400.0;
      y_pos := 25.0;
```

```
    END;
  END;

connections:
  %% Non-graphical connections
%  TempOut := ColdCZonA.M.Mout.Tkout;
  TempOut := ColdCZonA.M.Tc;
  % The temperature at the outlet of secondary side
  % Use for temperature sensor connection

  %% Graphical connections
  C2 ISA Base::Connection WITH
    HotInlet.Fout AT S2F.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [324.0, 249.0; 200.0, 250.0];
  END;
  C3 ISA Base::Connection WITH
    S2F.Fout AT HotC.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [249.0, 250.0; 125.0, 250.0];
  END;
  C4 ISA Base::Connection WITH
    HotC.Fout AT HotF.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [174.0, 250.0; 50.0, 249.0];
  END;
  C5 ISA Base::Connection WITH
    HotF.Fout AT OutH;
    bpoints TYPE STATIC Matrix[3, 2] := [99.0, 249.0; 50.0, 249.0;
50.0, 300.0];
  END;
  C6 ISA Base::Connection WITH
    HotC.QinZonC AT HeatFZonC.hin;
    bpoints TYPE STATIC Matrix[4, 2] := [125.0, 225.0; 125.0, 214.0;
100.0, 214.0; 100.0, 150.0];
  END;
  C7 ISA Base::Connection WITH
    HotC.QinZonB AT HeatFZonB.hin;
    bpoints TYPE STATIC Matrix[4, 2] := [150.0, 225.0; 150.0, 212.0;
199.0, 212.0; 199.0, 150.0];
  END;
  C8 ISA Base::Connection WITH
    HotC.QinZonA AT HeatFZonA.hin;
    bpoints TYPE STATIC Matrix[5, 2] := [174.0, 225.0; 174.0, 218.0;
176.0, 218.0; 300.0, 218.0; 300.0, 150.0];
  END;
  C9 ISA Base::Connection WITH
    HeatFZonC.hout AT ColdCZonC.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [100.0, 199.0; 100.0, 100.0];
  END;
  C10 ISA Base::Connection WITH
    HeatFZonB.hout AT ColdCZonB.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [199.0, 199.0; 199.0, 99.0];
  END;
  C11 ISA Base::Connection WITH
```

```
    HeatFZonA.hout AT ColdCZonA.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [300.0, 199.0; 299.0, 99.0];
  END;
  C12 ISA Base::Connection WITH
    InC AT ColdCZonC.Fin;
    bpoints TYPE STATIC Matrix[5, 2] := [75.0, 0.0; 75.0, 23.0; 47.0,
23.0; 47.0, 100.0; 75.0, 100.0];
  END;
  C13 ISA Base::Connection WITH
    ColdCZonC.Fout AT ColdFZonC.Fin;
    bpoints TYPE STATIC Matrix[6, 2] := [124.0, 100.0; 134.0, 100.0;
134.0, 71.0; 114.0, 71.0; 114.0, 49.0; 125.0, 49.0];
  END;
  C14 ISA Base::Connection WITH
    ColdFZonC.Fout AT ColdCZonB.Fin;
    bpoints TYPE STATIC Matrix[6, 2] := [174.0, 49.0; 187.0, 49.0;
187.0, 71.0; 167.0, 71.0; 167.0, 99.0; 175.0, 99.0];
  END;
  C15 ISA Base::Connection WITH
    ColdCZonB.Fout AT ColdFZonB.Fin;
    bpoints TYPE STATIC Matrix[6, 2] := [224.0, 99.0; 234.0, 99.0;
234.0, 71.0; 216.0, 71.0; 216.0, 50.0; 225.0, 50.0];
  END;
  C17 ISA Base::Connection WITH
    ColdFZonB.Fout AT ColdCZonA.Fin;
    bpoints TYPE STATIC Matrix[6, 2] := [274.0, 50.0; 283.0, 50.0;
283.0, 72.0; 266.0, 72.0; 266.0, 99.0; 275.0, 99.0];
  END;
  C18 ISA Base::Connection WITH
    ColdCZonA.Fout AT ColdFZonA.Fin;
    bpoints TYPE STATIC Matrix[6, 2] := [324.0, 99.0; 336.0, 99.0;
336.0, 71.0; 294.0, 71.0; 294.0, 49.0; 300.0, 49.0];
  END;
  C20 ISA Base::Connection WITH
    ColdFZonA.Fout AT OutC;
    bpoints TYPE STATIC Matrix[2, 2] := [349.0, 49.0; 350.0, -1.0];
  END;
  C1 ISA Base::Connection WITH
    InH AT HotInlet.Fin;
    bpoints TYPE STATIC Matrix[3, 2] := [349.0, 299.0; 349.0, 250.0;
275.0, 250.0];
  END;
END;
```

D.4 SteamTrapFM

```
SteamTrapFM ISA K2ClassTreeLib::UnitGC WITH
                      %%Base::Model WITH
  %% A model of a steam trap with float mechanism
  %%
  %% Assumptions: constant volume of the exchanger
  %%
  %%              no work interaction.
  %% Model Use:   given mass flow directions.
  %% States:      enthalpy (h).
  %%
  %% Medium:      Steam(Flash) and water.
  %% Model type:  full.
  %%
  %% Made by Ola Lõfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "SteamTrap";
  END;

submodels:
  TrapValve ISA ventiler::WaterValveCFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 151.0;
      y_pos := 226.0;
    END;
  END;
  TrapC ISA vvx::PlateTrapCompartmentFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 250.0;
      y_pos := 225.0;
    END;
  END;
  TrapHeatLoss ISA vvx::PlateTrapHeatResistorFM WITH
    Graphic ISA super::Graphic WITH
      x_pos := 251.0;
      y_pos := 151.0;
    END;
  END;

terminals:
  TrapIn ISA K2TerminalLib::FlowInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400.0;
      y_pos := 150.0;
    END;
  END;
  TrapOut ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 0.0;
```

```
      y_pos := 150.0;
    END;
  END;

connections:
  C1 ISA Base::Connection WITH
    TrapC.Fout AT TrapValve.Fin;
    bpoints TYPE STATIC Matrix[2, 2] := [274.0, 224.0; 138.0, 225.0];
  END;
  C2 ISA Base::Connection WITH
    TrapC.TrapOpening AT TrapValve.y;
    bpoints TYPE STATIC Matrix[3, 2] := [225.0, 200.0; 150.0, 200.0;
150.0, 217.0];
  END;

  C3 ISA Base::Connection WITH
    TrapValve.Fout AT TrapOut;
    bpoints TYPE STATIC Matrix[4, 2] := [162.0, 225.0; 79.0, 225.0;
79.0, 150.0; -1.0, 150.0];
  END;
  C4 ISA Base::Connection WITH
    TrapC.Fin AT TrapIn;
    bpoints TYPE STATIC Matrix[4, 2] := [225.0, 224.0; 313.0, 224.0;
313.0, 150.0; 400.0, 150.0];
  END;

  C5 ISA Base::Connection WITH
    TrapHeatLoss.hin AT TrapC.Qin;
    bpoints TYPE STATIC Matrix[2, 2] := [250.0, 151.0; 250.0, 225.0];
  END;
END;
```

D.5 PlateSteamCompartmentFM

```
PlateSteamCompartmentFM ISA K2CompartmentLib::CompartmentIC WITH
  %% A control volume model of a
  %%   superheated steam medium based on
  %%   dynamic energy and mass balances.
  %%
  %% Assumptions: constant volume,
  %%              homogenous mixed,
  %%              no momentum description,
  %%              no work interaction.
  %% Model Use:   given mass flow directions,
  %%              mass flows are described elsewhere.
  %% States:      pressure (p),
  %%              enthalpy (h).
  %% Medium:      superheated steam.
  %% Model type:  full model.
```

```
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icon:
  Graphic ISA base::Layout WITH
    bitmap TYPE String := "SteamCompartment";
  END;

parameters:
  V ISA Parameter;

submodels:
  Fin ISA super::Fin WITH
    M ISA K2TerminalLib::SteamMediumTC;
  END;
  Fout ISA super::Fout WITH
    M ISA K2TerminalLib::SteamMediumTC;
  END;

variables:
  % states: p = pressure, h = enthalpy
  p ISA Variable;
  h ISA Variable;
  % balance derivitives
  dm, de ISA Variable;
  dV ISA Variable;
  % auxilary variables
  K1, K2 ISA Variable;
  T11, T12, T13 ISA Variable;
  T21, T22, T23 ISA Variable;

equations:
  % ------------------------------------- mass balance
  dm = Fin.w - Fout.w;
  % ------------------------------------- energy balance
  de = Fin.w*Fin.h - Fout.w*Fout.h + Qin.R.q;
  dV := 0;
  % ------------------------------------- auxiliary variables
  K1 = M.Mout.ap*M.Mout.rho + M.Mout.ah;
  K2 = M.Mout.rho*M.Mout.rho/p + M.Mout.ah;
  T11 = (M.Mout.rho + M.Mout.ah*h)/K1/V;
  T12 = -M.Mout.ah/K1/V;
  T13 = -p/K1/V*K2;
  T21 = (T11 - h/V)/M.Mout.rho;
  T22 = (1 - M.Mout.ah/K1)/M.Mout.rho/V;
  T23 = (1 - K2/K1)*p/M.Mout.rho/V;
  % -------------------- transform into pressure and enthalpy
  p' = T11*dm + T12*de + T13*dV;
  Fin.p = p;
  Fout.p = p;
  h' = T21*dm + T22*de + T23*dV;
  Fout.h = h;
```

```
medium:
  M ISA K2MediumLib::SuperheatedSteamMM;

medium_connections:
  %------------------------- medium state to medium model
  M.Min.p    := p;
  M.Min.hout := Fout.h;
  M.Min.hin  := Fin.h;
  %------------------------- medium temp to heat transfer
  Qin.R.Tin  := M.Mout.Tkin;
  Qin.R.Tout := M.Mout.Tkout;
  % ----------------------------------- medium terminals
  Fin.M.T    := M.Mout.Tkin;
  Fout.M.T   := TP(p);
    % Saturated steam temperature

heat_medium_connections:
  Qin.M.p    = p;
  Qin.M.w    = Fout.w;
  Qin.M.Gmix := [0;0;0;0;0;0;1.0];
END;
```

D.6 PlateWaterCompartmentFM

```
PlateWaterCompartmentFM ISA K2CompartmentLib::CompartmentIC WITH
  %% A control volume model of a liquid
  %%   medium (water) based on
  %%   dynamic energy and static mass balances.
  %%
  %% Assumptions: constant volume,
  %%              homogenous mixed,
  %%              incompressible (static pressure balance),
  %%              no momentum description,
  %%              no heat interaction,
  %%              no work interaction.
  %% Model Use:   given mass flow direction,
  %%              mass flows are described elsewhere.
  %% States:      enthalpy (h).
  %% Medium:      subcooled water (h < 4.1e5).
  %% Model type:  full.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "WaterCompartment";
  END;
```

```
parameters:
  V ISA Parameter;
  height, plateh ISA Parameter;

variables:
  % states: p = pressure, h = enthalpy
  h ISA Variable;
  p ISA Variable;
  % balance derivitives
  dm ISA Variable;
  de ISA Variable;

equations:
  % ------------------------------------- mass balance (static)
  Fin.w = Fout.w;
  dm := 0;
  % ------------------------------------- energy balance (dynamic)
  de = Fin.w*Fin.h - Fout.w*Fout.h + Qin.R.q;
  % ------------------------------------- state equations (enthalpy)
  Fin.p = p;
  Fout.p = p;
  h' = 1/(M.Mout.rho*V)*de;
  Fout.h = h;

medium:
  M ISA K2MediumLib::WaterConstMM;

medium_connections:
  %------------------------- medium state to medium model
  M.Min.p    = p;
  M.Min.hout = Fout.h;
  M.Min.hin  = Fin.h;
  %------------------------- medium temp to heat transfer
  Qin.R.Tin  = M.Mout.Tkin;
  Qin.R.Tout = M.Mout.Tkout;
  % ------------------------------------- medium terminals
  Fin.M.z    := height;
  Fout.M.z   := height+plateh;

heat_medium_connections:
  Qin.M.p    := p;
  Qin.M.w    := Fout.w;
  Qin.M.Gmix := [0;0;0;0;0;0;1.0];
END;
```

D.7 PlateCompartmentIC

```
PlateCompartmentIC ISA K2ClassTreeLib::CompartmentCC WITH
  %% Interface class of a compartment model with
  %%   one inflow and one outflow.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997

icon:
  Graphic ISA super::Graphic WITH
    bitmap TYPE String := "WaterCompartment";
  END;

terminals:
  Fin ISA K2TerminalLib::FlowInTC WITH
    M ISA K2TerminalLib::SteamMediumTC; % Change to steam media
    Graphic ISA super::Graphic WITH
      x_pos := 1;
      y_pos := 151;
    END;
  END;
  Fout ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400;
      y_pos := 151;
    END;
  END;
  Qin ISA PlateHeatTransferInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 201;
      y_pos := 151;
      invisible := 1;
    END;
  END;
  HTran ISA RecordTerminal WITH
      % Contains the effective heat transfer area and the
      % saturation temperature of the steam flowing in at
      % the primary side of the PHE.
    Area,Tin ISA SimpleOutput;
    Graphic ISA Base::Layout WITH
      x_pos := 1;
      y_pos := 1;
    END;
  END;

END;
```

D.8 PlateFlashCompartmentFM

```
PlateFlashCompartmentFM ISA PlateCompartmentIC WITH
  %% A control volume model of a mixture of
  %%   saturated water and steam described by
  %%   dynamic energy and mass balances.
  %%
  %% Assumptions: constant volume,
  %%              homogenous mixed,
  %%              no work interaction.
  %% Model Use:   given mass flow directions,
  %%              mass flows are described elsewhere.
  %% States:      pressure (p),
  %%              enthalpy (h).
  %% Medium:      saturated water and steam.
  %% Model type:  full.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "FlashCompartment";
  END;


parameters:
  V, height, plateh, TotArea, NrPlates ISA Parameter;

variables:
  p ISA Variable;                        % Pressure
  h ISA Variable;                        % Enthalpy
  dm, de ISA Variable;                   % Balance equation vars
  dV, K1, K2 ISA Variable;               % aux variables
  T11, T12, T13 ISA Variable;
  T21, T22, T23 ISA Variable;
  level ISA Variable;
  hproc, levelweight ISA Variable;
    % hproc: Enthalpy mix out (hw -> h)
    % levelweight: weightingfactor on level depending on the pressure
  ackmassa ISA Variable WITH initial := 0; END;
    % A test variable

equations:
  levelweight = -0.33/1.037*atan(60*(p/101325)-62)+0.506;
  level = levelweight*plateh*(1 - M.Mout2.alpha);
    % The level of condensate
  hproc = -0.35/1.037*atan(35*(level/plateh)-6.825)+0.519;
    % Desides the mix of water and steam enthalphy that
    % flows out of the primary side.
  HTran.Area =(NrPlates-2) * TotArea*(plateh-level)/plateh;
    % Calculation of the effective heat transfer area.
    % (Area free from condensate)
```

```
  % ------------------------------------- mass balance
  dm = Fin.w - Fout.w;
  ackmassa' = dm;
  % ------------------------------------- energy balance
  de = Fin.w*Fin.h - Fout.w*Fout.h + Qin.R.q;
  dV := 0;
  % ------------------------------------- auxiliary variables
  K1 = M.Mout.ap*M.Mout.rho + M.Mout.ah;
  K2 = M.Mout.rho*M.Mout.rho/p + M.Mout.ah;
  T11 = (M.Mout.rho + M.Mout.ah*h)/K1/V;
  T12 = -M.Mout.ah/K1/V;
  T13 = -p/K1/V*K2;
  T21 = (T11 - h/V)/M.Mout.rho;
  T22 = (1 - M.Mout.ah/K1)/M.Mout.rho/V;
  T23 = (1 - K2/K1)*p/M.Mout.rho/V;
  % ------------------------- pressure/enthalpy transformation
  p' = T11*dm + T12*de + T13*dV;
  Fin.p = p;
  Fout.p = M.Mout2.rhow*K2BasicLib::g*level+p;
  h' = T21*dm + T22*de + T23*dV;
  Fout.h = M.Mout2.hw + (h - M.Mout2.hw)* hproc;
      % Enthalpy mix out (hw -> h)

medium:
  M ISA K2MediumLib::SaturatedWaterSteamMM;

medium_connections:
  % ------------------------------------- medium states
  M.Min.p := p;
  M.Min.hin := Fin.h;
  M.Min.hout := h;
  %------------------------- medium temp to heat transfer
  Qin.R.Tout   := M.Mout.Tkout;
  % ------------------------------------- medium terminals
  HTran.Tin := Fin.M.T;
  Fout.M.z := height;
END;
```

D.9 PlateTrapCompartmentIC

```
PlateTrapCompartmentIC ISA K2ClassTreeLib::CompartmentCC WITH
  %% Interface class of a compartment model with
  %%   one inflow and one outflow.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icon:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "FlashCompartment";
```

```
  END;
terminals:
  Fin ISA K2TerminalLib::FlowInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 1;
      y_pos := 151;
    END;
  END;
  Fout ISA K2TerminalLib::FlowOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400;
      y_pos := 151;
    END;
  END;
  Qin ISA Base::RecordTerminal WITH
     R ISA Base::RecordTerminal WITH
       q ISA K2TerminalLib::HeatRateTC;
       Tout ISA K2TerminalLib::TemperatureTC;
       Tin ISA K2TerminalLib::TemperatureTC;
     END;
    Graphic ISA super::Graphic WITH
      x_pos := 201;
      y_pos := 151;
      invisible := 1;
    END;
  END;
  TrapOpening ISA SimpleOutput WITH
    % Contains the trap valve opening position setpoint
    Graphic ISA Base::Layout WITH
      x_pos := 1;
      y_pos := 1;
    END;
  END;
END;
```

D.10 PlateTrapCompartmentFM

```
PlateTrapCompartmentFM ISA PlateTrapCompartmentIC WITH
  %% A control volume model of a mixture of
  %%   saturated water and steam described by
  %%   dynamic energy and mass balances.
  %%
  %% Assumptions: constant volume,
  %%              homogenous mixed,
  %%              no work interaction.
  %% Model Use:   given mass flow directions,
  %%              mass flows are described elsewhere.
  %% States:      pressure (p),
  %%              enthalpy (h).
  %% Medium:      saturated water and steam.
```

```
  %% Model type:  full.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

parameters:
  V, height ISA Parameter;
  trapheight ISA Parameter WITH default := 0.14; END;
  TimeTrap ISA Parameter WITH default := 1.0; END;

variables:
  yRef ISA Variable WITH initial := 0.0; END;
    % The float mechanism setpoint to the trap
    % opening position. yRef is fed into a function
    % with a time constant TimeTrap, limiting the
    % the derivatives.
  p ISA Variable;                        % Pressure
  h ISA Variable;                        % Enthalpy
  dm, de ISA Variable;                   % Balance equation vars
  dV, K1, K2 ISA Variable;               % aux variables
  T11, T12, T13 ISA Variable;
  T21, T22, T23 ISA Variable;
  level ISA Variable;
    % The level of condensate in the steam trap
  ackmassa ISA Variable WITH initial := 0; END;
    % Test variable
  TrapProc ISA Variable WITH initial := 0; END;
    % Trap opening position

  %% Media description variables used for transition between
  %% the two mediatypes SaturatedWaterSteamMM and WaterVarMM.
  %%
  Full TYPE DISCRETE Integer;
    % Indicates if the trap is full i.e. maximum level
  rho TYPE Real;
  rhow TYPE Real;
  ah TYPE Real;
  ap TYPE Real;
  alpha TYPE Real;
  Tkin TYPE Real;
  Tkout TYPE Real;

equations:
  % ------------------------------------ media calculations
  WHEN ((NOT Full) AND (h < Msat.Mout2.hw)) DO
    NEW(Full) := 1;
  END;
  WHEN (Full AND (h > Msat.Mout2.hw)) DO
    New(Full) := 0;
  END;
  alpha := IF Full THEN 0 ELSE Msat.Mout2.alpha;
  ah := IF Full THEN Mwat.Mout.ah ELSE Msat.Mout.ah;
  ap := IF Full THEN Mwat.Mout.ap ELSE Msat.Mout.ap;
```

```
  rho := IF Full THEN Mwat.Mout.rho ELSE Msat.Mout.rho;
  rhow := IF Full THEN Mwat.Mout.rho ELSE Msat.Mout2.rhow;
  Tkin := IF Full THEN Mwat.Mout.Tkin ELSE Msat.Mout.Tkin;
  Tkout := IF Full THEN Mwat.Mout.Tkout ELSE Msat.Mout.Tkout;

  % -------------------------------------
  level = trapheight*(1 - alpha);        % Water level
  yRef := IF (level<(trapheight/2)) THEN 0 ELSE
             (level-(trapheight/2))/(trapheight/2);
  TimeTrap* TrapProc' +  TrapProc = yref;
  TrapOpening := TrapProc;
  % ------------------------------------- mass balance
  dm = Fin.w - Fout.w;
  ackmassa' = dm;
  % ------------------------------------- energy balance
  de = Fin.w*Fin.h - Fout.w*Fout.h + Qin.R.q;
  dV := 0;
  % ------------------------------------- auxiliary variables
  K1 = ap*rho + ah;
  K2 = rho*rho/p + ah;
  T11 = (rho + ah*h)/K1/V;
  T12 = -ah/K1/V;
  T13 = -p/K1/V*K2;
  T21 = (T11 - h/V)/rho;
  T22 = (1 - ah/K1)/rho/V;
  T23 = (1 - K2/K1)*p/rho/V;
  % ------------------------------------- pressure/enthalpy transformation
  p' = T11*dm + T12*de + T13*dV;
  Fin.p = p;
  Fout.p = rhow*K2BasicLib::g*level+p;
  h' = T21*dm + T22*de + T23*dV;
  Fout.h = h;

medium:
  Msat ISA K2MediumLib::SaturatedWaterSteamMM;
  Mwat ISA K2MediumLib::WaterVarMM;

medium_connections:
  % ------------------------------------- medium states
  Msat.Min.p := p;
  Msat.Min.hin := Fin.h;
  Msat.Min.hout := h;
  Mwat.Min.p := p;
  Mwat.Min.hin := Fin.h;
  Mwat.Min.hout := h;
  %------------------------- medium temp to heat transfer
  Qin.R.Tin    := Tkin;
  Qin.R.Tout   := Tkout;
  % ------------------------------------- medium terminals
  Fout.M.z := height;
  Fin.M.z := Fout.M.z;
END;
```

D.11 PlateTrapHeatResistorIC

```
PlateTrapHeatResistorIC ISA K2ClassTreeLib::HeatResistorCC WITH
%% This is an interface class for
%%   heat resistor models.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09

icon:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "TrapHeatResistor";
  END;

terminals:
  Hin ISA Base::RecordTerminal WITH
    R ISA Base::RecordTerminal WITH
      q ISA K2TerminalLib::HeatRateTC;
      Tout ISA K2TerminalLib::TemperatureTC;
      Tin ISA K2TerminalLib::TemperatureTC;
    END;
    Graphic ISA super::Graphic WITH
      x_pos := 201;
      y_pos := 160;
    END;
  END;

END;
```

D.12 PlateTrapHeatResistorFM

```
PlateTrapHeatResistorFM ISA PlateTrapHeatResistorIC WITH
%% A complex heat transfer model with
%%   separate submodels for convective
%%   and conductive heat resistance.
%%
%% Connect the SteamTrap to Hin
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09


functions:
  dTm ISA K2BasicLib::LogMean WITH
    x = Hin.R.Tin - 303;
    y = Hin.R.Tout - 293;
  END;

parameters:
  k ISA Parameter WITH default := 500;END;
  Area ISA Parameter WITH default := 0.02; END;
```

```
equations:
  Hin.R.q := dTm*Area*k;
END;
```

D.13 PlateHeatResistorIC

```
PlateHeatResistorIC ISA K2ClassTreeLib::HeatResistorCC WITH
%% This is an interface class for
%%   heat resistor models.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09

icon:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "HeatResistor";
  END;
terminals:
  Hin  ISA PlateHeatTransferInTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 201;
      y_pos := 1;
    END;
  END;
  Hout ISA K2TerminalLib::HeatTransferOutTC WITH
    Graphic ISA super::Graphic WITH
      x_pos := 201;
      y_pos := 300;
    END;
  END;
  HTran ISA RecordTerminal WITH
    Area,Tin ISA SimpleInput;
    Graphic ISA Base::Layout WITH
      x_pos := 1;
      y_pos := 150;
    END;
  END;
END;
```

D.14 PlateHeatResistorFM

```
PlateHeatResistorFM ISA PlateHeatResistorIC WITH
%% A complex heat transfer model with
%%   separate submodels for convective
%%   and conductive heat resistance.
%%
%% Connect the steamside (primary) to Hin
%% and the waterside (secondary) to Hout
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09


functions:
  dTm ISA K2BasicLib::LogMean WITH
    x = HTran.Tin - Hout.R.Tout;
    y = Hin.R.Tout - Hout.R.Tin;
  END;

parameters:
  lambda, alpha, Wtemp, ToutC, H, rs, DT,
    alphaSteam, dH, rhoK, lambdaK, etaK,
    TwaterC, cpW, lambdaW, etaW, PrW, ReW, NuW TYPE Real;
  k ISA Variable WITH initial := 33.0;END;

  delta ISA Parameter WITH default := 0.4e-3; END;
  NbrOfWaterChan ISA Parameter WITH default := 11;END;
  ChanInletArea ISA Parameter WITH default := 2e-4; END;
  CRe ISA Parameter WITH default := 0.079; END;
  nRe ISA Parameter WITH default := 0.75; END;
  nPr ISA Parameter WITH default := 0.4; END;

equations:
  ToutC := Hin.R.Tout - 273.15;

% Calculation of alphaSteam
  lambdaK := 0.57 + 0.00177*ToutC - 0.00000657*ToutC^2;
  etaK := 0.02666e-3*exp(537.62/(128.01 + ToutC));
  rs := 350.4e+3*(374.35-(HTran.Tin-273.15))^0.3317;
  DT := HTran.Tin - Wtemp;
  alphaSteam := 1.06*(lambdaK^3*rhoK^2*K2BasicLib::g*rs/
              (etaK*H*DT))^0.25*(H/dH)^0.134;

% Calculation of alpha Water
  TwaterC := (Hout.R.Tin + Hout.R.Tout)/2 - 273.15;
  cpW := (4.22-0.00171*TwaterC+0.0000160*TwaterC^2)*1e3;
  lambdaW := 0.57+0.00177*TwaterC-0.00000657*TwaterC^2;
  etaW := 0.02666e-3*exp(537.62/(128.01+TwaterC));
  PrW := cpW*etaW/lambdaW;
  ReW := Hout.M.w*dH/(NbrOfWaterChan*etaW*ChanInletArea);
  NuW := CRe*ReW^nRe*PrW^nPr;
  alpha := NuW*lambdaW/dH;
```

```
  Wtemp = ((HTran.Tin+Hin.R.Tout)/2 + (Hout.R.Tin+Hout.R.Tout)/2)/2;
  lambda = 13.03 + 0.012*(Wtemp-K2BasicLib::T0);  % AISI 316 SS
  k := (1/(delta/lambda + 1/alpha + 1/alphaSteam));
  Hin.R.q := 0.85*dTm*HTran.Area*k;
  Hout.R.q := Hin.R.q;
END;
```

D.15 SteamValveFM

```
SteamValveFM ISA K2FlowLib::SteamFlowResistorFM WITH
  %% A valve model for Steam
  %%
  %% Logarithmic valve. Rangeability 50:1
  %%
  %% Assumptions:  No valve loss
  %%
  %% Medium:       steam (almost superheated)
  %% Model type:   full.
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA super::Graphic WITH
    bitmap TYPE String := "WaterValve";
  END;

variables:
  kvo, kvr TYPE Real;
  Area ISA Variable;

parameters:
  y ISA Parameter WITH default := 0.7; END;
    %Valve opening, 0..1
  kvs ISA Parameter WITH default := 10.0; END;
  R ISA Parameter WITH default := 50.0; END;

equations:
 kvr = kvs / R;
 kvo := kvr;
 Area := kvo * exp(ln(kvs/kvo) * y )/3.6e4;
END;
```

D.16 SteamValveCFM

```
SteamValveCFM ISA SteamValveFM WITH
  %% Controlable SteamValve
  %%
  %% The valve position can be given 100 diskrete setpoints.
  %% The setpoints are feeded to a function with a time constant.
  %% This simulates a controlsignal with 100 diskrete levels.
  %%
  %% Ola Löfgren & Patrik Svensson 1997-09

terminals:
  yc ISA SimpleInput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 200.0;
      y_pos := 250.0;
    END;
  END;

parameters:
  Tvalve ISA Parameter WITH default := 1.0; END;

variables:
  y ISA Variable WITH initial := 0.0; END;
  yround ISA Variable WITH initial := 0.0; END;

equations:
  yround := ROUND(100*yc)/100;
  Tvalve * y' + y = yround;
END;
```

D.17 KE73kvs100FM

```
KE73kvs100FM ISA SteamValveCFM WITH
  %% A valve model for Steam with kvs = 63
  %% Spirax Sarco KE73.
  %%
  %% Assumptions:  No valve loss
  %%
  %% Medium:       steam (almost superheated)
  %% Model type:   full.
  %%
  %% Ola Löfgren & Patrik Svensson 1997-09


icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "KE73kvs100";
  END;
```

```
terminals:
  Fin ISA super::Fin WITH
    Graphic ISA Base::Layout WITH
      x_pos := 0;
      y_pos := 200;
    END;
  END;
  Fout ISA super::Fout WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400;
      y_pos := 200;
    END;
  END;

parameters:
  kvs ISA Parameter WITH default := 100.0; END;

END;
```

D.18 KE31kvs25FM

```
KE31kvs25FM ISA SteamValveCFM WITH
  %% A valve model for Steam with kvs = 25
  %% Spirax Sarco KE31.
  %%
  %% Assumptions:  No valve loss
  %%
  %% Medium:       steam (almost superheated)
  %% Model type:   full.
  %%
  %% Ola Löfgren & Patrik Svensson 1997-09


icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "KE31kvs25";
  END;

terminals:
  Fin ISA super::Fin WITH
    Graphic ISA Base::Layout WITH
      x_pos := 0;
      y_pos := 200;
    END;
  END;
  Fout ISA super::Fout WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400;
      y_pos := 200;
    END;
```

```
  END;

parameters:
  kvs ISA Parameter WITH default := 25.0; END;

END;
```

D.19 KE73kvs63FM

```
KE73kvs63FM ISA SteamValveCFM WITH
  %% A valve model for Steam with kvs = 63
  %% Spirax Sarco KE73.
  %%
  %% Assumptions:  No valve loss
  %%
  %% Medium:       steam (almost superheated)
  %% Model type:   full.
  %%
  %% Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "KE73kvs63";
  END;

terminals:
  Fin ISA super::Fin WITH
    Graphic ISA Base::Layout WITH
      x_pos := 0;
      y_pos := 200;
    END;
  END;
  Fout ISA super::Fout WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400;
      y_pos := 200;
    END;
  END;

parameters:
  kvs ISA Parameter WITH default := 63.0; END;

END;
```

D.20 PlateSteamFlowResistorFM

```
PlateSteamFlowResistorFM ISA K2FlowLib::FlowResistorFM WITH
%% A resistor model of a liquid based on
%%   static energy and mass balances.
%%
%% Assumptions: constant enthalpy
%%              no heat interaction,
%%              no work interaction.
%%              constant valve loss
%% Medium:      steam (superheated).
%% Model type:  full.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09

icon:
  Graphic ISA super::Graphic WITH
    bitmap TYPE String := "S2FFlowResistor";
  END;

submodels:
  Fin ISA super::Fin WITH
    M ISA K2TerminalLib::SteamMediumTC;
  END;
  Fout ISA super::Fout WITH
    M ISA K2TerminalLib::SteamMediumTC;
  END;

temperature_propagation:
  Fout.M.T = Fin.M.T;    % Propagates the temperature unchanged

loss_factors:
  zv ISA K2FlowLib::ValveLossFactorFunction;

equations:
  DeltaZ = 0;            % neglect piezometric pressure

medium:
  M ISA K2MediumLib::SteamFlowMM;
END;
```

D.21 PlateFlowResistorFM

```
PlateFlowResistorFM ISA K2FlowLib::FlowResistorIC WITH
%% A resistor model of a liquid based on
%%   static energy and mass balances.
%% Model referes to a PlateHeatExchanger.
%%
%% Assumptions: constant enthalpy
%%              no heat interaction,
%%              no work interaction.
%% Medium:      undefined (default water)
%% Model type:  full - height specifications.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09

variables:
  w ISA K2BasicLib::FlowVC;
  rho, my ISA Variable;
  NbrOfWaterChan ISA Parameter;
  DeltaP ISA K2BasicLib::PressureVC;
     % piezometric pressure drop

equations:
% ------------------------------- mass balance
  Fin.w = w;
  Fout.w = w;
% ------------------------------- enthalpy balance
  Fout.h = Fin.h;
% ------------------------------- mechanical energy balance
  DeltaP = 506.765e+3*(w/NbrOfWaterChan)^1.850;
% ------------------------------- auxiliary variables
  DeltaP = Fin.p-Fout.p;

medium:
  M ISA K2MediumLib::WaterFlowMM;

medium_connections:
  M.min.p := Fin.p;
  M.min.h := Fin.h;
  rho := M.Mout.rho;
  my := M.Mout.my;
END;
```

D.22 PlateCondFlowResistorFM

```
PlateCondFlowResistorFM ISA K2FlowLib::FlowResistorIC WITH
%% A resistor model of a liquid based on
%%   static energy and mass balances.
%% Cleaned up and added piezometric pressure drop, 9605, Jonas.
%%
%% Assumptions: constant enthalpy
%%              no heat interaction,
%%              no work interaction.
%% Medium:      undefined (default water)
%% Model type:  full - height specifications.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09
%% No backflow allowed.

parameters:
  length, diameter ISA Parameter;
  Area ISA Parameter WITH
    value := sqr(diameter)*K2BasicLib::pi/4;
  END;

variables:
  w ISA K2BasicLib::FlowVC;
  rho, my, DeltaZ ISA Variable;
  DP, DeltaP ISA K2BasicLib::PressureVC;
     % piezometric pressure drop

equations:
% ------------------------------- mass balance
  Fin.w = w;
  Fout.w = w;
% ------------------------------- enthalpy balance
  Fout.h = Fin.h;
% ------------------------------- mechanical energy balance
  w = Area*sqrt(2*rho*DeltaP/zv);
% ------------------------------- auxiliary variables
  DP = Fin.p-Fout.p + rho*K2BasicLib::g*DeltaZ;
    % The pressure balance (negative values possible
  DeltaP = if DP <= 0 THEN 0 ELSE DP;
    % Only positive DeltaP allowed <=> no backflow

loss_factors:
  zv ISA K2FlowLib::TubeLossFactor WITH
    Fi ISA K2FlowLib::TurbulentFrictionFactor;
    diameter := outer::diameter;
    length := outer::length;
  END;

medium:
  M ISA K2MediumLib::WaterFlowMM;
```

```
medium_connections:
  M.min.p := Fin.p;
  M.min.h := Fin.h;
  rho := M.Mout.rho;
  my := M.Mout.my;
END;
```

D.23 PlateFlashFlowResistorFM

```
PlateFlashFlowResistorFM ISA PlateCondFlowResistorFM WITH
%% A resistor model of a liquid based on
%%   static energy and mass balances.
%%
%% Assumptions: constant enthalpy
%%              no heat interaction,
%%              no work interaction.
%% Medium:      saturated water and steam mixture
%% Model type:  full.
%%
%% Modified by Ola Löfgren & Patrik Svensson 1997-09

equations:
  DeltaZ = Fin.M.z-Fout.M.z;
     % piezometric pressure

medium:
  M ISA K2MediumLib::FlashFlowMM;
END;
```

D.24 WaterValveCFM

```
WaterValveCFM ISA vvx::PlateFlashFlowResistorFM WITH
  %% Controlable WaterValve
  %%
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09

icons:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "WaterValve";
  END;

terminals:
  y ISA SimpleInput WITH        % Valve opening, 0..1
    Graphic ISA Base::Layout WITH
      x_pos := 200.0;
      y_pos := 50.0;
    END;
  END;
```

```
variables:
  Area ISA Variable;
  Area := A0*y;

parameters:
  A0 ISA Parameter WITH
    value := sqr(diameter)*K2BasicLib::pi/4;
  END;

submodels:
  zv ISA K2FlowLib::ValveLossFactorFunction;

medium:
  M ISA K2MediumLib::WaterFlowMM;
END;
```

D.25 PlateHeatTransferInTC

```
PlateHeatTransferInTC ISA Base::RecordTerminal WITH
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09
  R ISA PlateHeatRateInTC;
END;
```

D.26 PlateHeatRateInTC

```
PlateHeatRateInTC ISA Base::RecordTerminal WITH
  %% Modified by Ola Löfgren & Patrik Svensson 1997-09
  q ISA K2TerminalLib::HeatRateTC;
  Tout ISA K2TerminalLib::TemperatureTC;
END;
```