

ISSN 0280-5316
ISRN LUTFD2/TFRT--5813--SE

Modeling and Control of Reaction Calorimeter System

David Lam
Linda Törner

Department of Automatic Control
Lund University
May 2008

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> May 2008	
		<i>Document Number</i> ISRN LUTFD2/TFRT--5813--SE	
<i>Author(s)</i> David Lam and Linda Törner		<i>Supervisor</i> Hubertus Tummescheit, Modelon AB, Lund Holger Nilsson and Jan-Bertil Jeppsson, Chemisens AB Lund. Per Hagander, Automatic Control Lund (Examiner)	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Modeling and Control of Reaction Calorimeter System (Modellering och reglering av kalorimetri process)			
<i>Abstract</i> In dealing with calorimetry, safety aspects are very important. It is therefore very useful to be able to study a process before carrying out experiments on it. A computer model can then be very useful to save time and money. The purpose of this thesis was to construct a computer model of the Chemisens reaction calorimeter, in the modeling software Dymola. For the purpose of the modeling, a new Dymola library was developed, specific for the calorimeter. The task was also to investigate the control of the calorimeter. The existing cascade control was implemented in the computer model as well as PID and LQG control. The results from the modeling were surprisingly good and the controllers also worked well			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 76	<i>Recipient's notes</i>	
<i>Security classification</i>			

Preface

We would like to thank our supervisors Johan Åkesson, Department of automatic control, LTH, Hubertus Tummescheit, Modelon AB and Holger Nilsson and Jan-Bertil Jeppsson, Chemisens AB.

In this thesis we have had use for many things that we have learned at the Engineering Physics program.

David Lam & Linda Törner, Lund March 2008.

Contents

1	Introduction	1
1.1	Background	1
1.2	Project task	1
2	Companies involved	3
2.1	Chemisens AB	3
2.2	Modelon AB	3
3	Software tools	5
3.1	Matlab & Simulink	5
3.2	Dymola	5
3.2.1	Modelica	6
3.2.2	Linearize	6
4	Modeling the calorimeter	7
4.1	Calorimetry	7
4.2	The calorimeter	7
4.3	Theoretical background	8
4.3.1	The Energy balance	9
4.3.2	The heat equation	10
4.3.3	Discretization of the 1D heat equation	10
4.3.4	Heat flow and heat flux rate	12
4.3.5	The peltier effect	13
4.4	The simple model	14
4.5	The advanced model	14
4.6	Nodes	15
4.7	Modeling the calorimeter in Dymola	15
4.7.1	The CalorimeterSimulation library	16
4.8	The complete calorimeter model	18
4.9	Verification of model library	20
4.9.1	ReactorBottom	20
4.9.2	ReactionMass	24
4.9.3	Peltier	25

4.9.4	HeatFlowSensor	25
4.9.5	ThermalConductor	28
4.9.6	ReactorWall	28
4.9.7	Conclusions	31
4.10	Calibration of the model	33
4.10.1	Experimental data	33
4.10.2	The calibration setup	35
4.10.3	Calibration results	36
5	Control of the calorimeter	39
5.1	Cascade control	39
5.1.1	The Chemisens cascade controller	40
5.2	PID-control	40
5.2.1	Ziegler Nichols step response method	41
5.2.2	Approximate MIGO design	42
5.2.3	Comparison of Ziegler Nichols and AMIGO	42
5.3	LQG	43
5.4	Comparison of the control methods	45
5.5	Results and discussion of control methods	45
5.5.1	Cascade control	45
5.5.2	PID control	46
5.5.3	LQG	47
5.5.4	Comparison	50
5.6	Conclusions	55
6	Conclusions and discussion	61
7	Future work	63
8	Experiences	65
	Bibliography	67
A	Model parameters	69
B	Peltier element performance	71

Chapter 1

Introduction

1.1 Background

For safety reasons many chemical producing companies depend on a detailed knowledge of their production processes. A chemical reaction absorbs or releases energy and an uncontrolled energy release can be very dangerous. There may be a sudden increase in temperature or pressure resulting in fire or even an explosion.

A tool for measuring the energy exchange in a reaction can therefore be very useful in gaining understanding of a certain chemical process. ChemiSens (see Section 2.1) has developed such a tool, a reaction calorimeter. The ChemiSens Reaction Calorimeter System can measure the energy exchange in a process, as well as the temperature and pressure.

However, to gain full understanding of an actual chemical production process more information is needed. Questions like "Can we transfer the information from the calorimeter to a large scale process?" or questions on security aspects of the production need to be answered. One way to answer them would be to simulate the calorimeter and the chemical in a computer before using the real calorimeter. This is an inexpensive and straight forward solution. One can easily vary the conditions of the process, like the amount of reactants and the energy supply, without wasting time and expensive chemicals. In our thesis we have constructed such a computer model.

Specifically Chemisens wants to use the computer model to simulate the experiments before performing them and also as a tool for improving the calorimeter. In the future Chemisens vision is to be able to offer customers a simulation software developed by Chemisens.

1.2 Project task

The main task of this master thesis was to develop a computer model of the ChemiSens reaction calorimeter system see Figure 1.1. An important

aspect of the modeling was to get familiarized with the calorimeter and the enclosed software. We were also to investigate the possibilities of improving the current cascade control algorithm in the calorimeter as well as testing other control methods, such as LQG and PID. The main part of the modeling was done in the object-oriented modeling software Dymola that uses the programming language Modelica¹. Matlab was also used for evaluating the control aspects of the model and for plotting.

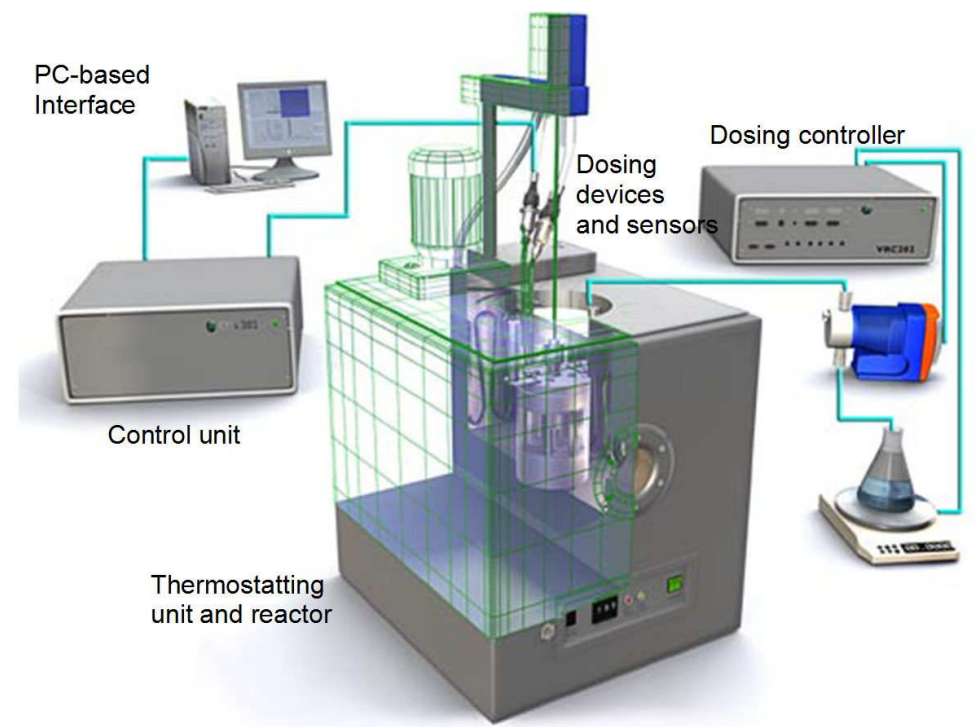


Figure 1.1: This is a picture of the real calorimeter system.

¹Modelica is an object-oriented modeling language designed to modeling of complex systems [2].

Chapter 2

Companies involved

This section contains a brief presentation of the companies involved in this master thesis and how they have contributed to this thesis.

2.1 Chemisens AB

This thesis was done on an assignment from Chemisens. Chemisens has developed and sells a reaction calorimeter that very accurately measures the heat exchange in a chemical reaction. Today there is not any computer simulation possibility in the enclosed software, for handling the equipment, that Chemisens sells. In the future Chemisens aims to develop their own software. A feature in this program could be the possibility to simulate the experiment before actually carrying it out. This is where our computer model could be used. Another usage will most likely be development and improvement of the calorimeter system.

Chemisens was founded by Holger Nilsson and Christer Silvegren in the seventies and the calorimeter is based on the results from their Ph.D thesis. The first reaction calorimeter was built in 1983. Chemisens vision is to offer reaction calorimeters based on the best possible measuring principles.

During our work with this thesis Chemisens has contributed with their knowledge on calorimetry as well as the opportunity to verify the computer model on the real reaction calorimeter.

2.2 Modelon AB

As already mentioned, the main part of the modeling was carried out in Dymola. Modelon is a company based in Lund that tailors Dymola libraries for their customers. Modelon provides physical modeling, simulation of dynamical systems and model based control design, of advanced complex technical systems. Modelon offers good knowledge in these three subject areas. The

modeling is based on the modeling and simulation tool Dymola¹ which in turn is based on the Modelica language.

In this thesis Modelon contributed with their knowledge in advanced modeling in Dymola.

¹Dymola modeling and simulation tool, a product of Dynasim AB

Chapter 3

Software tools

This chapter contains a brief description of the software used to create the models. In this project a number of different simulation tools have been used. Perhaps the most widely known is Matlab. Another simulation tool that was used is Dymola.

3.1 Matlab & Simulink

Matlab is an advanced numerical computing program. It can be used for heavy mathematical computations as well as for analysis and simulation of dynamic systems. This is especially useful in control. There is even a special control toolbox in Matlab called Simulink. Using Simulink, one can build a model of a particular system using drag and drop blocks. The system can then be analysed from a control point of view using the control commands in Matlab. This was very useful in our thesis.

3.2 Dymola

This program is designed for modeling and simulation of dynamic systems. One of the advantages with Dymola compared to Matlab is that one does not need to write the state space equations for a system built in Dymola. Parts describing the system are simply connected to make up a model and you can then run your simulation. Dymola will then by itself calculate the state space model. Interaction between Matlab and Dymola is easy and very useful. A linearized Dymola model can be imported to Matlab and analyzed with the control commands of Matlab. From a control point of view this gives a better understanding of the limitations and of structure of the model. Simulation data can also be exported from Dymola and then imported in Matlab as the plot function in Matlab is more manageable.

The models in Dymola are built by assembling defined objects from the Dymola libraries. This is done in a graphical interface by simple drag and

drop. The user can also create new objects and libraries and existing objects can also be altered. The objects are then connected by connectors. A connector in Dymola can have different types, for example Heat flow, current, temperature. Only objects with the same connectors can be put together.

3.2.1 Modelica

Dymola is programmed in a language called Modelica [2]. The main difference between Modelica and other programming languages is that Modelica is specifically designed for modeling, rather than programming in general. Like Java, Modelica is an object oriented programming language. For those who is not familiar with this here is a short explanation.

In Object Oriented Programming (OOP) the program is divided into different classes. The programmer decides how the objects are allowed to interact with each other. A class can inherit from another class and get the characteristics of that class. That way, a class can be used in many instances of a program. A class can also use an interface. An interface contains certain preset properties. The standard Dymola interfaces used in this thesis are HeatPort, RealInput and RealOutput. The HeatPort interface has two variables temperature [K] and heat flow rate [W]. Heat flow rate is the amount of energy passing through the heat port per unit time. The RealInput/Output has one real input/output signal

Modelica is built on component-based noncausal modeling. In simple terms this means that one can implement equations in the model without caring about the inputs and outputs.

3.2.2 Linearize

The *Linearize* command in Dymola creates a linearized version of the current model. This data is stored in a .mat-file that can be imported into Matlab. The file, apart from the A , B , C and D matrices, also contains information about what the states (x), the inputs (u) and the outputs (y) are in the model. With the control commands of Matlab, like *bode*, *margin* and *pole*, the stability and general behaviour can be explored. For comparison, the linearized model can then be imported back into Dymola. With a special block, StateSpace, a Dymola simulation of the linearized system can be performed. These results can then be directly compared to the behaviour of the original system, giving a picture of the accuracy of the linearized model.

Chapter 4

Modeling the calorimeter

This chapter treats the modeling of the reaction calorimeter. In the two first sections the background of the calorimeter is explained. Then a simple model of the calorimeter is derived from the temperature balance equation and implemented in the Matlab Simulink environment. In the following section the model is made more complex and implemented in the software Dymola using the programming language Modelica [4].

4.1 Calorimetry

Calorimetry is a scientific method for measuring the heat or physical changes of a chemical reaction. These measurements are made in a calorimeter where properties such as temperature, energy changes and pressure can be measured.

Many companies use the calorimeter in safety studying aspects of their processes. What happens, for example, if the cooling system suddenly shut down? This can be tested in the controlled environment of a calorimeter. The knowledge gained from the calorimeter experiment can then be used to better understand a particular process and it can also be used to ensure safety margins of a process. Under what conditions might sudden increases in temperature or pressure occur? This can be evaluated using calorimetry.

4.2 The calorimeter

There are a number of different calorimeters specialised in different types of chemical measurements. There are for example constant pressure calorimeters, differential scanning calorimeters, high energy particle calorimeters and reaction calorimeters.

In a reaction calorimeter the heat exchange can be measured by integrating heat flow over time. The heat flow is measured in the heat flow sensor inside the reactor bottom. In the reactor, the temperature, heat flow and

pressure can be monitored. The speed of the reaction, and thus temperature and pressure, can also be controlled via an electrical heater at the bottom of the vessel and a Peltier element at the bottom. Through the, lid reactants can be added during experiments. A stirrer ensures that the reactants are well mixed at all times. The speed of the stirring can be adjusted.

As mentioned before, Chemisens has developed and fabricates a reaction calorimeter. A basic picture of the principle of a reaction calorimeter is shown in Figure 4.1.

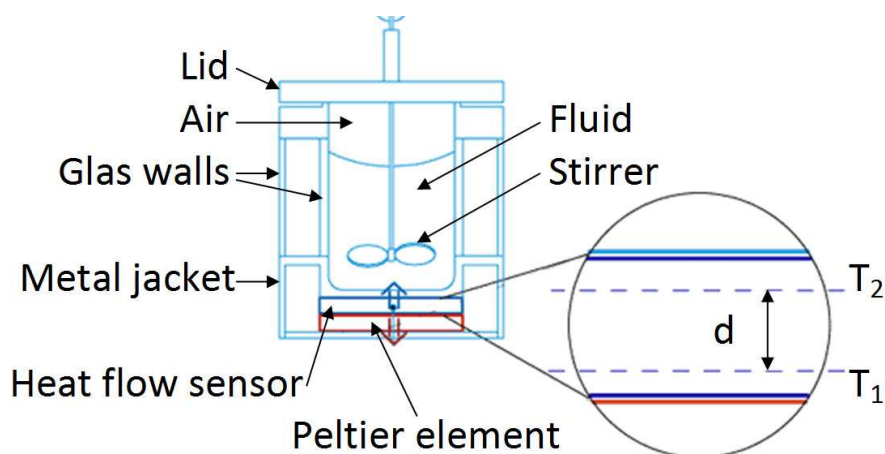


Figure 4.1: The principle of the reaction calorimeter that Chemisens has developed.

In the simple picture in Figure 4.1 the calorimeter consists of the following parts: an insulated lid, inner and outer glass walls, the bottom made of a metal jacket containing the heat flow sensor and the peltier element. The air between the lid and fluid and the glass walls insulates the calorimeter so that all heat exchange happens through the bottom. This insulation is of course not perfect so the reactor is put in a water container that is kept at a temperature 0.2°C above the set reference temperature. The water container also ensures a constant temperature environment.

The calorimeter is a slow system since the heat conductance in the reactor bottom is a rather slow process. This is due to the many layers of material between the peltier element and the fluid in the reactor. The time delay in the heat conductance is something that later needs to be taken into account in the control of the calorimeter.

4.3 Theoretical background

In this section the theoretical background is discussed and presented. The first part discusses a simple energy balance and assumptions that can be

made in the case of heating a fluid. The second and the third parts discuss the heat equation that describes the heat distribution in a material and how to implement this into the computer. The last part handle the notations used in heat calculations and about the peltier element.

4.3.1 The Energy balance

The simplest way to model the temperature change in a jar with a fluid, the reactor in the real case, is by an energy balance. The change in "process energy", i.e. a chemical reaction that produces heat, can be written by a simple energy balance like

$$accumulated = in + produced - out$$

where the accumulated energy are the sum of the uncontrolled produced energy and the added or subtracted energy that can be controlled. There are of course some assumptions that are to be made for the fluid. They are the following

1. Mass remains as fluid.
2. Ideal stirring.
3. Constant density.
4. Perfect insulated from the surroundings.
5. Constant heat capacity.

The first point implies to that the fluid not will vaporize or get stiff. The second point means that we got good stirring in the fluid and in that way has the same temperature in the whole fluid. The third point assumes that the fluid density will remain the same despite the temperature. The fourth point means that no uncontrolled energy is leaking to the outside of the jar. The last point follows from the first and third point and assumes that the fluid not will change the capacity for stored energy in the fluid. An equation for the temperature change can now be written based on these assumptions. The equation is given by

$$mc \frac{dT}{dt} = Q_{in} + Q_{prod} - Q_{out} \quad (4.1)$$

where m is the mass in kg for the fluid, c is the heat capacity in J/kg^oC for the fluid, Q_{in} is the supplied energy in J, Q_{prod} is the released energy in J (as in a thermal reaction) and Q_{out} the energy flow out in J that can be controlled. The energy balance is essential for understanding the modeling of the fluid in the reaction calorimeter. The description on how to model the fluid in the jar has now been made. In the next section the temperature distribution in the wall of the jar will be investigated.

4.3.2 The heat equation

In reality the whole jar is not insulated from the surroundings since one wants to be able to calculate the heat transported out from the jar. There will be a temperature gradient in the wall of the jar if there is a temperatures difference between the inside and outside of the jar. The temperature distribution in a material can be described by the heat equation. The heat equation [1] for a one dimensional heat flow is given by

$$\frac{\partial T(x, t)}{\partial t} - a \frac{\partial^2 T(x, t)}{\partial x^2} = 0 \quad (4.2)$$

where $a = \frac{\lambda}{\rho c}$ is the heat diffusivity a quantity depending on the material density , specific heat capacity and thermal conductivity.

$$\begin{aligned} \rho &= \text{density} && (kg/m^3) \\ c &= \text{specific heat capacity} && (J/kg \text{ } ^\circ C) \\ \lambda &= \text{thermal conductivity} && (J/m s \text{ } ^\circ C) \\ T &= \text{temperature at } x \text{ and time } t && (^\circ C) \end{aligned}$$

By Fourier's law the heat flow \mathbf{j} in a isotropic material becomes as

$$\mathbf{j} = -\lambda \frac{\partial T}{\partial x}$$

The heat flow through the boundary then yield as $\mathbf{j} \cdot \mathbf{n}$ (see Figure 4.2) and gives us the boundary condition at $x = 0$

$$-\lambda \frac{\partial T}{\partial \mathbf{n}} = \alpha_1 (T_1^y - T_1) = \mathbf{j} \cdot \mathbf{n} \quad (4.3)$$

where T_1^y is the temperature on the outer left side of the wall and

$$\alpha_i = \text{heat transfer coefficient} \quad (J/m^2 s)$$

at boundary i .

4.3.3 Discretization of the 1D heat equation

In order to be able to implement the heat equation in Dymola discretization of the heat equation is needed to be able to describe the temperature in the material. A common used method is called the finite difference method (FDM) [3] where an approximation for the partial space derivate is made. This is made with a second order difference approximation. The approximation for the room dependent part is given by

$$\frac{\partial^2 T_i}{\partial x^2} \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2}, 1 \leq i \leq N \quad (4.4)$$

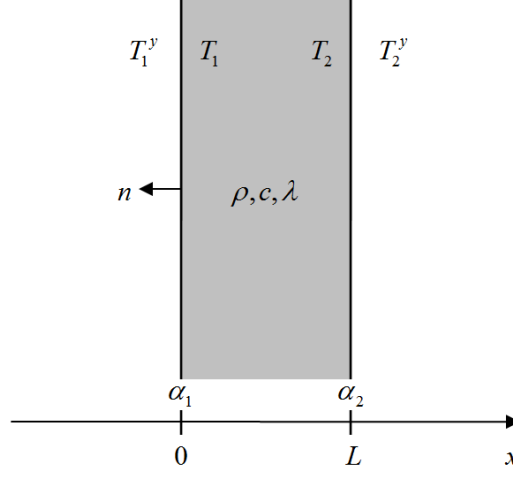


Figure 4.2: Wall with thickness L and heat transfer coefficients α_1 and α_2 at the boundaries.

where $T_1 \dots T_N$ are the temperatures at the nodes, if we choose to have N number of nodal points and

$$\Delta x = \frac{L}{N-1}$$

is the step length if the wall material thickness is L (see Figure 4.3).

The approximation for the Neumann boundary is given by

$$\frac{dT}{dx} \approx \frac{T_{i+1} - T_{i-1}}{2\Delta x} \quad (4.5)$$

for the node number $i = 1, N$. Using 4.4 and 4.5 in 4.2 and 4.3 gives us the approximation for the heat equation

$$\frac{dT_i}{dt} - a \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} = 0, \quad 2 \leq i \leq N-1 \quad (4.6)$$

and the boundary conditions as

$$\begin{aligned} -\lambda \frac{T_2 - T_1^y}{2\Delta x} &= \alpha_1 (T_1^y - T_1) = \frac{Q_1}{A} \\ -\lambda \frac{T_N^y - T_{N-1}}{2\Delta x} &= \alpha_2 (T_N - T_N^y) = \frac{Q_2}{A} \end{aligned} \quad (4.7)$$

where Q_1 and Q_2 are the heat flow through the wall at $x = 0$ and $x = L$ respectively. A is the cross sectional area of the wall. The temperatures T_1^y and

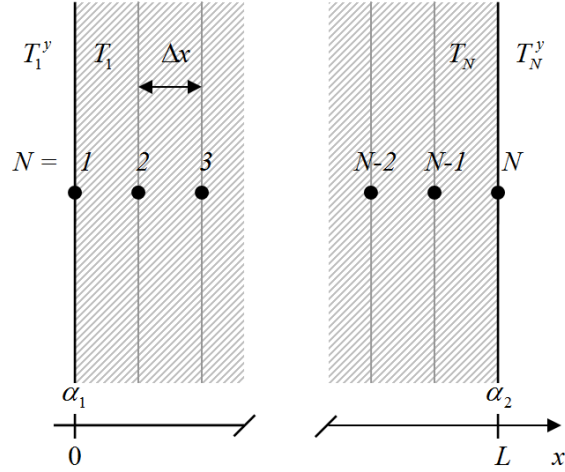


Figure 4.3: The node distribution in a wall with thickness L and N number of nodes. Where T_i^y corresponds to the temperature outside the wall for the node i and T_i the temperature at the inside for the same node.

T_N^y are assumed to be known, but it is unlikely that they are. To solve this a first order difference approximation for the outer boundary temperatures is used, according to

$$T_i = \frac{T_{i-1} + T_{i+1}}{2}$$

There by the temperatures T_1^y and T_N^y in the boundary equation 4.7 can be eliminated. The temperatures for the boundaries are then given by

$$\begin{aligned} T_1 &= T_2 + \frac{\Delta x}{A\lambda} Q_1 \\ T_N &= T_{N-1} + \frac{\Delta x}{A\lambda} Q_2 \end{aligned} \quad (4.8)$$

not containing the outer temperatures T_1^y and T_N^y .

4.3.4 Heat flow and heat flux rate

Heat flow and heat flux are often used when working with enthalpy changes. They are both describing an energy flow. But the difference between them is that heat flow has the unit $[W]$ while heat flux has the unit $[W/m^2]$. They are both used and it depends on what purpose one have. In the continuation of this report the heat flow rate are going to be used. The heat flow in a material is described by the heat conduction equation

$$Q = \frac{\lambda}{d} A \Delta T t \quad (4.9)$$

where λ is the thermal conductivity in $\text{J/ms}^\circ\text{C}$ for the material, d is the thickness in m, A the cross section in m^2 for the heat flow, ΔT the temperature variance in $^\circ\text{C}$ in the material and t the time in s.

Heat flow through a cross section with the heat transfer coefficient α is given by the equation

$$Q = \alpha A \Delta T t \quad (4.10)$$

This is used when there is some kind of resistance when heat is passed through two materials.

4.3.5 The peltier effect

The function of the Peltier element is built on the principles of the peltier effect which was discovered by Jean Peliter in 1834. The peltier effect is the phenomenon that one can create a heat difference from an electric voltage. It is typically made from two semi conductors of n- and p-type. By connecting the semi conductors and then applying a current, one can transfer heat from one junction to the other thus obtaining one cold and one warm side of the element. This is often used in thermoelectric cooling.

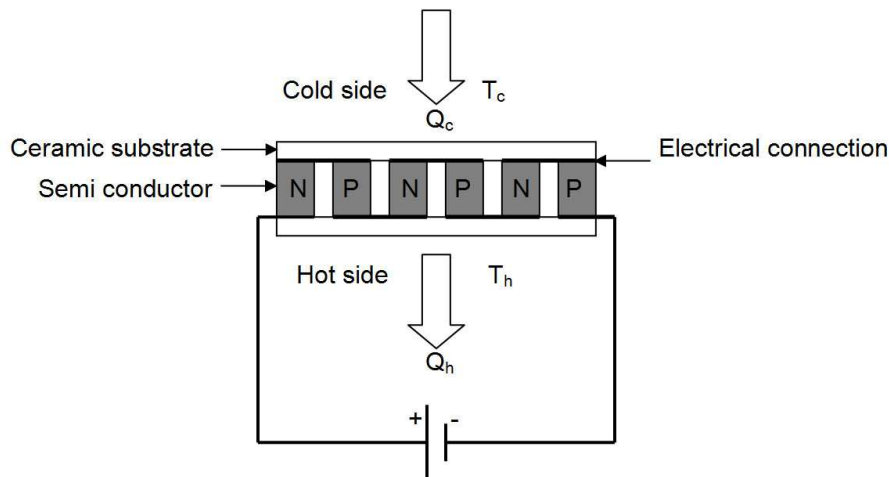


Figure 4.4: The principle of the peltier element.

The peltier element in the Chemisens calorimeter can be described by the following equations. The heat pumped on the cold side Q_c is given by the equation

$$Q_c = S_m T_c I - \frac{1}{2} I^2 R_m - K_m \Delta T \quad (4.11)$$

where S_m is the module Seebeck coefficient in V/K, T_c the temperature on the cold side in K, I the input current in A, R_m the module resistance in Ω , K_m the module thermal conductance in W/K and ΔT the temperature difference $T_h - T_c$ in K. The heat pumped to the hot side is equal to

$$Q_h = Q_c + P_{in} \quad (4.12)$$

Where an extra term P_{in} from the heat generated by the thermal resistivity is added. The equation for Q_c prescribes the heat flow in to the peltier element. Simulating an ideal peltier with excellent cooling on the hot side can be done by setting the coefficient $K_m = 0$.

4.4 The simple model

The first model was derived by paper and pen and was based on the a energy balance 4.1. The goal of this first model was to develop an understanding of the calorimeter and what sort of behaviour that could be expected. Before building the Matlab models the heat equations were set up by hand. Basically was this just a model for heating/cooling water in an insulated container. In this case there was no added energy Q_{in} , only produced Q_{prod} and subtracted Q_{out} energy that was controlled. The temperature change was then given by

$$\frac{dT}{dt} = kQ_{prod} - kQ_{out}$$

where $k = \frac{1}{\rho c V}$. This was implemented in Matlab Simulink and tested. The input signal for the produced energy was changed and the aim was to come back to the reference temperature as quickly as possible. This was done with a PI-controller. This simple model gave some basic understanding of the calorimeter. It was quickly discovered that Matlab was not the best modeling tool for this task. In Matlab all the states and signals have to be defined by the user, this would be very difficult in a large complex model with a large amount of states.

4.5 The advanced model

A new model was developed in Dymola by using the mathematical heat equation (4.2), since the first model based on an energy balance was too simple.

The reactor bottom was modeled as a cylindrical plate with heat conduction in the axial direction and discretization was also made in the same direction. The mathematical model is given by the equations 4.2 and 4.3 as

$$\begin{aligned}\frac{\partial T(x, t)}{\partial t} - a \frac{\partial^2 T(x, t)}{\partial x^2} &= 0 \\ -\lambda \frac{\partial T}{\partial x} &= Q_{prod}, \quad x = 0 \\ -\lambda \frac{\partial T}{\partial x} &= \alpha_1(T_{heatsink} - T_N), \quad x = L\end{aligned}$$

The plate was given a connector, one on each side. A heat flow rate Q_{prod} , assumed to be controlled by a known signal, on the upper side and a heat sink with constant temperature $T_{heatsink}$ to the other side was then attached. These were used when testing the model. The model was the first part in the *CalorimeterSimulation* library.

4.6 Nodes

The algorithms for calculating the temperature distribution throughout the different parts of the calorimeter is naturally discrete. The temperature is calculated at different points in, for example the reactor bottom. Such a point is called a node. Ideally, every element should have an infinite number of nodes. This of course is not possible in reality. An important part of the model building was therefore to evaluate how many nodes was needed to get an accurate but reasonably fast model. This issue is discussed in Section 4.9.

4.7 Modeling the calorimeter in Dymola

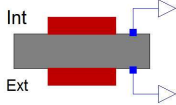
Modeling the reaction calorimeter using the Dymola program was done by dividing the calorimeter into suitable components. The following classes were defined:

- Reactor bottom
- Heat sink
- Reaction mass
- Peltier
- Heat flow sensor
- Thermal conductor
- Reactor wall

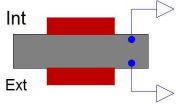
None of these elements existed in the standard Dymola library. However, a good starting point could be found in the free library ThermoPower [5] developed by Francesco Casella. These elements were not perfectly adapted to the task, the elements were for example not discretized and the output was heat flux not heat flow which would be more suitable for this task. Therefore a substantial part of the modeling was to write new elements. These parts later made up the model library.

4.7.1 The CalorimeterSimulation library

In this section the model classes will be presented and discussed.

 **HeatEqSolver** solves one-dimensional heat equation for a plate or a cylinder. The temperature distribution is described by the discretized Fourier's equation (4.6) and the boundary temperatures (4.8). By choosing the parameter *geomOpt* the user can decide to solve the equation in a plate or in a cylinder. For the cylinder case the parameter *h* is also used.

Parameters: *Nr* is the number of discretization nodes, *geomOpt* is the choice for heat flow geometry, *r* is the plate or inner cylinder radius, *t* is the bottom or wall thickness, *l* is the cylinder wall height used together with the *geomOpt* option, ρ is the material density, *c* is the material specific heat capacity, λ is the material thermal conductivity, $T_{startint}$ is the start temperature at the inner node, $T_{startext}$ is the start temperature at the last node and *initOpt* is the steady state initialization option.

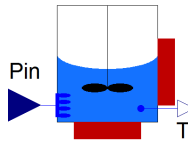
 **ReactorBottom** is a thermal model for a one dimensional heat transfer in a circular metal plate with a specified thickness *d*. The element contains the *heatEqSolver* object that solves the heat equation. The model is axial-symmetric and has one node in the radial direction and a specified number *N* of nodes in the axial-direction chosen by the user and uses two heat ports to simulate the connection of the surfaces to another material. Leaving one heat port unconnected means thermal insulated on that side. The model allows the user to set material parameters, dimensions, number of nodes and initial temperatures.

Parameters: *Nr* is the number of discretizing nodes, *r* is the bottom radius, *t* is the bottom thickness, ρ is the reactor material density, *c* is the reactor Material specific heat capacity, λ is the reactor material thermal conductivity, T_{init} temperature start value and *initOpt* is the steady state initialization option.



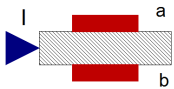
HeatSink is like an electrical ground. But in this case there is no current instead a heat flow. This means that the element can have infinite heat flow into it without changing its temperature, i.e. the temperature will remain the same no matter of what. The model is simply created by using a heat port with a prescribed temperature. The user can set this temperature parameter.

Parameters: T the heat sink temperature.



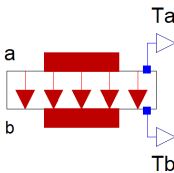
ReactionMass is describing a heat source in a fluid as a chemical reaction. This is done by modeling the temperature for a fluid in a cylindrical tank by the equation (4.1). The model has one realinput port to simulate the reaction power released in the fluid and two heat ports one for the cylinder bottom and one for the cylinder wall. The parameters for the model are fluid parameters, cylinder radius, initial temperature and thin-film resistance for the fluid.

Parameters: m is the chemical mass, c is the chemical material specific heatcapacity, h is the cylinder flange height, ρ is the chemical material density, α_{bottom} is the heat conductance coefficient for the bottom, α_{wall} is the heat conductance coefficient for the wall, r is the cylinder bottom radius, T_{init} is the initial temperature of the chemical and *steadyStateStart* is the steady state initialization option.



Peltier is describing the pumped heat and the temperature for a peltier element. This is done by the known equations for the heat flow on the cold and hot sides (4.11). The model has one realinput port for controlling the current for the element and one heat port on each side of the element. The parameters S_m , R_m and K_m can be changed. An additional option *currentCorrection* is available to adjust or invert the input current, that can be necessary.

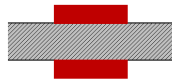
Parameters: S_m is the module Seebeck coefficient, R_m is the module resistance, K_m is the module thermal conductance and *currentCorrection* adjusts the input current.



HeatFlowSensor is a sensor that measures the heat flow out from the reactor base. The heat flow can be calculated from the arising temperature gradient in the sensor. The element consists of three HeatEqSolver elements giving the heat distribution in the element. The model has two heat ports and two realoutput ports delivering the temperatures on each side of the element. The number of nodes in the element and the initial temperature can be defined by the user.

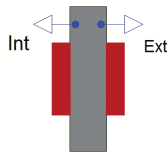
Parameters: Nr is the number of axial nodes in each layer. The total number of nodes will be $3*Nr-2$, r is the sensor cross-section radius, T_{init}

is the start temperature value in the sensor and *initOpt* is the steady state initialization option.



ThermalConductor is a model for transferring heat without storing it and uses the ThermalConductor element from the standard Dymola library. The element is used to model the fact that there is not perfect thermal conductivity between the surfaces when mounting them. The heat flow is given by the equation (4.9). The element has two heat ports.

Parameters: k is the thermal conductivity of the material, r is the cylinder radius, t is the thickness of the conductor.



ReactorWall is like the reactor bottom model except that this model describes the heat flow through a cylinder so the discretization is made in the radial direction. It contains a HeatEqSolver object.

Parameters: Nr is the number of nodes, r is the inner wall radius, t is the wall thickness, h is the cylinder wall height, ρ is the material density, c is the material specific heat capacity, λ is the material thermal conductivity, T_{init} is the start temperature at the inner wall, $T_{startext}$ is the start temperature t the last node and *initOpt* is the steady state initialization option.

4.8 The complete calorimeter model

The complete calorimeter model is shown in Figure 4.5 and can be compared to the calorimeter sketch in Figure 4.1. The mass of a fluid in the reactor is modeled with the ReactionMass element. The inner and outer reactor bottom and the reactor wall are modeled with the elements ReactorBottom and ReactorWall. The reference thermostat is modeled as a heat sink that can absorb all the produced heat and is therefore done by the element HeatSink. The space between the inner and outer bottom consists of a heat flow sensor and a peltier element. At last are there some thermal conductors added between some elements modeling heat paste used when assembling the reactor. The two thermal conductors between the heat sink and the wall and the outer reactor bottom are modeling the film resistance between the reactor surface and the fluid in the reference thermostat.

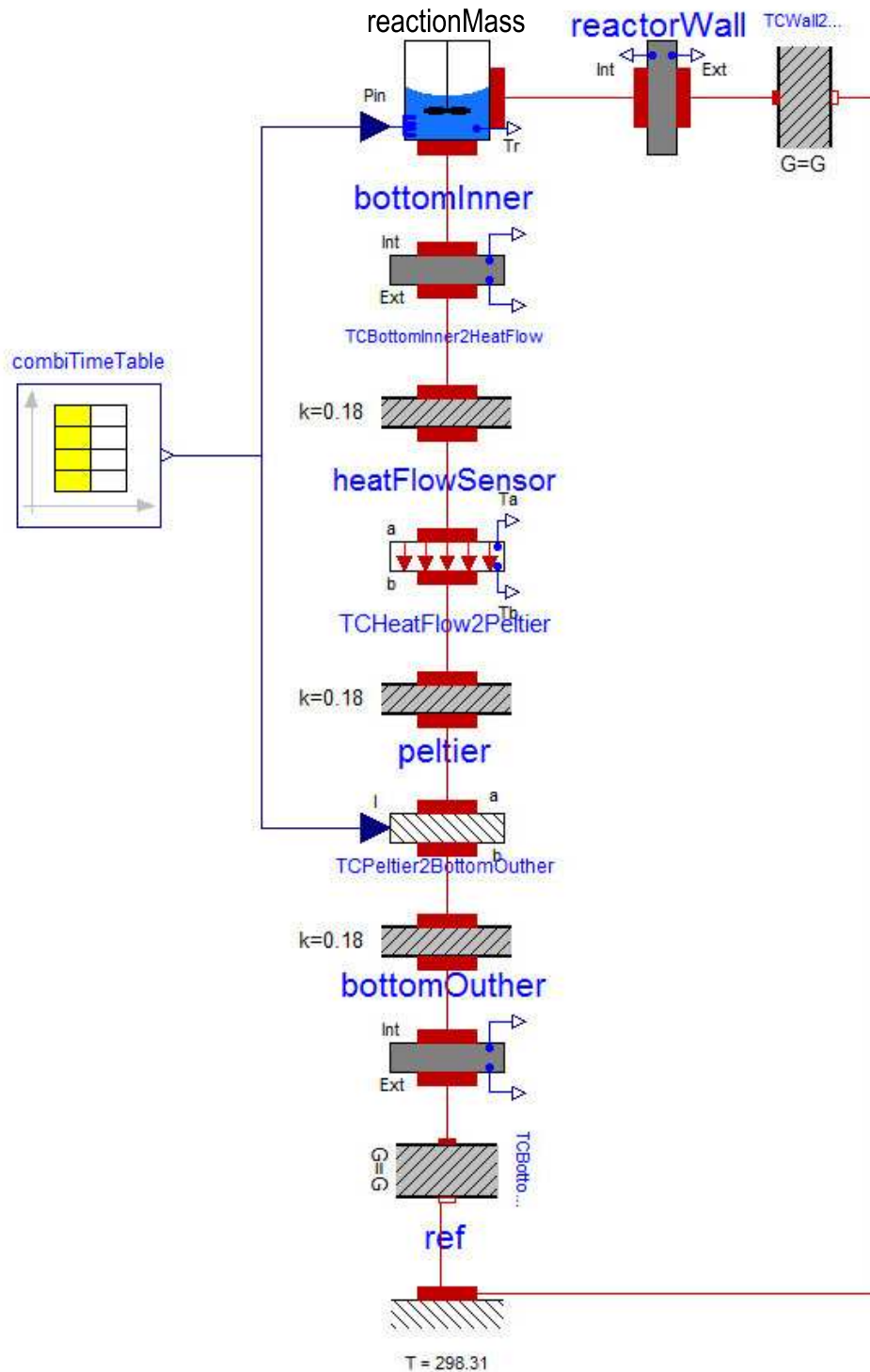


Figure 4.5: The calorimeter model created in Dymola using the CalorimeterSimulation library. A description is made in section 4.8.

4.9 Verification of model library

To verify the model library the elements were tested one by one on simple cases where the results can be calculated.

4.9.1 ReactorBottom

It is of course important that the heat equation is implemented correctly in the reactor bottom. To verify this one can add a known amount of energy to the reactor bottom. The resulting increase in temperature can then be calculated as

$$T_1 = \frac{Q}{\rho c V} + T_0 = 301.44K$$

where T_1 is the resulting temperature, T_0 the temperature from the beginning and Q is the total amount of energy added.

In Dymola this is done by connecting a prescribedHeatFlow element from the standard Dymola library to a ReactorBottom element. The setup is shown in the Figure 4.6. The prescribedHeatFlow element generates a heat flow from a given input signal. In this case two step signals were added to a pulse, generating a total power. The reactor bottom was given the dimensions, initial temperature and the material parameters for stainless steel according to Table 4.1. Then the steady-state temperature is calculated to 301.44 K. In the experiment the importance of the number of nodes in the element was also evaluated. Higher numbers of nodes give a more accurate solution. The test was performed with $N = 5, 20, 100, 200$ number of nodes. The result of the experiment is shown in Figure 4.7.

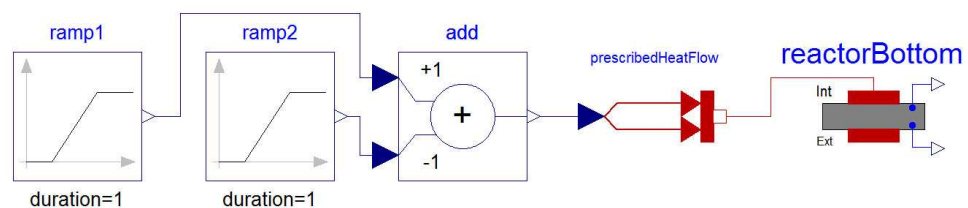


Figure 4.6: The Dymola setup for the test of adding a known amount of energy.

In steady state any number of nodes should give the same result. This was tested by doing a step response on a reactor bottom connected to a heat sink, see Figure 4.8. The result shown in Figure 4.9 shows the mean temperature in the reactor bottom element, as expected all the curves tend towards the same steady state value.

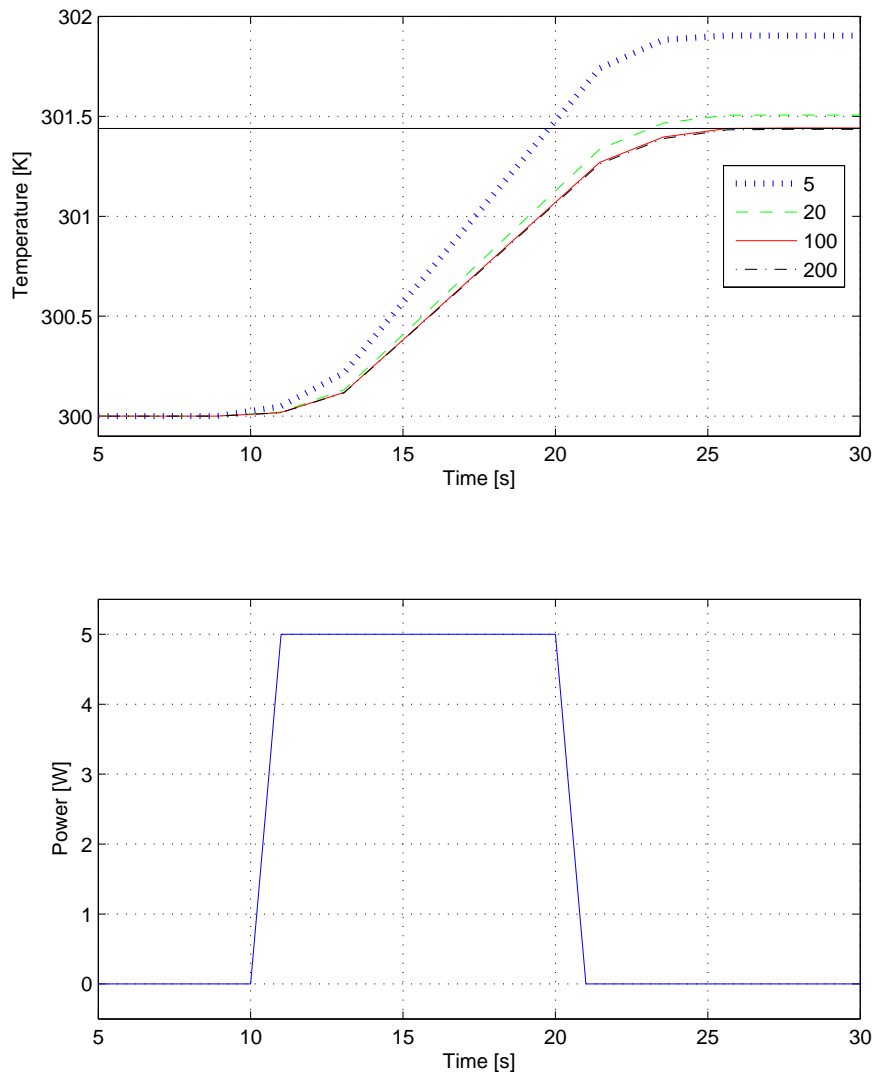


Figure 4.7: Simulation of different number of nodes in the reactor bottom. Higher number of nodes gives a more accurate solution. The steady-state temperature is 301.44 K which is described best by 100 nodes or more.

Parameter	Value	Unit
Q	50	W
r	275	mm
t	4	mm
T_0	300	K
ρ	8010	(kg/m^3)
c	460	$(J/kg \cdot K)$
λ	13.8	$(W/m \cdot K)$
$T_{heatsink}$	300	K
steadyStateStart	false	

Table 4.1: Simulation parameters for the reactor bottom test.

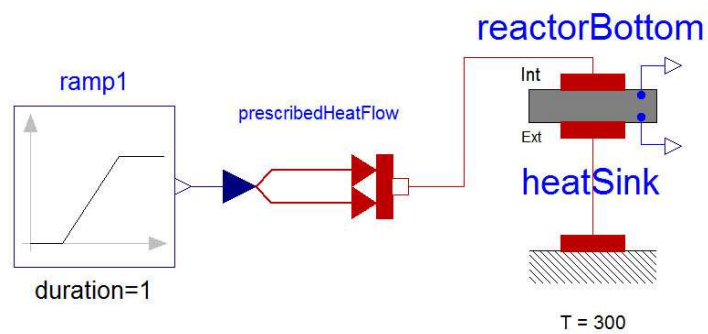


Figure 4.8: The Dymola setup of the steady state step response.

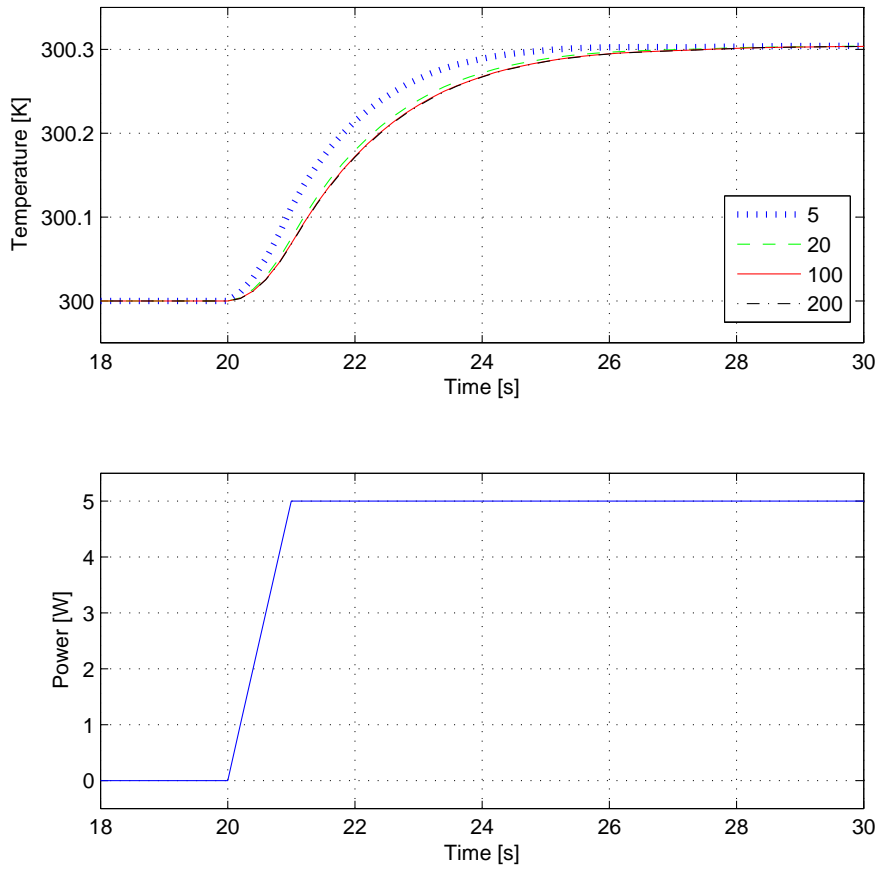


Figure 4.9: Simulation of a step response for different number of nodes in the reactor bottom. The upper curve shows the mean temperature in the object.

4.9.2 ReactionMass

In the reaction mass element, it is important that the heat balance equation is correctly implemented. This was done by setting the input signal to a constant for a certain time thus giving an certain amount of power, see Figure 4.10. The temperature change can be calculated, using the parameters in Table 4.2 in the equation

$$T_1 = \frac{Q}{mc} + T_{init} = 300.48K$$

. If the heat flow out is zero the result should be a constant temperature offset when equilibrium has been reached. The simulation parameters are shown in Table 4.2. The result is shown in Figure 4.12 and as expected it can be seen that there is a temperature offset.

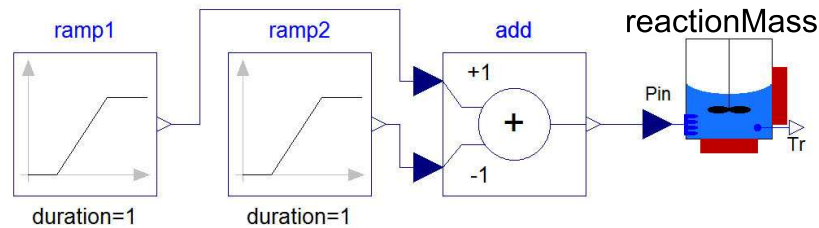


Figure 4.10: The Dymola setup for the reaction mass test of adding a known amount of energy.

Parameter	Value	Unit
m	0.1	Kg
c	4190	J/KgK
h	0.01	m
r	0.0275	m
α_{bottom}	500	W/m ² K
α_{wall}	500	W/m ² K
T_{init}	300	K
steadyStateStart	false	
$T_{heatsink}$	300	K

Table 4.2: Simulation parameters for the reaction mass test.

In the next case the model was tested with heat flow different from zero. This was done by connecting the model to a heat sink with constant temperature. The setup is shown in Figure 4.11. First a temperature change occurs, then the same temperature as the heat sink is reached, see Figure 4.12.

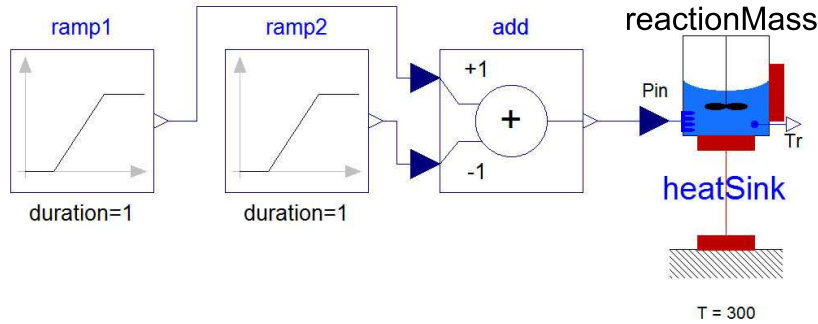


Figure 4.11: The Dymola setup for the reaction mass test of adding a known amount of energy with a heat sink attached.

4.9.3 Peltier

Testing the peltier element model was done by using constant current I to the model and connecting two heat sinks to each side of the element creating a temperature difference ΔT , the setup is shown in Figure 4.13. The setup parameters are shown in Table 4.3. The results are shown in Table 4.4 are compared to plots from the manufacturer [10] see Appendix B.1. Simulation was done at the temperature T_h 50°C on the hot side, the cold side temperature T_c was then varied. The current I was set to 6 A. The simulated values are close to the manufacturer values except for $\Delta T = 70$. It is unclear why.

Parameter	Value	Unit
T_h	323.15	K
I	6	A
S_m	0.05343	V/K
R_m	2.4796	Ω
K_m	0.5204	W/K
currentCorrection	1	

Table 4.3: Simulation parameters for the peltier element test. S_m is the module Seebeck coefficient, R_m is the module resistance, K_m is the module thermal conductance and currentCorrection adjusts the current signal.

4.9.4 HeatFlowSensor

The sensor gives the heat flow rate through the material. To test if the element behaves as a sensor, a test was setup in Dymola according to Figure 4.14. This was done by connecting two heat sinks at different temperatures

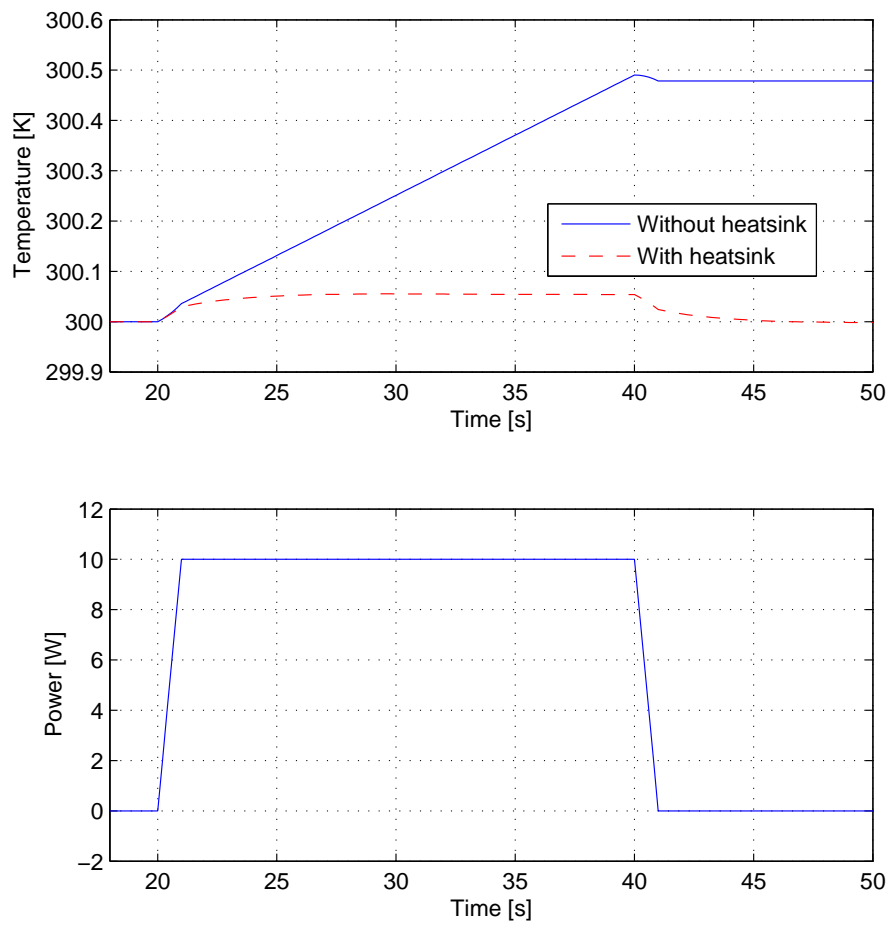


Figure 4.12: Simulation of the fluid temperature in the reaction mass for the cases without and with a heat sink.

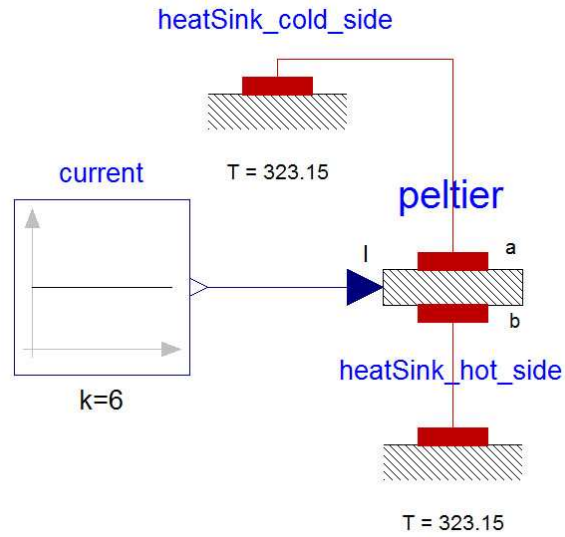


Figure 4.13: The Dymola setup of the peltier test.

ΔT	$Q_{c_{ref}}$ [W]	$Q_{c_{sim}}$ [W]
0	57	59
10	49	50.6
20	41	42.1
30	33	33.5
40	25	25.3
50	17	16.9
60	9	8.5
70	1	0.1

Table 4.4: Simulation of the pumped heat $Q_{c_{sim}}$ compared with the $Q_{c_{ref}}$ from the manufacturer.

to each side of the element. Thus creating a temperature gradient in the sensor element. This was simulated with values shown in Table 4.5 and the result can be seen Figure 4.15 as expected there is a temperature gradient over the element. The element can therefore be used to measure the heat flow.

Parameter	Value	Unit
N_r	100	
r	0.0275	m
T_{init}	300	K
initOpt	false	
T_{cold}	310	K
T_{hot}	300	K

Table 4.5: Simulation parameters for the heat flow sensor test.

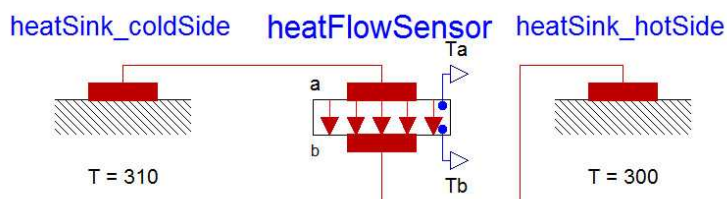


Figure 4.14: The Dymola setup for test of heat flow sensor.

4.9.5 ThermalConductor

The thermal conductor reduces the heat flow rate through two surfaces placed together. This was tested by placing a reaction mass element and a heat sink element on either side of the thermal conductor, see Figure 4.16. The result is shown in the Figure 4.17. Large k is good thermal conductance. The overshoot for the curve with $k=1000$ probably depends on errors in the numerical integration. The simulation parameters are the same as in Table 4.2, except for k , r and d which can be seen in Table 4.6.

4.9.6 ReactorWall

This test is made the same way as the first test for the reactor bottom. The importance of the number of nodes in the element is evaluated. The setup and the result are shown in Figures 4.18 and 4.19. The simulation parameters are shown in Table 4.7. The accuracy for 20 nodes or more is

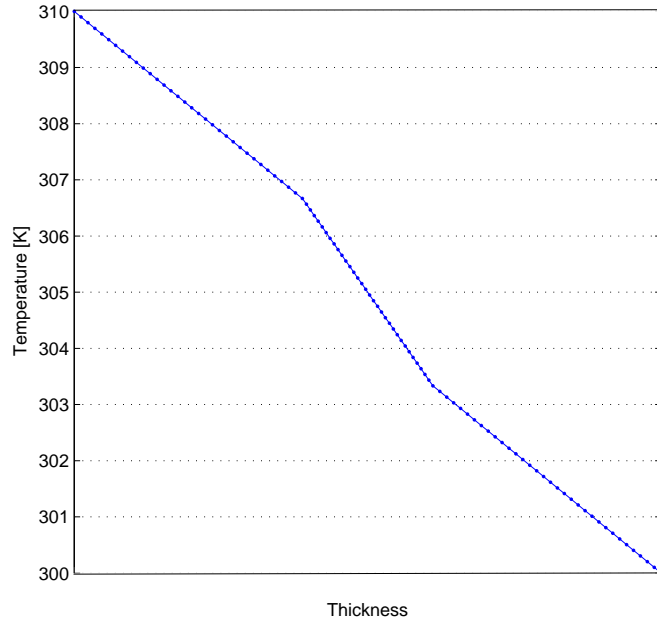


Figure 4.15: Simulation of the heat flow sensor.

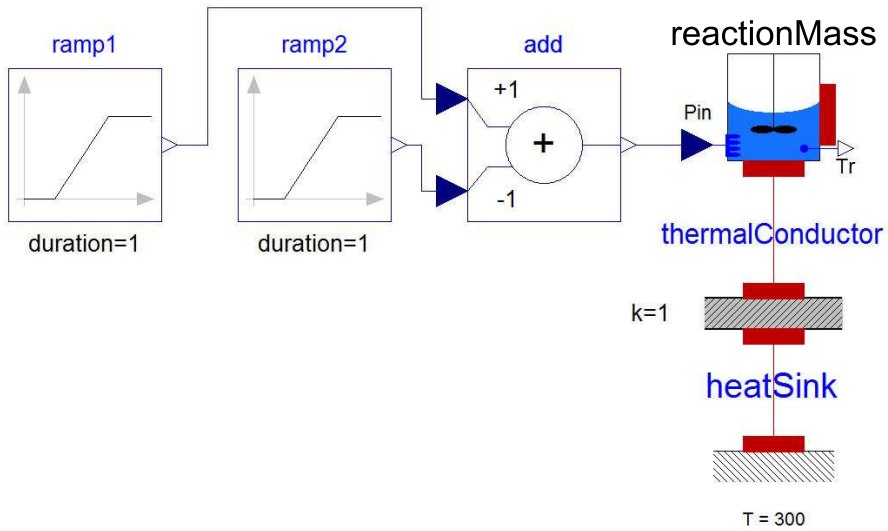


Figure 4.16: The Dymola setup of the thermal conductor test.

Parameter	Value	Unit
k	0.18	W/m ² K
r	0.0275	m
d	0.00001	m

Table 4.6: Simulation parameters for the thermal conductor element in the thermal conductor test.

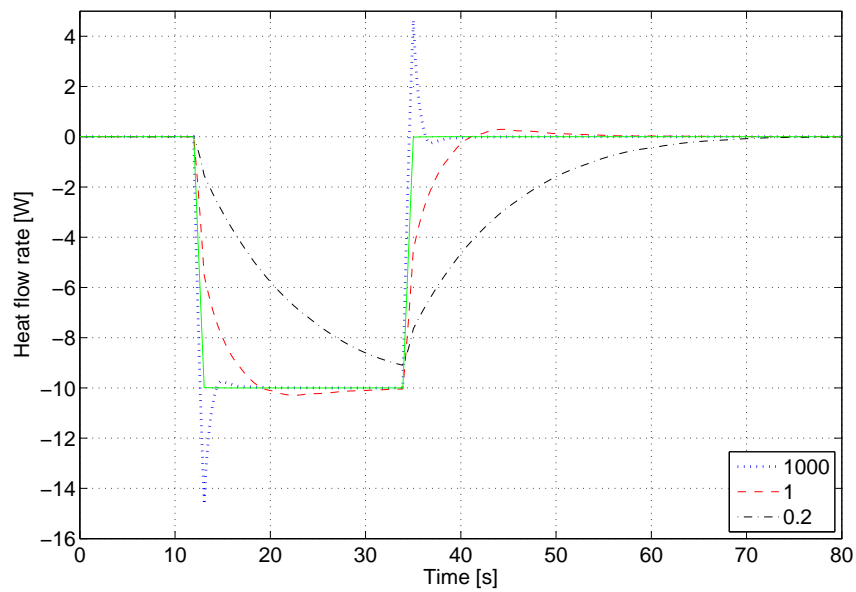


Figure 4.17: Simulation of the thermal conductor with different values of the thermal conductivity coefficient k.

about the same. There is a know error in this model regarding the heat flow. This is caused by the discretization in the radial direction. The heat flow on the outer surface is about 7 % higher than the input flow. The explanation is likely that the energy equation not is energy preserving. This should not cause any problem while the important part is on the inner side and the reference bath is assumed to be able to take care of superfluous heat.

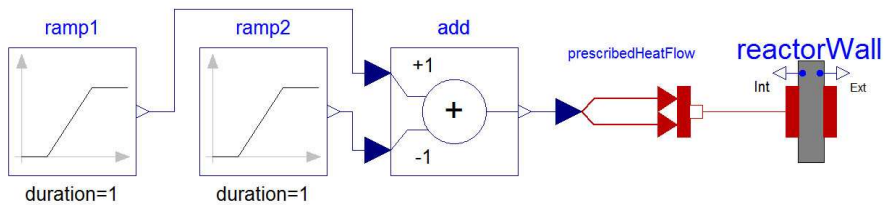


Figure 4.18: The Dymola setup for the reactor wall test.

Parameter	Value	Unit
r	0.0275	m
t	0.002	m
h	0.01	m
ρ	8010	kg/m ³
c	460	J/kgK
λ	13.8	W/mK
T_{init}	300	K
$T_{startext}$	300	K
initOpt	false	

Table 4.7: Simulation parameters for the reactor wall test.

4.9.7 Conclusions

The tests made on the model elements all seem to show that the elements behave correctly. As seen in the test on the reactor bottom, the heat flow sensor and the reactor wall, the number of nodes in the elements are of importance. 100 nodes or more give a good result. Naturally a large number of nodes give a better result, however too large a model may have too long execution time. Therefore 20 nodes per element seem like a reasonable choice. As more simulations were run even 20 nodes per element was too time demanding, especially in the model calibration part, it was therefore adjusted to 15 nodes per element. The difference between 15 and 20 nodes

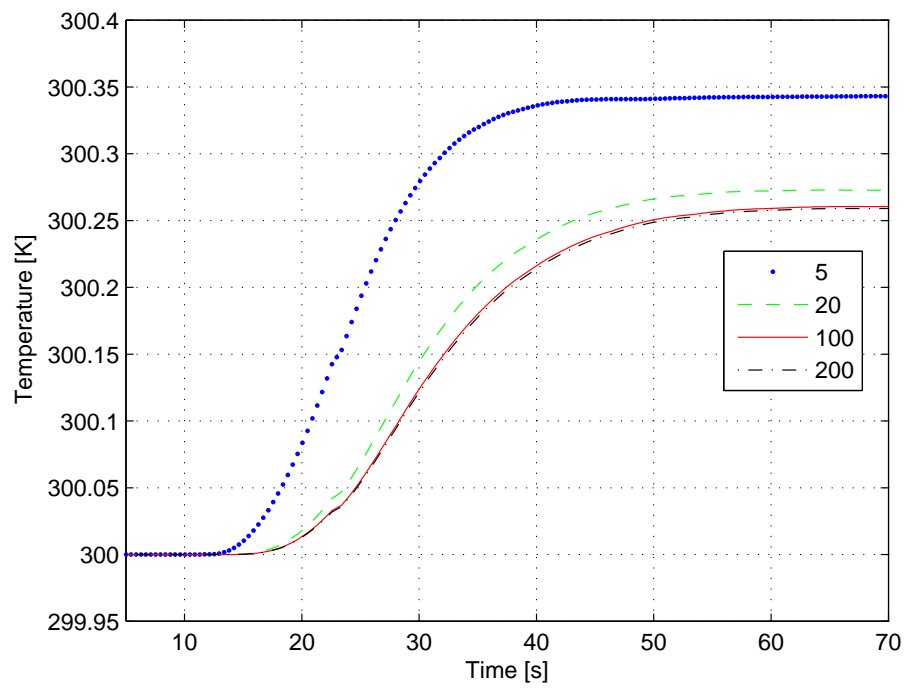


Figure 4.19: Simulation of the temperature in the reactor wall with different number of nodes.

should give an error in temperature of less than 0.5°C , this can be reasoned from Figure 4.7.

4.10 Calibration of the model

To get an idea of how accurate the Dymola model is, experiments were performed on the real calorimeter. These tests showed that the Dymola model was more ideal than the real model. For example, the peltier element in the model was much more efficient than the real peltier element. The heat conductance in the model did not seem to be correct compared to the real setup. Also, the exact heat conductance coefficients were not known. These parameters depend on several factors. For example between the fluid and the reactor wall or bottom the heat conductance depends on the fluid velocity at the contact surface. Model calibration was therefore made to get a more accurate model.

This was done by the additional Dymola option Calibration in the package *Design*. The model calibration option *Calibrate* was used on the calorimeter model. Calibrations are necessary for each fluid or chemicals that are to be used in the reactor, due to the differences in material properties for different materials.

4.10.1 Experimental data

In order to calibrate the Dymola model, data from the real calorimeter was needed. A schematic picture of the experimental setup is shown in Figure 4.20.

In the experiment the reactor was filled with 100 g of the fluid. The controller in the control unit was disabled and the current I to the peltier element was manually controlled. The current was set to a constant value to avoid condensation on the inside surface of the lid. The condensation may otherwise cause heat losses through the lid. After activating the current but before the actual experiments begun the reactor was allowed to reach equilibrium. To simulate a chemical reaction with heat production an electrical heater in the reaction calorimeter was used. The heater was changed sequentially. The changes were made as a pulse and a step. The pulse is interesting to see how fast the system responds, it is also useful for the calibration. The step response is interesting to see how the system responds to a sudden increase of heat production.

The experiments were performed with water and with ethylene glycol. Ethylene glycol is an oil with higher viscosity than water. The oil was tested because it is interesting to see how well the model work on another substance than water. A comparison between the material parameters for water and ethylene glycol is made in Table 4.8.

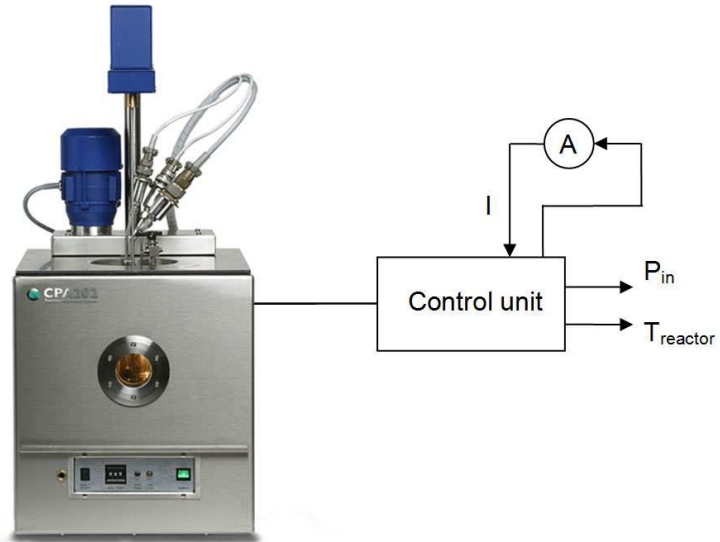


Figure 4.20: Simple picture of the real experimental setup.

Parameter	Water	Ethylene glycol
ρ [kg/m ³]	998	1118
λ [W/mK]	0.6	0.2
c [J/kgK]	4190	2303
ν [Ns/m ²]	0.0010	0.0036

Table 4.8: Material parameters for water and ethylene glycol. ρ is the density, λ is the heat conductivity, c is the specific heat capacity and ν is the viscosity.

By mistake the experiments were made at different currents, $I = 0.5$ A for water and $I = 1$ A for ethylene glycol.

Several experiments were made but only two were successful. The other ones failed because of mistakes made during the experiments. For example different reactors were used at different experiments making it impossible to compare the results. The reactors are unique in their performance. It would have been desirable with at least two sets of data to be able to verify the Dymola calibration. This was discovered late in the work and there was no time to make new experiments. Every experiment took more than half a day.

4.10.2 The calibration setup

The calibration was started with the model using water in the reactor. Experimental data for the reactor temperature T_r , the input power P_{in} to the reaction mass and the current I to the peltier element were used as reference data and input signals. The model setup was made to the given specifications (see Table A.1).

A hard task was to find out which parameters to calibrate. At first, parameters that were considered unknown or uncertain were adjusted and tested. "trial and error" calibrations were also made, just to see how much effect the parameter had. The first guess of parameters that were considered as unknown or uncertain were the heat transfer coefficients for the bottom α_{bottom} and the wall α_{wall} . Trying to calibrate these parameters was unsuccessful. The parameter α_{wall} got different values between 1000 to 10000 in every calibration under the same conditions. The only difference was the max value that was set for the heat conductance in the calibration. This was not realistic. The influence on the reactor temperature was small, only a few degrees. The problem seemed to be the thermal conductivity in the wall ($\lambda = 13.8$) that was the limit for how much power that could pass through the wall. Adjusting the wall specifications would solve the problem and the parameters that were considered as uncertain were the wall height h and the wall thickness t due to a very complex geometry. The wall geometry was assumed from the beginning very simple and as solid but would in this case show to be wrong.

The results for the calibrated parameters, heat transfer coefficients for the reactor bottom α_{bottom} and reactor wall α_{wall} and the wall thickness t and height h , is presented in Table 4.9. The careful reader may notice that there is a rather large difference between the calibrated values of water and ethylene glycol for α_{bottom} . This can be explained by the fact that different reactors were used for the calibration experiments. The error between the experimental result and the calibrated result was less than 0.3 °C, this is shown in Figure 4.21 and 4.22. As seen in the figures the calibration results are very good. The curves are practically on top of each other.

4.10.3 Calibration results

The figures presented in this section are the result of the model calibration of the models using water and ethylene glycol.

Parameter	Calibrated value	Nominal value	Unit
<u>Water</u>			
α_{bottom}	709.367	250	W/m ² K
α_{wall}	504.677	500	W/m ² K
t	0.0417267	0.02	m
h	0.0466921	0.016	m
<u>Ethylene glycol</u>			
α_{bottom}	250	250	W/m ² K
α_{wall}	500	500	W/m ² K
t	0.0268912	0.02	m
h	0.0226459	0.016	m

Table 4.9: This table shows the calibrated and nominal parameters for the models with water and ethylene glycol.

4.10 Calibration of the model

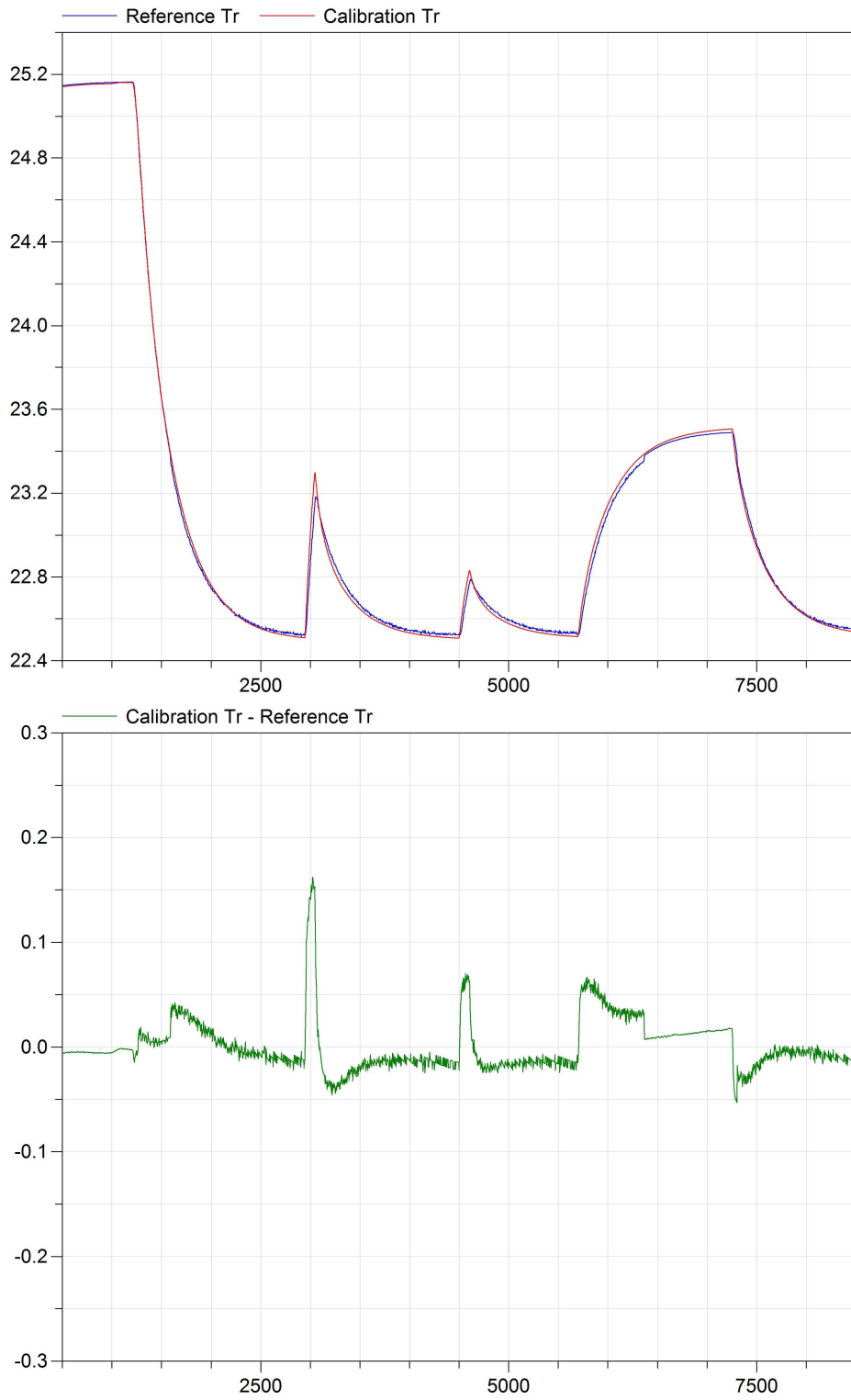


Figure 4.21: Simulation of the calibrated model with water for the constant current $I = 0.5$ A. The second plot is the error between the reference plot and the calibrated model plot. The y-axis is temperature [$^{\circ}C$] and the x-axis is time [s].

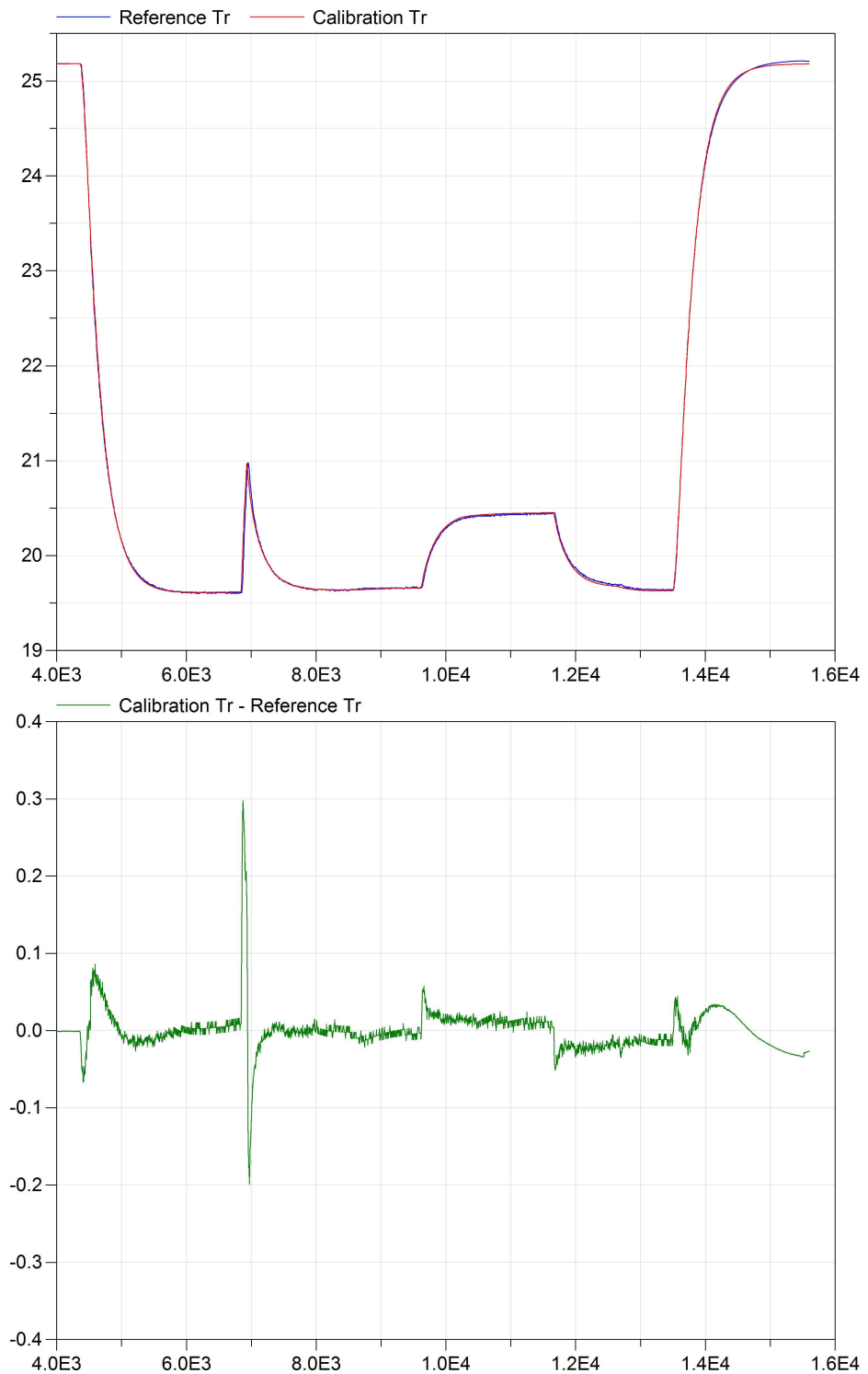


Figure 4.22: Simulation of the calibrated model with ethylene glycol for the constant current $I = 1$ A. The second plot is the error between the reference plot and the calibrated model plot. The y-axis is temperature [$^{\circ}C$] and the x-axis is time [s].

Chapter 5

Control of the calorimeter

This chapter deals with the control of the calorimeter. First the cascade control implemented by Chemisens is discussed. Then two alternative control algorithms are evaluated, PID control and LQG. These methods were only tested on the computer model, not on the real calorimeter. The controllers are described one at a time and in the final section they are discussed and compared.

5.1 Cascade control

The idea behind cascade control [6] is to use two controllers and let the first controller give the reference value for the second controller, the general structure is shown in Figure 5.1. This gives faster control for systems with build in time delays, like the calorimeter. However, one more controller requires one more measurement and this is not always possible.

The general method for tuning a cascade controller is by first disconnecting the outer loop (PID1) and tuning the inner loop (PID2). When a satisfactory behaviour of the inner loop is achieved, the outer loop is connected and both of the controllers are tuned together.

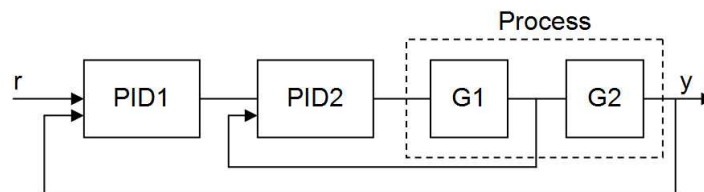


Figure 5.1: Basic structure of a cascade controller. r is the reference value and y is the output signal. PID1 is the outer loop controller, PID2 is the inner loop controller and G1, G2 presents the process.

5.1.1 The Chemisens cascade controller

Now that the theory behind the cascade control has been established the cascade control of the calorimeter will be described. In the calorimeter setup the inner controller is a P-controller and the outer is a PI-controller, this is shown in Figure 5.2. The outer controller minimizes error $e_1 = T_{ref} - T_{reactor}$ and gives a set value for the inner controller. The inner loop minimizes the error $e_2 = u_1 - (T_{ref} - T_{sensor})$. The control signal u_2 is then the current to the peltier element. The inner loop is faster because the measurement T_{sensor} is closer to the peltier element. The inner controller compensates for the fact that the control signal u_1 could be too large. This is because the response from the temperature measurement $T_{reactor}$ is delayed. Cascade control is therefore a very appropriate control method for the calorimeter.

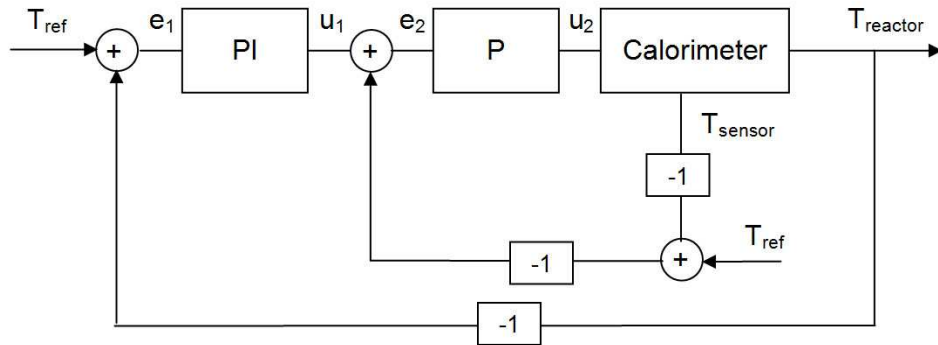


Figure 5.2: The control structure of the cascade control. T_{ref} is the reference temperature, T_{sensor} is the temperature for the heat flow sensor and $T_{reactor}$ is the temperature in the fluid.

To find out if the control algorithm for the calorimeter could be improved a couple of other controllers were also implemented in the model. Next a PID-control scheme was tested.

5.2 PID-control

Since PID control is such a well known and widely used control method it was natural to try to implement it on the calorimeter. The controller is easy to tune since there are few parameters. The basic PID structure with negative feedback is shown in Figure 5.3. There are many tuning methods for PID control. Two tuning methods are used in this project, Ziegler Nichols and Approximate MIGO design.

5.2.1 Ziegler Nichols step response method

One of the most used tuning methods for PID control is the Ziegler Nichols [7] step response method. In this method two parameters are obtained from the step response of the open system, see Figure 5.4. From these parameters the control parameters K , T_i and T_d can then be calculated according to Table 5.1.

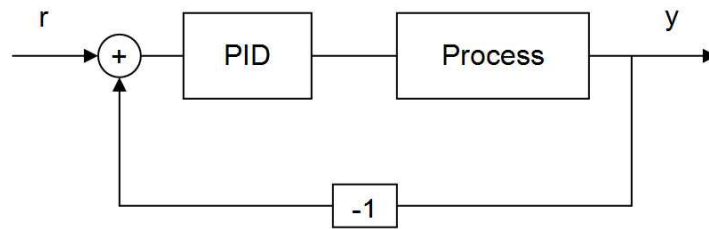


Figure 5.3: Basic structure for PID control of a process.

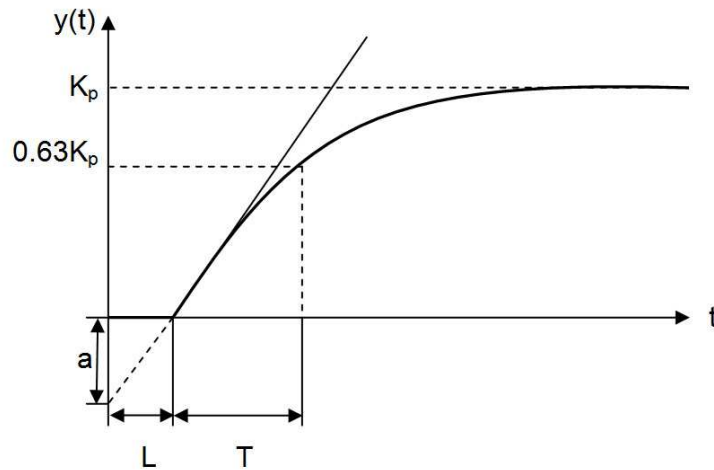


Figure 5.4: Characterisation of a step response for calculation of PID parameters in Ziegler Nichols step response method and AMIGO design method. a and L are used for Ziegler Nichols and L , T and K_p are used for AMIGO.

Ziegler Nichols step response method is a very simple method that only gives a rough idea of the controller parameters. A second more advanced method was therefore also tested.

Controller	aK	T _i /L	T _d /L
P	1		
PI	0.9	3	
PID	1.2	2	L/2

Table 5.1: Controller parameters for Ziegler Nichols step response method. K, T_i and T_d are the controller parameters.

5.2.2 Approximate MIGO design

AMIGO is a more elaborate method for tuning a PID controller. The tuning method is presented in Advanced PID control [8]. Approximate MIGO design is a simplification of the more advanced MIGO method, MIGO stands for M Constrained Integral Gain Optimization. Simply put, the MIGO method calculates PID parameters for a batch of different processes based on the information from them. All these sets of PID parameters are then used to obtain a set of simple tuning rules. These rules are called the Approximate MIGO method. These equations are given by

$$\begin{aligned}K &= \frac{1}{K_p} \left(0.2 + 0.45 \frac{T}{L} \right) \\T_i &= \frac{0.4L + 0.8T}{L + 0.1T} L \\T_d &= \frac{0.5LT}{0.3L + T}\end{aligned}$$

where the parameters T, L and K_p are obtained from the Figure 5.4. The advantage of this method to the Zeigler-Nichols method presented before is that AMGIO design uses more process information and thus requires less tuning once the parameters are calculated.

5.2.3 Comparison of Ziegler Nichols and AMIGO

The Ziegler Nichols method is very simple a widely used and the underlying theory is less complex in comparison to AMIGO. AMIGO is recently developed and therefore not as well known. Both methods are based on a step response and the process of obtaining the parameters is similar. Since AMIGO uses more process information the control parameters should be more reliable than those obtained from Ziegler Nichols.

5.3 LQG

The last control method presented is Linear Quadratic Gaussian control. LQG is a combination of a Kalman filter (Linear Quadratic Estimator) and a state feedback controller (Linear Quadratic Regulator). The Kalman filter predicts the states recursively and the state feedback controller calculates the optimal feedback control law that minimizes the cost function for the system. To deal with step disturbances one can introduce an integrator state.

An optimal Kalman filter estimates the states according to

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + K(y - \hat{y}) \\ \hat{y} &= C\hat{x}\end{aligned}\quad (5.1)$$

where

$$K = (PC^T + R_{12})R_2^{-1}$$

minimizes the error covariance for the system. $P > 0$ is the solution to

$$R_1 + AP + PA^T - (PC^T + R_{12})R_2^{-1}(PC^T + R_{12})^T = 0 \quad (5.2)$$

and R_1, R_{12}, R_2 are weights chosen to minimize the error covariance.

The next step is to calculate a controller based on the Kalman filter. Consider the linear continuous time system described by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\quad (5.3)$$

with the start value $x(0) = x_0$ and a cost function defined as

$$J = \int_0^\infty \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T Q \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \quad (5.4)$$

The Linear Quadratic Regulator is then given by the feedback control law

$$u = -Lx \quad (5.5)$$

where

$$L = Q_2^{-1}(SB + Q_{12})^T$$

minimizes the cost function J 5.4 and S is given by the algebraic Riccati equation

$$Q_1 + A^T S + SA - (SB + Q_{12})Q_2^{-1}(SB + Q_{12})^T = 0 \quad (5.6)$$

The optimal controller u will then give a stable closed-loop system if the weights Q_1, Q_{12}, Q_2 are chosen in such a way that

$$Q = \begin{pmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{pmatrix} \quad (5.7)$$

is positive semi definite. The rules of thumb for choosing the weights are to

- set $Q_{12} = 0$ and make Q_1 and Q_2 diagonal
- set $Q_1 = C^T C$ and Q_2 to be small

R has the same structure as Q and should also be positive definite. In the case of choosing the weights for R set $R_{12} = 0$ and make R_1 and R_2 diagonal. The simplest way is to let R be the identity matrix.

A drawback with this LQG controller is that it does not handle step disturbances and changes in the reference signal. A step input will give a constant error that the regulator cannot handle. The solution is to introduce integral action. This is done by treating the error $e = r - y$ as a load disturbance d . To create a controller that handles load disturbances one need an observer that takes the disturbance into account. This is done by introducing the augmented system [9]

$$\begin{aligned} \dot{x}_e &= \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} x_e + \begin{bmatrix} B \\ 0 \end{bmatrix} u = A_e x_e + B_e u \\ y &= \begin{bmatrix} C & 0 \end{bmatrix} x_e = C_e x_e \end{aligned} \quad (5.8)$$

where $x_e = [x^T d^T]^T$. By using the augmented system in the equations for a Kalman filter one will get the optimal gain $K_e = [K_x^T K_d^T]^T$. The state-space system for the controller is then described by

$$\begin{aligned} \begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{d}} \end{bmatrix} &= \begin{bmatrix} A - BL_x - K_x C & 0 \\ -K_d C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} K_x & BL_r \\ K_d & 0 \end{bmatrix} \begin{bmatrix} y \\ r \end{bmatrix} \\ u &= - \begin{bmatrix} L_x & 1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} 0 & L_r \end{bmatrix} \begin{bmatrix} y \\ r \end{bmatrix} = -L_e \hat{x}_e + L_r r \end{aligned} \quad (5.9)$$

where u is the control signal with integral action and r is the reference value that the controller will try to establish. The controller will have the equilibrium for $y = r$ if the gain L_r is chosen to be

$$L_r = (C(-A + BL_x)^{-1}B)^{-1} \quad (5.10)$$

The structure is shown in Figure 5.5.

The advantage of LQG control is that since the process is modelled in the controller it should in theory be possible to get very good control for almost any system. However in reality it can be difficult to implement since LQG often requires a high order model to get good control. In reality there is a compromise between the order of the model and limitations in the hardware. For example if the process model has too many states it would be too demanding to estimate the states in the Kalman filter.

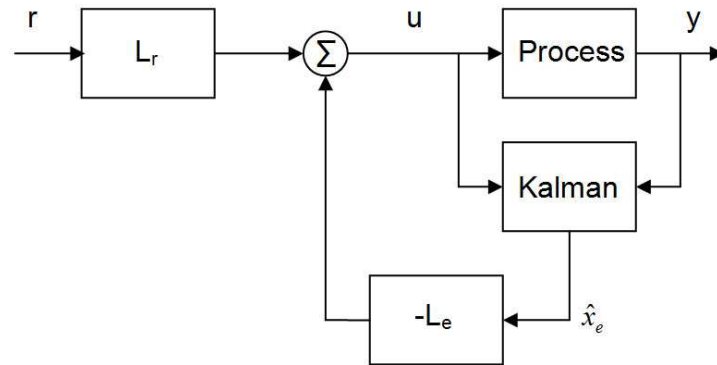


Figure 5.5: Basic structure for LQG control with a disturbance observer. The controller has the equilibrium r if L_r is chosen according to Equation 5.10.

5.4 Comparison of the control methods

The different control methods all have advantages and disadvantages. The cascade control method has the advantage of two measurements. This is good when one measured signal is less reliable or if there is a time delay in the system. The response of the system will then be faster. The need for two signals can also be a disadvantage since a second measurement is not always available.

The PID controller is easy to implement and understand. This method works well for a large number of processes. A disadvantage is that the derivative part can cause large variations in the output signal for a noisy measurement.

LQG has the advantage that if it can be implemented, it results in a good controller for that system. However LQG cannot be used on any system. For example if the system is too large the estimations may be too much for the hardware.

The conclusion is the most important is to choose a control method suitable for the process at hand.

5.5 Results and discussion of control methods

In this section the results from the implementation of the control methods are presented and discussed.

5.5.1 Cascade control

The controller parameters can be seen in Table 5.2 below. The result is a fast controller that efficiently handles load disturbances and reference changes.

The simulated controller is slightly faster than the real process. This can be expected, as the model is an ideal version of the real process.

$K_{innerloop}$	$K_{outerloop}$	$Ti_{outerloop}$
2	9	50

Table 5.2: Controller parameters for the cascade controller.

5.5.2 PID control

The results from the two tuning methods Ziegler Nichols and Approximate MIGO design are presented in this section.

Ziegler-Nichols step response method

The Ziegler Nichols method uses information from the open-loop step response. The response is described by the two parameters a and L , see Figure 5.4 and Table 5.3. A tangent is drawn along the point of the step response with the steepest slope. The intersection of the tangent with the coordinate axes then determines a and L . This is not an exact method as it is difficult to determine the point with the steepest slope. The calculated parameters are therefore not definite but give a good starting point for further manual tuning of the controller.

The PID parameters were calculated from a step response on the water model and then tested on the water model and the ethylene glycol model. During the tuning the aim was to find one set of parameters that would work for both of the models. This was done by testing parameters alternately on the two models.

The original parameters, shown in Table 5.4, did not give satisfactory results. The controller did not manage to compensate for the disturbances in either model and saturated repeatedly. The final tuned parameters also found in Table 5.4 gave better results.

a	L
0.1	30

Table 5.3: The parameters a and L for the Ziegler Nichols step response method

AMIGO

The results from the AMIGO method are presented in Table 5.5 and Table 5.6. The untuned parameters worked, although not very well. The controller

	P	I	D
Untuned parameters	12	60	450
Tuned parameters	2	60	60

Table 5.4: The controller parameters for Ziegler Nichols step response method.

was too slow.

a	L	K_p	T
0.1	30	1.66	90

Table 5.5: The AMIGO parameters a, L, K_p and T for the system step response.

	K	T_i	T_d
Untuned parameters	0.93	64.6	13.6
Tuned parameters	2	20	60

Table 5.6: Controller parameters for the AMIGO method. Tuned and untuned parameters.

In Figure 5.6 the bode diagram of the untuned and tuned Ziegler Nichols and AMIGO controllers are shown. To the left the open loop systems for the untuned parameters are shown. It can be seen that the AMIGO controller behaves better for frequencies up to 1 radians per second. To the right the tuned controllers are shown. The difference between the controllers are now of course very small. It may seem that the AMIG controller is not as good as before but this is due to the fact that the tuned controller should work for water and ethylene glycol.

Because the difference between the Ziegler Nichols and AMIGO controllers are so small from now on only the AMIGO controller will be discussed in mentioning the PID-controller.

5.5.3 LQG

As mentioned before the number of states in the model was a big factor in the design of the LQG controller. The original model contained 79 states corresponding to 15 nodes per discretized element. The model was then reduced in Matlab, with the command Minreal, giving 21 states. Figure 5.7 shows that the reduced system behaves as the unreduced.

Another important part of the LQG design was the choice of the weight matrices. Since the system is so large it is impossible to weights in the Q

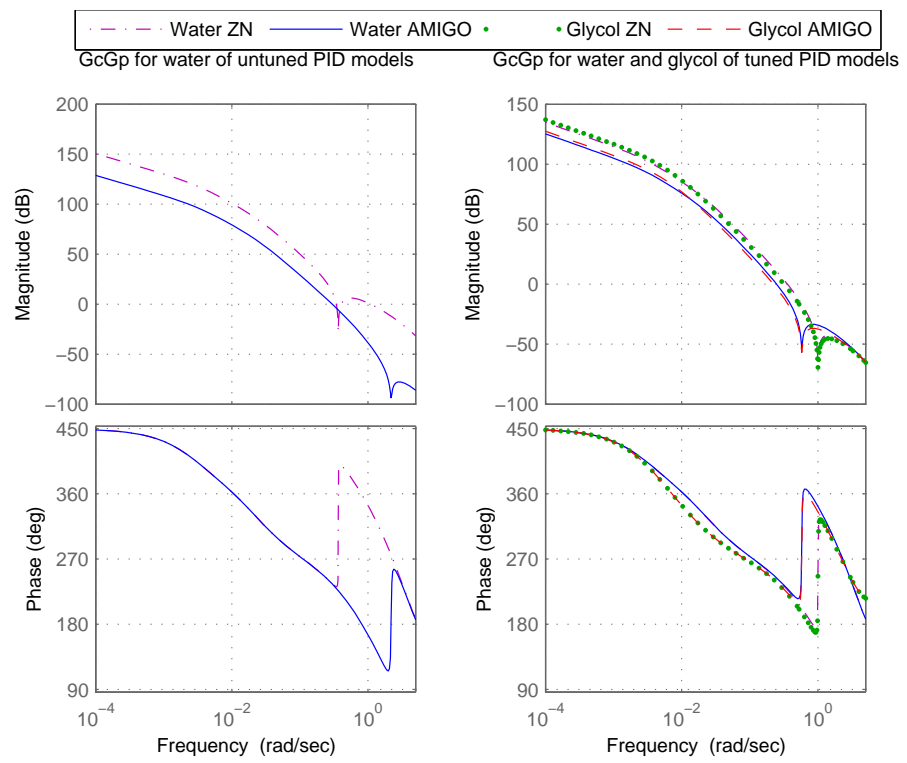


Figure 5.6: The bode diagrams for the PID controllers. To the left the untuned controller and to the right the tuned.

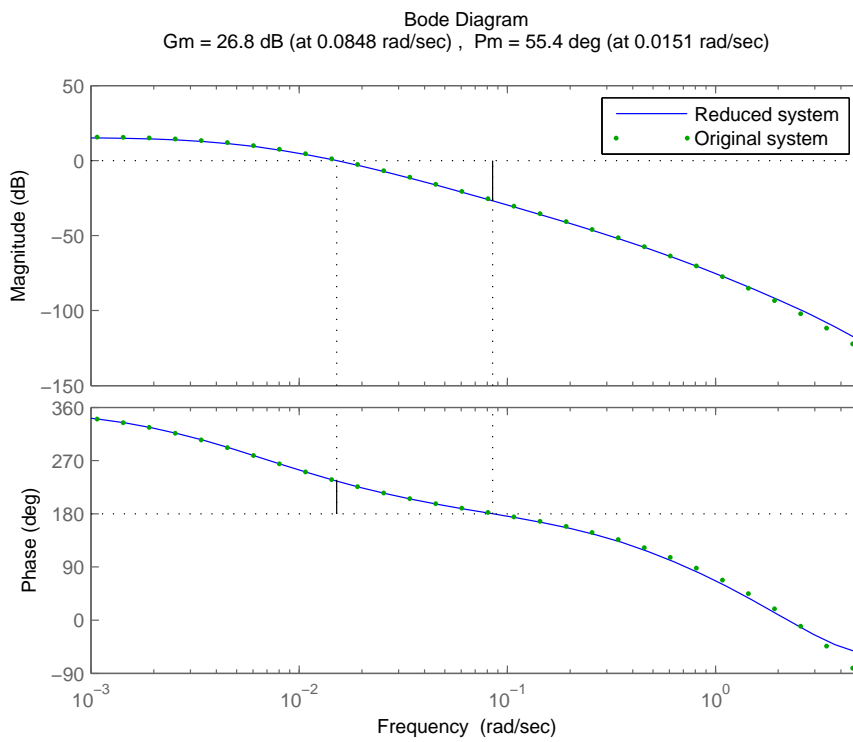


Figure 5.7: The bode diagrams for the reduced and unreduced systems.

matrix individually. The rule of thumb $Q_1 = C^T C$ was therefore used and Q_2 was set to a very small value. In the case of choosing R this was done according to the rule of thumb $R = I$. This worked very well.

5.5.4 Comparison

In this section the control methods are compared to each other. In Figures 5.8 and 5.9 the open loop bode diagrams of models with water and ethylene glycol are shown. The cascade and LQG controllers both work well. The PID controller has larger gain and is therefore fast. It also has a dip around 0.6 radians per second, this can be explained by zeros close to the imaginary axis. There is no apparent difference between the behaviour of the water and ethylene glycol models.

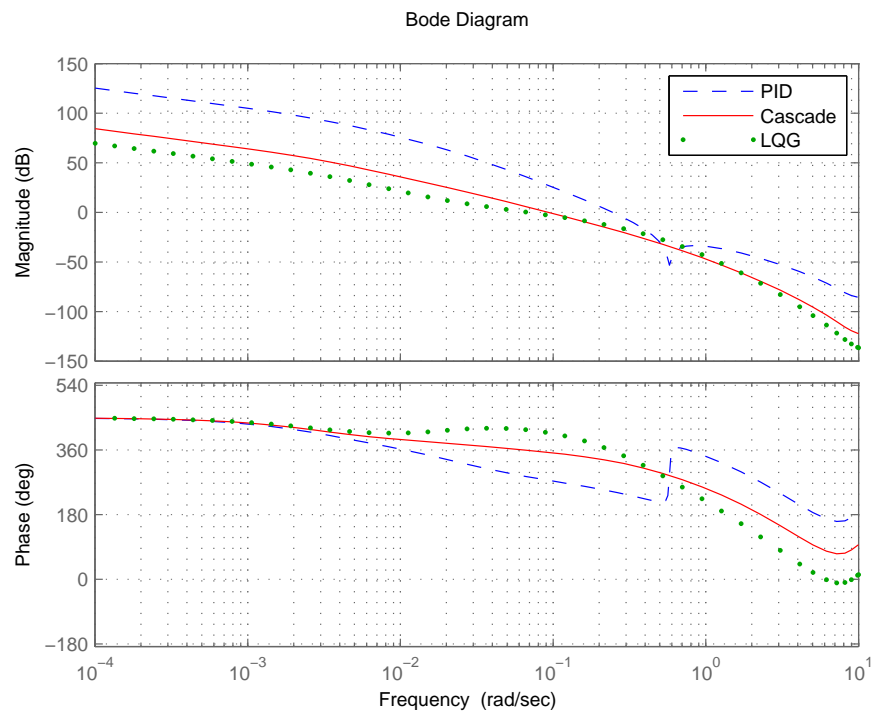


Figure 5.8: The bode diagram of the controlled open loop for water.

By plotting the Gang of Four one can gain a better understanding of the process. The gang of four is a set of transfer functions defined from measurement noise n and load disturbance d to output signal y and control signal u . See Figure 5.10.

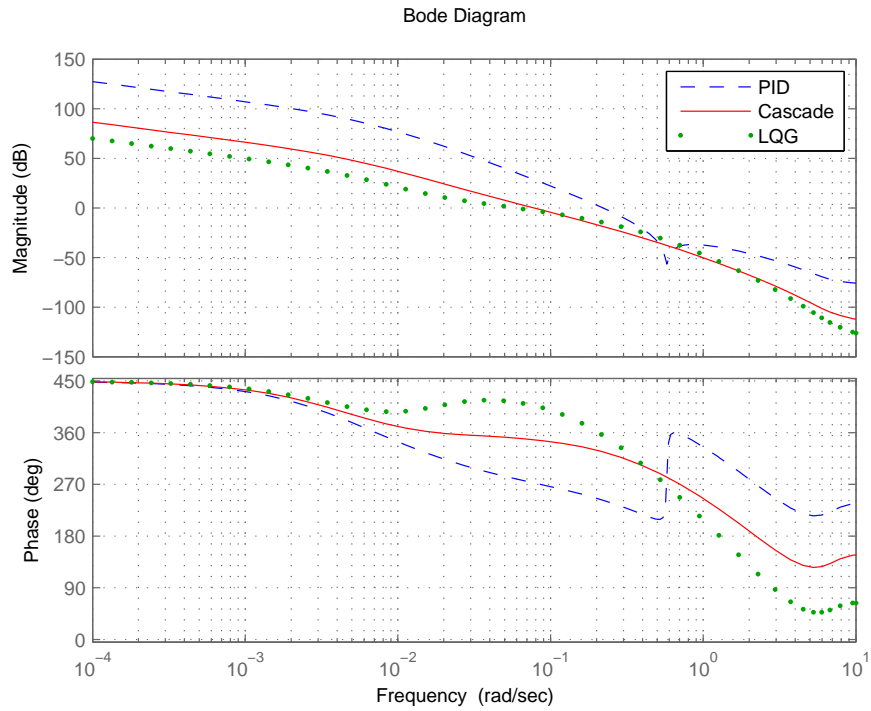


Figure 5.9: The bode diagram of the controlled open loop for ethylene glycol.

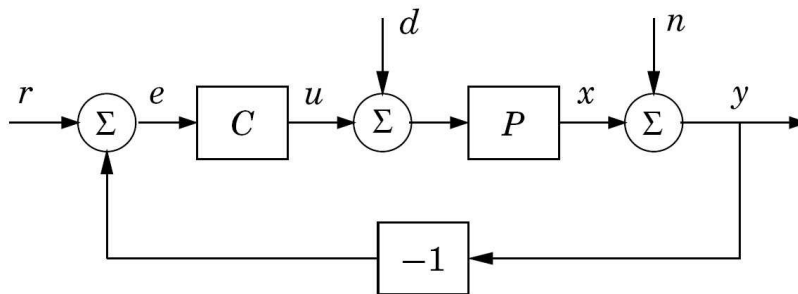


Figure 5.10: Simple block diagram of a basic feedback loop.

$\frac{1}{1+PC}$, $n \rightarrow y$, *the sensitivity function.*

$\frac{PC}{1+PC}$, $d \rightarrow u$, *the complementary sensitivity function.*

$\frac{P}{1+PC}$, $d \rightarrow y$, *the load disturbance sensitivity function.*

$\frac{C}{1+PC}$, $n \rightarrow u$, *the noise sensitivity function.*

The gang of four shows how well the process handles different types of disturbances. Perhaps the most important is the sensitivity function S and the complementary sensitivity function T . The sensitivity function tells how well the system deals with load disturbances and should be small for low frequencies. The complementary sensitivity function indicates how sensitive the process is to noise and should be small for high frequencies.

In Figures 5.11 and 5.12 the gang of four are plotted for water and ethylene glycol models. The sensitivity function can be seen in the upper left plot and all controllers handles load disturbances well, as they are small for low frequencies. The complementary sensitivity function can be seen in the upper right corner and is decreasing for high frequencies for all controllers. The lower left plot shows how well load disturbances are suppressed on the measurement signal and the PID controller seems to handle this best. As can be seen in the lower right plot the control signals in the PID and LQG controllers are quite sensitive to measurement noise. Only the cascade control seems to handle this type of disturbance well. Again there is no apparent difference between the behaviour of the water and ethylene glycol models. In total the cascade controller handles disturbance and measurement noise the best.

It is also interesting to study the process in the time domain. All previous plots have been made on the linearized model. The following plots however have been made on the nonlinearized model, in Dymola. The time domain studies were done by performing disturbance step responses and reference changes on the process. As can be seen Figures 5.13 and 5.14 the controllers all handle a disturbance step change well with a reasonable control signal. The LQG controller does not work as well for the water model as for the ethylene glycol model. The same is true for the cascade controller implemented in the water model. The work in this thesis has not been able to explain why. It would be interesting to further investigate this. The controllers handle the reference changes well, as is shown in the Figures 5.15 and 5.16. The careful reader may notice that the control signal saturates at 3.6 A. A limiter was connected before the input signal to the peltier element. This is because the real peltier element has a limit in performance after 3.6 A, in theory the peltier element can handle currents up to 6 A. However the limitation does not seem to affect the performance of the controllers and cause any integrator windup. This because the PID and cascade controllers

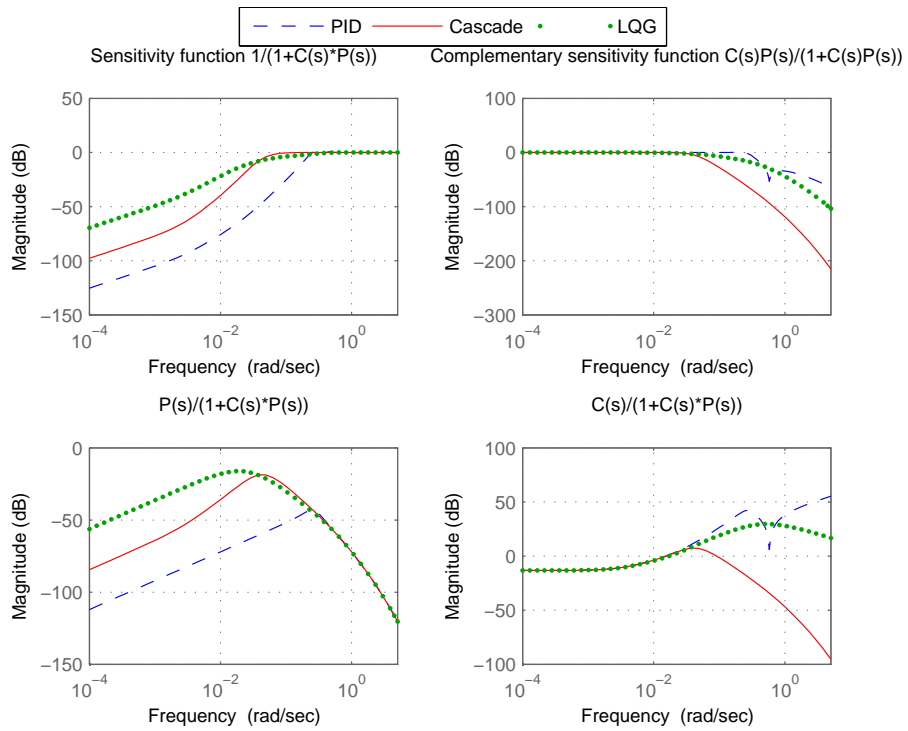


Figure 5.11: The plot of Gang of Four. For water.

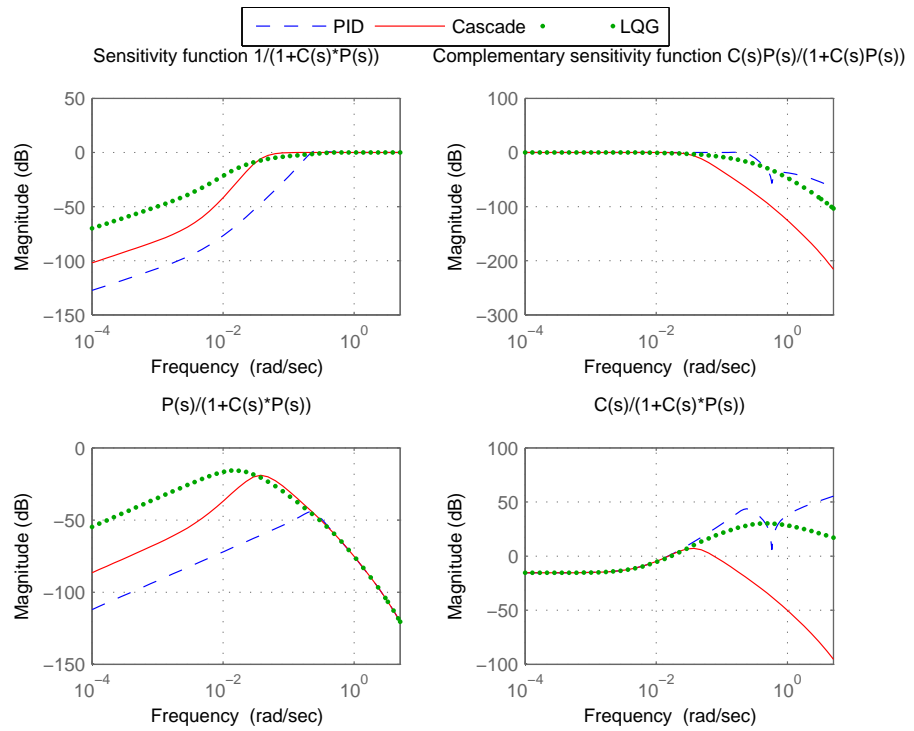


Figure 5.12: The plot of Gang of Four. For ethylene glycol.

were implemented with antiwindup.

5.6 Conclusions

In comparison the existing cascade controller implemented in the calorimeter is very good. One can therefore understand why this control structure was chosen. The process is well suited for cascade control. Because the process is slow it is favourable to have an inner faster loop.

The first control method tested was PID control. As mentioned before two tuning methods were used, Ziegler Nichols step response method and AMIGO tuning. Both methods require substantial tuning. But at least the original AMIGO parameters manage to control the process. For the tuned systems the AMIGO process is slightly better. It is possible to find a set of PID parameters that works well for both water and ethylene glycol. From the tuning process the experience was that if the parameters worked for ethylene glycol, they would also work for water. One might therefore argue that it would have been better to calculate the original parameters from the ethylene glycol, rather than water as was done here. The reason this was not done is that it was decided until quite late that ethylene glycol should be included in the thesis.

The LQG controller is also good, but could be hard to implement in reality as it requires a model of quite high order, at least 21, to function. Attempts at further reducing the model order were made, but they were unsuccessful. There are methods for reducing the model further, but these are not treated here as they are well beyond this thesis.

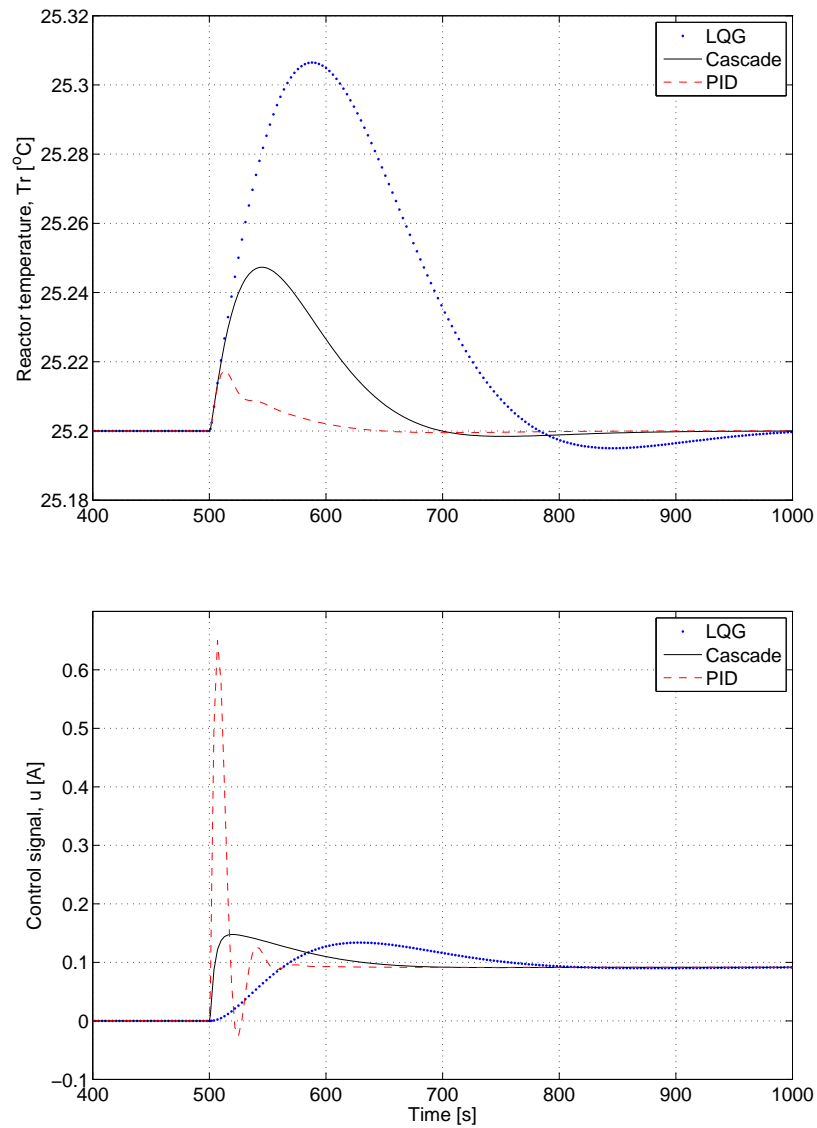


Figure 5.13: The step responses, with water in the reactor, for Tr with the controllers, LQG, cascade and PID. The second plot is the control signal u .

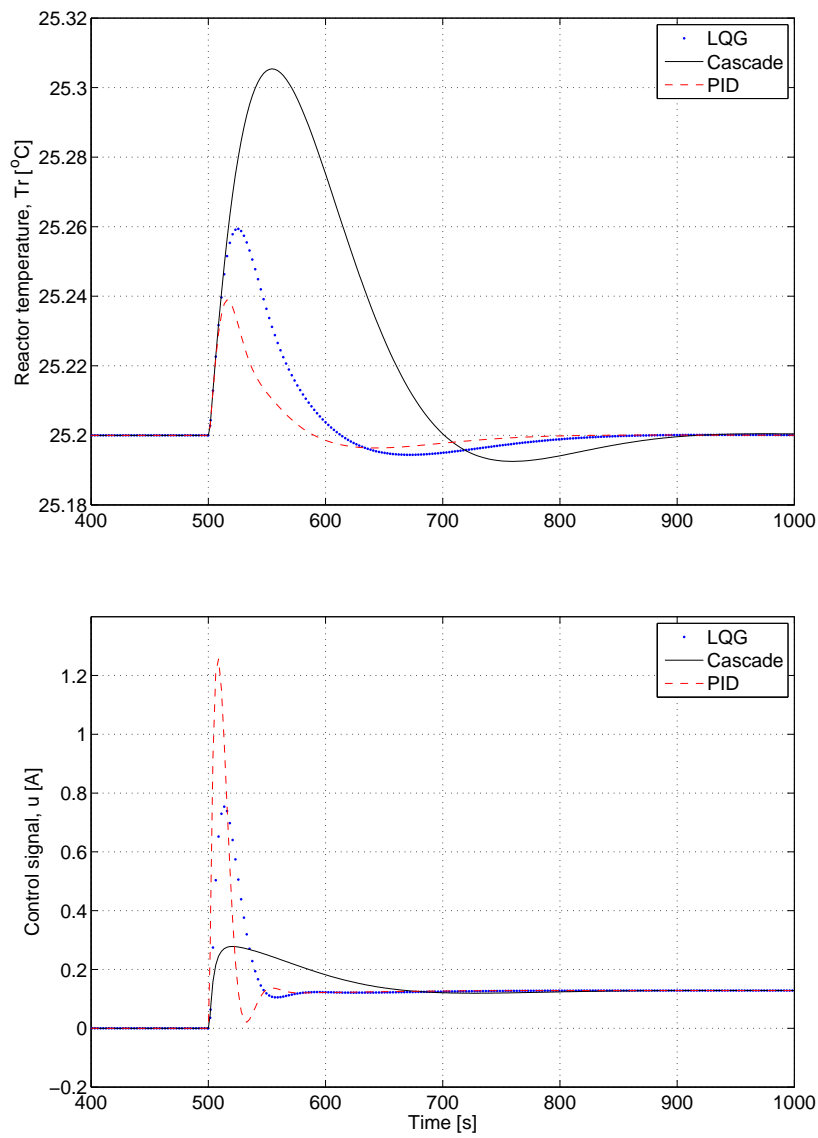


Figure 5.14: The step responses, with ethylene glycole in the reactor, for Tr with the controllers, LQG, cascade and PID. The second plot is the control signal u .

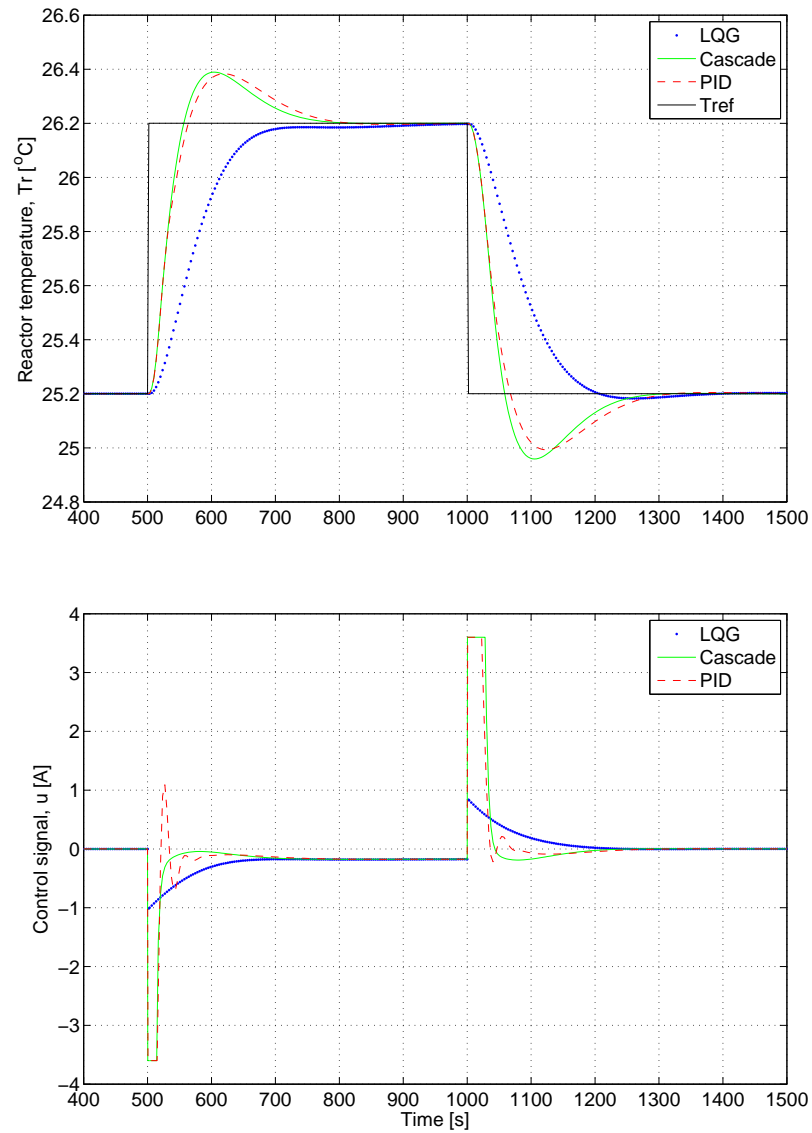


Figure 5.15: Reference change, with water in the reactor, for the set temperature in the reactor for Tr with the controllers, LQG, cascade and PID. The second plot is the control signal u .

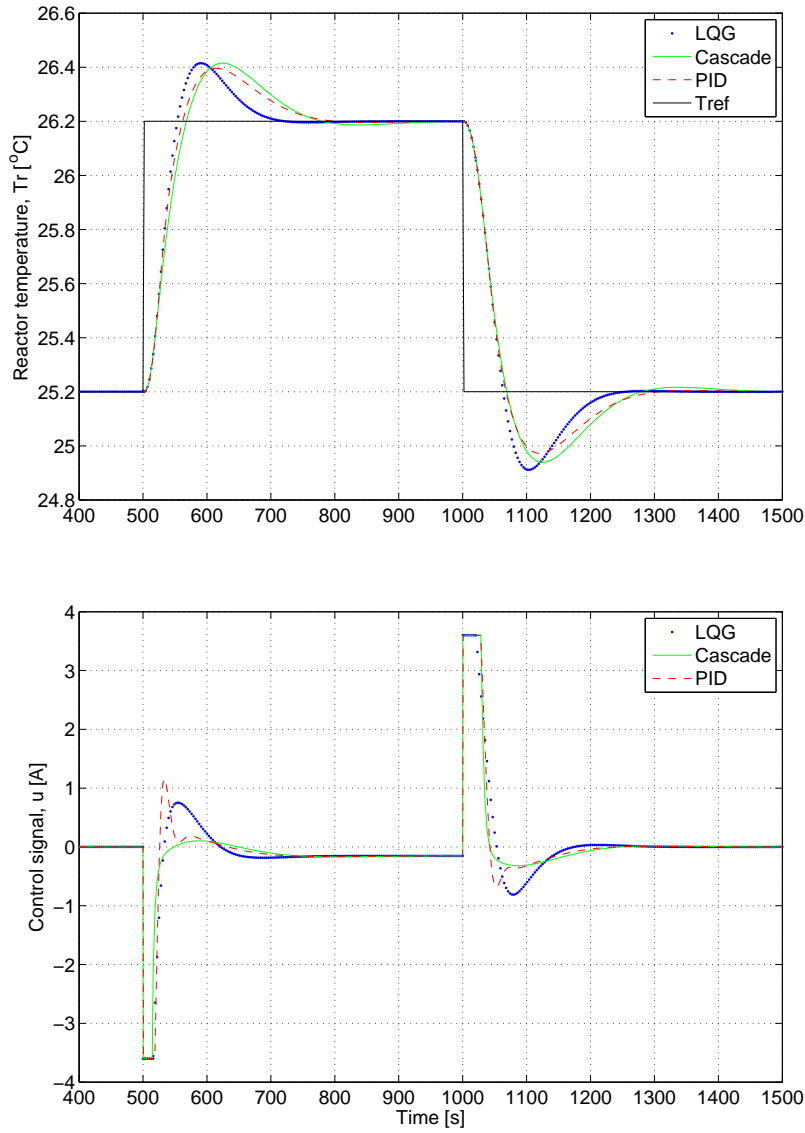


Figure 5.16: Reference change, with ethylene glycole in the reactor, for the set temperature in the reactor for Tr with the controllers, LQG, cascade and PID. The second plot is the control signal u .

Chapter 6

Conclusions and discussion

The modeling of the calorimeter has not been an easy task. Many things had to be considered and reflected about. It is difficult to know which parts of the calorimeter that are the most important. For example, there was some discussion between our supervisors whether it was necessary to model the heat flow in two dimensions or not. In the end the one dimensional model turned out to be very good. The model calibration improved the accuracy of the model a lot.

The choice of discretization in the elements was in the end chosen to be about 15 nodes in each element. This was because of not wanting a too large system to simulate in Dymola. The testing of the model element for example the reactor bottom also showed that about 20 nodes gave a pretty good result for a pulse heating. It also showed that the number of nodes not had so much affect for a step response. It might be possible to further reduce the total number of states if the heat flow sensor element simplified. Because the sensor consists of three layers the total number of nodes is almost multiplied by three, despite that the thickness is about the same as for the reactor bottom and the reactor wall.

The model calibration was also a hard task since the decision for which parameters to calibrate were to be made. The first decision trying to calibrate the heat conducting coefficients α_{bottom} and α_{wall} did not turn out that good. These parameters were calibrated to be very large approximately 1000-10000 depending on the calibration setup. Trying to calibrate these parameters gave no result. And the reason was that the limitation was the heat conducting coefficient in the bottom and the wall. The calibration of the wall dimensions was much more successfully since the assumption of a simple solid wall geometry was not completely correct. Calibration of both the heat conducting coefficients and the dimension of the wall gave good results.

One mistake that was made was that only one useful experimental data set for each fluid was obtained. This was discovered in the end. More

experiments were made but they were not performed on the same reactors and some small mistakes were made. Every reactor is an individual and an experiment performed on one reactor can therefore not be directly compared to an experiment made on another reactor. A second set of data would have been used to verify the model calibrations.

Some time was spent on investigating the control of the calorimeter. The conclusion is that the existing cascade control works very well. The other control methods, PID and LQG, were only implemented on the Dymola model. It would have been interesting to see how well they would work on the real calorimeter. It is interesting to see that the simple PID controller once again performs very well.

To get better control, the substance in the reactor needs to be considered. Especially the LQG controller, which is model based, needs to be retuned for every new substance. The cascade control however does not seem to be that sensitive to different substances. Thus being the most suitable controller for the calorimeter.

Chapter 7

Future work

In the future Chemisens, will hopefully be able to use the calorimeter model to develop a simulation software.

Some attempts at building a two dimensional model were made when the calibrations of the one dimensional model failed. However calibration problems were solved before the two dimensional model was completely finished. It would be interesting to know if two dimensions could substantially improve the model.

It might also be necessary to further look at the model reduction. The existing model is still quite large.

Another possibility for future work is to implement the PID and LQG controllers on the real process. It would be interesting to see if they work as well in reality as in the model.

Chapter 8

Experiences

In the thesis we learned that getting familiar to a new problem can take a lot of time. For example just learning Dymola took quite some time. Also studying all the background theory took time. It is also important to think through the experiments before carrying them out. This way a lot of time can be saved.

A serious time plan for this thesis was made in the beginning but as always it had to be revised again and again. What we have learned is that it is worth while to spend some time on really planning out the work.

Bibliography

- [1] Gunnar Sparr Annika Sparr. *Kontinuerliga system*. Studentlitteratur, Lund, Andra upplagan, 2000.
- [2] Modelica and the Modelica Association, 2008-02-25. *Modelica*.
<http://www.modelica.org>
- [3] Baehr, H.D., Stephan, K. *Wärme- und Stoffübertragung*. Springer Verlag, 2 edition, page 200-210, 1996.
- [4] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. IEEE Press, 2004.
- [5] Francesco Casella, 2008-03-11. *The ThermoPower library Home Page*.
<http://home.dei.polimi.it/casella/thermopower/>
- [6] Krister Forsman. *Reglerteknik för processindustrin*. Studentlitteratur, Lund, 2005.
- [7] Tore Hägglund. *Reglerteknik AK Föreläsningar*. Institutionen för reglerteknik Lunds tekniska högskola, Lund, 2003.
- [8] Karl J Åström, Tore Hägglund. *Advanced PID Control*. Instrumentation, Systems and Automation Society, 2006.
- [9] Johan Åkesson, Per Hagander. *Integral action - a disturbance observer approach*.
- [10] Ferrotec Corporation, Ferrotec Single Stage Thermoelectric Modules, 2007-09-20. *Peltier Cooler Model 9500/127/060 B*.
<http://www.ferrotec.com/products/thermal/modules/moduleDetail.php?id=70>

Appendix A

Model parameters

	Parameter	Value	Unit
Common parameters	r	0.0275	m
	Nr	20	
	initOpt	SteadyState	
	Tinit	298.33	°C
	G	400	W/K
	Tref	298.33	°C
	m	0.1	Kg
	specificHeatCapacity for water	4190	J/Kg.K
	specificHeatCapacity for ethyleneglycol	2303	J/Kg.K
ReactorBottomInner	density	8010	Kg/m ³
	specificHeatCapacity	460	J/Kg.K
	thermalConductivity	13.8	W/m.K
	t	0.004	m
ThermalConductor	k	0.18	W/m ² .K
	t	0.00001	m
Peltier	Sm	0.05253	V/K
	Rm	2.3553	Ω
	Km	0.5074	W/K
	currentCorrection	1/3.39	
ReactorBottomOuter	specificHeatCapacity	380	Kg/m ³
	density	8400	Kg/m ³
	thermalConductivity	115	W/m.K
ReactorWall	specificHeatCapacity	460	Kg/m ³
	density	8010	Kg/m ³
	thermalConductivity	13.8	W/m.K

Table A.1: Parameters used in the model for the calibration setup.

Appendix B

Peltier element performance

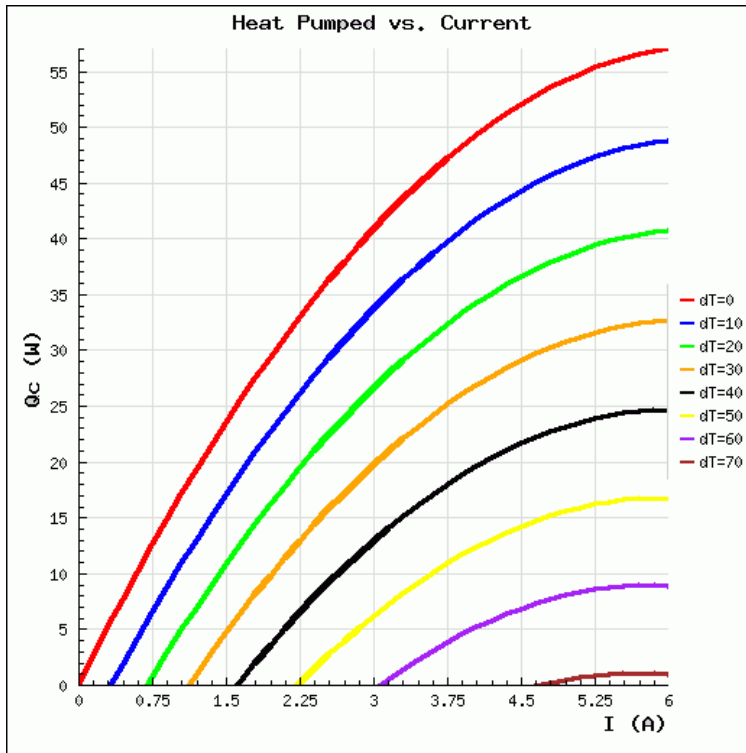


Figure B.1: The performance of Peltier Cooler Model 9500/127/060 B [10].