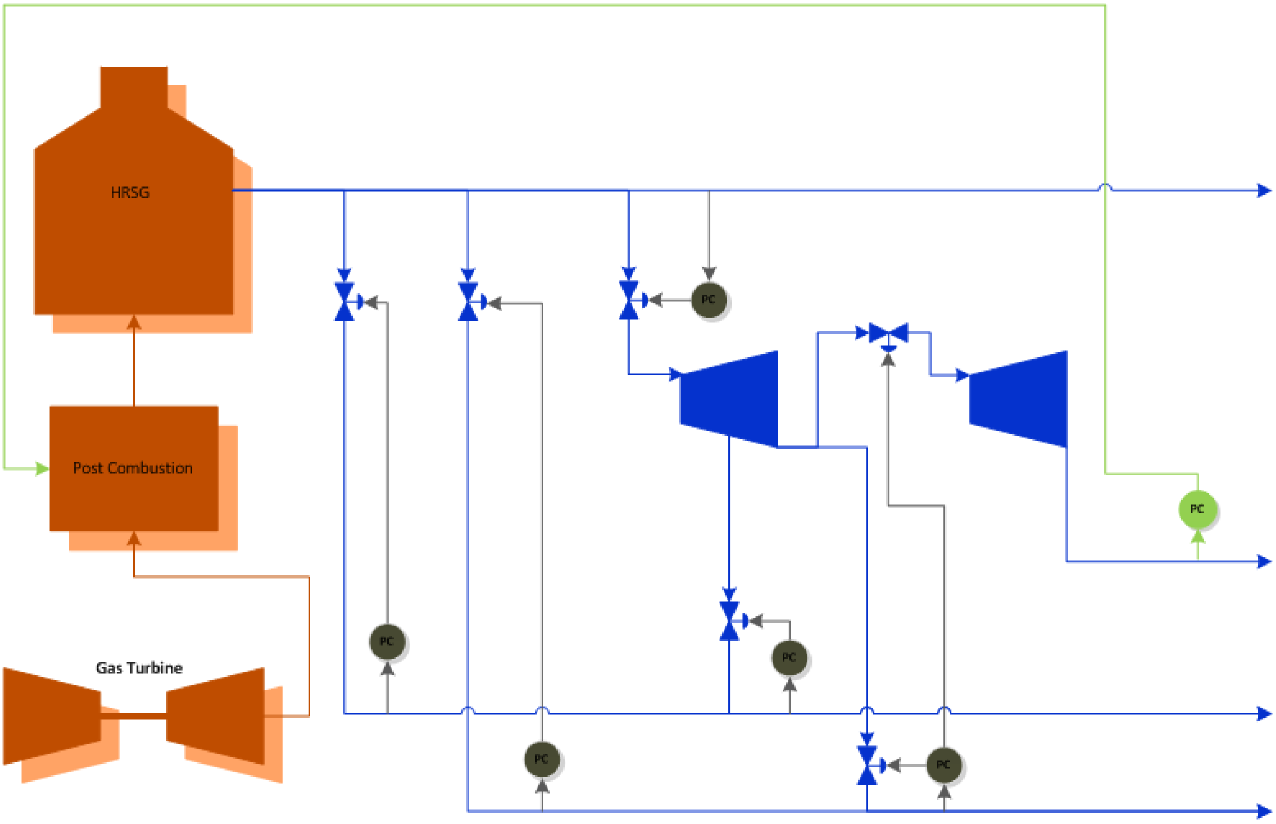


Control of a Process with Large Time Constants and Significant Time Delay

Modeling and Control of Post Combustion Capabilities to Improve Steam Production in a Utility Plant.

Master's thesis in Process control

HENRIK GRANBERG

Department of Signals and System
Division of Automatic Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013
Master's thesis 2013:EX002

MASTER'S THESIS IN PROCESS CONTROL

Control of a Process with Large Time Constants and Significant Time Delay

Modeling and Control of Post Combustion Capabilities to Improve Steam Production in a Utility Plant.

HENRIK GRANBERG

Department of Signals and System
Division of Automatic Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2013

Control of a Process with Large Time Constants and
Significant Time Delay
Modeling and Control of Post Combustion Capabilities to Improve Steam
Production in a Utility Plant.
HENRIK GRANBERG

Master's thesis 2013:EX002
ISSN 1652-8557
Department of Signals and System
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Some explanation

Chalmers Reproservice
Gothenburg, Sweden 2013

Control of a Process with Large Time Constants and
Significant Time Delay
Modeling and Control of Post Combustion Capabilities to Improve Steam
Production in a Utility Plant.
Master's thesis in Process control
HENRIK GRANBERG
Department of Signals and System
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology

Abstract

In a process plants utility system the steam demand can change rapidly. To reduce the amount of emissions to the environment and reduce the cost of fuel an efficient control system is needed to compensate for disturbances in the process.

In this thesis the control of a utility system with large time constants and significant time delay is studied. The utility plant consists of a gas turbine with post combustion capabilities for production of high pressure steam. The high pressure steam is expanded to low pressure steam with help of turbines before it condensates or is used by the process. The problem is to control the post combustion based on the flow to the condenser, which implies large time constants and significant time delay. A model of the utility plant was developed with help of the software Dymola, which can simulate large processes with many coupled equations. Different tuning methods for PI and PID controllers were then analysed with help of the model to investigate the responses to reference changes and load disturbances. It is concluded that the easiest method to tune a controller is by using the Lambda method. However the responses to reference changes and load disturbance is slow. A PID controller can be used for improving the performance and by using a relay to determine the parameters of the controller, a stable and fast controller can be designed. Including a Smith predictor with the controller will reduce the problems with overshoot/undershoot at reference changes and load disturbance, but a good model is necessary to give good performance. The influence of measurement noise has different effects on different controllers and was studied in regard to control activity. This indicates that PID controllers often have problems with noise, especially PID controllers with Smith predictor. Therefore, filtering of the signal is an important factor to consider.

Keywords: Control, PID, Smith, Dymola, Time delay, Post combustion, Steam network

Acknowledgements

I would like to thank the company Solvina and their employers for the opportunity to do a master thesis with them. Special thanks go to my supervisor Peter Dahlström for all the help during the thesis and Carl Ressel for his feedback.

At Chalmers, I would like to thank my exterminator Torsten Wik at the department of Signals and Systems for his feedback and help.

Göteborg, 24 January, 2013

Henrik Granberg

Nomenclature

Roman upper case letters

A	Total heat transfer area [m^2]
Cp_i	Heat capacity for component $_i$ [$\mathrm{Jkg}^{-1}\mathrm{K}^{-1}$]
$C(s)$	Transfer function for controller
F_i	Mass flow for component $_i$ [kgs^{-1}]
HRSG	Heat recovery steam generator
K	Static gain of the process
K_p	Controller gain
K_i	Integral gain
K_{inf}	Gain for high frequencies
$L(s)$	Transfer function from error signal(e) to output signal(y)
M_s	Maximum Sensitivity function
M_t	Maximum complementary sensitivity function
$P(s)$	Transfer function for process
Q	Heat flow [W]
$S(s)$	Sensitivity function
T_i	Temperature for component $_i$ [K]
T	Time constant of the process s
$T(s)$	Complementary sensitivity function
T_i	Controller integral time constant [s]
T_d	Controller derivative time constant [s]
T_f	Controller derivative filter time constant [s]
U	Overall heat transfer coefficient [$\mathrm{Wm}^{-2}\mathrm{K}^{-1}$]
V	Volume [m^3]

Roman lower case letters

f	Index for flue gas
h_i	Enthalpy for component $_i$ [Jkg^{-1}]
in	Index for inlet
out	Index for outlet
w	Index for water

Greek Letters

ΔT_{lm}	Logarithmic mean temperature [K]
λ_f	The lambda factor for control using lambda tuning method
ρ	Density of meduim [$\mathrm{kg\,m}^3$]
τ_d	Time delay constant of the process s

Contents

Abstract **i**

Acknowledgements **ii**

Nomenclature **iii**

Contents **v**

1 Introduction **1**

1.1 Purpose and Aim . . . 1

1.2 Limitations . . . 1

1.3 Method and Literature Review . . . 2

1.4 Outline . . . 3

2 Utility Plant Description **4**

3 Theory **6**

3.1 Modelling . . . 6

3.1.1 Dymola and Modelica . . . 6

3.1.2 Solvers . . . 7

3.2 Process . . . 7

3.2.1 Mediums . . . 7

3.2.2 Combustion . . . 7

3.2.3 Heat Transfer . . . 8

3.2.4 Pressure drops . . . 9

3.2.5 Steam Tanks . . . 9

3.2.6 Turbines . . . 9

3.2.7 Grand Composite Curve . . . 10

3.3 Control . . . 10

3.3.1 Process Model . . . 10

3.3.2 PI(D) Controller . . . 11

3.3.3 Process In Marginally Stable Condition . . . 15

3.3.4 Cascaded Control . . . 15

3.3.5 Smith Prediction Control . . . 16

3.3.6 Anti Windup . . . 17

3.3.7 Noise . . . 18

4 Model **19**

4.1 Process Models . . . 19

4.1.1 Gas Turbine . . . 19

4.1.2 Post Combustion . . . 20

4.1.3 HRSG . . . 21

4.1.4 Steam Network . . . 22

4.1.5 Time Delays . . . 23

4.2 Control model . . . 24

4.2.1 Post Combustion . . . 24

4.2.2 HRSG . . . 24
4.2.3 Steam Network . . . 24

5 Results **25**
5.1 Process Model . . . 25
5.2 PI(D) Control Design Based on Transfer Function Model . . . 27
5.2.1 Model Margin PI Controller . . . 27
5.2.2 Model Margin PID controller . . . 28
5.2.3 Model Z-N PID Controller . . . 29
5.2.4 Model κ_{180} PID Controller . . . 29
5.2.5 Comparison of Model Design Strategy . . . 29
5.3 PI(D) Control Design Based on Experimental Data . . . 31
5.3.1 Exp-Data Lambda PI controller . . . 31
5.3.2 Exp-Data Z-N PID Controller . . . 32
5.3.3 Exp-Data κ_{180} PID Controller . . . 32
5.3.4 Comparison of Control Design Using Experimental Data . . . 32
5.4 Smith Prediction Control . . . 34
5.4.1 PI Controller with Smith Prediction . . . 34
5.4.2 PID Controller with Smith Prediction . . . 36
5.4.3 Comparison of Smith Controllers . . . 37
5.5 Response to Measurement Noise . . . 38
5.6 Overall Comparison of Controllers . . . 39

6 Discussion **40**
6.1 Dymola Model and Transfer Function Model . . . 40
6.2 Control Design . . . 40

7 Conclusions **42**
7.1 Further Work . . . 42

References **43**

Appendix A Dymola **45**
A.1 Dymola Interface . . . 45
A.2 Gas turbine . . . 47
A.3 Post Combustion . . . 48
A.4 HRSG . . . 49
A.5 Steam Network . . . 50
A.6 Complete Model . . . 51

Appendix B GCC **52**

1 Introduction

In today's society, energy conservation is an important subject because of both economic and environmental reasons. Fuel prices are increasing which makes companies review their energy demand to reduce costs. At the same time, harder legislations makes companies invest in equipment that is more efficient. Control optimisation is one area, where a small investment cost can result in reduced energy demand without any new equipment.

The engineering company Solvina has discovered problems in an existing utility plant for production of steam for a chemical process plant. During the years, the utility plant has been modified and upgraded its capabilities of producing steam in different stages. The last investment was in a gas turbine with post combustion capabilities. Their aim is to not produce more steam than necessary, and to achieve this the post combustion is automatically controlled. The current control configuration results in an unstable control of the post combustion and to solve this, an excess of steam is produced resulting in a higher fuel demand than necessary. An alternative solution is to reduce the response of the controller, which makes the post combustion respond slower on changed steam demand. This was not an option because it made the steam network too vulnerable to pressure drops.

The problem arises from long time constants and significant time delay, which make the process hard to control. PI and PID controllers are the most common controllers in process industries due to easy tuning. More advanced controllers are rarely used due to the difficulties to characterise the process with a model. However, frequently controlled parameters e.g. flow, pressure and temperature are often quite easy to describe with a simple transfer function. In electrical and mechanical processes an accurate model is more easily derived. The problems with time delays are in these areas often solved with help of Smith prediction, where a model predicts the correct value after the time delay and therefore reduces overshoot/undershoot that otherwise appear for PI and PID controllers. Another difference from fluid/thermal processes and electrical/mechanical processes is that fluid/thermal processes often has a larger error on the measured variable due to less accurate measurement equipment. The less accurate measurement equipment introduce a higher noise factor, which can make the process harder to control.

1.1 Purpose and Aim

The purpose of this thesis is to investigate how a fluid/thermal processes with large time constants and significant time delay can be controlled for optimal performance. To achieve this the purpose is divided into interim aims:

- Create a model of the existing utility plant with large time constants and significant time delay.
- Design different controllers for processes with large time constants and significant time delay and evaluate them for response, stability and measurement noise.
- Derive a control design strategy for fluid/thermal processes with large time constants and significant time delay.

1.2 Limitations

The gas turbine with post combustion capabilities is a part of a bigger utility system consisting of several boilers and turbines. The influence of these equipments on the gas turbine and the steam network is not considered in this thesis.

The purpose was not to study startup or shutdown behaviours in the utility plant's equipment, which had resulted in a different modelling and control approach.

The utility plant includes many different equipments and their controllers. Any faulty tuning of these controllers could potentially result in the observed unstable behaviour of the post combustion controller. However, due to lack of information of these equipment, no further study could be done to remove them as a possible problem.

In this thesis, all equipment is assumed to behave as expected and the considered problem is only the tuning of the post combustion controller.

Disturbances are often measurable, through information further downstream in the steam network. This often results in other design strategies, but in this project the disturbances are assumed to be non-measurable.

1.3 Method and Literature Review

To develop a model of the utility plant, a number of different software can be used, such as Dymola[1] and Simulink [2]. In this thesis the multi-engineering modelling and simulation platform Dymola is used. Dymola was chosen because of its abilities to handle large process simulations and its ''model" structure. The model structure makes it possible to reuse models in different projects. Equipment models had already been developed by the engineering company Solvina. These models have been modified to achieve similar performance as the utility plant described later in Section 2. The models are often based on simple theory about the equipment and include only a small amount of time dependent dynamics. This resulted in a model with small time constants and delays. To make a model with more time dependent dynamics a number of different dynamics were implemented, e.g. steam pipe delay, first order transfer functions and ''simple" time delays (see Chapter 4). At last a validation of the complete model was done to make sure that the model resembles the actual plant.

To determine possible controllers for the post combustion, a literature study was made. Literature suggested that for standard PI and PID controllers, the recommended tuning method is with help of stability margins [3]. Stability margins analyse the controllers performance based on how the controller respond to additional gain and phase shift in the process. The stability margin method is used in this thesis to tune PI and PID controllers with help of a first order transfer function model with time delay.

If no model of the process is available, the literature suggested to put the process in semi-unstable behaviour to determine the process characteristics [3, 4]. This can be done in many different ways. One easy method is to use a relay and thereafter use a standard tuning method [3, 4]. There are various standard tuning methods available. The Lambda method is a popular method for process industries, because of its simple procedure to determine the PI controllers parameters based on a small amount of measurement data. Zeigler-Nicholas method of tuning PID controllers is a method which often results in a fast controller, at the expense of a high signal activity. The κ_{180} tuning method is a more advanced tuning method, often used for its good response and stability [3]. In this thesis the characteristics of the Dymola model is determined through a relay experiment. The tuning methods above were then used to determine the parameters for PI and PID controllers.

More advanced controllers for processes with time delay is often based on the Smith prediction theory [5] where a process model is used to avoid overshoot/undershoot. This controller often assumes a constant time delay. If the process doesn't have a constant time delay the performance of the controller is reduced. Recent studies in Smith prediction indicate that the modelled time delay can be expressed as variable [6]. There are variants of Smith prediction controller, where both PI and PID controllers can be used. The Smith prediction controller is often implemented with a filter to increase the performance and/or reduce the influence of measurement noise [7, 5]. In this project the time delay was constant, which made it possible to not implement the time delay as a variable. However different Smith prediction controllers with both PI and PID controllers were evaluated. Two different filters were also evaluated to increase the performance.

To investigate the performance of the different controllers the response to load disturbance, reference changes and measurement noise were studied. These three performance evaluators were chosen for their abilities to summarise most of the problems in process control for fluid/thermal processes with large time constants and significant time delay.

At last the result from the evaluation of the controllers was used to derive a design procedure for control of fluid/thermal processes with long time constants and significant time delay.

1.4 Outline

This report is divided into 7 chapters. After the introduction, Chapter 2 provides the essential information available about the utility plant.

Chapter 3 explains how the software Dymola works for modelling of large processes, along with a description of the theory behind the different models used in this project. This thesis main objective is to design the post combustion controller. Therefore, an extensive theory section for control design ends this chapter.

The model development in Dymola is presented in Chapter 4. This include a description of the problems during the modelling part of this project along with the design strategy for the different equipments.

Chapter 5 presents the result of this thesis. The different controllers designed in this thesis are evaluated based on response, signal activity, stability and measurement noise.

Chapters 6 discuss the results and the modelling part of this thesis to develop a design strategy for processes with large time constants and significant time delay.

At last, Chapter 7 explains the controller design conclusions along with suggested further investigations regarding this project.

2 Utility Plant Description

The utility plant consists of a gas turbine with post combustion capability, a heat recovery steam generator (HRSG) and a steam network with turbines. Because of efficiency reasons the gas turbine runs at 100 % of its capacity all the time. The post combustion unit controls the steam demand, where maximum post combustion increase the steam production by 50 %. In the heat recovery steam generator the flue gas is heat exchanged with help of an economiser, an evaporator and a superheater. The heat recovery steam generator can handle a high temperature difference in the flue gas temperature to produce different amount of high-pressure steam.

The steam network consists of a long pipe network before the high-pressure steam expands through a two-stage turbine. The steam turbine has the capability to extract intermediate pressure steam. After the first turbine, the steam can be used as low-pressure steam or be expanded through a second steam turbine. The outgoing flow from the second turbine is of no use in the process and is therefore condensed. The steam network has bypass lines to ensure correct pressure in the steam lines. The need for intermediate pressure steam is around $15 - 20$ % and the demand for low pressure steam is $65 - 80$ %.

A schematic view of the utility plant is illustrated in Figure 2.0.1.The combustion part of the plant is in red, including the gas turbine, post combustion unit and the HRSG. Blue represents the steam network, illustrating the two steam turbines and the valves controlling the flow in the plant. In the plant, there are a number of different controllers that control both the steam network and the different equipments. The controller studied in this thesis is the one controlling the post combustion with help of measurements from the outgoing flow from the second turbine. This controller loop is illustrated in green. The other controllers are used mainly to ensure correct pressures in the steam network and is illustrated in brown. The figure also displays the maximum/minimum flow rates of steam and the set points of pressure at different locations.

The desired operational mode is to use the first steam turbine as much as possible and use the second one as little as possible, to produce as much steam as possible with lowest amount of fuel. Detailed information about mass flow rates, temperatures, pressures and efficiencies for the different equipment are presented in Table 2.0.1.

Table 2.0.1: *Performance data for gas turbine, post Combustion, HRSG, turbines and steam network.*

Gas turbine	Mass flow	$130\,\mathrm{kg\,s^{-1}}$
	Exhaust temperature	540 °C
Post combustion	Max exhaust temperature	700 °C
HRSG	Maximum mass flow	$24\,\mathrm{kg\,s^{-1}}$
	Minimum mass flow	$16\,\mathrm{kg\,s^{-1}}$
	Pressure	64 bar
	Temperature	525 °C
Turbines	Inlet Pressure	63 bar
	Intermediate pressure	34 bar
	Low pressure	8.6 bar
	Condensate pressure	0.4 bar
	Maximum mass flow stage 1 and 2 for 1^{st} turbine	$24\,\mathrm{kg\,s^{-1}}$
	Maximum mass flow for 2^{nd} turbine	$11\,\mathrm{kg\,s^{-1}}$
	Minimum mass flow for 2^{nd} turbine	$2\,\mathrm{kg\,s^{-1}}$
	Isotropic efficiency	75 %
Steam network	High pressure	62 bar
	Intermediate pressure	12 bar
	Low pressure	6 bar
	Condensate pressure	0.4 bar

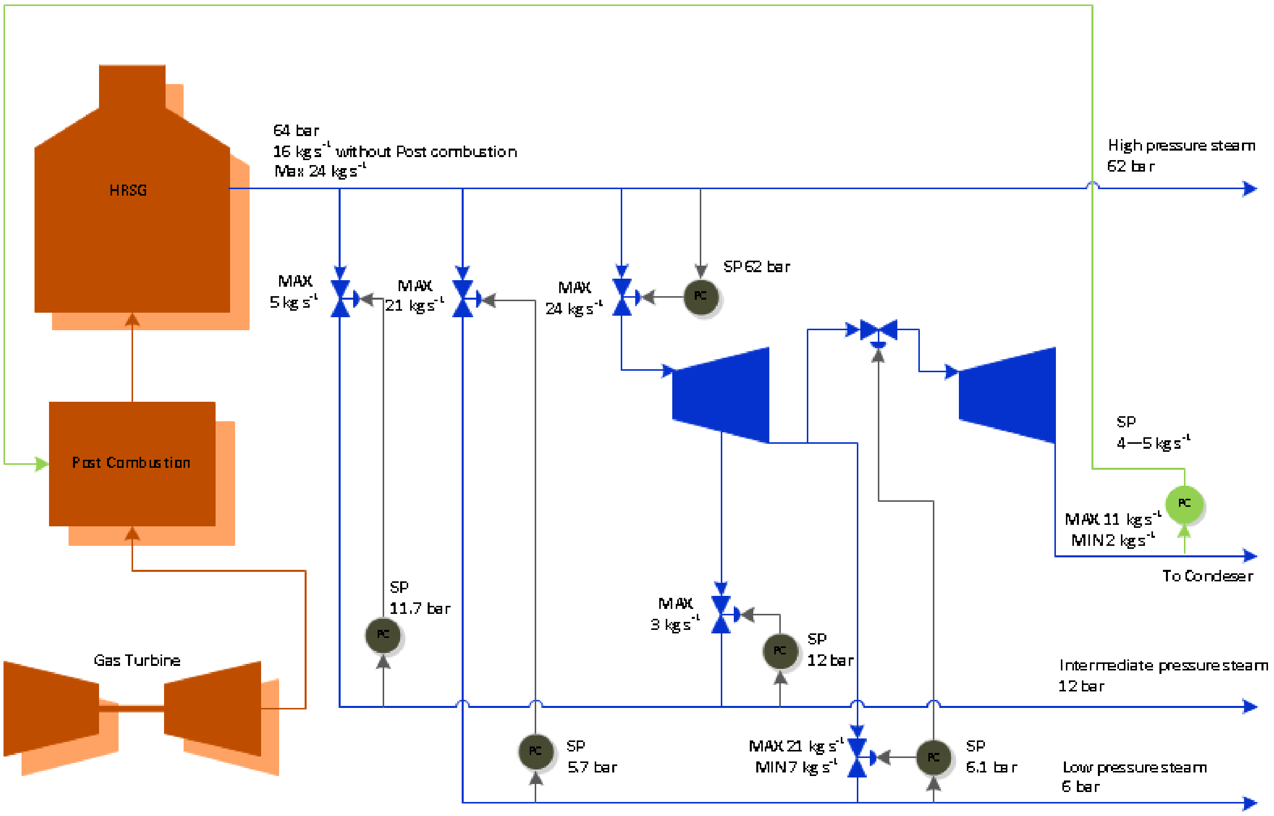


Figure 2.0.1: *Schematic overview of the utility plant. In red are the gas turbine with post combustion capabilities and the HRSG, in blue is the steam network and in brown/green the control of the utility plant. Details of maximum/minimum flow rates and pressure set points is seen next to the equipment. The controller studied in this thesis is the one highlighted in green, where it controls the post combustion through flow measurements from the second turbine.*

3 Theory

The theory behind this thesis includes both mathematical models of process equipment and theory from process control. The solver also uses advanced algorithms to solve the differential equations. Because of the extensive theory behind the process modelling and the solver algorithms, which contain a high number of equations, only a brief description of the theory is described in this report. The control theory is more notoriously described to achieve a better understanding for control of processes with large time constants and significant time delay.

3.1 Modelling

To model complex processes a number of different approaches can be used with different resolutions and costs. The choice depends on what properties the model should describe. For controller design and tuning, the most common way is to use a transfer function or a state space model to describe the process. To derive an exact transfer function, the physics behind the process is required. The time dependent equations are then Laplace transformed to describe the output signal as a result of an incoming signal. If the equations have non-linear behaviour the equations needs to be linearised around an operating point. A transfer function is needed for each of the parameters in the equipment, which can result in a very large model [3].

Another method is to stay in the state space environment and numerically solve the equations in each time step. This method makes it easier to model non-linear relations. The resolution depends on how much dynamics the equations are describing, e.g. can flow through a pipe be modelled as the flow going into the pipe, is also going out at the same time. At steady state and over a large time interval this assumption is correct. However, in a short time interval this is not correct due to time delays for the flow traveling from the inlet to the outlet. The controllers performance depend on the transient solution and not on the steady state solution so for control purposes the resolution is of great importance. To tune controllers in time space environment one method is to produce a step response, and from the response create a simple transfer function model of the process [3, 1].

3.1.1 Dymola and Modelica

Dymola is suitable for modelling of complex physical systems. Its main advantage is the model approach where components can be reused in different projects in an easy way. Therefore, a large community can develop models and share them with each other to reduce the cost of making new models each time. Dymola has a graphical interface that reduces the learning time along with a lucid presentation of the model (see Appendix A) [1].

Dymola includes a standard library of properties for different species in both liquid and gaseous form. The model can then request the properties required, e.g. enthalpy, temperature, pressure and density. To write equations the language Modelica is used. Other good properties is that it can handle real-time simulations and can be used with a number of different programs, such as Simulink [1].

Dymola uses Modelica, which is an object oriented and equation based modelling language. Modelica reduces the cost of rewriting equations substantially, because of its abilities to manage equations written on implicit form. The equations can be written in any order, Modelica then decides in which order the variables shall be solved [1].

3.1.2 Solvers

Dymola has a number of different solver algorithms to choose from, divided into two categories:

- Single step algorithms are algorithms that starts all over on every step. This results in lower cost for restarting the solver after an event. Single step algorithms should be used when there are many events occurring during simulations [1].
- Multi step algorithms base the next step on previous results. This results in a higher cost for restarting after an event, but reduces the simulation time. Therefore, multi step solvers should be used when there are not so many events [1].

Dassl is a multi-step algorithm that uses previous result to estimate the new result, which makes the iteration fast. It can handle variable step size and use error tolerances to decrease time steps if necessary. One disadvantage is that the solver needs a very close guess to not diverge at the initialisation step of the simulation [1]. This solving algorithm is the one used in this project.

The most favourable start position is in a steady state. To achieve steady state the initial value needs to be in balance with each others to avoid time derivatives of the parameters. If the initial values is calculated incorrectly and there exist strong non-linear behaviour there is a chance of a diverged initialisation [1].

3.2 Process

The process theory describes the basic theory used in the Dymola model including purpose, requirements and limitations. It will not give the complete information of the theory behind the models, but should give enough to get an understanding of the modelling process. When the equations result in dynamics or are of great importance for understanding, the theory is described in greater detail.

3.2.1 Mediums

For calculations with water as medium, Dymola use the IF97 standard developed by International Association of the Properties of Water and Steam to determine the properties for water. The library can handle liquid, steam and phase changes. The properties for water are calculated with help of the law of Gibbs free energy, and are valid in the temperature region $0\ ^{\circ}\mathrm{C} < T < 800\ ^{\circ}\mathrm{C}$ and pressure region $p < 1000\,\mathrm{bar}$. To determine all thermal properties of the water the library require only two thermal properties [8].

The flue gas package in Dymola can handle common flue gas components and air. This includes oxygen, water, nitrogen, carbon monoxide, nitrogen oxide, sulfur dioxide, sulphuric acid and carbon dioxide. It is based on ideal gas mixture theory and valid in the temperature range of: $-73\ ^{\circ}\mathrm{C} < T < 5000\ ^{\circ}\mathrm{C}$. The data used to calculate the thermodynamic properties comes from a NASA report [9]. The method to determine the mixed gas properties require the composition and two thermal properties [1]. This give an overall "Mixed gas" behaviour that is based on composition, pressure and temperature.

3.2.2 Combustion

In the utility plant, the combustion natural gas is performed in both a gas turbine and in a post combustion unit. Natural gas can often be approximated to consist of 90 % methane, 8 % ethane and 2 % hydrogen sulfide [10]. There are many different reaction expressions derived to model the combustion of methane. The one used in this thesis was developed by D. G. Nicol et al [11], where the theory can predict the reaction of methane to carbon oxide, carbon dioxide, water, nitrogen and nitrogen oxide. The reaction rates are only valid for a pressure of 1 bar, an inlet temperature of 650 K and an excess air/fuel ratio of $1.4 - 2.2$. The flue gas package cannot handle methane, ethane and hydrogen sulfide. Therefore, methane and ethane are assumed to react instantaneously to carbon oxide and water. The hydrogen sulfide is also assumed to react instantaneously to sulfur dioxide. These assumptions are probably not valid with the reaction rate expression derived by D. G. Nicol et al. [11] due to composition problems of the flue gas described later in Section 4.1.1 where this theory predicts an unrealistically high content of carbon oxide.

The required input to calculate the reaction of natural gas is fuel input and excess air/fuel ratio along with the composition and thermal properties of the gas it will burn. The volume of the combustion chamber is also necessary.

3.2.3 Heat Transfer

Transfer of heat from flue gas to water is usually performed in a heat recovery steam generator. There are many different designs of a HRSG but in general, they consist of a superheater, evaporator and an economiser. A schematic view of a HRSG is illustrated in Figure 3.2.1. Feed water is pressurised and heated through an economiser to a temperature below the saturated temperature. To reach saturated condition, the water is mixed in a steam drum with saturated steam. The saturated water in the drum is evaporated to saturated steam in an evaporator. The saturated steam in the steam drum is at last superheated to the desired temperature [12].

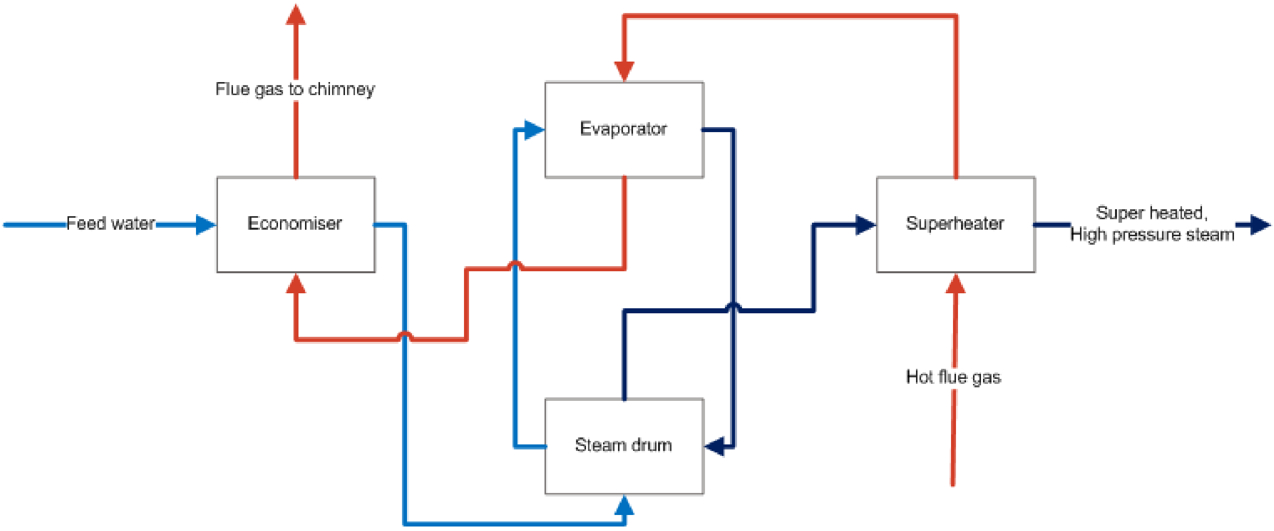


Figure 3.2.1: *The basic of heat exchaning between flue gas and water in a HRSG.*

Transfer of heat from flue gas to water in the HRSG is performed in separate steps due to the different properties of water at different states. The heat transfer theory is based on an overall heat transfer coefficient U. To solve the heat transfer Q between the flue gas and the water, a heat transfer equation is required;

$$Q = UA\Delta T_{lm}, \tag{3.2.1}$$

together with a heat balance on the water side,

$$Q = F_w(h_{w,out} - h_{w,in}) = F_w Cp_w(T_{w,out} - T_{w,in}), \tag{3.2.2}$$

and a heat balance on the flue gas side,

$$Q = F_g(h_{g,out} - h_{g,in}) = F_g Cp_g(T_{g,out} - T_{g,in}). \tag{3.2.3}$$

The logarithmic mean temperature is defined as:

$$\Delta T_{lm} = \frac{\Delta T_{Hot} - \Delta T_{Cold}}{\ln \frac{\Delta T_{Hot}}{\Delta T_{Cold}}},$$

where:

$$\Delta T_{Hot} = T_{g,in} - T_{w,in}$$

$$\Delta T_{Cold} = T_{g,out} - T_{w,out}$$

If the heat transfer is occurring in one phase the heat transfer coefficient can be approximated as constant due to the similar thermal properties of water. To determine the sizing of the equipment, in this case the area A, the following parameters needs to be decided: mass flows, temperatures, pressures, volumes, heat capacities and heat transfer coefficient at the desired operating point. To increase/decrease the heat transfer the temperature of the flue gas can be changed [12].

3.2.4 Pressure drops

Pressure drops in the process model is modelled with help of valves. The theory for pressure losses in steam valves is based on theory from Fisher [13]. The pressure drops for different valves are calculated through valve specific properties and information about the steam. The most important factors are:

- The ratio of specific heats: If steam 1.33 can be assumed.
- The piping geometry factor: If no knowledge of the pipe and valve diameter it can be estimated to 1.
- The rated pressure drop factor: If valve type not known it can be estimated to 0.5.
- The inlet design pressure.
- The design pressure drop.
- The design mass flow.

This theory can predict how the pressure drop and mass flow changes for different positions of the valve. The limitations with Fishers valve theory are that it cannot handle liquids or two-phase mediums. It is derived for compressible medium such as steam, but similar theory with some modification can be used also for liquids.

3.2.5 Steam Tanks

Tanks are used to model the transient behaviour in the process. These tanks implement a volume in pipes, heat exchangers and steam turbines. With help of the volume of the tanks and knowledge of the density and the internal energy of the steam, it simulates a transient behaviour of changing mass flow. The most important equations are:

The transient mass balance:

$$\frac{\mathrm{d}(\rho_{out}V)}{\mathrm{d}t} = F_{in} - F_{out} \tag{3.2.4}$$

The transient heat balance:

$$\frac{\mathrm{d}(h_{out}\rho_{out}V)}{\mathrm{d}t} = Q_{in} - Q_{out} \tag{3.2.5}$$

3.2.6 Turbines

The model for steam turbines are based on theory developed by Stodola [14]. The required inputs are the isotropic efficiency and the Stodola machine constant. Stodolas equation for part loads assumes constant density in the turbine, which can be correct if the pressure drop in the turbine is low. The theory does not compensate for changes in isotropic efficiency during part load operations.

3.2.7 Grand Composite Curve

A grand composite curve (GCC) is a tool to estimate the heat transfer and required temperatures for heat transfer between flue gas and water. The mass flow of both the flue gas and the water are required along with the enthalpies of water at different states. The flue gas is assumed to have a constant heat capacity of $1.05\,\mathrm{kJ\,kg^{-1}K^{-1}}$. The GCC is then constructed with help of heat balances. For more details about the construction of a GCC, see Appendix B.

3.3 Control

The theory necessary for control of processes with large time constants and significant time delay is presented in this section. In Figure 3.3.1 a basic presentation of how control of a process is done with help of feedback. The controller $C(s)$ controls the process $P(s)$. The output y from the process is measured and feedback to the controller. There are other methods to control processes but this is the most common structure used in process industry [3].

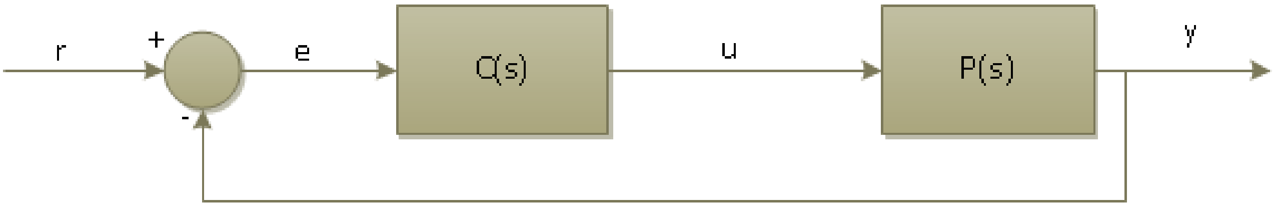


Figure 3.3.1: *Flowchart of control with help of feedback: r is the desired value of y, e is the error from y to r, C(s) is the controller, u is the control signal, P(s) is the process to control and y is the measured value from the process.*

For discrete controllers the sampling interval is decided trough the timescale of the process, but in general, discrete controller with a sampling interval faster than 200 ms can be assumed to have the same properties as a continuous controller. This can be assumed for most of the controllers available today [15]. If not, the design of the controller require information about the discrete behaviour of the controller and the process.

3.3.1 Process Model

For a simple process with real poles the process often has one dominating pole. When this occur a transfer function model can then often be predicted using experimental data to determine a first order model with time delay [3]:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{1+sT}e^{-s\tau_d} \tag{3.3.1}$$

The constants can be determined with help of a step response experiment around the operating point. The process gain K is determined from the change in y divided by the change of u. The time constant T is the time the process take to reach 63 % of its final y value minus the time delay. The time delay τ_d is the time it takes for the process to respond to an input step change [3]. The time delay is expressed as:

$$P_d(s) = e^{-s\tau_d}$$

Because of the exponential factor, numerical problems sometimes occurs. To make a polynomial expression an infinite number of poles is required to describe the time delay. One method is to use the Padé approximation [3]:

$$P_d(s) = e^{-sT_d} = \frac{1+\beta_1 s + ... + \beta_n s^n}{1+\alpha_1 s + ... + \alpha_n s^n} \tag{3.3.2}$$

For $n = 2$ the Padé approximation gives the following time delay transfer function [3]:

$$P_d(s) = \frac{12 - 6s\tau_d + (s\tau_d)^2}{12 + 6s\tau_d + (s\tau_d)^2}$$

The $n = 2$ results in a transfer function where the real phase of the time delay deviate a maximum of $1\,°$ down at phase of $-100\,°$. The expansion criterion is decided by analysing the total transfer function for the process. It is of importance to have a good approximation until the total phase shift of the controller together with the process is $-180\,°$, but in general most processes can be approximated with a second order approximation [3].

When time analysis is performed, the Padé approximation is not enough due to a jump to 1 (second order approximation) at a step response along with oscillating behaviour during the time delay. To decrease these problems, further dynamics in the approximation is included in form of a time constant of the same magnitude as the time delay [3]:

$$P_c(s) = \frac{1}{1 + s\tau_d} \tag{3.3.3}$$

This also makes the approximation more exact at higher frequencies and if the process model only has one pole, overshoot/undershoot would newer occur theoretically [3]. Therefore, the extra pole from the time delay increases the dynamics in the model.

The final transfer function model used to describe a process with large time constants and significant time delay is on the form:

$$P(s) = \frac{K}{sT + 1} \cdot \frac{1}{s\tau_d + 1} \cdot \frac{12 - 6s\tau_d + (s\tau_d)^2}{12 + 6s\tau_d + (s\tau_d)^2} \tag{3.3.4}$$

3.3.2 PI(D) Controller

The most common controller in process industry is the PI(D) controller. The different parts of the controller have different purposes:

- The P-part is the proportional part of the controller also called gain. Increasing the P-part results in a faster system response and increased compensation of process disturbances but to the cost of increased control activity and reduced stability margins [3].
- The I-part is the integral effect of the controller. Increasing the I-part results in a faster response at low frequencies along with decreased stability margins. It is necessary to use integral action to prevent that static errors remain [3].
- The D-part gives the derivative effect of the controller. Increasing the D-part may result in better stability margins and faster reduction of overshoots/undershoots, but it also results in higher high frequency gain along with increased control activity. If measurement noise is present in the process, the D-part can make the control signal jumpy and the D-part is therefore often filtered [3].

The theoretical continuous PID controller can be written as:

$$u(t) = \underbrace{K_p e(t)}_{\text{P}} + \underbrace{\frac{K_p}{T_i}\int^t e(s)ds}_{\text{I}} + \underbrace{K_p T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}}_{\text{D}} \tag{3.3.5}$$

where $u(t)$ is the control signal out from the controller, $e(t)$ is the error between reference and measurement, K_p is the gain constant, T_i is the integral time constant and T_d is the derivative time constant [15].

The PID controller is usually implemented in discrete time. If the sampling interval is low enough it can still be designed in continuous time, because of the simpler analysis of the controller [15]. This results in the PID controller in continuous parallel form with filter on the derivative part to reduce impacts from measurement noise [3]:

$$C_{PID}(s) = \frac{u(s)}{e(s)} = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_f s}\right) \tag{3.3.6}$$

where T_f is the derivative filter time. For a block presentation describing the different parts of the PID controller, see Figure 3.3.2.

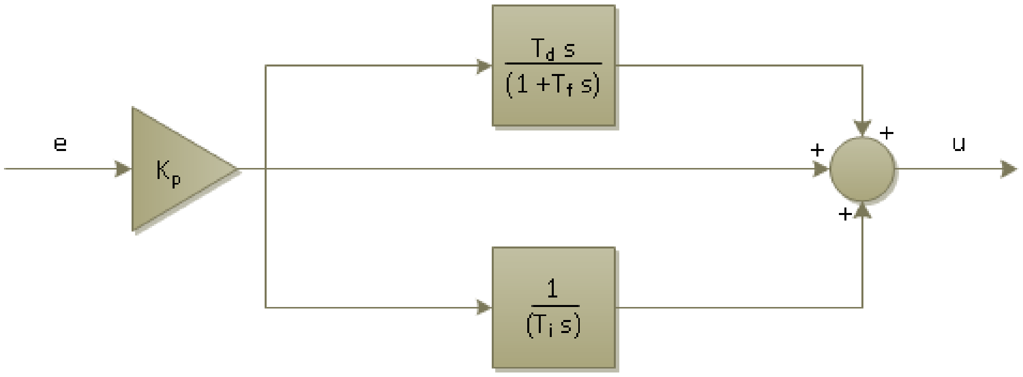


Figure 3.3.2: *Block presentation of a PID controller: e is the error between reference and measurement and u is the control signal.*

Stability margins

To determine the constants in the PID controller there are a number of different ways. One is to study the margin of the closed loop stability. For simplifications the controller along with the stable process is defined as the open loop transfer function

$$L(s) = C(s)P(s). \tag{3.3.7}$$

The closed loop transfer function from r to y is

$$P_{ry}(s) = \frac{L(s)}{1 + L(s)}. \tag{3.3.8}$$

For processes, the stability margins can be evaluated through Nyquist criteria which is illustrated in Figure 3.3.3 Its created through plotting $L(j\omega)$ for $\omega = 0$ to $\omega = \infty$. From this figure the margins of the closed loop transfer function can be evaluated.

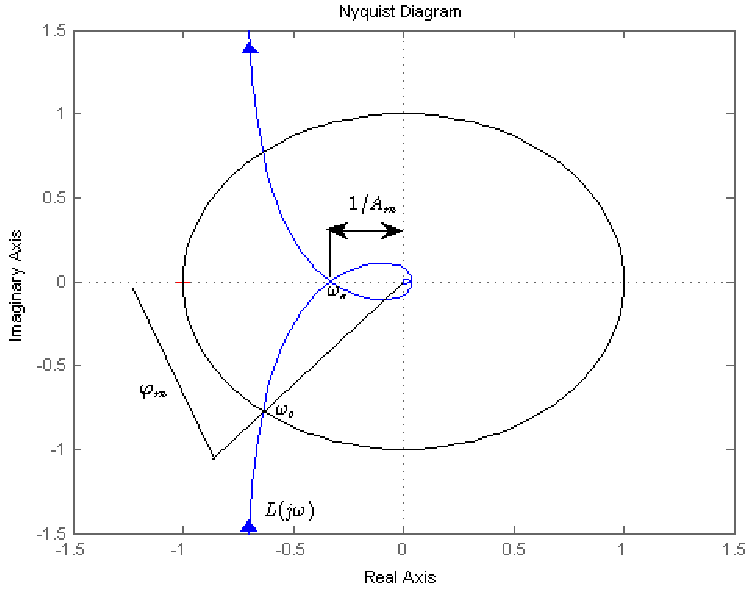


Figure 3.3.3: *Nyquist plot visualising the margins of the closed loop transfer function.*

The amplitude margin A_m and phase margin φ are measurements of how much additional gain and phase the system can handle before it becomes unstable. Amplitude margin and phase margin only ensure that the transfer function is safe from -1 at ω_c and ω_π. Therefore, some additional criteria is required if the system approach -1 at some other frequency. This can be studied using the sensitivity function $S(s)$ and the complementary sensitivity function $T(s)$ [3].

The sensitivity function is defined as

$$S(s) = \frac{1}{1 + L(s)}. \tag{3.3.9}$$

The maximum of $S(s)$ for all frequencies, i.e.

$$M_S = \max_{\omega} |S(j\omega)|, \tag{3.3.10}$$

is a measurement of how far $L(j\omega)$ is from the point $(-1, 0)$ at minimum. If $M_S = 1.7$ an amplitude margin of at least 2.4(7.6 dB) is achieved for all frequencies. The relationship between A_m and M_S is [3]:

$$A_m \geq \frac{M_S}{M_S - 1} \tag{3.3.11}$$

The complementary sensitivity function is a function describing the output behavior after a change in reference:

$$T(s) = 1 - S(s) \tag{3.3.12}$$

The maximum of $T(s)$ for all frequencies, i.e.

$$M_T = \max_{\omega} |T(j\omega)| \tag{3.3.13}$$

is a measurement of the maximum response peak for the closed loop system. The relationship between φ_m and M_T is:

$$\varphi_m \geq 2 \arcsin \frac{1}{2M_T} \tag{3.3.14}$$

If $M_T = 1.3$ a phase margin of at least 45 ° is achieved [3].

$K_i = K_p/T_i$ is the so called integral gain. The higher it is, the faster the system responds to low frequency disturbances.

$K_\infty = |C(\infty)|$ is a measurement how the signal activity behaves for high frequency disturbances. A higher value gives higher gain for measurement disturbances. Therefore, a balance of K_∞ and K_i needs to be decided to achieve a proper response on reference changes, load disturbances and measurement noise [3].

$\beta \propto K_\infty/K_i$ is a measurement of the control activity [3].

Bode diagram is a frequency plot for a transfer function; it is a lucid way to study the phase and amplitude margin for the process. It shows graphically how the transfer function behaves for different frequencies and thereby the stability of the system [3].

Lambda tuning method

The Lambda method is a method used frequently in process control. It is a fast design method that requires small amount of data. It needs a step response experiment to determine the characteristics of the process. The time constant from the step response is then used for cancellation of the slowest pole with help of the controllers integral part, therefore $T_i = T$. The static gain is calculated using [16]:

$$K_p = \frac{T}{K(\lambda + L)}$$

λ is then chosen as a factor of the highest constant:

$$\lambda = \lambda_f T$$

Basic guideline is that, $\lambda_f = 1$ results in fast control of the process and $\lambda_f = 3$ results in slow control of the process. The controller is then tuned by trial and error of the parameter λ_f. The only control parameter that changes with changed λ_f is the parameter K_p [16].

Ziegler-Nichols tuning method

There are a number of different dimension strategies for PID controllers. One of them is the Ziegler-Nichols method, developed almost a century ago. Ziegler-Nichols method is still used to determine parameters for PID controllers. It requires knowledge of the frequency ω_{P180} where the process has a phase of 180 ° and the corresponding gain $|P(j\omega_{P180})|$ at this frequency. ω_{P180} and $|P(j\omega_{P180})|$ can be determined by increasing a proportional gain K_p until the process become marginally stable and starts to oscillate with constant amplitude (see Section 3.3.3). The parameters can then be calculated from [3]:

$$|P(j\omega_{P180})| = \frac{1}{K_{po}} \text{ and } \omega_{P180} = \frac{2\pi}{T_o}$$

where K_{po} is the gain where the process become marginally stable and T_o is the oscillation period.

The parameters for the PID controller are then determined by:

$$K_p = \frac{0.6}{|P(j\omega_{P180})|}\ , T_i = \frac{\pi}{\omega_{P180}},\ T_d = \frac{T_i}{4} \text{ and } T_f = \frac{T_d}{10}$$

T_f is usually quite small with this tuning method and results in a high high-frequency gain and high control activity [3].

κ_{180} tuning method

A more recent method is the κ_{180} method. As the Ziegler-Nichols method it is based a on ω_{P180} and $|P(j\omega_{P180})|$, but also uses the low frequency gain $|P(0)|$. It has been verified for a number of different processes and usually the stability margins M_T and M_S deviate less than 10 % from the stability margins method with a $M_T \leq 1.3$ and $M_S \leq 1.7$. At the same time the gain is maximised to an extent of 5 % from the optimal value [3].

The parameters is found through:

$$\begin{aligned}
K_i &= \omega_{P180}|P(0)|^{-1}(0.13 - 0.16\kappa_{180}^{-1} - 0.007\kappa_{180}^{-2}) \\
\tau &= ((0.4 + 0.75\kappa_{180})\omega_{P180})^{-1} \\
\zeta &= 0.75 \\
K_\infty &= (4 + \kappa_{180}^{-1})P(0)^{-1} \\
\beta &= K_\infty(\tau K_i)^{-1} \\
T_f &= \tau\beta^{-1} \\
T_i &= 2\zeta\tau - T_f \\
T_d &= \tau^2 T_i^{-1} - T_f \\
K_p &= K_i T_i
\end{aligned}$$

3.3.3 Process In Marginally Stable Condition

To determine the frequency ω_{P180} and the corresponding gain $|P(\omega_{P180})|$ at this frequency, the process is required to be in a marginally stable condition. This can be achieved by increasing the gain of a P controller until the process starts too oscillate [3]:

$$K_p|P(\omega_{P180})| = -1$$

This is often hard to achieve for processes with slow dynamics therefore a relay can be used instead of increasing the gain. The relay creates a self oscillation for all processes that at some point have a phase shift of $-180°$ or more, e.g. time delay processes [3].

The relay creates a square wave to the process. If analysed in frequency space the Fourier series results in a sinus wave. If higher order harmonies are assumed to be filtered away by the process, we get the following time response:

$$y(t) = a\sin(\omega_{P180}t)$$

The relay is configured with an amplitude d around the operating point to achieve the desired amplitude of the process, a. Then a self oscillation will occur for the process with a period time T_0, and we get:

$$\omega_{P180} = \frac{2\pi}{T_0}$$

and

$$|P(j\omega_{P180})| = \frac{\pi a}{4d}$$

With a step response experiment the static gain at the operating point is estimated from:

$$P(0) = \frac{\Delta y}{\Delta u}$$

and κ_{180} is then calculated from:

$$\kappa_{180} = \frac{|P(j\omega_{P180})|}{P(0)}$$

The PID parameters can then be estimated using the κ_{180} tuning method or Z-N tuning method. This is only valid for processes with a $\kappa_{180} \geq 0.1$. When measurement noise is present there can be problems at the sign change and therefore is the relay often implemented with a small hysteresis to overcome this problem [3, 4].

3.3.4 Cascaded Control

Consist of two or more cascaded systems with the possibility to measure between them, it is usually a good idea to consider cascaded control. An inner feedback loop is then applied over the faster subsystem to improve the response to disturbances. In Figure 3.3.4 a scheme of the control is illustrated. If a load disturbance occurs in the inner loop, the inner controller reacts and compensates for this disturbance much faster than the outer loop should have done. Cascaded control requires more measurement points, but load disturbances are compensated for much faster [3].

A variant of cascaded control, often used in process plants, is the three point boiler controller. In a boiler with steam generation the dome level is controlled. But if the steam demand increase rapidly or the feed water looses pressure it can result in problems. The faster subsystem measures the change of feed water into the boiler and steam out of the boiler and compensate rapidly to changes. The dome level is used to compensate for measurement errors due to flow measurements and to have the drum at safe water/steam level.

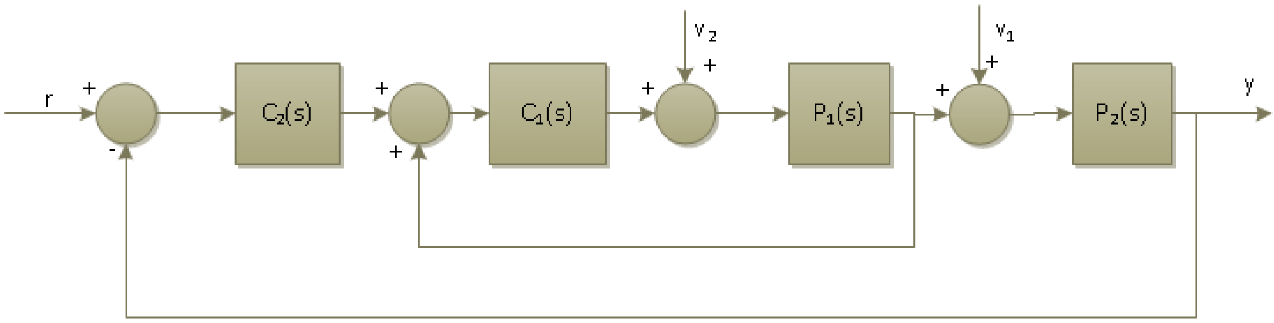


Figure 3.3.4: *Flowchart of cascaded process control. If the inner control loop is tuned faster than the outer control loop the response to load disturbances can improve significant.*

3.3.5 Smith Prediction Control

Controllers with Smith predictor compensate for time delays in the process. Theoretically if tuned perfectly, it removes overshoot/undershoot in the response to reference changes and load disturbances due to time delay. If there is a time delay in the process, the standard PI(D) controller will use outdated data from the feedback loop during the time delay. If a model instead predict where the process should be if there was no time delay, the controller can control the process without overshoot/undershoot. A schematic overview of a controller with Smith predictor is illustrated in Figure 3.3.5 [5].

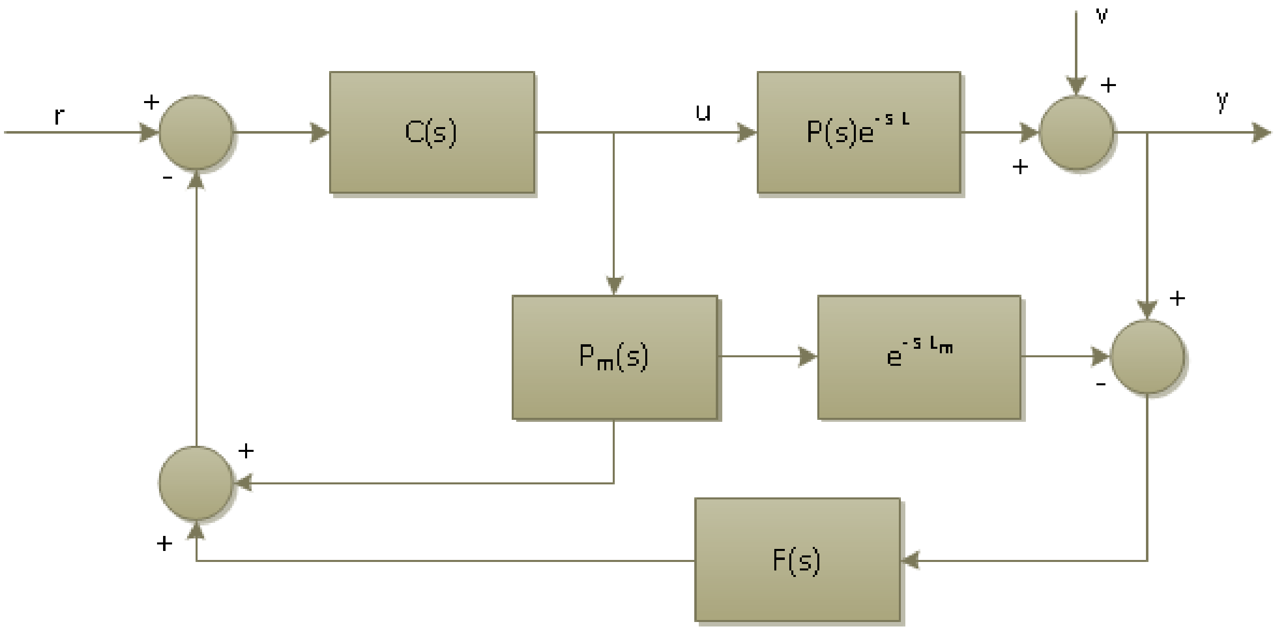


Figure 3.3.5: *Schematics overview of a controller with Smith predictor: $C(s)$ is a PI(D) controller, $P(s)e^{-sL}$ is the real process, $P_m(s)$ is the process model without time delay, e^{-sL_m} is the modelled time delay and L_m is the model time delay constant. $F(s)$ is a filter to increase stability or/and response.*

Expressing $y(s)$ in form of $r(s)$ and $v(s)$ for a perfect model ($P(s) = P_m(s)$ and $L = L_m$) results in:

$$y(s) = \frac{C(s)P(s)e^{-sL}}{1 + C(s)P(s)} r(s) + \frac{1 + C(s)P(s)(1 - F(s)e^{-sL})}{1 + C(s)P(s)} v(s) \tag{3.3.15}$$

The filter $F(s)$ will only have effect on measurement noise/load disturbances $v(s)$. If $F(s) = e^{s\tau}$ the response rejection is optimal. This can not be implemented due to its positive time delay, but an approximation can be done with help of a phase lead approximation [7]:

$$e^{\tau s} \approx \frac{1 + B(s)}{1 + B(s)e^{-\tau s}}, \tag{3.3.16}$$

where τ is the model time delay and $B(s)$ is a low pass filter

$$B(s) = \frac{K_{lp}}{1 + T_{lp}s} \tag{3.3.17}$$

Other filters can also be implemented to improve stability and disturbance compensation.

First order filter:

$$F(S) = \frac{K_F}{1 + T_F s} \tag{3.3.18}$$

Filter for better rejection of low frequency disturbances

$$F = \frac{1 + T_{F1}s}{1 + T_{F2}s} \tag{3.3.19}$$

The controller is designed from u from y to have good margins, where the loop depends on many different components:

$$u(s) = \frac{C(s)}{1 - F(s)C(s)P(s)e^{-sL} + C(s)P(s)}r(s) + \frac{F(s)C(s)}{1 - F(s)C(s)P(s)e^{-sL} + C(s)P(s)}y(s) \tag{3.3.20}$$

3.3.6 Anti Windup

When processes have limitations on the control signal, windup can occur, e.g. a valve cannot be open more than completely and less than closed. If the controller calculates a control signal larger than the maximum value the integral part will accumulate an error. To solve this, a inner feedback can be introduced in the controller to reduce the effects on windup. A PID controller with anti windup is seen in Figure 3.3.6 [15].

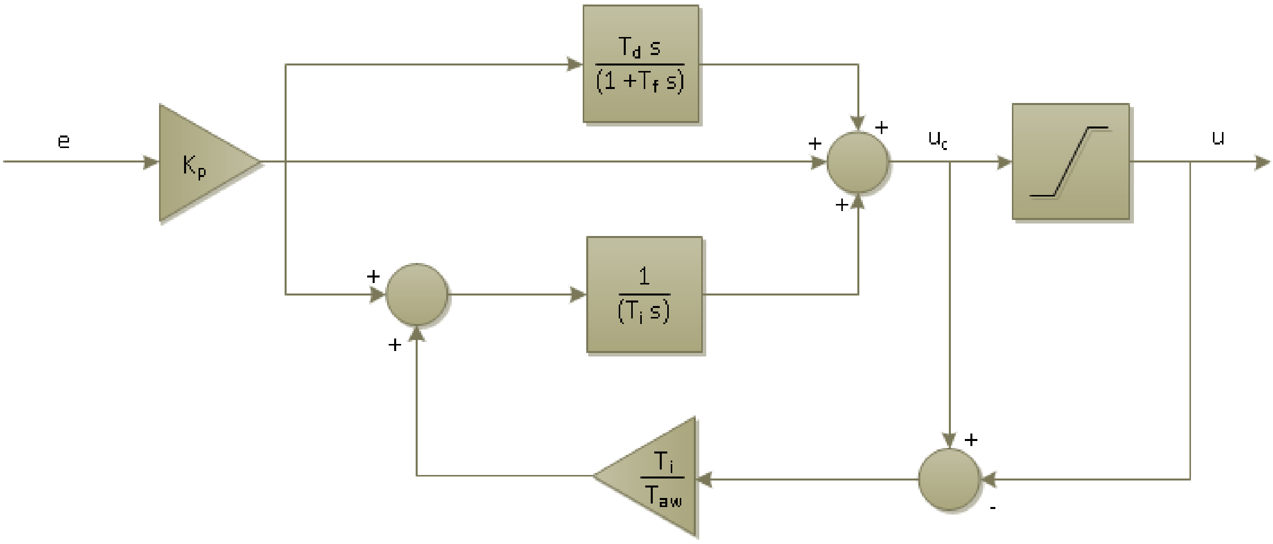


Figure 3.3.6: *Flowchart of a PID controller with anti-windup: e is the error and u is the control signal. The saturation-block enforce constraints on maximum and minimum control signal. If there is a difference between* u_c *and u the anti- windup compensates the integral action to not accumulate an error, and thereby give faster response when changes are made in the process.*

The windup constant T_{aw} needs to be determined. There is no general design procedure but the value of T_{aw} depends on how fast the integral action shall be reset versus the measurement noise. Optimal is to have a low value on T_{aw}, this result in fast windup compensation, but measurement noise can accidentally reset the integral part [15].

3.3.7 Noise

If the noise from the measurements can be approximated as random, it can be modeled as white noise. The main specifications are then the mean vector and the covariance matrix [15, 17]:

$$\mu = \mathbb{E}[\mathbf{w}] = 0$$

$$\mathbf{R}_{ww} = \mathbb{E}[\mathbf{w}\mathbf{w}^T] = \sigma^2\mathbf{I}$$

This can be implemented in discrete time through two uniform number generators that will create two sets of random numbers with help of the modulus function. A normal number generator then combines these sets to a continuous signal and normalises the signal to a zero mean and a variance of σ^2 [17].

4 Model

The model of the utility plant was developed with help of Dymola, where the modelling was divided into several process models. In these models, smaller parts were modelled separately, to be put together at the end as a working model over the utility plant. In this chapter the derivation and validation of the models are described. Solvina has developed most of the models, but some modifications were required to make them correspond to the actual utility plant.

To get a deeper understanding of the Dymola model see Appendix A, where the graphic interface together with the text based interface are described.

4.1 Process Models

The process model was divided into four different parts: gas turbine, post combustion, HRSG and steam network, where each part was modelled separately. When the different models were completed, they were put together and the control system was developed. The main objective was to resemble the utility plant as accurately as possible. At steady state, mass and heat balances were a tool to verify the models. The transient behaviour was hard to verify because of lack of information about the equipment in the utility plant. This made the equations and the physics behind them more important. This section presents a summary of the model development along with verifications made to achieve a "correct" behaviour of the models.

4.1.1 Gas Turbine

In the utility plant, the gas turbine runs at full load. This made it possible to simplify the modelling of the gas turbine to a model with constant boundary conditions at the flue gas outlet. The mass flow and temperature was known, but the composition of the flue gas was unknown. With help of the combustion model, the flue gas composition was calculated. In Table 4.1.1 the specifications of the flue gas at the outlet is presented when the volume of the combustion chamber is assumed to be 1 m^3. The result from this indicates a high amount of carbon monoxide that does not react. Usually the carbon monoxide content is less than 10 ppm [18]. The NOx formation is also too low. Usually the flue gas have around 25 ppm in the outlet of the gas turbine [19].

To solve the incorrect compositions the volume of the reactor volume was increased to an unrealistic level, this because it was assumed to be more important to have a correct composition than a correct reactor volume due to the differences in heat capacity of the flue gases. To achieve a better composition, the volume of the gas turbine were increased to 10000 m^3. The unrealisticly high reactor volume did not change the transient behaviour of the gas turbine, which was validated through step response experiments. The composition is presented in Table 4.1.1 together with flow rate and temperature out from the gas turbine.

The schematics of the gas turbine model in Dymola can be seen in Appendix A.2 where the components are also described.

Table 4.1.1: *Combustion data from Dymola model for different reactor volumes. When the volume is realistic there are too high amount of un-reacted carbon monoxide. The NO formation is also less than expected.*

Flue gas information at outlet	Gas turbine	Gas turbine	Post Combustion
Fuel input [kg s^{-1}]	2.12	1.82	0.6
Excess air ratio	3.32	3.91	
Volume 10 [m^3]	1	10 000	10 000
Mass flow [kg s^{-1}]	132.1	132.1	132.4
Temperature [°C]	542	542	691
O_2 [w-%]	15.6	15.9	14.3
H_2O [w-%]	3.52	3.01	4.00
N_2 [w-%]	78.0	78.6	78.4
CO [w-ppm]	17 900	11.6	43.8
NO [w-ppm]	0	0	0
SO_2 [w-%]	0.128	0.110	0.146
CO_2 [w-%]	0.97	2.37	3.14

4.1.2 Post Combustion

The post combustion unit increases the temperature of the flue gas. This is done through combustion of additional fuel together with the flue gas from the gas turbine, which contain an excess of oxygen. The exit temperature of the flue gas after the post combustion unit was not known. Therefore, a Grand composite curve (GCC) was designed to study the process heat demand (see Appendix B). The GCC can be seen in Figure 4.1.2 which imply that the required temperature out from the post combustion is around 700 °C. The combustion model was adjusted to not exceed this temperature with help of a fuel input limitation. For flue gas composition, mass flow rate and temperature at full post combustion see Table 4.1.1. The fuel demand for an increased steam production from 16 kg s^{-1} to 24 kg s^{-1} is around 32 %.

To make the Post Combustion time dependent a first order transfer function with a gain of 0.6 and a time constant of 60 s was introduced together with a time delay of 5 s for changes in fuel input. The values were set to achieve dynamics in the model for the purpose of control. The real numbers are hard to estimate without data from the actual process and no similar data could be found. The temperature profile is seen in Figure 4.1.1 for a change from minimum post combustion to maximum post combustion at 200 s.

The post combustion model in Dymola is similar to the gas turbine model and is further described in Appendix A.3.

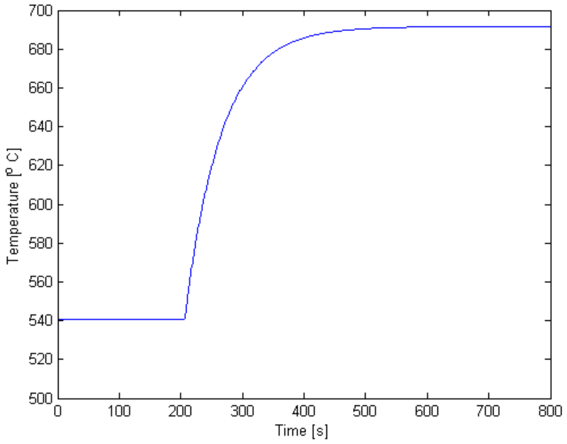


Figure 4.1.1: *Step response at* 200 s *displaying the temperature in the flue gas out from the post combustion unit. The time delay and the time it takes to reach full effect are introduced to make the model more time dependent.*

4.1.3 HRSG

The HRSG model consists of (for Dymola model see Appendix A.4):

- Economiser, heating the feed water to a user defined temperature.
- Drum, heat the flow from the economiser to saturation temperature with help of the energy in the steam.
- Evaporator, evaporates the saturated water.
- Superheater, heats the saturated steam from the drum to a user defined temperature.
- HRSG control system, control the feed water flow, the steam flow and the level in the drum.

The HRSG was designed to produce steam at 63 bar and 525 °C. Flue gas and water temperatures at the design point were determined with help of the grand composite curve seen in Figure 4.1.2. The heat transfer coefficient was estimated to be 90 $\mathrm{Wm^{-2}K^{-1}}$ in the heat transfer equipment [20].

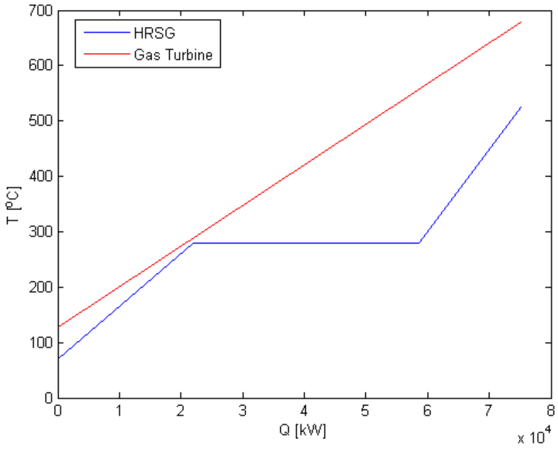


Figure 4.1.2: *Grand Composite Curve over the process at full post combustion. Red line describe the energy in the flue gas. The blue line indicates the heat demand at economiser, evaporator and super heater. It Indicates that the process total heat demand is around* 70 MW *along with a required flue gas temperature of at least* 690 °C.

In the original HRSG model there were little time dynamics. To get more time dependent dynamics in the HRSG, flue gas tanks were implemented with a large volume, which resulted in delays at both flue gas side and water side. The temperature in the superheater, evaporator and economiser are seen in Figure 4.1.3. The delays can be explained by the time it takes for the flue gas to pass the HRSG. The time delay was constructed so that a significant share of the time delay comes from the HRSG unit. The influence of delays due to the HRSG is explained further in Section 4.1.5.

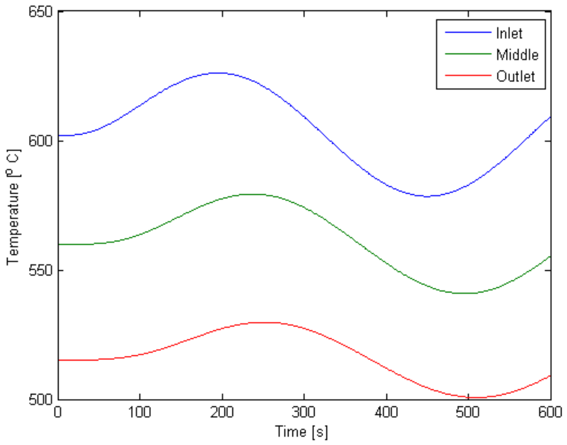


Figure 4.1.3: *Flue gas temperature profile for the superheater: It shows the constructed time delay at load changes in the form of temperature delay.*

4.1.4 Steam Network

The steam network was constructed using the overview of the utility plant illustrated in Figure 2.0.1. For the Dymola model see Appendix A.5. Tanks were required when streams were divided or combined. Because the solver handle two streams easier if the solver can use the derivative for small differences. This did not influence the result in any major way.

The valves were constructed to handle the maximum flow and produce a correct pressure drop. The opening and closing time for all valves was approximated to 20 s. In this project no information about the valves was supplied so standard valves was used. The choice of valves can have large effect on the steam network dynamics, due to differences in properties of different valves. The position of the valves is decided with a control signal between 0 and 1, where 0 is closed and 1 is fully open.

The steam turbines isotropic efficiency was 75 %. Together with the full load design point the Stodola machine constant could be estimated for each turbine. This resulted in a correct model for the full load case at steady state. The part load calculations can probably be questionable due to a large extent of simplifications made in the Stodola model.

The steam network has a long steam network from the HRSG to the turbines. To introduce time dependent dynamics, a steam pipe delay was constructed with help of pressure drops together with tanks. It was assumed that the pipe length was about 2 km long, the most common pipe size at current flow rate is according to 0.3 m TLV[21]. The estimated pressure drop was 4.5 bar and the total volume of the pipes was estimated to be 150 m^3. The pressure drop and volume was divided into five steps with one pressure drop and one tank in each. Including more tanks and pressure drops did not significant change the process dynamics. The constructed time delay is further analysed in Section 4.1.5.

4.1.5 Time Delays

The main contributors to the total time delay in the final model are illustrated in Figure 4.1.4. The largest one is due to the HRSG unit that provide an estimated time delay of 25 s, the second largest is due to the model without constructed time delay. This includes dynamics of valves and responses of controllers. This sums up to 17 s. The time delay due to long piping is estimated to 8 s. The figure also shows that the time constants do not depend on added time delay. The biggest contributor to the time constant is the post combustion unit. The shape of the response is different, where the final model has more linear behaviour than the model without added time delay.

The reason for adding time delays was to make the process harder to control and also more realistic. The exact values of the time delays in different equipment cannot be verified due to lack of data for the process.

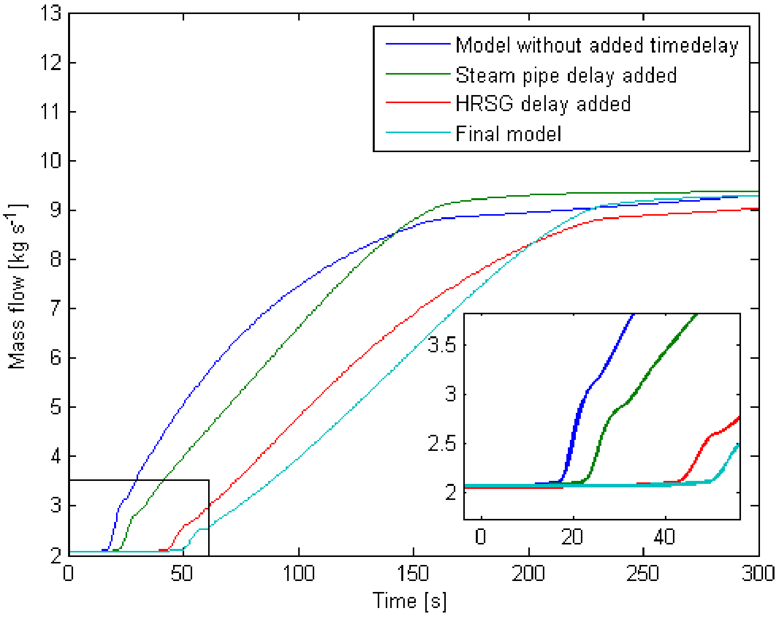


Figure 4.1.4: *Step response from minimum post combustion to maximum post combustion, for different dynamics are added to the process. The HRSG is the main contributor to the delay, the second largest is the dynamic of the system modelled without added time delay and the steam pipe delay give a minor influence to the time delay.*

4.2 Control model

For control of the process, a number of different controllers were required. The control of interest is the post combustion controller illustrated in Figure 2.0.1. Therefore the other controllers required in this project got less attention. The other controllers was designed with a high gain ($>$ 10 times higher than the control for the post combustion) to reduce the influence on the result of the post combustion control. An overview of the controllers in Dymola is presented in Appendix A.6.

4.2.1 Post Combustion

The controller studied in this thesis is the post combustion controller. A number of different controllers was implemented and tuned to see the effect of different tuning methods. The results are presented in Chapter 5.

Some characteristics was the same for all controllers. One thing was that the controllers were assumed to have a delay of 2 s, due to slow measurement equipment along with the internal delay of the controller. Another property was that the controllers had 1 as upper limit of control signal representing full effect on post combustion and 0 as the lower limit of the control signal, representing no post combustion.

4.2.2 HRSG

A 3-point boiler PI controller was used to control the pump supplying feed water to the HRSG unit. It was used to control that the dome level of the HRSG was on a stable level. The faster loop measures the flow in to and out from the HRSG to rapidly compensate for changes in steam demand.

The controller was tuned to be faster than the post combustion controller and the dome level set point was set on half full. During simulations in Dymola the dome level had some small oscillations during steam changes, but stabilised quite fast.

4.2.3 Steam Network

To control that the steam network keeps the correct pressures, additional controllers were required all being PI controllers and tuned fast. This is assumed as a good approximation because the controllers cannot be unstable in the simulations due to no measurement noise and no dead time between their measurement point and control point. In real life there are delays, measurement noise and non-linearities that will influence the control performance and possibly make these controllers unstable if not tuned correctly. The set points were supplied by the plant description to keep correct pressures in the steam network.

Between the second turbine and the 6 bar steam network there is one controller controlling two valves. There was no information on how this was implemented in the actual utility plant. For simple reasons the valves were modelled to be open one minus the other one. This achieved a correct flow and pressure characteristic for the flow through the second turbine. There was also a limitation in how far the second valve could be closed to always keep the minimum flow through the second turbine.

5 Results

In this chapter the results from the project are presented. It includes determination of a transfer function model, characterisation of the system and the design and evaluation of possible controllers to control fluid/thermal processes with large time constants and significant time delay.

Methods used in this project to dimension PI(D) controllers are: the Lambda method, the Zeigler-Nichols method, the κ_{180} method and with help of stability margins. To further increase performance of the controllers, Smith prediction was implemented for its reduction of overshoot/undershoot.

When the derivative part is included in the controller, measurement noise can have large effects on the controllers performance. Therefore, measurement noise was studied for PI(D) controllers with and without Smith Prediction.

To evaluate the controllers four different events was studied, called Setup 1:

- Steam demand increase: $5\,\mathrm{kg\,s^{-1}}$ at $500\,\mathrm{s}$. This indicates how the process responds on disturbances when the flow through the second turbine is low. It represents a load disturbance that requires the bypass valves to be open to achieve correct pressures in the steam network.
- Steam demand decrease: $5\,\mathrm{kg\,s^{-1}}$ at $2000\,\mathrm{s}$, This indicate how the process respond on disturbances in form of reduced steam demand.
- Reference value change: From $4\,\mathrm{kg\,s^{-1}}$ to $8\,\mathrm{kg\,s^{-1}}$ at $3500\,\mathrm{s}$. This indicates how fast the system responds to manual setpoint changes.
- Reference value change: From $8\,\mathrm{kg\,s^{-1}}$ to $4\,\mathrm{kg\,s^{-1}}$ at $5000\,\mathrm{s}$. This indicates how fast the system responds to manual setpoint changes in the other direction.

5.1 Process Model

In this section the derivation of the transfer function model is explained. This model was used for design of optimal controllers and to calculate system performance, such as phase margin and amplitude margin. It was also used for the Smith prediction controllers as a model to evaluate the Smith prediction.

A step response experiment was done in Dymola to determine a simple transfer function model for the flow to the second turbine. With no process disturbances, a step response from minimum post-combustion to maximum post-combustion was made. This resulted in the dynamics shown in Figure 5.1.1. The behavior looks as for a first order process with time delay, but has a steeper start of the step response and a less steep end of the step response. It also indicates that the process only has poles that are real, due to the stable behavior.

The step response was used to determine a transfer function model with the parameters: $K \approx 9$, $T \approx 120$ and $L \approx 50$. With the Padé approximation in Section 3.3 this gives:

$$G(s) = \frac{9}{120+1} \cdot \frac{1}{(50+1)} \cdot \frac{12 - 6 \cdot 50s + (50s)^2}{12 + 6 \cdot 50s + (50s)^2} \tag{5.1.1}$$

A comparison of the transfer function model and the Dymola model was done for verification (see Figure 5.1.1). The transfer function indicates high similarities in the beginning of the step response, as expected. The time constant influenced behavior deviate from the model some at the end of the step.

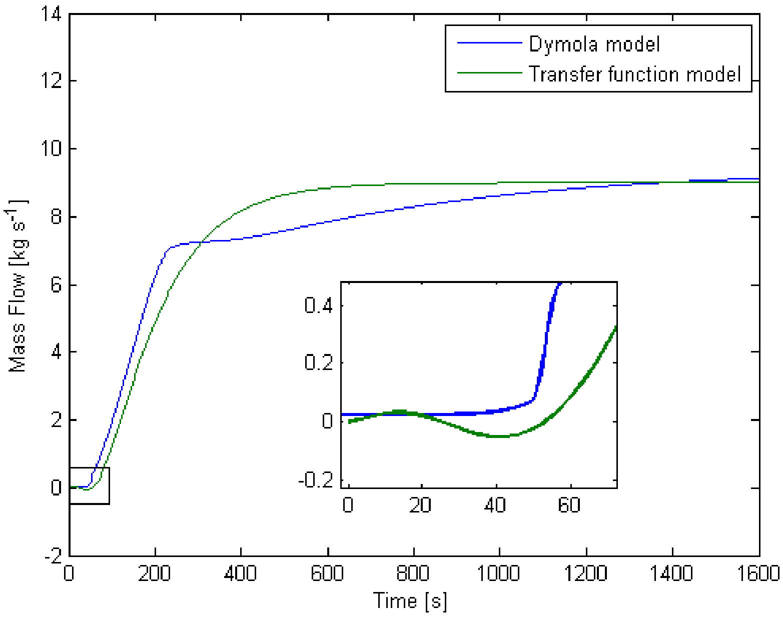


Figure 5.1.1: *Step response from Dymola model and transfer function model. The Dymola model has a behavior that resembles a first order process except from the rapid response in the beginning and slower at the end of the step. The transfer function model shows large similarities in the beginning of the step but some differences at the end. The oscillation behavior in the transfer function model at the beginning of the step is due to the Padé approximation.*

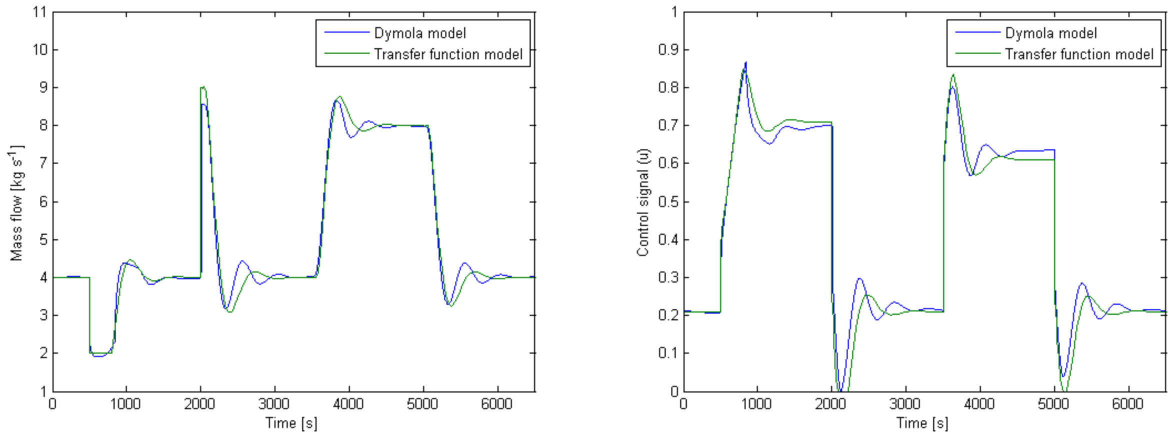
5.2 PI(D) Control Design Based on Transfer Function Model

In this section the transfer function model is used to dimension and analyse the design of one PI controller and several PID controllers using Zeigler-Nichols, κ_{180} and stability margin tuning. These controllers were then implemented in Dymola to see the responses to load disturbances and reference changes. A comparison explaining the performance of the controllers in simulations in Dymola ends this section.

5.2.1 Model Margin PI Controller

The PI controller was designed to have a maximum sensitivity function to be less than 1.7 and to have a maximal complementary sensitivity function less than 1.3. With these condition the controller achieved a phase margin of at least 45 ° and an amplitude margin of at least 7.6 db. These conditions were then used to optimise the controller to have as high gain as possible. For more details about this controllers constants, stability and performance, see Section 5.2.5.

To verify that the transfer function model behaves as the Dymola model, a simulation was made with the transfer function model to compare it with the Dymola results (see Figure 5.2.1). The responses of the transfer function model and the Dymola model are quite similar: The response to a load disturbance shows an overshot of the response but stabilises quite fast, which indicate a stable controller. Also the control signal behaves similarly for the Dymola model and the transfer function model. This made it possible to assume that the transfer function model corresponds well with the Dymola model.



(a) (b)

Figure 5.2.1: *Responses from simulations according to Setup 1: (a) Mass flow to condenser: The simulation indicates a declining oscillation in the response from both models. (b) Control signal from the PI controller: The behavior is similar during the simulations, through with peaks slightly higher for the Dymola model simulation.*

5.2.2 Model Margin PID controller

To improve the response at load disturbances a PID controller was dimensioned using the same stability margins as the PI controller. The control activity was optimised to have as little control activity as possible to get a fast response. This was done using beta optimisation where the high frequency gain was compared to the compensation factor for low frequency load disturbances, this can be seen in Figure 5.2.2. Where β is a measurement on how much control activity the PID controller has. It indicates that the increased response for increased control activity decay after $\beta = 10$. Therefore β was set to 10. For more details about this controllers constants, stability and performance, see Section 5.2.5.

In Figure 5.2.3 the increased performance of a PID controller in form of a much faster control signal gives a faster response to load disturbances in the process. The responses of the Dymola model and the transfer function model are very fairly when comparing the control signal.

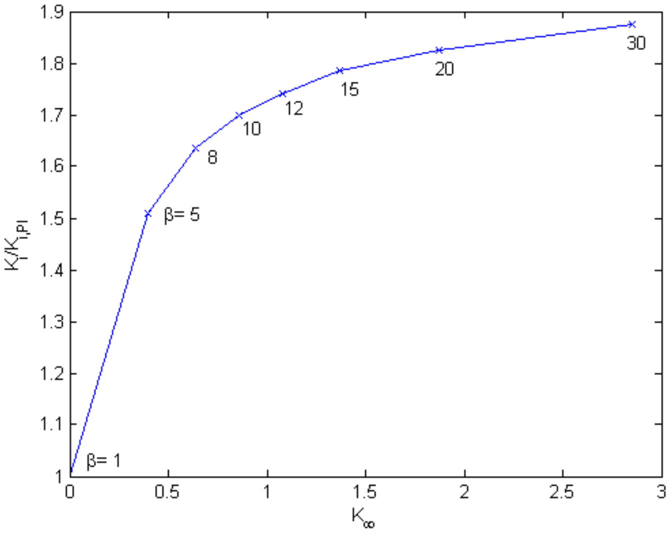
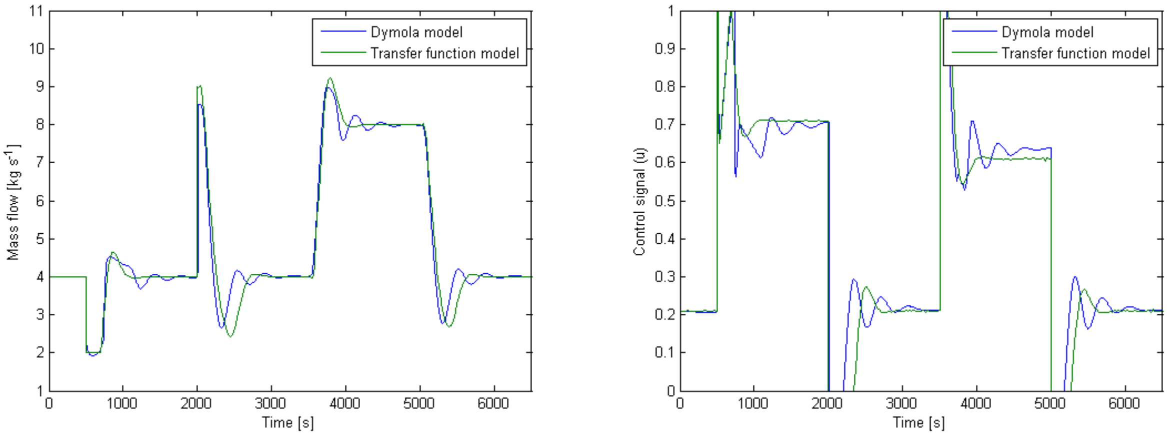


Figure 5.2.2: *The control activity versus the gain: K_i is normalised with a PI controller. It shows that the most optimal β for smallest control activity versus the gain is around $\beta = 10$.*



(a) (b)

Figure 5.2.3: *Responses from simulations according to Setup 1: (a) Mass flow to condenser: The simulation indicates a declining oscillation in the response from both models. (b) Control signal from the PID controller: The Dymola model has more unstable control signal due to more oscillations.*

5.2.3 Model Z-N PID Controller

There are a lot of different tuning rules given by the control community. One old method is the Z-N method. Using this method to determine the parameters of the PID controller gave a high gain of the controller with lower filtering on the derivative part of the controller and ideally much lower amplitude margin. For more details about this controllers constants, stability and performance see Section 5.2.5.

5.2.4 Model κ_{180} PID Controller

One more recent method to determine PID controller parameters is the κ_{180} method. In this method the parameters becomes similar to the model margin PID controller. The integral constant becomes slightly lower and the derivative part becomes a bit higher but with more low frequency filter. For more details about this controllers constants, stability and performance see Section 5.2.5.

5.2.5 Comparison of Model Design Strategy

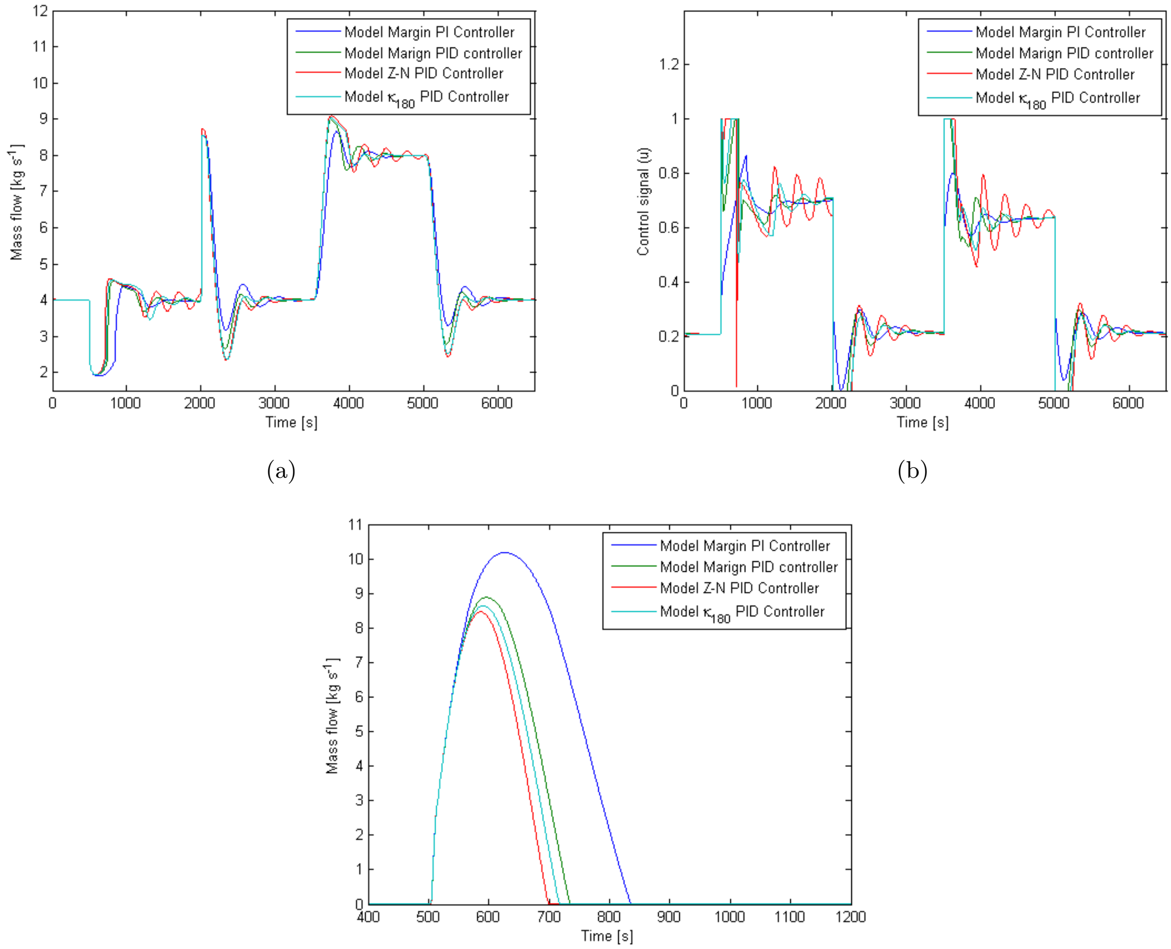
To evaluate the controllers constructed with help of the transfer function model, the response from simulations according to Setup 1 were analysed. The response, the control activity and the need for bypass is plotted see Figure 5.2.4. The response to reference changes and reduced steam demand, results in no bypass, all controllers except the model Z-N PID controller has the same time to reach stable conditions.

When bypass was required the model margin PI controller responded much slower to load disturbance, resulting in much more need for bypass. The fastest one was the model Z-N PID controller, which decreased the time for bypass with 37 % compared to the model margin PI controller. The comparison indicates that derivative action improve the response to load disturbances and that the model Z-N PID controller is less stable than the other controllers.

In Table 5.2.1, the parameters for the different controllers are presented. The gain is highest for the PID controllers. The filter time constant for the PID controllers differ some where the model κ_{180} has a higher filter constant and the model Z-N PID controller has the lowest filter constant. The amplitude margin for the Z-N method is a lot lower than the others. The phase margin is almost the same for all controllers, but slightly less for the model Z-N PID controller.

Table 5.2.1: *Table comparing transfer function based tuning methods. The controllers constants and the margins of the controllers is presented.*

Controller	K_p	T_i	T_d	T_f	M_S	M_T	β	ϕ_m	A_m
Margin PI	0.0912	119	-	-	1.70	1.12	-	53.1	10.4
Margin PID	0.176	131	29.4	6.89	1.70	1.18	10.0	50.2	9.50
Z-N PID	0.282	143	35.7	3.56	2.30	1.53	3.04	42.5	5.70
κ_{180} PID	0.177	107	46.6	11.2	1.79	1.23	6.99	49.5	7.70



(a)

(b)

(c)

Figure 5.2.4: *Responses from simulations according to Setup 1: (a) Mass flow to condenser: The three different PID controllers have almost the same behavior, but the model Z-N PID controller has more oscillations than the others. The model margin PI controller has a slower response when bypass is required. (b) Control activity: The model Z-N PID controller does not reduce the oscillation as fast as the other controllers. (c) . Bypass mass flow for first load disturbance: Indicates that model Z-N controller is the fastest one and the model margin PI controller is the slowest one.*

5.3 PI(D) Control Design Based on Experimental Data

Different procedure can be used to determine PI(D) controllers parameters when the model of the process is unknown. One easy way is to use the lambda method. The lambda method only requires a step response experiment. To make a more advanced design a relay experiment can be used to determine ω_{P180} and $P(j\omega_{P180})$ This can then be used with the κ_{180} tuning method or the Z-N tuning method. In this section both these methods are used to tune the PI(D) controller.

The relay setup was to control the process from a control signal from 0.1 and 0.3 this resulted in the behavior shown in Figure 5.3.1. A step response was also done at the operating point. With the data the properties of the process could be determined to: $\omega_{P180} = 0.0198$, $P(j\omega_{P180}) = 2.7096$ and $\kappa_{180} = 0.2705$.

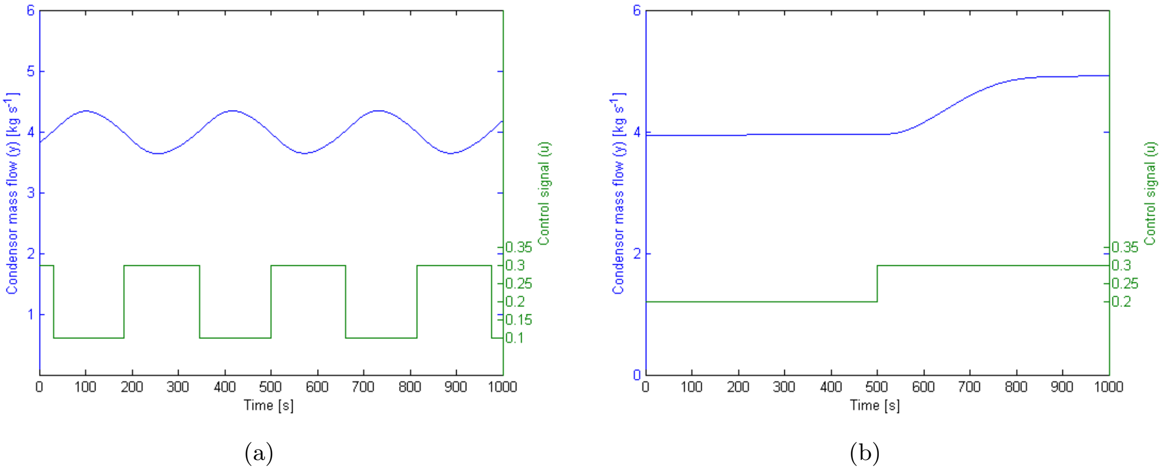


Figure 5.3.1: *(a) Self-oscillation due to relay implementation. The result can be used to find* ω_{P180}, $P(j\omega_{P180})$ *and* κ_{180}. *(b) Step response at operating point to determine stationary gain.*

5.3.1 Exp-Data Lambda PI controller

In Figure 5.3.2 the gain of the exp-Data Lambda PI controller is plotted for different values of λ_f. When $\lambda_f = 0.8$ the gain is the same as for the model margin PI controller, with good stability margins. $\lambda_f < 1$ is usually described as a very fast controller. The integral time constant for the controller was 120 s. For more details about this controllers constants, stability and performance see Section 5.3.4.

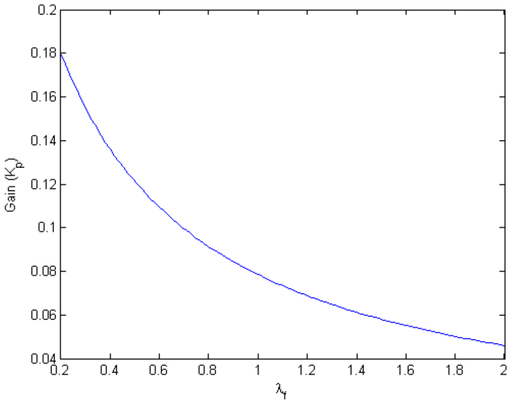


Figure 5.3.2: *The gain of the PI controller for different values on* λ_f. *When* $\lambda_f = 1$ *the gain was* 0.078, *which gave a slower controller than the model margin PI controller.*

5.3.2 Exp-Data Z-N PID Controller

With help ω_{180} and $P(j\omega_{180})$ the parameters of the controller was determined. The use of experimental data, compared to using the model, results in higher gain of the controller with less filtering on the derivative part. If the same values are used on the transfer function model the amplitude margin decrease. Because of the high gain it becomes one of the fastest controllers. For more details about this controllers constants, stability and performance see Section 5.3.4.

5.3.3 Exp-Data κ_{180} PID Controller

The relay experiment resulted in $\kappa_{180} = 0.2705$ and with help of this the controllers constants were determined. The controller compared to the model based κ_{180} controller, results in lower gain, higher integral time constant, higher derivative time constant and almost the same filter constant as for the derivative part. For more details about this controllers constants, stability and performance see Section 5.3.4.

5.3.4 Comparison of Control Design Using Experimental Data

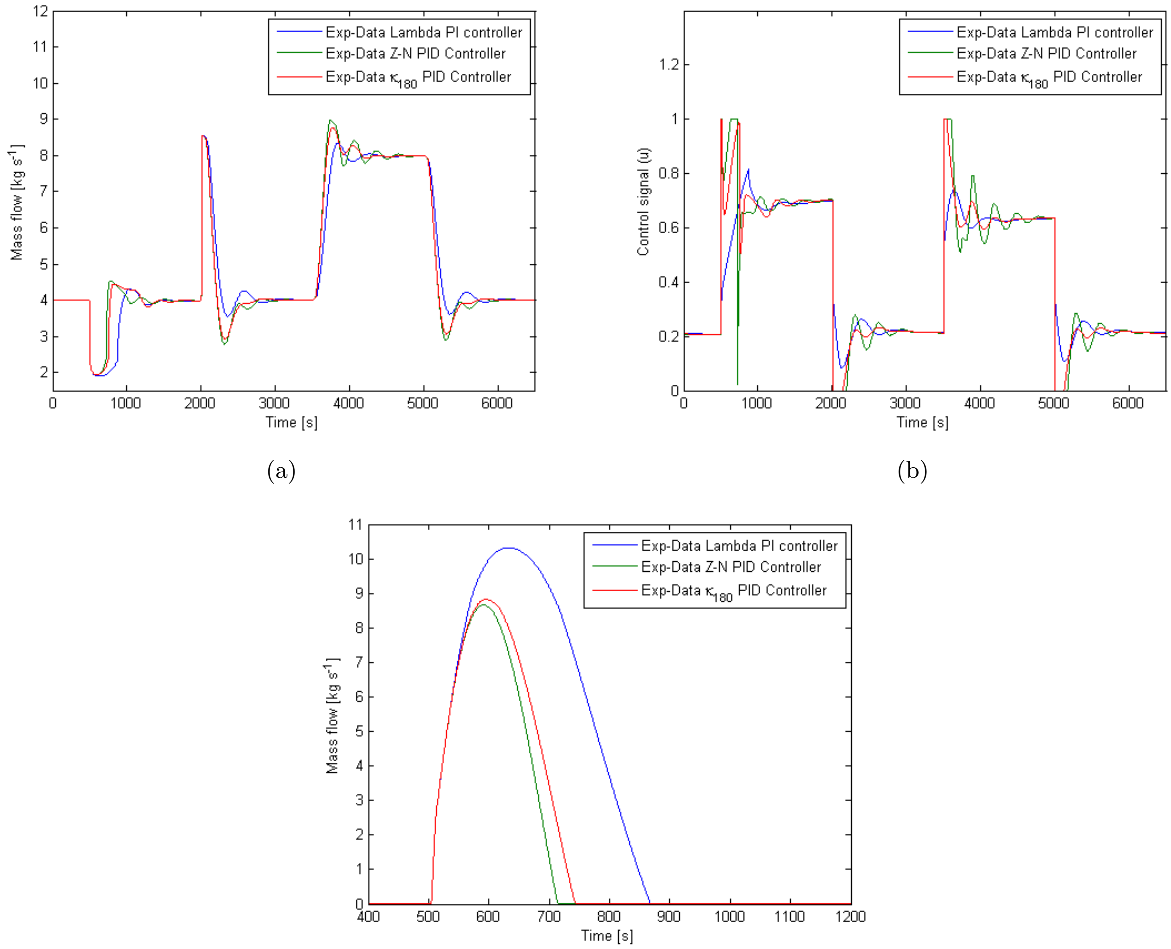
To evaluate the controllers constructed with help of the experimental data the responses from simulations according to Setup 1 were analysed. The responses, the control activity and the need for bypass are plotted in Figure 5.3.3.

The results are similar to tuning based on the transfer function. As before, the Z-N PID controller has larger oscillations in both signal activity and measurement value, but are also the fastest ones to compensate for load disturbances when bypass is required. The lambda method use bypass for another 200 s due to the low gain in the PI controller. The Exp-Data κ_{180} PID controller behaves almost the same as the transfer function based κ_{180} controller.

In Table 5.3.1 the parameters for the different controllers can be seen. The gain is highest for the PID controllers and Z-N has the highest of them. The amplitude margin for the Z-N method is a lot lower than the others but the margins have increased for all methods compared to the transfer function tuning methods. The phase margins are almost the same for all the controllers, and are also higher than for the transfer function tuned controllers.

Table 5.3.1: *Table comparing experimentally tuned methods. The controllers constants are in the first four columns. The last four indicate the margins of the controllers, calculated using the transfer function model.*

Exp-Data	K_p	T_i	T_d	T_f	M_S	M_T	ϕ_m	A_m
$\lambda_f = 1$ PI	0.0781	120	-	-	1.57	1.03	58.1	11.8
Z-N PID	0.221	159	39.8	3.98	1.86	1.06	56.8	7.50
κ_{180} PID	0.142	115	50.2	11.3	1.57	1.11	57.1	9.30



(a)

(b)

(c)

Figure 5.3.3: *Responses from simulations according to Setup 1: (a) Mass flow to condenser: The PID controllers have almost the same behavior, but the exp-data Z-N PID controller has a little more oscillations than the others. The exp-data Lambda PI controller has a slower response when bypass is required. (b) Control activity: The exp-data κ_{180} PID controller has faster oscillation compensation. (c) Bypass mass flow for first load disturbance: Indicates that the exp-data Z-N PID controller is the fastest one and the exp-data Lambda PI controller is the slowest one.*

5.4 Smith Prediction Control

Smith prediction control is used to avoid overshoot/undershoot and oscillation when controlling of processes with large time delays. The model implemented is the one described in Section 3.3.5. This Smith prediction was implemented for two PI- and one PID- controller.

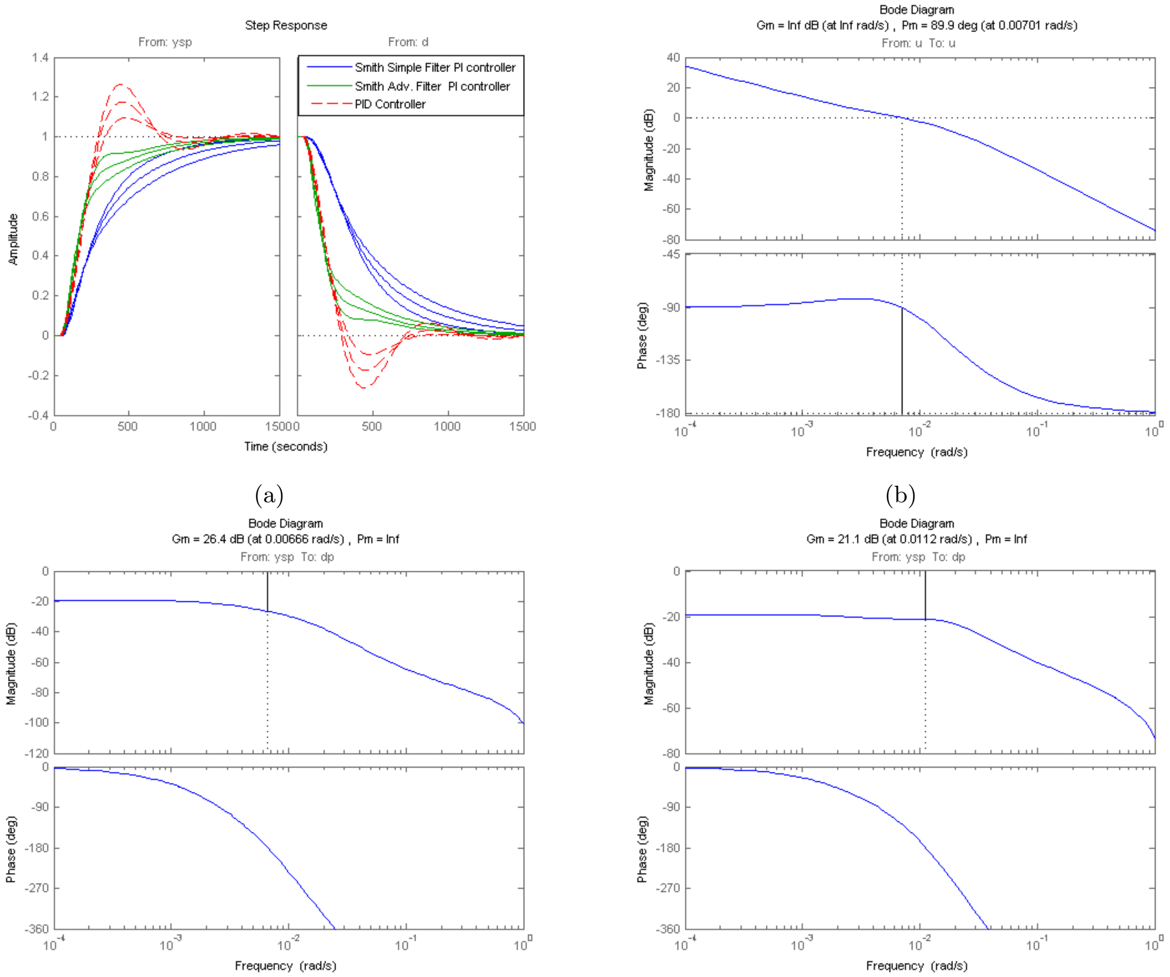
5.4.1 PI Controller with Smith Prediction

The design of PI controller was made a little different than before. The phase margin was set for 90 ° and a crossover frequency was then iterated for good stability margins. The PI controllers parameters was iterated to: $K_p = 0.132$, $T_i = 240$.

Noise often have big impacts on the Smith predictor, therefore two filters were evaluated: The first filter was a simple filter using $T_{F1} = 10$ and $T_{F2} = 100$. The second filter was an advanced filter using a prediction theory for expressing a positive time delay.

The result for the transfer function model and two mismatch models are displayed in Figure 5.4.1. Both controllers make the system respond without any overshoot. If the more advanced filter is present the system respond much faster to reference and load disturbances, but the time it takes before the process reach its target is about the same.

The stability margins are very good for the inner loop with a phase margin on 90 ° and the outer loop has ideally an infinite amplitude margin but if mismatched, the stability can be studied and a large amplitude margin of 21.1 db is achieved for the mismatch model so there are still high margins. The model mismatch is for 10 % of the parameters original value (K, T, τ_d) in both directions (lower/higher). The response from a standard PID controller indicates that the Smith prediction PI controller and the PID controller have the same time to reach the final value, but with help of the Smith prediction the overshoot/undershoot is avoided. For more details about this controllers performance see Section 5.4.3.



(a)

(b)

(c)

(d)

Figure 5.4.1: *(a) Theoretical step responses for Smith predictor with PI controllers and one PID controller from reference signal and from load change: The Smith predictor removes the overshoot/undershoot ideally, but also for mismatched models. (b) Frequency plot for inner loop: The margin is high in the inner loop (c) Frequency plot for outer loop and simple filter when the model is mismatched: The margins are good for the mismatched model. (d) Frequency plot for outer loop and advanced filter when the model is mismatched: The margins are good for the mismatched model, but the amplitude margin is lower than for the simple filter.*

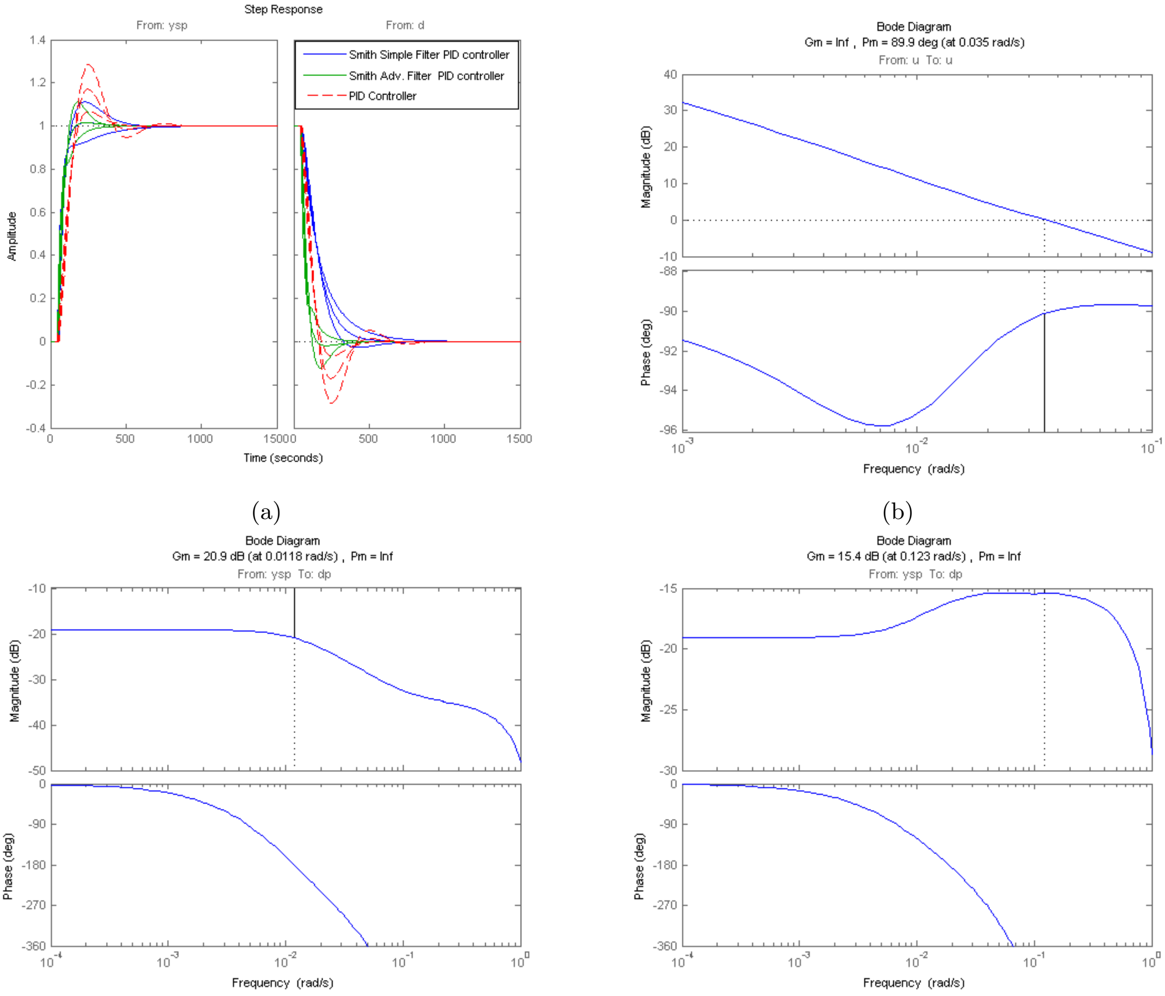
5.4.2 PID Controller with Smith Prediction

If a PID controller is implemented instead of a PI controller the performance increases further. The PID controller was implemented with a phase margin of 90 ° and the crossing frequency for the total loop was iterated to have good margins. The PID parameters was iterated to: $K_p = 0.663$, $T_i = 144$, $T_d = 36$ and $T_f = 10$. The filters are the same as for the Smith PI controller. The results are shown in Figure 5.4.2.

For reference changes both filters behaves better than a standard PID controller. If no mismatch, the controller stabilises much faster. For load disturbances, the Smith predictor with an advanced filter performs much better.

As with the PI Smith controller the margins are good for both inner and outer loops, but there is a slightly smaller amplitude margin for the outer loop when the advanced filter is used.

The model mismatch is 10 % of the parameters original value (K, T, L) in both directions (lower/higher). For more details about this controllers performance see Section 5.4.3.



(a) (b)

(c) (d)

Figure 5.4.2: *(a) Theoretical step response for Smith predictor with PID controllers and one PID controller from reference signal and from load change: The improved response for PID controllers compared to PI controllers is significant. The mismatched models have a small overshoot/undershoot, where the Smith predictor with simple filter has the smallest one. (b) Frequency plot for inner loop: The margin is high in the inner loop as for the PI controller.(c) Frequency plot for outer loop and simple filter when the model is mismatched: The margins are good for the mismatched model. (d) Frequency plot for outer loop and advanced filter when the model is mismatched: The filter has problems for high frequencies, but the margins are good*

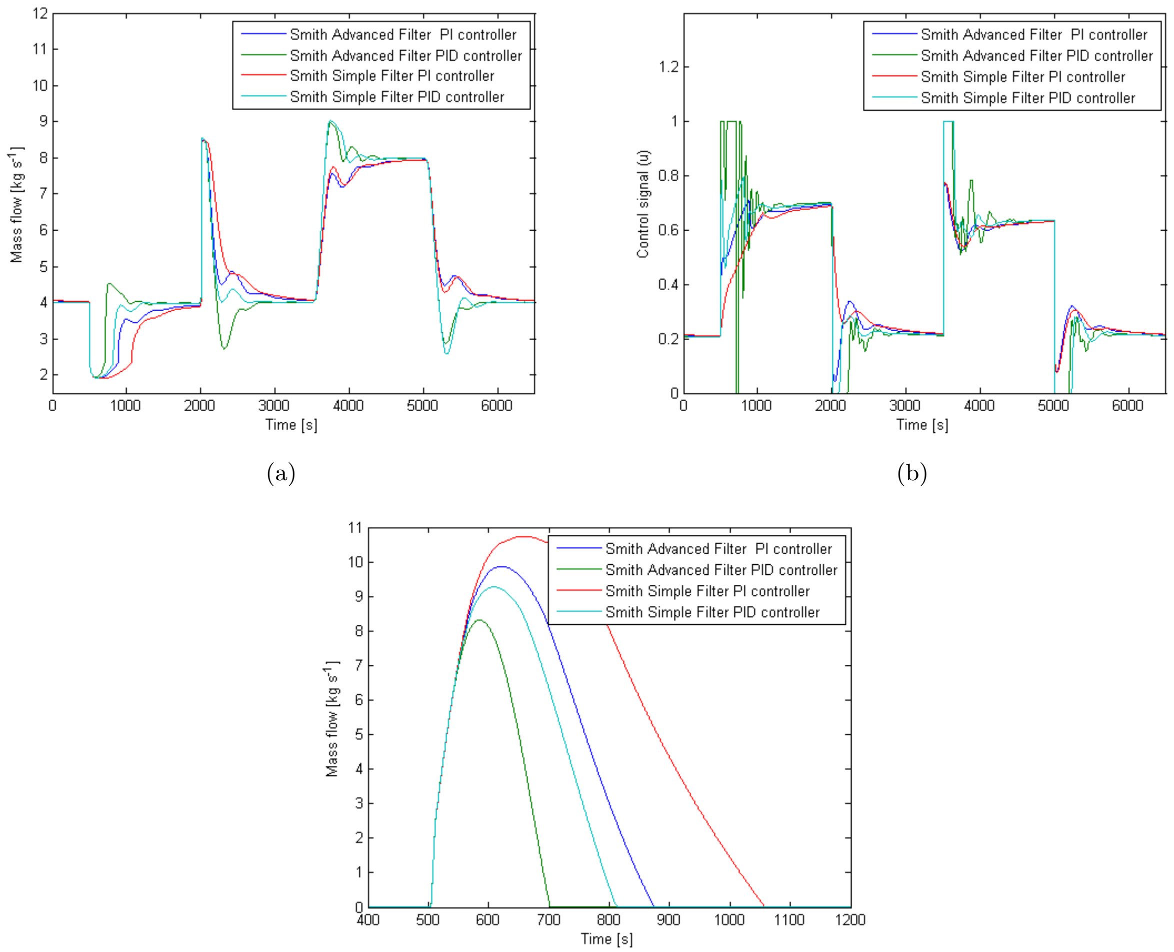
5.4.3 Comparison of Smith Controllers

To evaluate the Smith prediction controllers performance the response from simulation in Dymola according to Setup 1 was analysed. The response, the control activity and the need for bypass are plotted in Figure 5.4.3.

The responses are not as good as expected, probably due to model mismatch. As can be seen, the best controller to reduce overshoot is the PI controller with simple filter. When bypass is require,d it takes a lot longer time for the controller to compensate the load disturbance.

The implementations with PID controllers makes the response fast, but with significant overshoots. There are no large oscillations as the standard PID controllers have. The control activity is high for the Smith PID controllers, and very high for the Smith advanced filter PID controller.

The only controller that performs well at load disturbances at low set points is the Smith advanced filter PID controller where the rest have much longer time before the load disturbance is compensated.



(a)

(b)

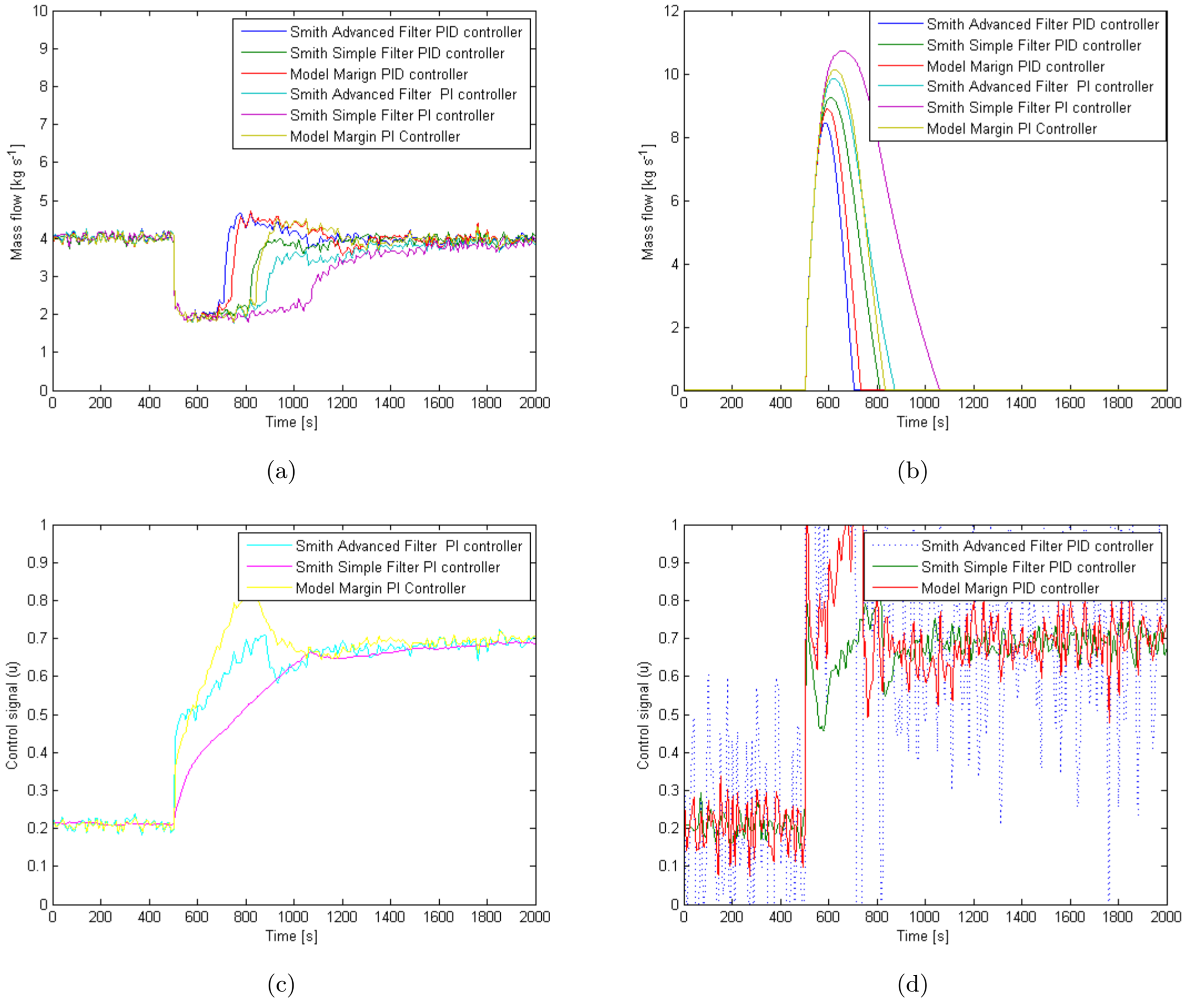
(c)

Figure 5.4.3: *Response from simulation according to Setup 1: (a) Mass flow to condenser: The PI controllers and the simple filter PID controller reduce the overshoot significantly. The PID with advanced filter reduces the oscillations, but have a big peak. (b) Control activity: The control activity is high for the advanced filter PID controller. (c) Bypass mass flow for first load disturbance: The only one performing as good as a standard PID controller is the Smith advanced filter controller. The rest requires substantially longer time for bypass.*

5.5 Response to Measurement Noise

To design controllers theoretically without any noise is relatively easy. For large processes there is always noise in measurements and from the process itself. To study how the controllers behaves in presence of noise a measurement noise was simulated in Dymola. Most of the controllers have a derivate filter constant around 10 s, which makes the variations small, but to see if larger noise have effect, a white noise with a standard deviation of 1 % on the flow measurement and a sampling interval at 10 s was implemented.

The result is presented in Figure 5.5.1. PI controllers are much more robust against measurement noise and small load disturbances due to the lack of a derivative part in the controller. With PID controllers the control activity increases. The PID controller overcompensate in the standard case every small noise duration too much which can be seen in the control signal. When the PID controller is implemented with Smith predictor and advanced filter the control signal is even more unstable. The simple filter handles the noise much better than the advanced filter.



(a)

(b)

(c)

(d)

Figure 5.5.1: *Responses to load disturbance which requires bypass: a) Mass flow to condenser: Similar behavior as for simulation without measurement noise. b) Bypass required: Similar behavior as for simulation without measurement noise. c) Control activity PI controllers: The PI controllers are stable with measurement noise, with only some small fluctuations. The Smith Simple filter PI controller is filtering the noise best. d) Control activity PID controllers: The Smith advanced filter PID controller becomes unstable in the control activity, with large fluctuations. The standard PID controller also has problems with high fluctuations. The PID controller with simple filter has low fluctuations.*

5.6 Overall Comparison of Controllers

The design of an optimal controller for a fluid/thermal process with large time constants and significant time delay depends mainly on three things:

- How fast response should the controller have to reference changes and load disturbances.
- How necessary is it to reduce overshoots/undershoots in reference changes and load disturbances.
- How much noise is it in the process.

If a transfer function model is known, an extensive analysis can be done to determine the PI and PID control parameters. If no model is available and the process is stable (real poles), a simple model can be derived to estimate these properties. The recommended method is to use margin based PI or PID tuning methods to determine the parameters of the controller. Z-N and κ_{180} tuning methods are not recommended when a process model is known.

To tune a process quickly with help of experimental data the lambda tuning method can be used. The method results in a slow controller, which is stable and is easily tuned and a step response is the only required data. It is a trial and error method where one parameter is changed, and is therefore a very fast tuning method. It produce low overshoot/undershoots due to the slow response and are very robust against noise.

If the response from a PI controller is too slow the recommended method is to make a relay experiment and then use the κ_{180} method to determine a good PID controller. It will produce an overshoot/undershoot, which is hard to reduce without slowing down the response. The influence of measurement noise affects the PID controllers more than a PI controller. This problem can be reduced by measurement filtering, but that will also reduce the response of the controller. The Z-N tuning method is not recommended due to oscillations in the control signal.

If no overshoot/undershoot is desired a slow tuned PI controller will work fine. If the response should also be fast, a Smith prediction controller is necessary. The design of the Smith prediction controller requires detailed and accurate process model. If the process model is mismatched, it can result in similar results as a standard PID controller. To have a fast response a Smith prediction PID controller is necessary. However, this controller is vulnerable for measurement noise and model mismatch. The best implementation for a fast and stable Smith prediction control is with a PI controller with a simple filter on the measurement value.

6 Discussion

This chapter discuss the model development in Dymola, along with a discussion of how well a transfer function model describes one variable of a fluid/thermal process. The control design is also discussed in terms as stability and response to reference changes, load disturbances and measurement noise.

6.1 Dymola Model and Transfer Function Model

To derive a model for a utility plant with help of Dymola, the physical relations of the equipment is required. Often is the physics simplified to describe the steady state solutions quite good, but for the transient behaviour, much of the time dependent dynamic is lost. This makes the control design questionable when the model is based on a Dymola model. To verify and modify the Dymola models to behave as expected in transient mode, an extensive amount of experimental data for the real equipment is required. In this project, the lack of data for the subsystems made it hard to verify that an accurate model was created.

Development of a model in Dymola requires a large amount of time, and also much information about the process. The model will give a lot of information that cannot be seen with experimental data if not every parameter is measured. Another advantage with the Dymola model is that different improvements and changes of the process can be analysed at a low cost.

The approximation of a process with a simple transfer function gave good results in this project, indicating that the Dymola model is quite linear. In real life, more nonlinear behaviour could influence the dynamics and therefore the control of the process. Overall it seems that a transfer function model of first order with time delay describe the process flow characteristics quite good, but further investigations are required to verify this result.

6.2 Control Design

PI controllers have a quite easy design procedure. If a model of the process is known, a design with good stability margin can easily be created. If instead data from a step response is known, the lambda method can be used to design the PI controller. The PI controller design usually gives a stable control with a slow response to reference changes and load disturbances. It handles measurement noise quite good without any filtering of the measured signal, which corresponds well with the theory about PI controllers.

PID controllers can substantially improve the responses to reference changes and load disturbance. The control activity is also increased, which can be both positive and negative. PID controllers usually require more information about the process to make a stable control. At minimum, a model of the process or a relay experiment is required. The difference between the different PID tuning methods is how much theory versus ‘’best practical guidelines” they are based on. In this project, the Z-N tuning method for PID controllers results in an oscillating behaviour regarding the control signal and is therefore not recommended. The κ_{180} tuning method behaves well for both a design based on a model of the process and with help of a relay experiment. It also has large similarities with the model margin method of tuning, which agree with the theory about the κ_{180} tuning method.

In this project the load disturbance behaves as measurement disturbances, this can be seen as the controller gets information of the load disturbances when they occur. The problem is that the controller does not know the time it has take for changes to be transported down in the process. This easily causes an overshoot/undershoot of the response. If no overshoot/undershoot is desired a model can be used to compensate for this problem. This was implemented with help of a Smith prediction controller, where the controller predicts a control signal for the process without time delay. Theoretically this control can be as fast as a PID controller, but without an overshoot/undershoot. This is described in the design of the Smith predictors. The results from simulations in Dymola indicate that the Smith predictor is vulnerable to model mismatch because overshoot/undershoots occur. However, then theoretically this this controller should only be used when the dynamics of the process are known. The transfer function model developed in this thesis is not sufficiently correct, but further tuning will possible increase the performance of the Smith prediction controller. The time delay is for most cases in control of fluid/thermal processes not constant and therefore, it is a good idea to study if a varied prediction of the time delay can increase the performance. This was not possible to do with the Dymola model because of its constant time delay.

Anti-windup can be crucial in situations where the control signal saturates. This is a problem that is particularly common in control of fluid/thermal processes. However, this was not analysed, due to that reference changes and load disturbances were chosen to not cause any large effects of windup. In a final control this is an import factor that needs to be further analysed.

7 Conclusions

The implemented models in Dymola are a great for modelling and study the steady state performance of a process. For transient performance the models in Dymola are not so accurate. To verify and include more transient behaviour in the models, extensive measurement data is required for the process. Therefore, the design of a control system without this data makes the design of the controllers questionable.

The design of an optimal controller for a process with significant time delay depends mainly on three criteria: How should the controller respond to reference changes and load disturbances, how necessary is it to avoid overshoots/undershoots of reference changes and at load disturbances, and how much noise is it in the process.

Smith prediction controller was studied extensively in this thesis. The conclusion is that if the model of the process is accurate, the Smith predictor with a PI controller can perform as good as a PID controller in responses to reference changes and load disturbances, without any overshoot/undershoot. Another important factor for Smith controllers is that they are sensitive to process and measurement noise. Therefore, a filter is required to handle small disturbances. In this thesis two filters were studied. The simple filter performed well in spite of measurement noise, but had a slow response. The more advanced filter had a fast response but may cause an less stable control signal due to measurement noise.

Noise has a big effect on the performance of the controllers, especially for PID controllers where the control activity increases substantially with measurement noise.

7.1 Further Work

Further investigations are required in the model development done in Dymola. With more information available about the equipment a more accurate model can be developed and thereby a more reliable controller. This includes more information on valves, turbines and HRSG. The transient behaviour in the post combustion is also very important to describe accurately.

The most interesting future work is to validate if the theoretically good smith predictor can be implemented in reality with a simplified process model for fluid/thermal processes with large time constants and significant time delay.

Anti-windup for the controllers was not studied in this thesis. For the final tuning of the controllers, anti-windup needs to be conducted to compensate for physical limitations in the equipment.

References

[1] Dynasim AB. *Dymola, Dynamic Model Laboratory, User Manual.* 5th ed. 2009.

[2] The MathWorks inc. *Simulink, User Manual.* 2012.

[3] Lennartson B. *Reglerteknikens grunder.* 4:7. Malmö: Holmbers, 2008.

[4] Åström K.J. and Hägglund T. "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins." In: *Automatica* 20.5 (1984), pp. 645–651.

[5] Ingimundarson A. and Hägglund T. "Robust tuning procedures of dead-time compensating controllers". In: *Control Engineering Practice* 9 (2001), pp. 1195–1208.

[6] Anne Nortcliffe and Jonathan Love. "Varying time delay Smith predictor process controller". In: *ISA Transactions* 43.1 (2004), pp. 61 –71.

[7] Huang H.P. et al. "A Modified Smith Predictor with an Approximate Inverse of Dead Time". In: *Journal of American Institute of Chemical Engineers* 36.7 (1990), pp. 1025–1031.

[8] International Association for the Properties of Water and Steam. *IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Stem.* 1997.

[9] McBride B.J., Zehe M.J., and Gordon S. *NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species.* NASA report TP-2002-211556. 2002.

[10] Naturalgas.org. *Overview of Natural Gas:Background, Accessed.* Dec. 2012. URL: `http://naturalgas.org/overview/background.asp`.

[11] Nicol D.G. et al. "Development of a Five-Step Global Methane Oxidation-NO Formation Mechanism for Lean-Premixed Gas Turbine Combustion". In: *Journal of Engineering for Gas Turbines and Power* 121.2 (1999), pp. 272–280.

[12] Department of Heat and Power Technology. *Design od Industrial Energy Equipment.* Gothenburg, Sweden: Chalmers, 2012.

[13] Fisher controls Internationals. *Control Valve Handbook.* 4th ed. 2005.

[14] Stodola A. *Steam and Gas Turbines, with a supplement on the prospects of the thermal prime mover, Vol. 1.* McGraw-Hill Book Company, Inc., 1927.

[15] Åström K.J and Wittenmark B. *Computer Controlled Systems.* 3rd ed. New Jersey: Prentice Hall inc, 2008.

[16] Hollender M. *Collaborative Process Automation Systems.* ISA, 2010.

[17] Aiordachioaie A. et al. "On the Noise Modelling and Simulation." In: *The Modelica Association Modelica 2006, September 4th-5th.* 2006, pp. 369–375651.

[18] Siemens. *Low Load Operational Flexibility for Siemens G-class Gas Turbines Accessed.* Dec. 2012. URL: `http://www.energy.siemens.com/co/pool/hq/energy-topics/pdfs/en/gas-turbines-power-plants/PowerGen08_LLOPFlexibilityforSiemenGClassGT.pdf`.

[19] Siemens. *Industrial Gas Turbines:The comprehensive product range from 5 to 50 megawatts Accessed.* Dec. 2012. URL: `http://www.energy.siemens.com/us/pool/hq/power-generation/gas-turbines/downloads/Industrial%20Gas%20Turbines/Industrial_Gas_Turbines_EN_new.pdf`.

[20] Sinnott R.K. *Coulson and Richardson's Chemical Engineering Volume 6 - Chemical Engineering Design (4th Edition).* Elsevier, 2005.

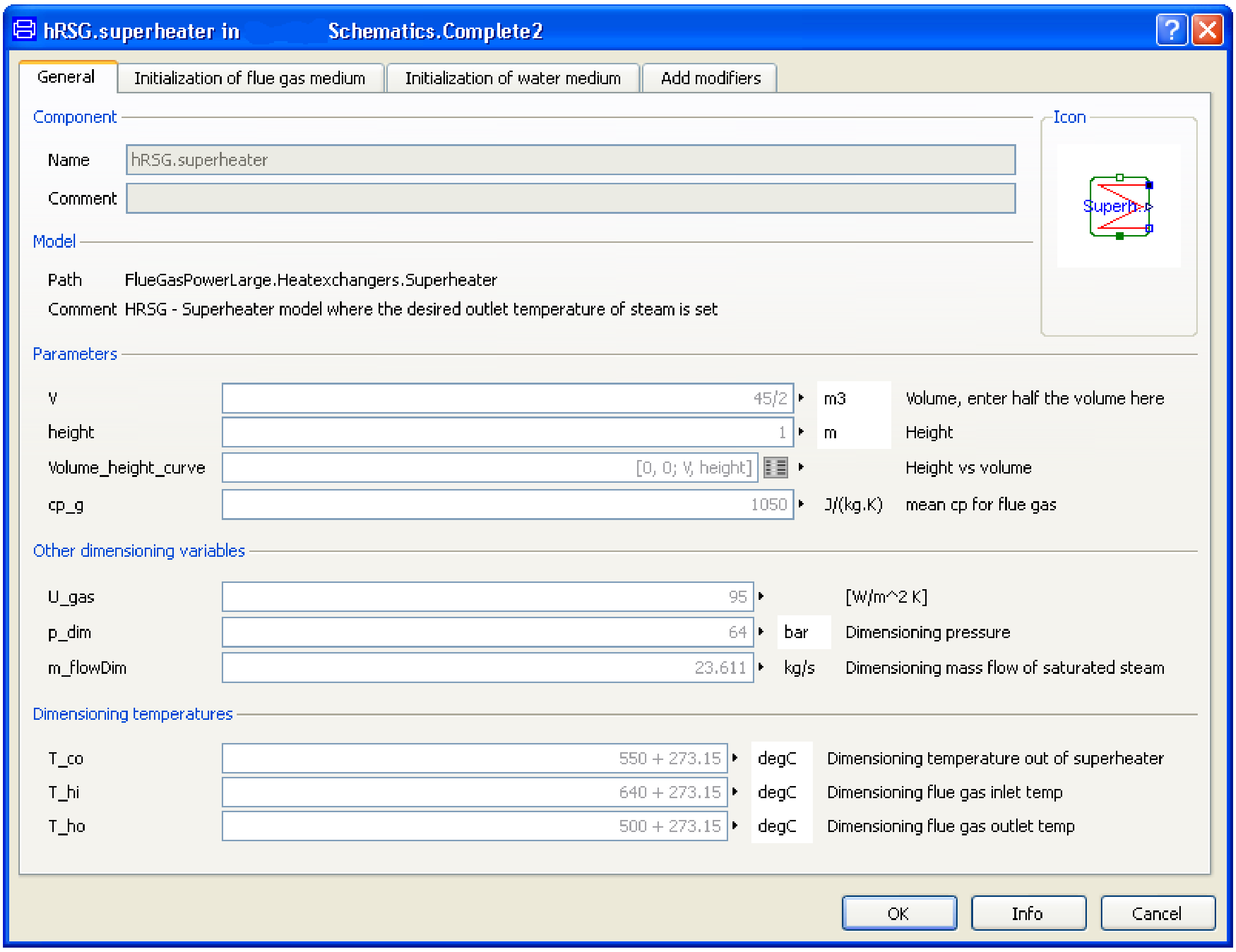
[21] TLV. A Steam Specialist Company. *Pipe Sizing by Pressure Loss for Steam Accessed.* Nov. 2012. URL: `http://www.tlv.com/global/TI/calculator/steam-pipe-sizing-by-pressure-loss.html`.

A Dymola

In this appendix the software Dymola is explained together with the model created to simulate the utility plant.

A.1 Dymola Interface

Dymola has both a graphically and a text based interface. These two are connected to each other and code created in text appears in the graphic interface and vice versa. In Figure A.1.1 the graphic interface is illustrated. The easy change of important parameters is displayed in a lucid way, which makes the program powerful. To define parameters and create equations the text based interface is used (see Figure A.1.2).


hRSG.superheater in Schematics.Complete2
General
Initialization of flue gas medium
Initialization of water medium
Add modifiers
Icon
Superh
OK
Info
Cancel

Component

Name hRSG.superheater

Comment

Model

Path FlueGasPowerLarge.Heatexchangers.Superheater

Comment HRSG - Superheater model where the desired outlet temperature of steam is set

Parameters

V	45/2	m3	Volume, enter half the volume here
height	1	m	Height
Volume_height_curve	[0, 0; V, height]		Height vs volume
cp_g	1050	J/(kg.K)	mean cp for flue gas

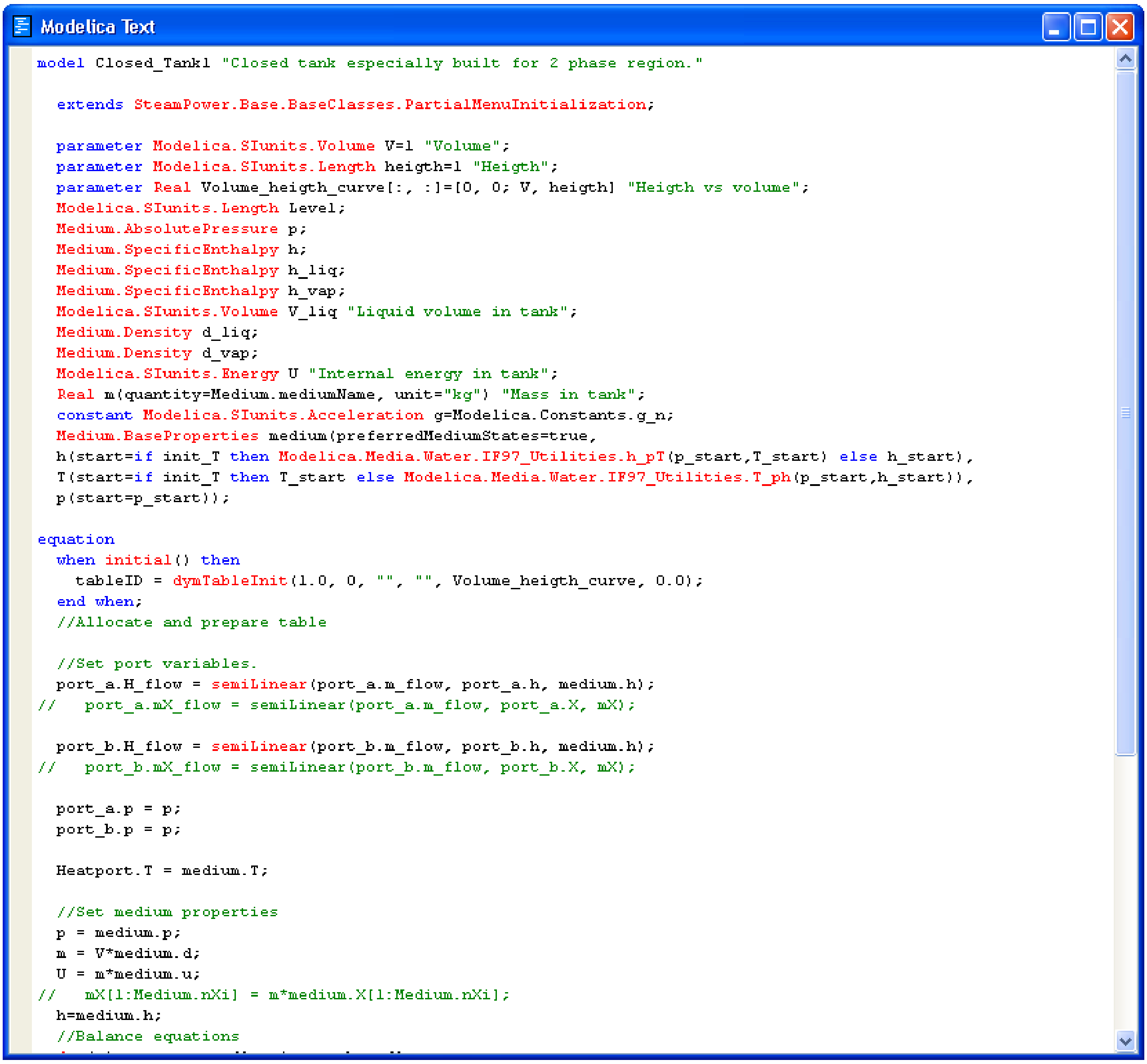
Other dimensioning variables

U_gas	95		[W/m^2 K]
p_dim	64	bar	Dimensioning pressure
m_flowDim	23.611	kg/s	Dimensioning mass flow of saturated steam

Dimensioning temperatures

T_co	550 + 273.15	degC	Dimensioning temperature out of superheater
T_hi	640 + 273.15	degC	Dimensioning flue gas inlet temp
T_ho	500 + 273.15	degC	Dimensioning flue gas outlet temp

Figure A.1.1: *The graphic interface of a super heater in Dymola illustrates the easy change of important parameters.*



```
model Closed_Tank1 "Closed tank especially built for 2 phase region."

  extends SteamPower.Base.BaseClasses.PartialMenuInitialization;

  parameter Modelica.SIunits.Volume V=1 "Volume";
  parameter Modelica.SIunits.Length heigth=1 "Heigth";
  parameter Real Volume_heigth_curve[:, :]=[0, 0; V, heigth] "Heigth vs volume";
  Modelica.SIunits.Length Level;
  Medium.AbsolutePressure p;
  Medium.SpecificEnthalpy h;
  Medium.SpecificEnthalpy h_liq;
  Medium.SpecificEnthalpy h_vap;
  Modelica.SIunits.Volume V_liq "Liquid volume in tank";
  Medium.Density d_liq;
  Medium.Density d_vap;
  Modelica.SIunits.Energy U "Internal energy in tank";
  Real m(quantity=Medium.mediumName, unit="kg") "Mass in tank";
  constant Modelica.SIunits.Acceleration g=Modelica.Constants.g_n;
  Medium.BaseProperties medium(preferredMediumStates=true,
  h(start=if init_T then Modelica.Media.Water.IF97_Utilities.h_pT(p_start,T_start) else h_start),
  T(start=if init_T then T_start else Modelica.Media.Water.IF97_Utilities.T_ph(p_start,h_start)),
  p(start=p_start));

equation
  when initial() then
    tableID = dymTableInit(1.0, 0, "", "", Volume_heigth_curve, 0.0);
  end when;
  //Allocate and prepare table

  //Set port variables.
  port_a.H_flow = semiLinear(port_a.m_flow, port_a.h, medium.h);
//   port_a.mX_flow = semiLinear(port_a.m_flow, port_a.X, mX);

  port_b.H_flow = semiLinear(port_b.m_flow, port_b.h, medium.h);
//   port_b.mX_flow = semiLinear(port_b.m_flow, port_b.X, mX);

  port_a.p = p;
  port_b.p = p;

  Heatport.T = medium.T;

  //Set medium properties
  p = medium.p;
  m = V*medium.d;
  U = m*medium.u;
//   mX[1:Medium.nXi] = m*medium.X[1:Medium.nXi];
  h=medium.h;
  //Balance equations
```

Figure A.1.2: *The text interface of a steam tank in Dymola illustrates how parameters are defined and how equations are written.*

A.2 Gas turbine

The model of the Gas turbine is illustrated in Figure A.2.1. It has two inputs, which are the amount of fuel and the air/fuel ratio. In this model the enthalpy of the gas is set to 0 to illustrate air at 25 °C. The different blocks are also models but in a lower level, where:

- The Fuel/air block calculates the amount of air necessary.
- AirInput block creates a flue gas stream including air.
- The combustion unit block where the air stream and FuelInput translates into flu gas.
- The green square is a connector, which makes the model easy connectable with other models that has the same connector. It includes the properties of the flue gas: composition, mass flow and enthalpy for the mixed gas.

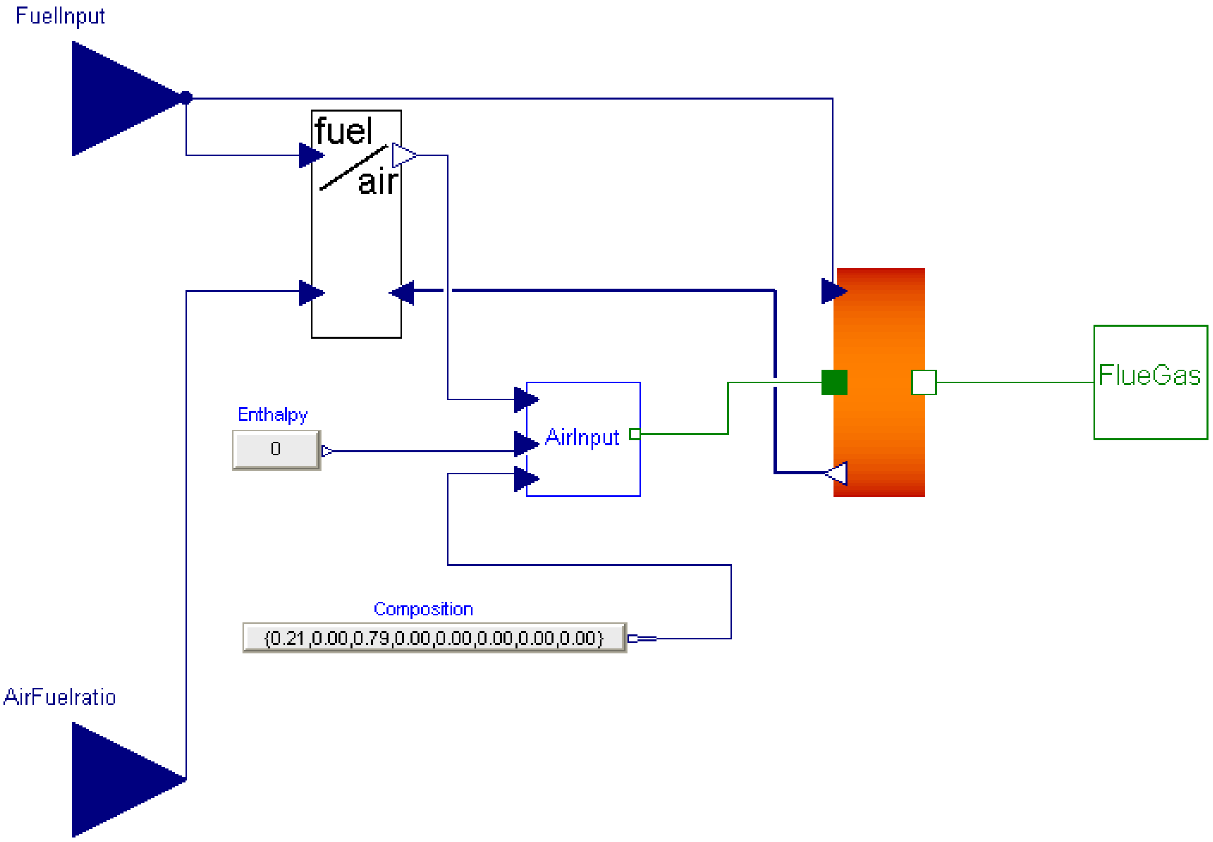


Figure A.2.1: *Dymola model of the gas turbine.*

A.3 Post Combustion

The Post combustion model has the same combustion chamber as the gas turbine, which illustrates the reuse of models. The connectors make the model easy to connect to both the gas turbine and the HRSG (see Figure A.3.1). In the figure, the dynamics of fuel input is created using mathematical functions. The first is a transfer function to make the response slower. The second is a limiter to limit the amount of maximum fuel input. The last is a time delay function.

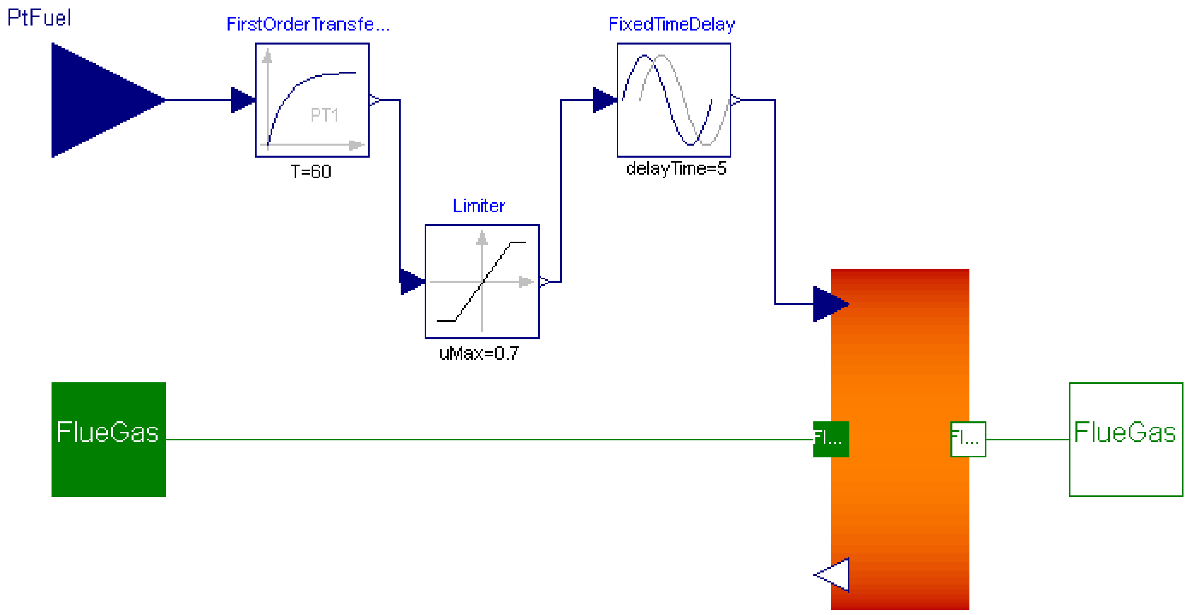


Figure A.3.1: *Dymola model of the post combustion.*

A.4 HRSG

In Figure A.4.1 the heat transfer between flue gas and water is presented. The connector from the flue gas is connected with the post combustion model and on the water side there are one connector for feed water and one for steam. The three heat exchangers are also displayed. The time dependent dynamics of the HRSG is created using flue gas tanks that result in time delay for transfer of heat in the model.

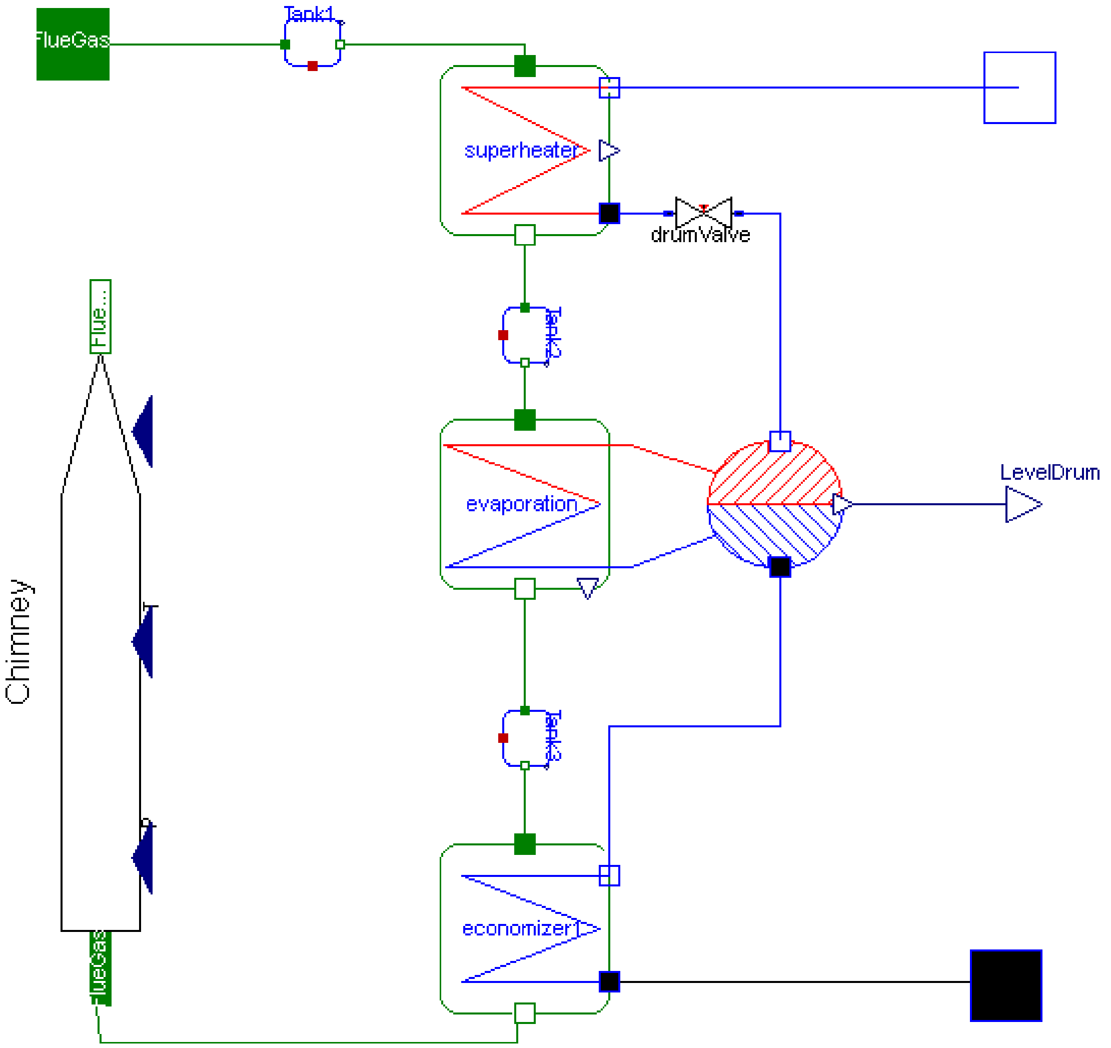


Figure A.4.1: *Dymola model of the HRSG.*

A.5 Steam Network

The Steam network is displayed in Figure A.5.1. The different models included are valves, turbines and the dynamics for pipe delay. The bound blocks are sinks for steam.

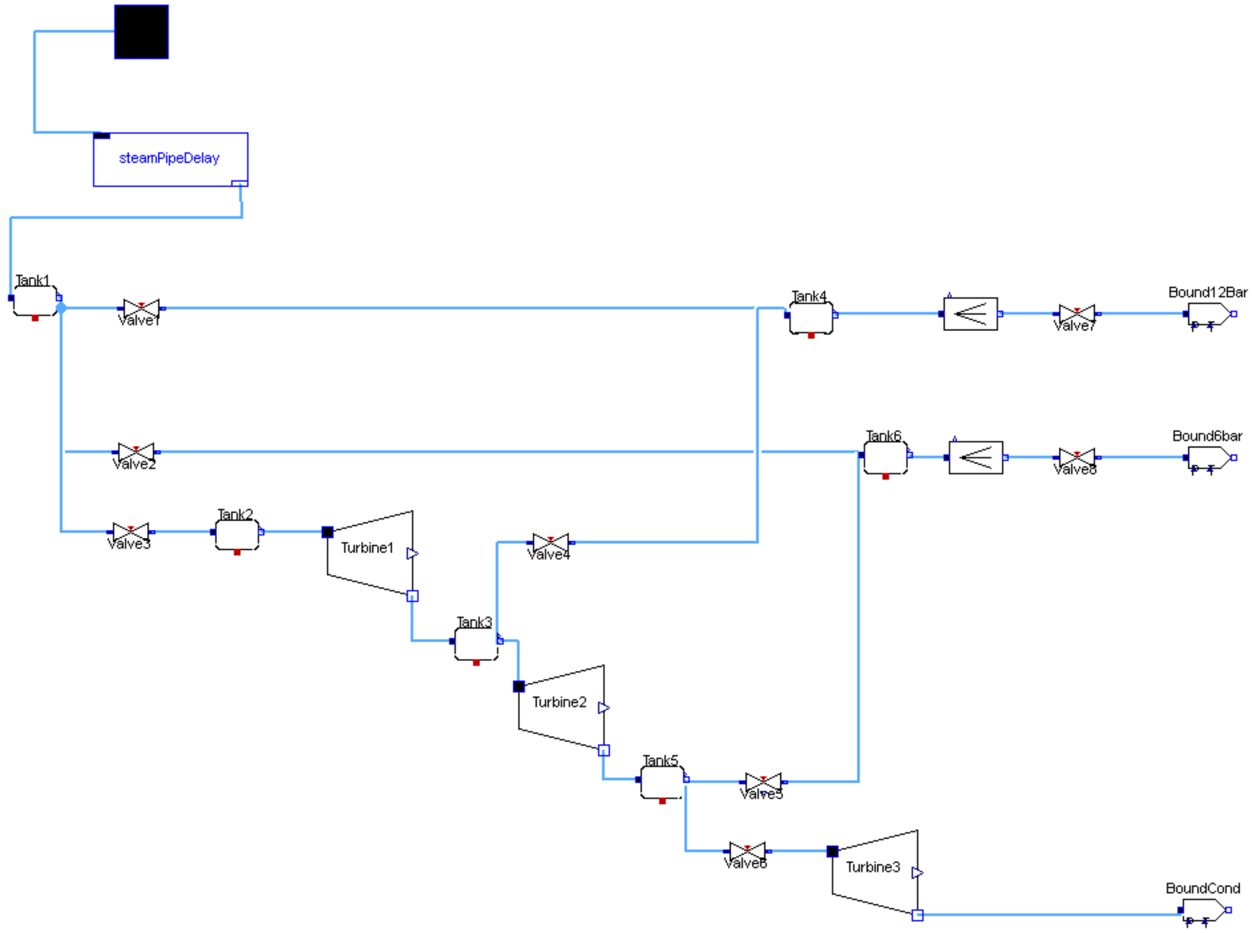


Figure A.5.1: *Dymola model of the steam network.*

A.6 Complete Model

The complete model with controllers is displayed in Figure A.6.1. All blocks are here connected. The square blue boxes represent controllers. Circle blokes represent measurement points in the steam network.

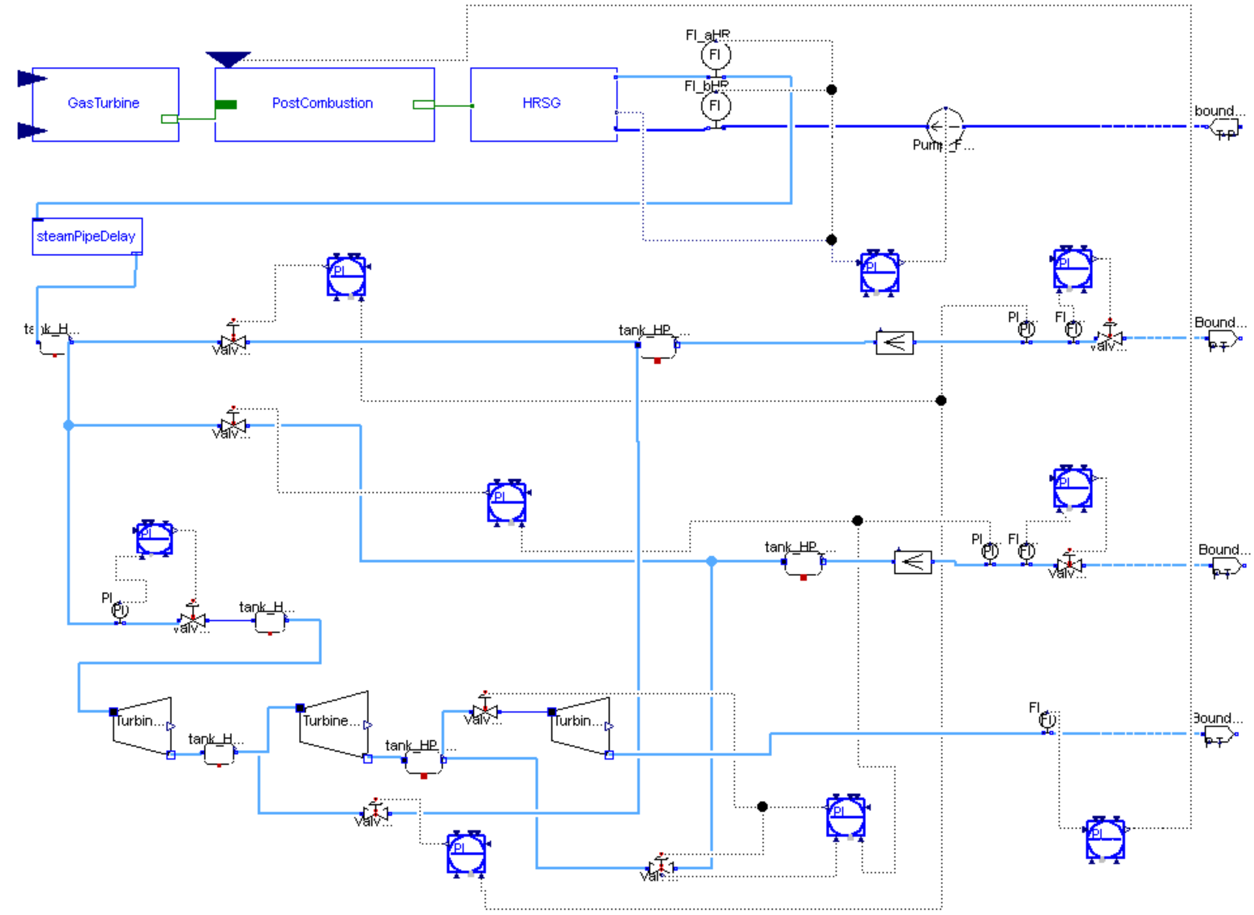


Figure A.6.1: *The final Dymola model.*

B GCC

This appendix explain the creation of a GCC with help of Matlab. Required package is Xsteam which provide steam properties.

```
%% Given data

P_MP=64;                          % [Bar] Pressure medium pressure steam
T_MPsat=Xsteam('Tsat_p',P_MP);    % [C] SatTemperature medium pressure steam
T_MP=525;                         % [C] Temperature medium pressure steam
    %superheated MP
P_LP= 63;                         % [Bar] Pressure low pressure steam
T_LPsat=Xsteam('Tsat_p',P_LP);    % [C] SatTemperature low pressure steam
T_LP=T_LPsat;                     % [C] Temperature low pressure steam
m_MP=85e3/3600;                   % [kg s^-1] Flow rate medium pressure steam
m_LP=0e3/3600;                    % [kg s^-1] Flow rate low pressure steam
T_FW=70;                          % [C] Temperature feed water

% Superheater
T_bSUP=T_MPsat; % [C] Temperature water before superheater
T_aSUP=T_MP;    % [C] Temperature water after superheater
Q_SUP=m_MP*(Xsteam('h_pT',P_MP,T_aSUP)-Xsteam('hV_p',P_MP)); % [kJ s^-1]
    %Heat load superheater

% Evaporator for medium pressure steam
T_bEVPMP=T_bSUP ;   % [C] Temperature water before evaporator for medium
    %pressure steam
T_aEVPMP=T_bEVPMP;  % [C] Temperature water after evaporator for medium
    %pressure steam
Q_EVPMP=m_MP*(Xsteam('hV_p',P_MP)-Xsteam('hL_p',P_MP)); % [kJ s^-1]
    %Heat load evaporator

% Economicer for medium pressure steam
T_aECOMP=T_bEVPMP;  % [C] Temperature water after economicer for medium
    %pressure steam
T_bECOMP=T_LP;      % [C] Temperature water before economicer for medium
    %pressure steam
Q_ECOMP=m_MP*(Xsteam('hL_p',P_MP)- Xsteam('h_pT',P_MP,T_bECOMP)); % [kJ s^-1]
    %Heat load economicer

% Evaporator for low pressure steam
T_bEVPLP=T_LP;      % [C] Temperature water before evaporator for low pressure
    %steam
T_aEVPLP=T_LP;      % [C] Temperature water after evaporator for low pressure
    %steam
Q_EVPLP=m_LP*(Xsteam('hV_p',P_LP)- Xsteam('hL_p',P_LP)); % [kJ s^-1] Heat
    %load evaporator

% Economicer for both pressure levels where the pressure i separated
T_aECOLPMP=T_LP;    % [C] Temperature water after economicer for low pressure
%steam
T_bECOLPMP=T_FW;    % [C] Temperature water before economicer for feed water
Q_ECOLPMP=m_LP*(Xsteam('hL_p',P_LP)- Xsteam('h_pT',P_LP,T_bECOLPMP))+ ...
    m_MP*(Xsteam('h_pT',P_MP,T_aECOLPMP)- Xsteam('h_pT',P_MP,T_bECOLPMP)) ;
    % [kJ s^-1] Heat load economicer for both pressure levels

% Total Process demand
Q_process=Q_ECOLPMP+Q_EVPLP+Q_ECOMP+Q_EVPMP+Q_SUP; % [kJ s^-1] Heat
    %load process total

%% Fluegas gasturbine

cp_FG=1.05;                      % [kJ kg^-1 K^-1] Heat capasity for fluegas
m_FG=[130 ];                     % [kg/s] Mass flow fluegas for gasturbines
T_exhNoSupfire=[540 ];           % [C] Exhaust temperature for gasturbines
T_stackNoSupfire=T_exhNoSupfire-Q_process./(m_FG.*cp_FG);
T_bECO2NoSupfire=T_stackNoSupfire+ (Q_ECOLPMP+Q_EVPLP+Q_ECOMP)./(m_FG.*cp_FG);
```

```
T_bECONoSupfire=T_stackNoSupfire+ Q_ECOLPMP./(m_FG.*cp_FG);
% Checking deltatmin at economicer inlet
for i = 1:1
if T_bECONoSupfire(i) <= T_aECOLPMP+10
    T_diff(i)=T_aECOLPMP+10-T_bECONoSupfire(i);             % Check difference
    %in celcius
    T_FGbECOLPMP(i)=T_bECONoSupfire(i)+T_diff(i);           % New flugas
    %temperature at economicer inlet
    T_Stack(i)=T_FGbECOLPMP(i)- Q_ECOLPMP/(m_FG(i)*cp_FG);  % Calculate new
    %stack temperaure
    T_exh(i)=T_Stack(i) +Q_process/(m_FG(i)*cp_FG);         % Calculate new
    %flugas outlet temperature
    q_Supfire(i)=m_FG(i)*cp_FG*(T_Stack(i)-T_stackNoSupfire(i)); % Calculate
    %suplementary fireing

else
    % needed to remain dimensions
    T_diff(i)=T_aECOLPMP+10-T_bECONoSupfire(i);
    T_FGbECOLPMP(i)=T_exhNoSupfire(i);
    T_Stack(i)=T_stackNoSupfire(i);
    T_exh(i)=T_exhNoSupfire(i);
    q_Supfire(i)=0;
end
end

%checking deltatmin at econimer outlet
for i = 1:1
if T_Stack(i) <= T_bECOLPMP+10
    T_diff2(i)=T_bECOLPMP+10-T_Stack(i);        % Check difference
    T_Stack(i)=T_Stack(i)+T_diff2(i);           % New stack emperature
    T_exh(i)=T_Stack(i) +Q_process/(m_FG(i)*cp_FG); % New flugas outlet
    %temperature
    q_Supfire(i)=m_FG(i)*cp_FG*(T_Stack(i)-T_stackNoSupfire(i)); % suplementary
    %fireing
    T_FGbECOLPMP(i)= Q_ECOLPMP/(m_FG(i)*cp_FG) +T_Stack(i);  % new
    %temperature at econimicer
else
    T_diff2(i)=T_bECOLPMP+10-T_Stack(i);
    T_Stack(i)=T_Stack(i);
    T_exh(i)=T_exh(i);
    q_Supfire(i)=q_Supfire(i);
    T_FGbECOLPMP(i)=T_FGbECOLPMP(i);
end
end
%checking deltatmin at econimer Mp outlet

for i = 1:1
if T_bECO2NoSupfire(i) <= T_aECOMP+10
    T_diff3(i)=T_aECOMP+10-T_bECO2NoSupfire(i);    % Check difference in
    %celcius
    T_FGbECOMP(i)=T_bECO2NoSupfire(i)+T_diff3(i);  % New flugas temperature
    %at economicer inlet
    T_Stack(i)=T_FGbECOMP(i)- (Q_ECOLPMP+Q_EVPLP+Q_ECOMP)/(m_FG(i)*cp_FG);
    % Calculate new stack temperaure
    T_exh(i)=T_Stack(i) +Q_process/(m_FG(i)*cp_FG);%Calculate new flugas
    %outlet temperature
    q_Supfire(i)=m_FG(i)*cp_FG*(T_Stack(i)-T_stackNoSupfire(i)); % Calculate
    %suplementary fireing

else
    % needed to remain dimensions
    T_diff3(i)=T_aECOMP+10-T_bECO2NoSupfire(i);
    T_FGbECOMP(i)=T_exhNoSupfire(i);
    T_Stack(i)=T_stackNoSupfire(i);
    T_exh(i)=T_exhNoSupfire(i);
    q_Supfire(i)=0;
end
end
% Obs check echust maximum temperature
T_gt=Q_process./(m_FG.*cp_FG)+T_Stack;   % [C] Flugas temperature slope
```

```
%% Ploting

figure(1)
plot([0 Q_ECOLPMP Q_ECOLPMP Q_ECOLPMP+Q_EVPLP Q_ECOLPMP+Q_EVPLP ...
    Q_ECOLPMP+Q_EVPLP+Q_ECOMP Q_ECOLPMP+Q_EVPLP+Q_ECOMP ...
    Q_ECOLPMP+Q_EVPLP+Q_ECOMP+Q_EVPMP Q_ECOLPMP+Q_EVPLP+Q_ECOMP+Q_EVPMP ...
    Q_ECOLPMP+Q_EVPLP+Q_ECOMP+Q_EVPMP+Q_SUP ] ...
    ,[T_bECOLPMP T_aECOLPMP T_bEVPLP T_aEVPLP T_bECOMP T_aECOMP T_bEVPMP ...
    T_aEVPMP T_bSUP T_aSUP])
hold on
plot([0  Q_process ],[T_Stack(1) T_gt(1)], 'r')
legend('HRSG','Gas Turbine');
xlabel('Q [kW]')
ylabel('T [^oC]')
```