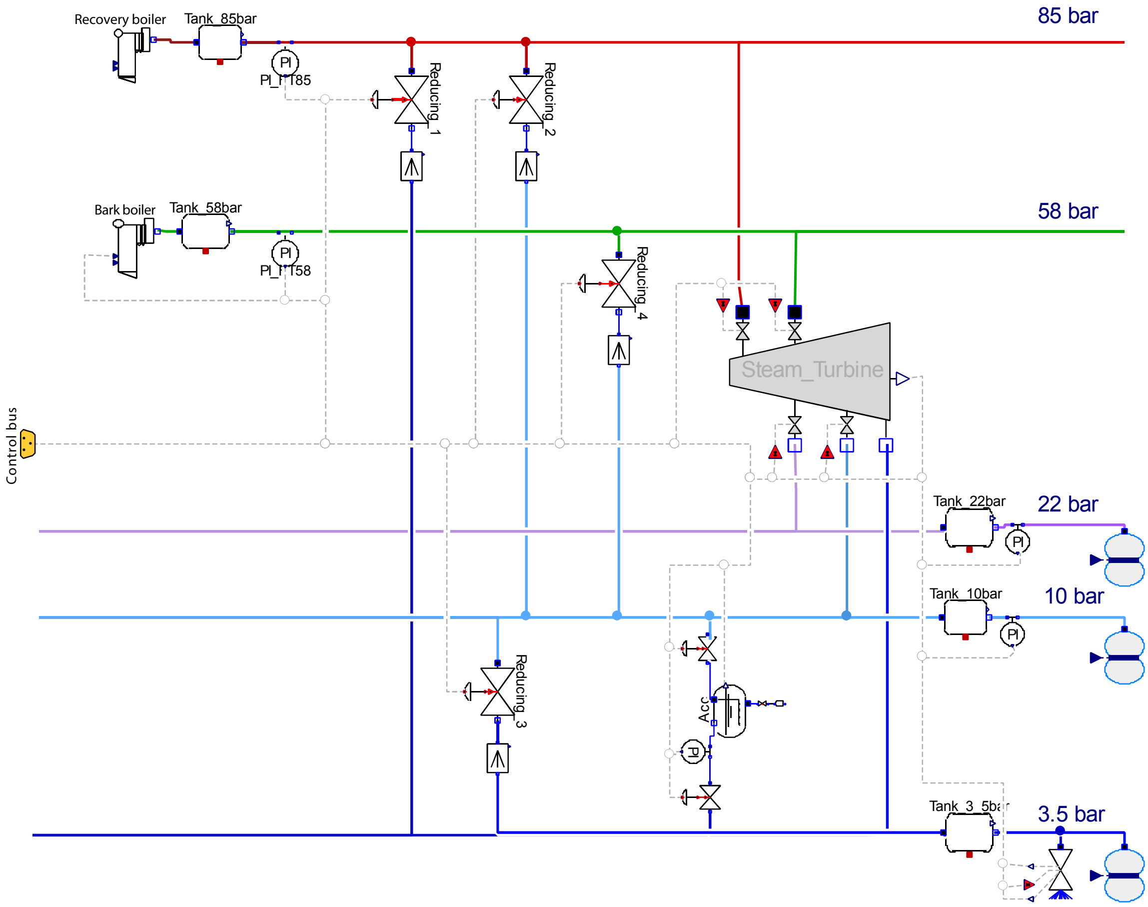


Modeling and Model Predictive Control of a Multilevel Steam Network

Master's thesis in Systems, Control and Mechatronics

DANIEL STENBACK

Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Master's thesis EX042/2013

MASTER'S THESIS IN SYSTEMS, CONTROL AND MECHATRONICS

Modeling and Model Predictive Control of a Multilevel Steam Network

DANIEL STENBACK

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

Göteborg, Sweden 2013

Modeling and Model Predictive Control of a Multilevel Steam Network
DANIEL STENBACK

Master's thesis EX042/2013
ISSN 1652-8557
Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Graphical representation of a Multilevel Steam Network Modelica model. For more information on the model see chapter 3 on page 7.

Chalmers Reproservice
Göteborg, Sweden 2013

Modeling and Model Predictive Control of a Multilevel Steam Network
Master's thesis in Systems, Control and Mechatronics
DANIEL STENBACK
Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology

Abstract

Decreasing environmental impact and cutting energy expenses are major reasons why the pulp and paper industry are interested in improving the control behaviour of their processes. The part of the process transferring most of the energy in a pulp and paper plant is the multilevel steam network. A poorly configured control system of multilevel steam network can lead to energy losses in form of venting excessive steam. A preferred approach to venting excessive steam is to decrease the production of steam if a decreased demand on steam can be predicted. If information regarding the upcoming steam demand is available, then this could be considered as a measurable disturbance for which a feedforward controller can be used to improve the energy efficiency of the process.

In this thesis a Modelica model of a multilevel steam network has been developed. This model is used to evaluate how Model Predictive Control (MPC) could complement or replace traditional parts of the control systems in a multilevel steam network and see how it could improve the energy efficiency by making use of the steam demand forecast.

Simulations have been made with the MPC Modelica implementation and the multilevel steam network in Dymola. The preventative actions that the MPC is taking when the feedforward feature is implemented shows that the MPC technique could be suitable for decreasing the production of steam when predicted disturbances are available. This decreases the amount of excessive heat that needs to be released and thus has the potential to both decrease the environmental impact and to cut costs.

Keywords: model predictive control, multilevel steam network, Modelica

Acknowledgements

I would like to express my gratitude to all the people who have been involved in this master thesis project:

My supervisor at Solvina AB, M.Sc. Carl W. Ressel for his support and suggestions important for the progress of this master's thesis work.

My examiner at Chalmers University of Technology, Assoc. Prof. Knut Åkesson for guidance and advice in key situations of the project.

The people at Solvina AB for making my time there a learning experience.

My parents for their support and encouragement throughout my studies and certainly this master's thesis work.

My friend Sten, for generously sharing of his versatile insights, cooking skills and unceasing support.

Contents

1 Introduction

Today's rising energy costs makes it profitable for industries to invest in projects and actions improving the energy efficiency. The pulp and paper industry are characterized by being one of the most energy consuming industries. It uses wood as raw material to turn it into pulp, which in turn is used for producing pulp products, such as paper. There are two ways to turn wood into pulp, either mechanically by grinding or chemically by using white liquor to break the bonds between lignin and cellulose.

The pulp and paper plant studied in this thesis is using the chemical method. The residual from this process is black liquor and in order to recover the chemicals used for producing white liquor, the black liquor is burned. The heat from this combustion is used for producing steam to supply the multilevel steam network with. Increasing the reuse of steam from the steam network could improve the energy efficiency and could lead to increased profits, as well as decreased environmental impact.

To optimize the use of energy in the steam and to minimize the difference between produced and consumed steam, there are some proven methods. A fully automatic method is to implement a controller with a plant wide perspective. This controller could control the system's use of steam turbines and boilers in a preventative manner thanks to forecasts of future steam demand and prizes of boiler fuel and electricity. A semi-automatic method is to install utilities for the operator. These utilities could predict the future state of the system and suggest actions to be taken by the operator. In this thesis we have been working on the first approach to make an existing steam network more energy efficient.

The model predictive control (MPC) technique has been chosen because for several reasons. MPC-applications is available in modern Distributed Control Systems (DCS-equipment). As claimed in [Jal06] MPC was successfully implemented in pulp and paper plants. Also there are successful results published in [Maj05] on using model predictive control for keeping the pressure steady in a multilevel steam network. In [Mac02] the author states three major reasons why MPC is successful in industrial control engineering; (i) it handles multivariable control problems naturally; (ii) it can take actuators limitations into account; and (iii) it allows operations close to constraints, such as limitations of the equipment, which usually is where it is most profitable to operate.

A time and cost effective method to design a controller is by modelling and simulation. The modelling language Modelica allows you to write equations describing relations between physical quantities in your process, without having to define the causality. The major drawback from this is the sensitivity for good start guesses when initiating the algebraic differential equation solver.

Modelica is an object oriented language. The models are compared with classes in a regular object oriented programming language and one class can include other subclasses or submodels. This opens for the use of model libraries and Modelica has an extensive integrated standard library.

Dymola is a graphical user interface and an interactive development environment for Modelica. In Dymola the Modelica models are presented graphically and if the model is well shaped the graphics can be helpful when navigating between the submodels. This puts some requirements on having some planning when building the model. The more intuitive the composition of the model is, the easier it is to understand and to adapt the model.

In this thesis we have developed a Modelica model in Dymola, of the dynamics in a multilevel steam network. The model has been used in the designing of the controller and for evaluating the performance.

1.1 Objective

The primary goal of this thesis is to investigate how a model predictive control design could optimize the balance between produced and consumed steam. The secondary goal is to search for a solution which will maximize the power output of the steam turbines. This requires a simulation model in order to be able to test and tune control parameters.

The simulation model should describe the essential dynamics of the multilevel steam network and the including equipment. The properties and physical quantities need to be basically alike. In addition, the existing control structure needs to be described in the simulation model in order to have the real control behaviour. The disturbance forecast needs to be modelled so that it can be used in the control that is going to be designed.

The aim of the project can be expressed as the two main parts; create a Modelica model of the steam system; and design the optimizing controller. This will also include testing and evaluation of the designed controller and validation of the dynamics in the model.

The work of designing the controller can be divided into the following parts. First of all, the finding of what measurable output is going to be the controlled output for the optimizing controller. This part also includes finding the variables to be the control signals. Possible variables which could constitute the control signals are not only direct inputs for actuators, but also set-points for the inner control loops. Another possible variable to control could be the limits on the inner control loops' control signals.

Second part of designing the controller is to make it use the available information of the upcoming needs of steam.

1.2 Purpose

The results of this thesis could serve as an indication of how complementing control strategies could help to improve the energy efficiency of a multilevel steam network at a pulp and paper plant. The essentials of the existing control system will remain in order to meet the commonly occurring reluctance to make major changes which could be costly. In addition, the existing control system meets the safety regulations which advantageously are worth to keep. If successful, the industry could increase the profit and decrease its environmental impact.

1.3 Limitations

The project is an evaluation of the concept of using model predictive control techniques to take forecast into account and will not serve as a complete investigation with a result ready to be implemented in an existing plant.

The model of the system will only describe characteristics crucial for the overall process controller. In this case the main focus has been to attain realistic steam pressures and flows. Other dynamics of the parts in the system of minor importance for the optimizing controller will not be modelled. The scenarios that the model is acquired to describe are major transients in pressures and flows during regular operation. The model will not describe characteristics of the steam networks during start up or similar special cases.

1.4 Contributions

The first contribution of this thesis is development of a Modelica model for a multi-level steam network. This was made possible by using existing Modelica models from the model library Steam Power developed by Solvina. With this model it is possible to simulate pressure disturbances, such as, variations of steam demand and emergency stop of the steam turbine. It is also possible to simulate the forecast of upcoming steam demand.

The second contribution is design and implementation of a MPC in Modelica. The MPC uses a prediction of the consumed steam to compensate for the disturbance before it occurs. The major aim of the approach is to improve the overall energy efficiency of the steam network.

The third contribution is an evaluation of the proposed approach of the simulated model, including a discussion of strength and weaknesses.

2 Plant Description

The pulp and paper industry is the branch of the forest industry that uses wood as raw material for producing pulp. Pulp is in turn used for producing paper, board and other cellulose-based products. Often both pulp and paper are obtained at the same mill but could be in separated facilities.

The plant used as a reference for this project is a pulp and paper plant located in Sweden. The plant produces around 396 000 air dry ton of pulp per annum (ADt/a). It has also a sawmill attached to it. There are plans to build an extension to the saw mill which would result in larger demands of steam. Today the plant produces around 50 MW of electricity and by expanding the saw mill the plan is to maximize the energy output from the steam turbine to 62 MW.

The steam drained from the steam turbine is in turn used in the manufacturing process. Stable pressure levels of the steam are vital for the paper quality.

2.1 Steam producing units

There are basically two ways of producing pulp, either mechanically by grinding or chemically. The bonds between lignin and cellulose in small wood logs are separated by using white liquor, which produces black liquor as residue. The chemicals for white liquor are recovered by burning the black liquor in recovery boilers. Heat from the combustion is used to produce superheated high pressured steam for supplying the multilevel steam network. Steam can also be produced by the bark boilers, which burns bark or oil.

With superheated steam means that the steam is heated up to a temperature slightly above the boiling point of the corresponding saturated water. The steam is then distributed at various pressures to different processes within the plant.

2.2 Multilevel steam network

Normally a pulp and paper mill has divided the steam network into three pressure level categories, high pressure (HP), medium pressure (MP) and low pressure (LP). These levels are often referred to as their nominal pressure given in bar(g). This unit is in gauge pressure, i.e. pressure in bars above atmospheric pressure, hereinafter written as only bar. This plant has five levels of pressure headers - two HP at 85 and 58 bar, two MP at 22 and 10 bar, and one LP at 3.5 bar. A schematic of the multilevel steam network can be seen in Figure 3.2. The pressure levels differ slightly between plants but are essentially the same.

The produced pressurized superheated steam needs to be reduced in pressure and temperature before it is suitable for the steam consuming units in the plant. This can be done in two ways; either efficiently through a steam turbine, or inefficiently through pressure-reducing valves and steam coolers, also known as desuperheaters. These possible paths that the steam can be distributed via are illustrated in Figure 2.1.

Between two of the pressure headers in the network it is useful to have an accumulator. The accumulator is basically a pressurized tank half filled with water. Typically the inlet of the accumulator is connected to the MP header and the outlet is connected to the LP header. By either loading or unloading the accumulator, pressure deviations can be avoided efficiently. For example when the load decreases on the MP header some of the inflow steam needs to be redirected to avoid overpressure. If the excess steam is used to load the accumulator it is possible to prevent the pressure disturbance from propagating to the other pressure headers.

Another reason to start charging the accumulator could be when the steam demand decreases on the LP header. To maintain the pressure on the MP header while loading the accumulator the flow of steam from the drain valve on the turbine will be increased. As a result, less steam will be drained into the LP header from the steam turbine and the pressure disturbance can be rejected.

Using steam turbines is also a way of securing the power supply for the paper mill. In case of disturbances on the external power grid, it may in some cases be favourable to disconnect the external power supply and only use self-produced electricity, i.e. island mode operation.

2.3 Steam consuming units

The produced steam is used in several different facilities such as, drying equipment, paper machines, and evaporators, among others. Steam consumption is commonly measured in tons per hour or kilo grams per second. The consumption rate for different facilities varies in magnitude over time. These changes interferes the preferred pressure level on each header in the system. Because these changes are intermittent they constitute disturbances for the pressure controller. Major changes in load on MP header are forecasted 40 minutes in advance and the forecast refines gradually until the disturbance occurs.

As a last resort for the steam there are so called condensers and blow out valves connected to the LP header. The dump condensers recycle water from the vapour which is more energy-saving than the blow out valve, which releases steam into the atmosphere.

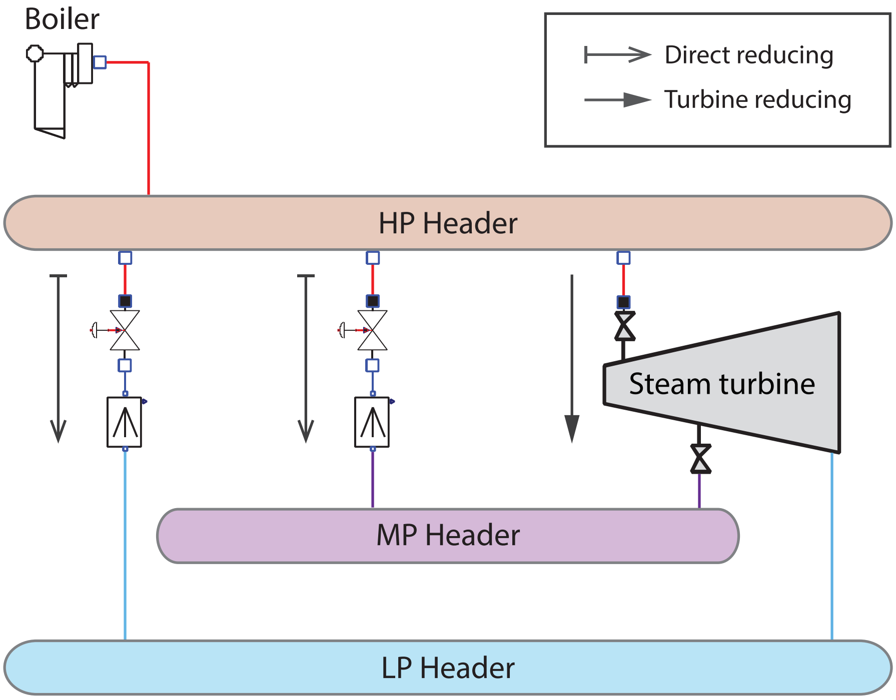


Figure 2.1: *Illustration on the different paths the steam can take.*

2.4 Steam pressure control

The pressure of the HP header is maintained by inflow of steam from the boiler output. While the pressure of the other pressure headers (MP and LP) are retained by steam from the HP header via let-down valves. In order to handle rapid changes in pressure on the HP header, pressure controllers are able to open the bypass valves for direct reduction of steam to the MP and LP headers. These valves could also be opened by the pressure controllers on the receiving headers, when there is a need for more steam downstream. If this is happening while the pressure runs low in the HP header, there are control loops for closing the let-down valves by overriding the other control signal. In this way the prioritization is realized and gives the HP header the highest priority. If there is a shortage of steam in the system, the pressure in the other headers will be lost prior to the HP header, starting with the LP header.

There are multiple control loops getting their feedback from the same pressure indicator on each pressure header. This feedback signal are compared with a reference signal individually biased from the set-point specified for each header, so that the controllers will start to operate the valves at different values of the feedback. During the time when the feedback value is far from the controller's reference signal the control signal will be saturated at either fully closed or fully opened valve, depending on the sign of the control error.

The Figure 2.2 is an illustrative example of how the signal selection is realized and shows the routing of the signals controlling the direct reducing valve between the HP and LP headers. In the Figure there are three proportional-integral controllers. Two controllers gets their feedback from the HP header and their reference signal originates from the same set-point (SP_{HP}) but are kept apart by their own bias terms ($Bias_{high}$) and ($Bias_{low}$). The selecting of which control signal from these three controllers that will control the valve is done by the MAX and MIN selectors.

The drawback of this control strategy is the lack of plant wide perspective of the steam balance which opens for the risk that some controllers are working against each other. How the control loops are made active at separate pressure intervals are discussed further in section 3.2.4.

2.5 Potential areas for improvements

Data gathered by Solvina, from the reference pulp and paper plant, shows where the system has potential to improve the energy efficiency. Measurements have been done under two different conditions, summer and winter. During the summer part of the year there is less demand of steam but because of the need of recovering chemicals the recovery boiler has to be running at full capacity. This is causing the steam balance to be positive. Although, the measurement data shows a negative steam balance because there is too much steam being vented.

Data from measurements done a winter day also shows that there is steam being dumped even during periods when more steam is consumed than produced. During the winter there is a higher demand for steam and it is also more profitable to sell electricity.

If there is a considerable negative steam balance for a longer time period, it will cause the steam turbine power output to decrease resulting in economic losses due to reduced sales of electricity. This is because steam will be directly reduced through reduction valves instead of being tapped into the steam turbine.

The direct reduction gives more desuperheated steam than the turbine would give from the same amount of superheated steam. This is because of the steam cooler (also known as desuperheater), which is mounted right after the pressure reducing valve. It ensures that the steam has the desired temperature by injecting a fine mist of water into the steam flow. Superheated steam then gives up heat to the water mist, which causes the temperature to drop and the amount of steam to increase.

The forecast of the major irregular changes in steam demand is not being used at all by the controllers. If the forecast would have been taken into account when controlling the multilevel steam network it could possibly prepare the system for upcoming changes in load. These preparations could be to adjust the reference signal for the preferred pressure in the headers such that the controllers create a small buffer. It could also be to prepare the pressure in the accumulator, to adjust the boiler fuel or a combination of those. These actions should be taken

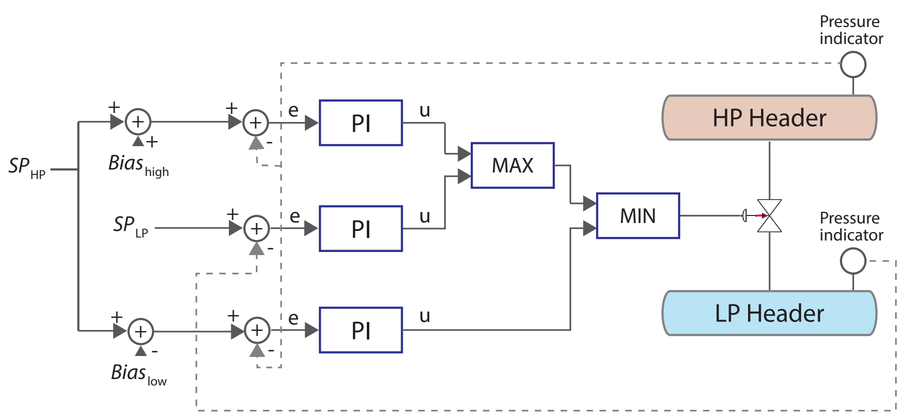


Figure 2.2: *Illustration of the choosing between control signals for a single actuator.*

Table 2.1: Potential aspects for energy efficiency improvements and its prioritisation.

Prio	Potential aspects of improvements
1	available information of upcoming steam demand is not used for preventative control behaviour
2	local control loops not considering overall steam balance

in order to handle fluctuations in load and prevent it from causing larger pressure deviations in the headers which could lead to unwanted venting of steam through the blow-out valve.

If the plan is to save steam, it is better to look for how to reduce the boiler fuel feed that is controllable in the first place. The uncontrollable fuel feed is the chemicals for recovery, which is needed regardless of the steam demand. Searching instead for a way how to decrease the need of low pressure steam could be ineffective because there is a risk that the turbine will produce the low pressured steam anyway because of the demands on producing electricity. The primarily aspects of the control behaviour in the system that potentially could be improved are summarised in Table 2.1.

3 Modelling the multilevel steam network

The process model is developed in the multi engineering software tool Dymola from Dassault Systems. It is a graphical user interface for the object oriented modelling language Modelica. In Dymola the Modelica models are visualized and can be edited both graphically and by writing Modelica code. It is also possible to translate and compile the Modelica model into an executable to be used for simulations. Dymola also provide an interface for plotting simulation results. Screenshots of Dymola's different views can be seen in Appendix A.1 at page 31.

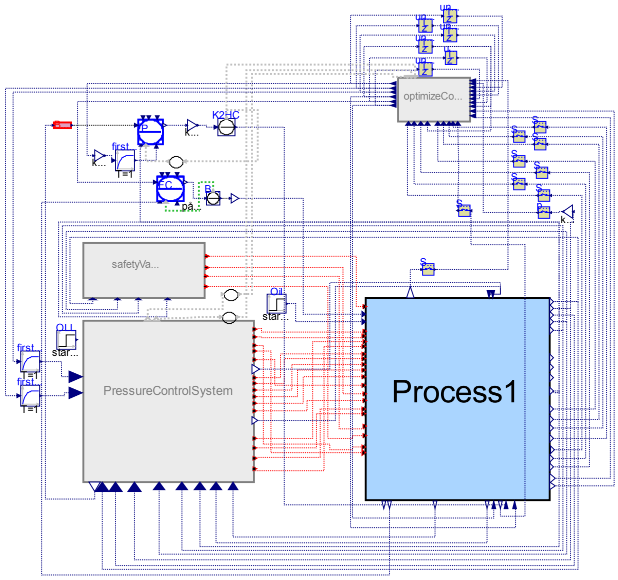
3.1 Modelica models

An advantage of creating the models as Modelica objects is the possibility to reuse them in other projects. As a consequence, it is possible to create user defined model libraries. Building Modelica models with components from existing model libraries speeds up the work.

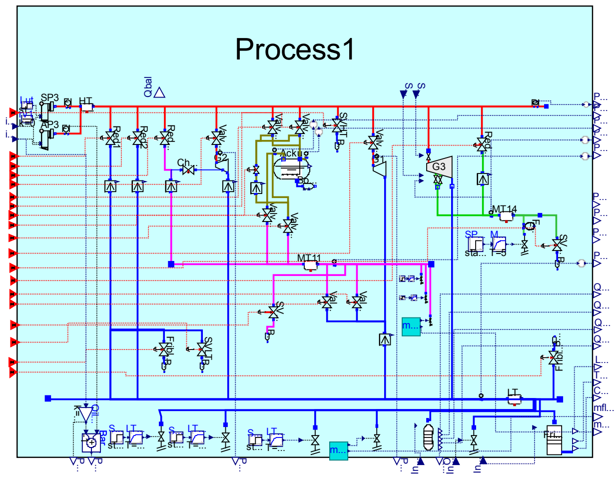
One Modelica library is the Steam Power package developed at the technical consultant company Solvina AB in Sweden. The package has an extensive selection of modelled components and associated accessories in a steam network. Another model library of parts in a steam network is the commercial *Thermal Power Library* from Modelon. This thesis makes use of Solvina's Steam Power package for developing the simulation model.

Dymola differs from many other data flow based modelling tools, such as MATLAB/Simulink, because the causality does not have to be defined. The different solvers used by Dymola when simulating will choose the causality, i.e. which side of the equality sign will define the other one, in an equation. This is depending on the evaluation order, also chosen by the solver.

There are many different integration algorithms used to solve differential equations provided with Dymola. The main differences between them are variable- or fixed-step size. Methods used by the solvers also differ and could, for example, be the Adams method or Runge-Kutta, according to[Dym]. If a model contain many events it is recommended to use a single-step



(a) *The process model are connected to the control models.*



(b) *All the red triangles to the left and all the white triangles to right are connectors for the process variables.*

Figure 3.1: *The use of non-expandable connectors can make the structure of the Modelica model complex.*

algorithm, such as *Radau IIa* rather than the multi-step integration algorithm *Dassl*. For further details on integration algorithms, see [Dym].

The Modelica languages are object oriented and it is possible to create base classes and functions, i.e. methods, as in standard object oriented programming languages. Both the base class can be assigned properties as well as each instance of the class. These objects are exchanging information through elements of the type called connector.

There are also expandable connectors called control buses. These connectors can have a yellow icon, as can be seen to left in Figure 3.2. The content of the control bus is defined by the variables connected to the instances of it. This way of arranging the signals simplifies the model structure especially for larger hierarchical models. In the Figure 3.1 there is a large model with several of layers of models connected through non expandable connectors. The advantage in using control buses can be shown by comparing the number of connectors in the process in Figure 3.1b, with the process model in Figure 3.2.

The model used for simulation throughout the project has been built by models from libraries, assembled together. These are models such as boilers, valves, turbines and accessories. The models which handles steam, uses the IF97 standard [WCD+00] developed by International Association to determine properties of water. It requires two thermal properties to calculate the properties of water, according to [WCD+00]. Detailed dynamics such as inertia of the steam flow or losses and changes in pressure along the steam pipes, are neglected due to the relative large time constants of interest for this thesis. The time constants of interest for the controllers are an interval from 1 second, as in the fastest steam valves, up to 10 minutes as in the controlled load boiler. For instance, the total volume of the part of the steam network with the same pressure level is modelled as a steam tank with approximately the corresponding volume.

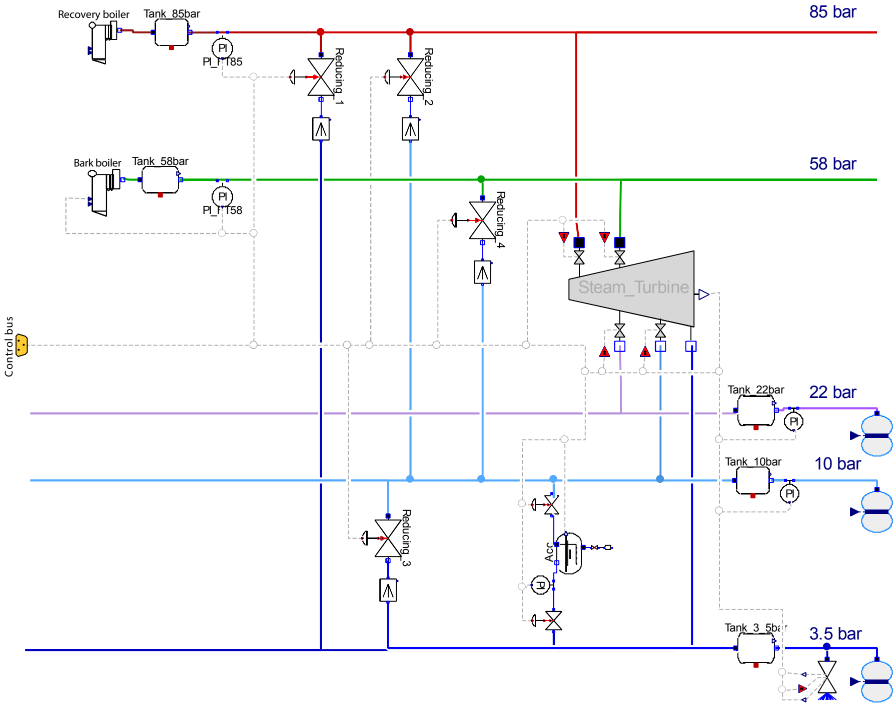


Figure 3.2: *The model of the multilevel steam network.*

3.2 Submodels of the process model

The parts in the multilevel steam network are modelled as submodels and assembled together they form the process model. From the perspective of the MPC even the first layer of control loops are considered as part of the controlled process. These submodels are described further in this section, starting with the accumulator.

3.2.1 Accumulator

The accumulator is modelled as a steam separation tank containing a medium that is in a two phase region, i.e. both in liquid and vapour form. Enthalpy and pressure in the tank is used to find the properties such as density, by help of the IF97 standard. The model contains equations describing the split between steam and water as,

$$V\rho = V_{liq}\rho_{liq} + (V - V_{liq})\,\rho_{vap} \tag{3.1}$$

where V is the total volume, ρ is the density of the medium, V_{liq} is the current volume of the liquid, ρ_{liq} is the density of the liquid and ρ_{vap} is the density of the vapour. When the split and the shape of the tank are known, the water level can be defined. Also equations for the rate of change in internal energy and mass of the medium is described as,

$$\frac{d}{dt}\left[V\rho\right] = \dot{m}_{port} \tag{3.2}$$

$$\frac{d}{dt}\left[mu\right] = \dot{h}_{port} \tag{3.3}$$

where $\dot{m}_{port}$ is the flow of steam, m is the mass, u is the internal energy of the medium and $\dot{h}_{port}$ is the flow of enthalpy. The parameters for configuring this model are; volume, height,

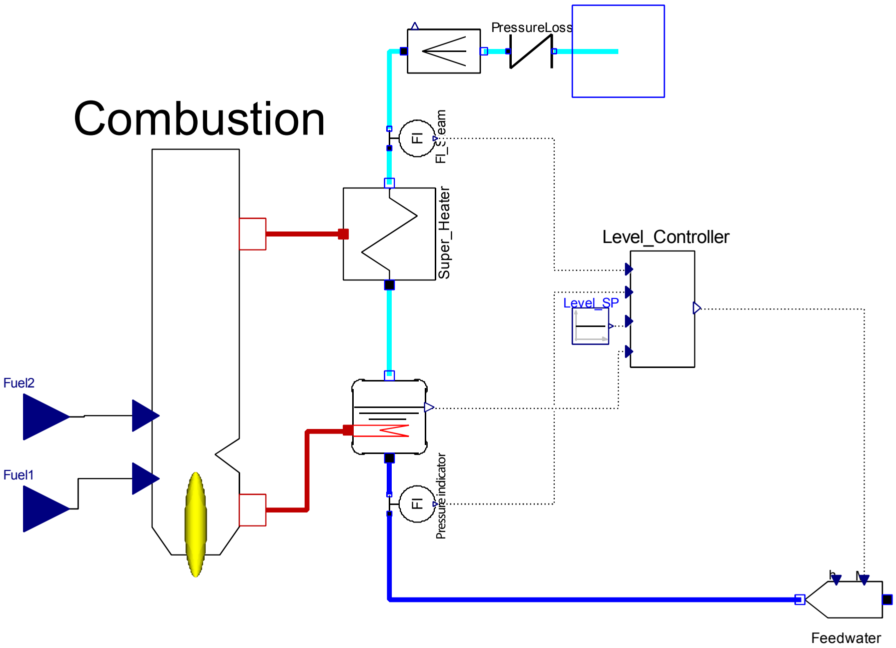


Figure 3.3: *Dymola model of the steam boiler.*

volume height curve, height of the inlet port from the bottom. These parameters have been copied from a simulation model of a similar process that has been verified.

3.2.2 Boilers

The steam is generated to the HP header by controlled load boiler models from Steam Power. Its symbol can be seen in Figure 3.4. The content of the boiler model can be seen in Figure 3.3. It consists of submodels for combustion, steam drum, water level controller, super heater and temperature controller.

The Boiler can be fed with two fuel variables to the combustion model. Here the fuel variables are converted to heat via some models describing the dynamics and time constants in the combustion. The heat variable is then connected to the steam drum model and the superheater model. Equations for energy and mass balances in these models are converting the heat variable into evaporated steam in form of mass flow and enthalpy. One output variable from the steam dorm is the level in the tank which is calculated based on pressure, enthalpy and the physical properties of the drum and water. By reading the variable for mass flow on the output as feed-forward signal, a level controller strives to keep the water level of half the steam dorm height by adjusting the inflow of water from an ideal temperature and flow source.

In order to avoid too high enthalpy of the mass flow generated by the superheater model there is a temperature control model which adds an extra amount of mass based on desired temperature and the current pressure. The timings of the boilers are approximately modelled such that a step on the fuel feed resulting in an increase of the steam output with 10 kg/s takes 1500 s until the output flow has stabilized again. The time constants are chosen to be like the verified simulation model.

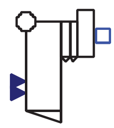


Figure 3.4: *The icon of the boiler model in the Dymola model.*

3.2.3 Steam consumers

The steam turbine models used in this project are based on theories developed by Stodola, A [SL27]. The configuration parameters for the model are the medium parameters and the Stodola machine constant (C_t) for the turbine. The Stodola equation used for defining the mass flow through the turbine ($\dot{m}$) is:

$$\dot{m} = C_t \frac{p_1}{T_1} \left(1 - \left(\frac{p_2}{p_1}\right)^2\right)^{0.5} \tag{3.4}$$

where p is the absolute pressure, T the temperature and index 1 and 2 are the in- and outflow respectively. The mechanical power output from the turbine is described as,

$$W_T = \dot{m} \left(h_{is,1} - h_2\right) \eta_{is} \tag{3.5}$$

where W_T is the power output, $\dot{m}$ is the mass flow through the turbine, h is the enthalpy, η is the efficiency, indexes 1 and 2 are the up- and downstream and index is means isentropic.

The steam consuming units on MP and LP pressure headers are modelled as a single model with an input for setting the desired flow of steam out from the steam network. The icon for this model can be seen in Figure 3.5. Each MP and LP pressure header has only one load model connected. These models consist of a valve and a flow controller for releasing steam to an ideal temperature and pressure sink.

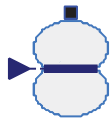


Figure 3.5: *The icon of the consumer model in the Dymola model.*

The input for the load model on the 10 bar MP header is connected to the Modelica model for generating the disturbance forecast. This model has two outputs. One output is the current disturbance signal which is fed into the load model. The other output is a vector signal fed into the controller. As parameter it takes a two column matrix where one is defining the time and the other the absolute value of steam flow. These points are then linearly interpolated to create a trajectory. The forecast vector is then points on this trajectory shifted in time with configurable distances.

3.2.4 Bias control

Each and every pressure level in the steam network has its own preferred pressure as set-point signal (SP). There are several control loops comparing their reference signal (r) with the same measured pressure as the controlled output to construct the control error. To separate these control loops and make them active at different value of the set-point, they have an individual bias added to the set-point. The Figure 3.6 is illustrating where the five different control loop's reference signal are relative to the set-point pressure for the HP header. There can also be seen in Figure 3.6 which pressure header is sacrificed first if there is a lack of steam in the HP header.

For those controllers where the control error has been large for a longer time, the control signal will be saturated. This is because the control signal is limited. Thanks to the anti-windup mechanism in the controllers they can remain saturated. The control signal has been inverted for the controllers with an outlet valve as the actuator. Because positive control error results in positive movement of the control signal and without inversion this would open the outlet

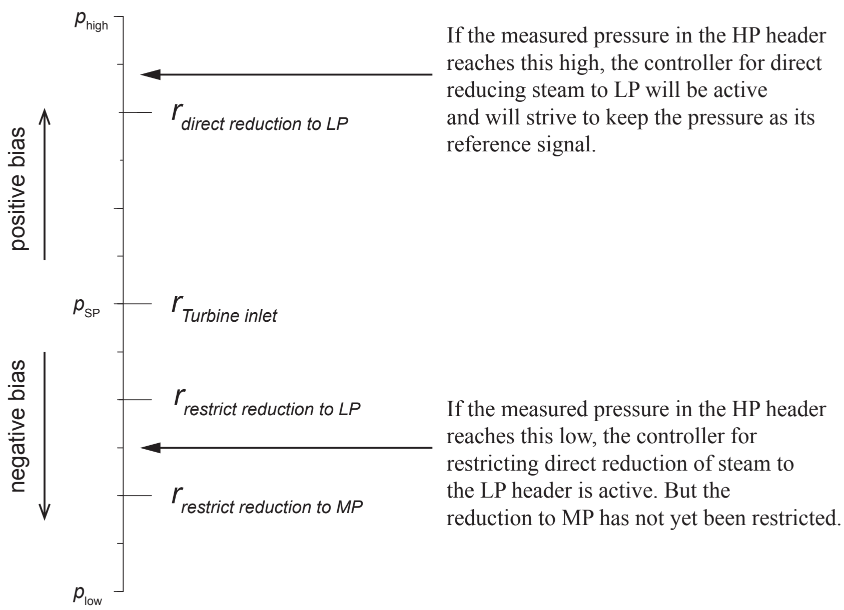


Figure 3.6: *The differentiation of reference signals* (r) *based on the set-point pressure* (p_{SP}) *for the HP header.*

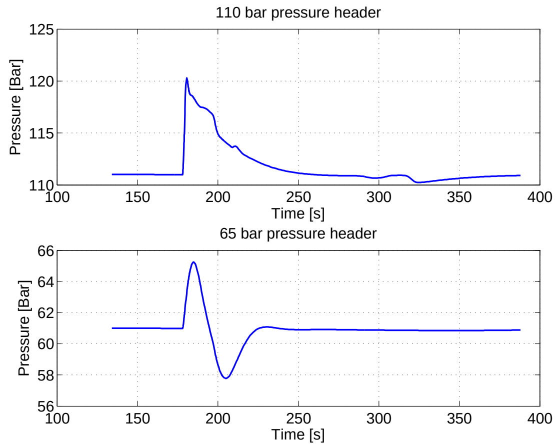
valve, instead of closing it so that the pressure will build up. This can also be done by using negative control gain instead.

The controllers are lumped together in one container Modelica model, which can be seen in Figure A.5. The layout of the graphical representation of all the controller and its wiring are inspired of a process scheme of a multilevel steam network. The controllers placement vertically is determined by their controlled output and their placement horizontally are determined by which actuator the control signal is wired to.

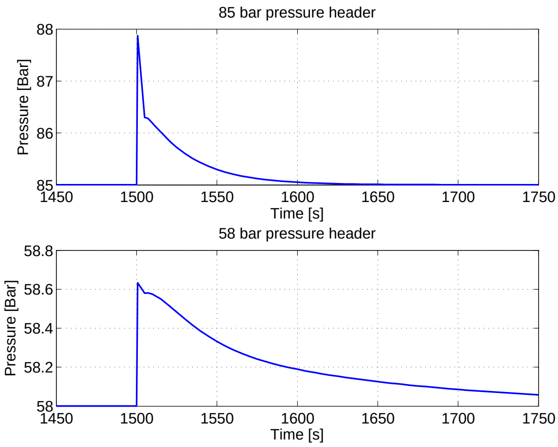
The interface of assigning the bias for each reference signal can be seen in Figure A.4 in appendix A.2 on page 33.

Depending on what the current pressure is at a pressure header some controllers have reached the limit for their control signal. If the pressure is for instance less than the controller's reference signal, with its bias added, then the control signal is zero, i.e. the signal for the outlet valve to be fully closed. This control loop is then inactive and will be until the pressure increases and the controller needs the valve to be opened to reduce the pressure.

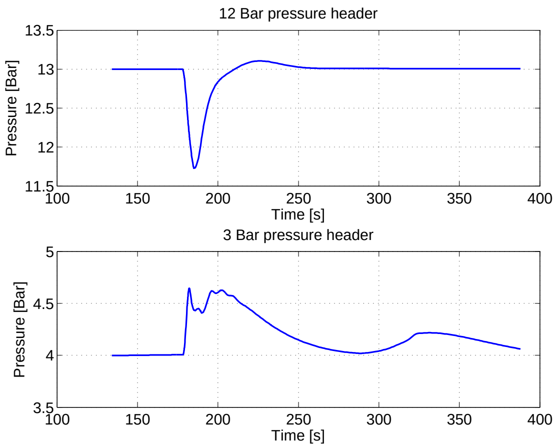
When the different control loops with bias terms are active the set-point signal is either over or under the nominal pressure for the certain pressure header. To avoid that this causes the actual pressure in the header to deviate from the nominal pressure, there is another controller adjusting the set-point signal for each pressure header with the aim to keep the pressure as close to the nominal as possible. This outer controller is referred to as master pressure controller and should approximately have five times longer time constant than the inner pressure controllers. The control structure is based on PI controllers. The tuning was done ad hoc to produce rapid and stable control responses, no analytical method was used.



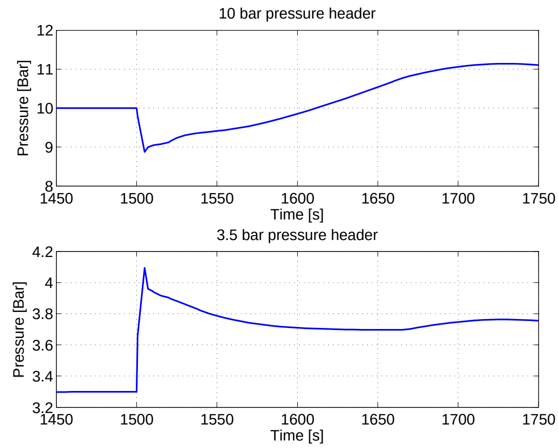
(a) *High pressure headers (110 bar and 65 bar).*



(b) *High pressure headers (85 bar and 58 bar).*



(c) *High pressure headers (12 bar and 3 bar).*



(d) *High pressure headers (10 bar and 3.5 bar).*

Figure 3.7: *Comparison of the pressure characteristics when the turbine is tripping.*

3.3 Model verification

One way to verify the characteristics of the system is to influence the process with disturbances such as turbine tripping. This will show how the multiple actuators are working to stabilize the pressures in the network and to lead the steam alternative ways.

When something goes wrong with the steam turbine it can be critical to stop the steam flow into the turbine as fast as possible. This is called turbine tripping. It is then important that the system quickly opens the bypass valves for the steam to escape in an alternative way. This disturbance is anyhow causing the different pressures in the pressure headers to deviate.

The characteristics of these deviations are compared with a similar already verified simulation model of another multilevel steam network. The reference model has been verified through comparisons with measurements done on the corresponding real plant. The physical properties such as volumes and components of the multilevel steam networks are similar and therefore suitable for comparison.

In Figure 3.7 on page 13, the different pressures in the pressure headers are plotted after a simulation with turbine tripping. On the left side, i.e. Figure 3.7a and 3.7c, there are simulation results taken from simulations of the reference model and on the right side, i.e. Figure 3.7b and 3.7d, there are data from the model developed by this thesis.

The actual pressure in the headers differs between the two steam networks. Also the magnitude of the pressure deviations on each pressure header are different between the reference steam network model and the one developed in this thesis. Also the shape of the pressure trajectories on 65 bar and 58 bar pressure headers differ and this is because the pressure controller on the reference plant uses feedforward technique, which the model developed in this thesis does not have. The important comparison of the pressure trajectories is the direction of the pressure deviation and the slope.

Both upper Figures 3.7a and 3.7b are measurements made on the HP headers. The pressures are increasing fast when the turbine closes its inlet valves but are after action taken by the controllers the pressure are decreasing again. There is a difference between how large the pressure deviation is on the 65 bar and the 58 bar pressure headers, because there is different flow of steam into the turbine from the headers.

The pressure in the upper graph in Figure 3.7c is decreasing right after the turbine is tripping. This is because the major flow into the 12 bar pressure header is from the drain valves on the turbines. After some seconds the valve for bypassing steam from the HP are opened and the pressure deviation are taken care of by the controllers and the actuators. The 3 bar LP are experiencing an increase of incoming steam unlike the 12 bar MP and this is because the bypass valve from the HP has opened. There is no use in dumping steam from HP header to MP header since it will end up in the LP header anyway.

A comment to make here is that the left curves for the MP and LP headers are settling on the nominal pressure again whilst the right pressure curves are settled on a slightly higher level. This is because the right model has master controllers which compensates for the bias and adjusts the set point such that the measured pressure will match the nominal. The left model is missing such controller. There is a possibility to get this master control functionality from the overall optimizing controller instead, this is discussed further down.

These pressure characteristics in the graphs to the left are reflected in the graphs on the right side. By observation the conclusion can be made that the controllers and actuators in both systems are prioritizing in a similar order and the pressure deviations are mitigated in the same way.

The 22 bar MP header of the model made in the thesis are not represented in any graph in the figure. This is because it will be lost if the turbine is tripping. The only way where steam can come into the 22 bar MP is from the drain valves on the turbine. Therefore this steam network is not relevant when studying the dynamics of the system when the turbine trip occurs.

4 Control strategies for the optimizing controller

The multilevel steam network system has some aspects where it could take care of the energy in a more efficient way as mentioned in section 2.5. By introducing some anticipation in the control of the system, it could be possible to reduce the amount of unused energy. One of the harder disturbances for the control system is the larger intermittent changes in load on the

MP header with 10 bar as nominal pressure. The rapid changes in load are the most common causes why the blow out valve is opened and energy is lost to the atmosphere.

When the demand of steam has decreased, it is desirable to produce less steam and vice versa. Out of the two steam producing boilers, only one is subject for control by changing the fuel feed. Only the recovery boiler can increase its production by adding extra fuel but because of the chemical recovery it cannot decrease its production below a certain level. The bark boiler on the other hand can be totally switched off as well as being operated at different rates. It produces high pressure superheated steam to the HP header with 58 bar as nominal pressure.

By using the load forecast, the controller could adjust the bark feed for the upcoming needs of the system. It is also possible to prepare the process for the upcoming disturbances by ensuring that there is steam available, such as the accumulator is loaded or perhaps the pressure in pressure header subject for the disturbance is slightly higher than normal.

4.1 Requirements on the controller

The controller needs to be able to control multiple output signals and make them follow reference signals. It should also be able to control the process with multiple control signals. These control signals often have physical limitations which the controller should take into account when controlling. For example if the control signal is the set point for the fuel feed to the boiler it can never go below zero and it is limited to a certain maximum rate of change. Also, the controller should adjust how active the control signal is to how much the actuator can take in order not to wear it out. Everything is to fulfil the mitigation requirements set by the actuators.

The controller needs to have some mechanism for making use of the different available forecasts. This means the controller needs to be able to predict the future states of the process somehow. Both the future needs of steam and the prices should be taken into account of the controller when predicting the future states.

4.2 An existing implementation of an optimizing controller with plant wide perspective

The goal for this thesis has been to design a controller using model predictive control. In the work of understanding the control problem a similar case has been studied. In this case the objectives are just about the same but the controller is not using model predictive control.

One solution for controlling the multilevel steam network with a plant wide perspective could be the solution developed by Solvina for a pulp and paper plant. The controller was implemented in *structured text* [Eks08] in a PLC .

The primary objective for the controller is to minimize the difference between produced steam and consumed steam. Secondary, the objective is to maximize the amount of produced electricity by the steam turbine when the circumstances are profitable. This is when the electricity price is higher than the costs for burning bark and venting steam. Prices for bark and electricity are manually configured as parameters for the controller.

The process is divided into several different operational situations. These are then realized in the structured text program as cases with logical expressions. For example an increase of fuel feed to the boiler can depend on following aspects, if the current steam balance is negative and there are good margins before reaching maximum capacity of venting steam.

The evaluation of the control situation is slightly more complex than the example. In addition, the controller is working with several parameters stating different physical limits, safety margins and preferred max and min values of different quantities in the system. There are also parameters for defining different operational scenarios such as temperature limits in the dump condenser which are depending on the temperature outdoors, i.e. the current season. These values and levels of thresholds are based on experience from test runs.

The controller is referred to as the Overall Optimizing Controller because its plant wide perspective and because it is the outer most controller in the cascade structure of controllers. The inner level of controllers is striving to keep the desired pressures in the different pressure headers. Because of this, the outermost controller needs to force the system to produce more steam when it is profitable despite the increased amount of vented steam. This is done by increasing the lower limit of the signal controlling the amount of fuel being fed into the boiler.

The actuator, which the outermost controller is acting on, are the controllable boiler, the dump condensers and the blow out valves. The control signals from the Overall Optimizing Controller are of incremental type and before they reaches the actuator it is converted based on current state of the process and depending on preconfigured parameters, such as, the target value for the pressure in the accumulator.

This control strategy opens up for highly configurable system with plenty of options to tune the system's behaviour. The structure is complex and the control signals are routed through multiple layers where they are influenced in different ways before they reach the actuators. Future adjustments to these *ad hoc* 'fixes' constituting the outermost controller can be hard and time consuming.

5 Control design using model predictive control

The MPC technique is using an internal model of the controlled process to find the next control signal. This internal model can generally be of three different types; Finite Impulse Response (FIR) model; Transfer function model; or state-space model. The FIR model is preferable due to its transparent description of the system's gain, delay and response time. The drawback is the restriction to only being able to handle stable processes. The transfer function models on the other hand can describe unstable plants as well but these models can turn into very complex structures when it comes to multi input multi output (MIMO) systems. Therefore the state-space representation of models with its simplicity in representing MIMO systems has gained its popularity in predictive control design [Mac02].

For each control cycle at time k_i the control algorithm predicts the future behaviour of the system within a limited time window, called the prediction horizon, N_p. The algorithm computes the trajectory of the control signal u that optimizes an objective function. System variables such as control effort, control error and the controlled process output y are represented in the objective function. These variables are manipulated by the decision variable, which in predictive control are the control signals. The length of the trajectory that constitutes as the decision variable can be less than but not longer than the prediction horizon and is called the control horizon, N_c. Although only the first value of the predicted control trajectory is sent to the process because this prediction procedure is repeated next control cycle again. This control strategy is often called receding horizon control, because the prediction window is moving in time.

The controller needs to have integral action in order to eliminate steady-state error. This is achieved by an algorithm that introduces an integrator for each output signal as a new state variable in the state-space model.

The cost function or objective function are formulated on a quadratic form and is a minimization problem. The advantages of formulating the optimization problem on "square root" form is because it turns the problem into a known optimization problem known as *Quadratic Programming* (or *QP*) problem. This means that it can be solved by standard algorithms. The termination can be guaranteed and it is also possible to estimate how long time it will take to solve it, according to [Mac02]. These properties are vital when it comes to control algorithms used on-line.

Predictive control's placement in the typical control structure in the process industries have become on a layer where it can replace a traditionally complex layer consisting of logic, overrides, decoupling networks and exception handling. One example of such *ad hoc* solutions to individual problems were given in section 4.2. This layer is above the local control layer which consists of proportional or proportional-integral controllers. The predictive control gets to deal with those conditions which cannot be handled by the simple 'one set-point, one control loop'. Because the predictive control is an integrated solution to handling these problems, it can provide dramatically better performance than the technology it is replacing, according to [Mac02].

In order to know the state of the system at each control cycle, an observer is used to estimate the state variables based on the process measurable output and the control signals. This estimation is done before each prediction so that the state of the internal model in the MPC reflects the real process.

Measurable disturbance are easily incorporated in predictive control as feedforward, according to [Mac02]. It is only a matter of changing the inner model to include the effects of the measured disturbances in the prediction of future outputs. The optimization algorithm for finding the future control signal will then take these into account as well.

5.1 The internal model of the controller

The internal model for the controller designed in this thesis, is created by the built in function for linearizing Modelica models, in Dymola. It provides a continuous time state-space model *(A,B,C,D)*. The MPC controller will be implemented in a computer environment as a discrete controller with a sample period. Therefore the internal model is discretized for the preferred sample period in Matlab. Then the model needs to be augmented so that the input becomes the difference of the control variable like

$$\Delta u(k) = u(k) - u(k-1). \tag{5.1}$$

This augmenting procedure gives the controller its integral action and is explained in detail in [Wan09]. The final model which is used by the MPC is

$$\hat{x}(k_i+1|k_i) = A\hat{x}(k_i|k_i) + B_u\Delta u(k_i) + B_v\Delta v(k_i) \tag{5.2}$$

$$\hat{y}(k_i|k_i) = C\hat{x}(k_i|k_i). \tag{5.3}$$

5.2 Prediction of the process output

The internal model is used to predict the future output based on the current state variables. The predicted variables are formulated in terms of current state variable information $x(k_i)$, future movement of the control $\Delta u(k_i + j)$ and the disturbance $\Delta v(k_i + m)$ where $j = 0, 1, \ldots, N_c - 1$ and $m = 0, 1, \ldots, N_p - 1$. The output are sequentially computed as

$$
\begin{aligned}
y\left(k_i + 1 \,|\, k_i\right) &= CAx(k_i) + CB_u\Delta u(k_i) + CB_v\Delta v(k_i) && (5.4)\\
y\left(k_i + 2 \,|\, k_i\right) &= CA^2x(k_i) + CAB_u\Delta u(k_i) + CAB_v\Delta v(k_i)\\
&\quad + CB_u\Delta u(k_i + 1) + CB_v\Delta v(k_i + 1) && (5.5)\\
y\left(k_i + 3 \,|\, k_i\right) &= CA^3x(k_i) + CA^2B_u\Delta u(k_i) + CA^2B_v\Delta v(k_i) + CAB_u\Delta u(k_i + 1)\\
&\quad + CAB_v\Delta v(k_i + 1) + CB_u\Delta u(k_i + 2) + CB_v\Delta v(k_i + 2) && (5.6)\\
&\;\;\vdots\\
y\left(k_i + N_p \,|\, k_i\right) &= CA^{N_p}x(k_i) + CA^{N_p-1}B_u\Delta u(k_i) + CA^{N_p-1}B_v\Delta v(k_i)\\
&\quad + CA^{N_p-2}B_u\Delta u(k_i + 1) + CA^{N_p-2}B_v\Delta v(k_i + 1)\\
&\quad + \ldots + CA^{N_p-N_c}B_u\Delta u(k_i + N_c - 1) + CA^{N_p-N_v}B_v\Delta v(k_i + N_v - 1) && (5.7)
\end{aligned}
$$

$$
\begin{aligned}
Y &= \left[y\left(k_i + 1 \,|\, k_i\right)^T \quad y\left(k_i + 2 \,|\, k_i\right)^T \quad y\left(k_i + 3 \,|\, k_i\right)^T \quad \ldots \quad y\left(k_i + N_p \,|\, k_i\right)^T\right]^T && (5.8)\\
\Delta U &= \left[\Delta u\left(k_i\right)^T \quad \Delta u\left(k_i + 1\right)^T \quad \ldots \quad \Delta u\left(k_i + N_c - 1\right)^T\right]^T && (5.9)\\
\Delta V &= \left[\Delta v\left(k_i\right)^T \quad \Delta v\left(k_i + 1\right)^T \quad \ldots \quad \Delta v\left(k_i + N_v - 1\right)^T\right]^T && (5.10)
\end{aligned}
$$

The predicted output at each prediction instance in (5.4) are expressed as the vector Y in (5.8). The future control movement and disturbance changes are expressed as the vectors ΔU in (5.9) and ΔV in (5.10). Using these vectors and rearranging equation (5.4) the predicted variables can be expressed in compact matrix form as

$$Y = Fx(k_i) + \Phi\Delta U + \Psi\Delta V \qquad (5.11)$$

where

$$
F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB_u & 0 & 0 & \ldots & 0 \\ CAB_u & CB_u & 0 & \ldots & 0 \\ CA^2B_u & CAB_u & CB_u & \ldots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B_u & CA^{N_p-2}B_u & CA^{N_p-3}B_u & \ldots & CA^{N_p-N_c}B_u \end{bmatrix};
$$

$$
\Psi = \begin{bmatrix} CB_v & 0 & 0 & \ldots & 0 \\ CAB_v & CB_v & 0 & \ldots & 0 \\ CA^2B_v & CAB_v & CB_v & \ldots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B_v & CA^{N_p-2}B_v & CA^{N_p-3}B_v & \ldots & CA^{N_p-N_v}B_v \end{bmatrix}.
$$

For each sample period the observer reads the process output, control signal and the disturbance to estimate the current state variables in the inner model. The equation for finding the estimated state variables are formulated as,

$$\hat{x}(k_i+1) = A\hat{x}(k_i) + B\Delta u(k_i) + V\Delta v(k_i) + K_{ob}\left(Y(k_i) - C\hat{x}\left(k_i\right)\right) \tag{5.12}$$

where K_{ob} is the observer gain calculated by the Matlab function dlqr. This function is solving an algebraic Riccati equation to find the optimal gain which minimizes the linear quadratic objective function. The function needs two weighting matrices (Q and R) as arguments in addition to the A and C matrices. These weight matrices can be chosen based on the covariance of the measurement disturbances (R) and the covariance of the input disturbance (Q). But for simplicity the weight matrices where chosen to be the unit matrix. By executing the command,

$$K_{ob} = \mathrm{dlqr}(A, C^T, Q, R);$$

The function returns the optimal observer gain, according to [MAT11].

5.3 Advantages of using Hildreth's Quadratic Programming Procedure

The objective function used for finding the optimal control law is formulated on quadratic form. The algorithm Hildreth's quadratic programming [Lue97], [WC78], is used to find the control law that minimizes the objective function.

The algorithm allows for putting constraints on the control signal, the rate of change of the control signal and the process output. This feature is useful when the actuator has physical limitations, e.g. if the actuator is a valve, then the control signal should not be more than fully closed or fully open.

Perhaps even more importantly, the iterative solution does not involve matrix inversion, so in the situation of conflict constraints, the algorithm still delivers a compromised, sub-optimal solution without being numerically unstable. This is particularly important in the real-time implementation of the predictive control system.

5.4 The optimized control signal

The objective of the predictive control is to bring the predictive output as close as possible to the set-point signal. In one optimization window is the given set-point signal $r(k_i)$ at sample time k_i assumed to remain constant. The number of controlled output is noted as q and each has a corresponding reference signal $r_j(k_i)$. These are formed into the vector

$$r(k_i) = \begin{bmatrix} r_1(k_i) & r_2(k_i) & \dots & r_q(k_i) \end{bmatrix}^T \tag{5.13}$$

with dimensions $q \times 1$. The references remain the same during the prediction window and therefore the vector

$$R_s = \overbrace{\begin{bmatrix} I_q & I_q & I_q & \dots & I_q \end{bmatrix}}^{N_p q}{}^T r(k_i) \tag{5.14}$$

needs to contain $r(k_i)$ repeated N_p times and forms the matrix of dimensions $N_p q \times q$. This vector will be compared with the predicted output vector Y in the objective function J.

$$J = (R_s - Y)^T (R_s - Y) \tag{5.15}$$

The decision variable of the objective function is the predicted incremental control signal $\Delta U(k_i)$. To prevent the control signal to be too large an extra term is added to the objective function.

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \tag{5.16}$$

Substituting the predicted output Y in J with the definition given in (5.11) and rearranging the equation as

$$\begin{aligned} J &= \left(R_s - Fx(k_i) - \Psi\Delta V(k_i)\right)^T \left(R_s - Fx(k_i) - \Psi\Delta V(k_i)\right) \\ &- 2\Delta U^T \Phi^T \left(R_s - Fx(k_i) - \Psi\Delta V(k_i)\right) + \Delta U^T \left(\Phi^T \Phi + \bar{R}\right) \Delta U \end{aligned} \tag{5.17}$$

it can be seen that the decision variable only affects the two last terms. The objective function is simplified to

$$J = \Delta U^T \left(\Phi^T \Phi + \bar{R}\right) \Delta U - 2\Delta U^T \Phi^T \left(R_s - Fx(k_i) - \Psi\Delta V(k_i)\right). \tag{5.18}$$

The optimal variable ΔU will contain the controls $\Delta u(k_i)$, $\Delta u(k_i + 1)$, $\Delta u(k_i + 2)$, ..., $\Delta u(k_i + N_c - 1)$ but with the receding horizon control principle, only the first control are implemented and the rest are ignored.

The constraints are built up by matrices and formulated to suit the quadratic programming routine.

$$M\Delta U \leq \gamma \tag{5.19}$$

The γ vector in equation (5.19) are the given limits and the M matrix are there to decide which control parameter or parameters in ΔU are applicable for a certain limit. Further details on how this is formulated can be found in the book [Wan09].

5.5 Choice of control inputs and outputs

The overall optimizing controller can control multiple process output signals and can affect multiple process inputs. The choice of process outputs are limited to only the variables that are measurable. For the modelled process there are several accessible variables that could possibly be the controlled output. There are also plenty of options for choosing the points where the controller are acting on. It could be on a valve directly or it could be the set point for an inner controller.

In [Maj05] model predictive pressure control of steam networks is discussed. The modelled process used in the article is similar to the process that is modelled in the thesis. It has a controlled load boiler, steam turbine, multiple pressure headers and an accumulator.

The solution described in the article involves two different MPC controllers. One called the boiler MPC and the other called the turbine MPC. As the name suggests, the controllers are acting on the boiler and the turbine inlet valves. The process outputs measured by the controllers are the MP and LP header pressures, flow of steam through the inlet valve to the turbine and the accumulator flow in and out.

The sample period was 5 s for the boiler MPC and 0.5 s for the turbine MPC. By this setup the MPC controllers are substituting the dedicated multiple control loops. The results in [Maj05] shows that MPC performs better and that the tuning of the system was simplified compared to conventional control structure based on PI controllers.

The control objectives in this thesis are to keep the multiple control loops and complement the system with a MPC for optimizing the system's performance from a plant wide perspective. The controller can then have longer sample period than the controllers in the article. The advantage from keeping the inner control loops and design the optimizing controller to act outside is the ease in implementing the controller and hopefully the costs will be kept low when the modifications of the system is of a limited extent.

Traditionally the indication of excess or shortage of steam is the load level in the accumulator. If the bias values are chosen smart the inner control loops will first load or unload the accumulator before opening the blow-out valve or a bypass valve. This makes the load level of the accumulator a candidate for being one of the controlled output signals used by the MPC. It is always preferable to vent as little steam as possible and this makes the flow of the blow-out valve a candidate for being one of the controlled process output. Then the MPC could track flow rate reference of as small as possible. The consequence of this could then be that the MPC is throttling down the steam producing units to avoid any venting of steam. This could lead to decreased flow through the steam turbines and less electrical power sold.

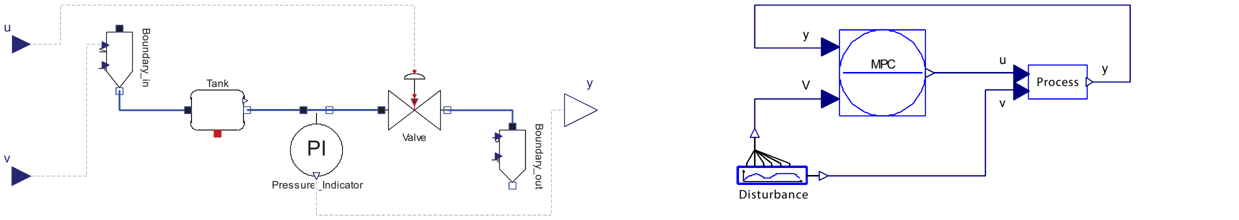
It would have been preferable to design the MPC such that it works with the objective of maximizing the power output of the steam turbines. This cannot be achieved by the MPC controller designed, because it is not enough to follow a reference signal for the power output when it is changing depending on the current loads on the MP and LP headers.

5.6 Simulation results

The simulation results described in this chapter are results from verifications of the different implementations done in Modelica in Dymola. First the simulation results from the verification of the MPC controller will be shown. Together with this it will also be demonstrated how the controller handles the disturbance forecast and how the control performance was improved. All this is demonstrated on a simpler process model consisting of a simpler steam flow where the controller needs to track a reference pressure.

5.6.1 Control law testing

The Modelica implementation of the solver in the MPC is based on the work done by Ressel [Res10]. It uses a built in function of Modelica to invert the matrices needed. The MPC controller, which calls this solver is also a revised and extended version of the one developed by Ressel [Res10]. It has been modified to be able to take the disturbance forecast as an input as well.



(a) *The model is used for testing the principles of MPC in this application.*

(b) *Dymola model of the simple process connected to the MPC and the disturbance model.*

Figure 5.1: *Screen shots of Dymola's graphical representation of the simple process model together with the closed loop model.*

Table 5.1: A table with the eigenvalues of the linearized Modelica model consisting of a tank and an outlet valve.

$eig(A)$
-0,013
-0,005
-1,000

The Modelica implementation of the MPC is a model that is configurable in many ways, so that it can be used in a variety of applications. The number of inputs and outputs are set dynamically and configured by parameters. The internal process model is loaded by importing the state-space matrices as parameters. This causes the Modelica model to be of two different sections; one consisting of the calculations done prior to the simulation; and the other section for code executed on-line during simulations. In this way, only the calculations needed are performed each control cycle and the other are prepared before.

The disturbance forecast is fed into the controller as a vector of configurable length. This vector is then transformed into a vector with the same length as the prediction horizon. The future disturbance absolute magnitude needs to be translated into the difference between the current and the previous value. Because information about the disturbance in between these points in time is unknown for the controller, it is simplified to be constant until the next time instance where the disturbance is estimated again. The vector with absolute values of the disturbance is time shifted every control cycle and only the vector elements corresponding with the timings of the forecast vector signal are updated. In this way the controller forms a view of what the disturbance could be like in between the forecasted time instances.

To test and verify the MPC design in Modelica, a simple steam process was developed and can be seen in Figure 5.1a. The disturbance is constituted by changes in the inflow of steam into the tank. The controlled process output was the pressure in the tank and the control signal was the opening of the outlet valve.

The linearization of this Modelica model was generated by Dymola and resulted in a state-space object of three states. The eigenvalues of the A matrix was all negative real numbers and can be seen in Table 5.1. The observability and controllability matrices of the system are of full rank, i.e. the system is both observable and controllable. Discretization and augmentation of the system is done in Matlab for a sample period of 1 second.

The disturbance forecast is an array of estimations of the upcoming disturbance with an interval of every twentieth second into the future. If the prediction horizon (N_p) is for instance 41 then three element signals will be used when transforming it to an input for the internal model. The current value, the value in twenty seconds and the value in 40 seconds are then transformed into a vector of length N_p.

The results from simulating the system with the MPC controlling the simple steam process can be seen in Figure 5.2. The system is under influence of disturbance in form of a step on the inflow rate of steam. The objective is to keep the pressure in the tank as steady as possible. Results from simulations with disturbance forecast taken into account by the optimizer in the MPC are plotted for different length of the horizons, i.e. the control horizon and the prediction horizon. It can be seen that the control signal is taking preventative actions before the disturbance even has affected the output, thanks to the feedforward.

5.6.2 Controlling the large process model with predictive control and feedforward

It has been shown that the MPC controller is able to make use of the disturbance forecast as feedforward on the case of the simplified steam process. In order to use this controller on the large multilevel steam network model some decisions needs to be made. As a first test the control signal will be the rate of fuel feed to the controlled load boiler and the controlled measurement signals are the pressure in the accumulator and the flow rate of steam through the blow out valve.

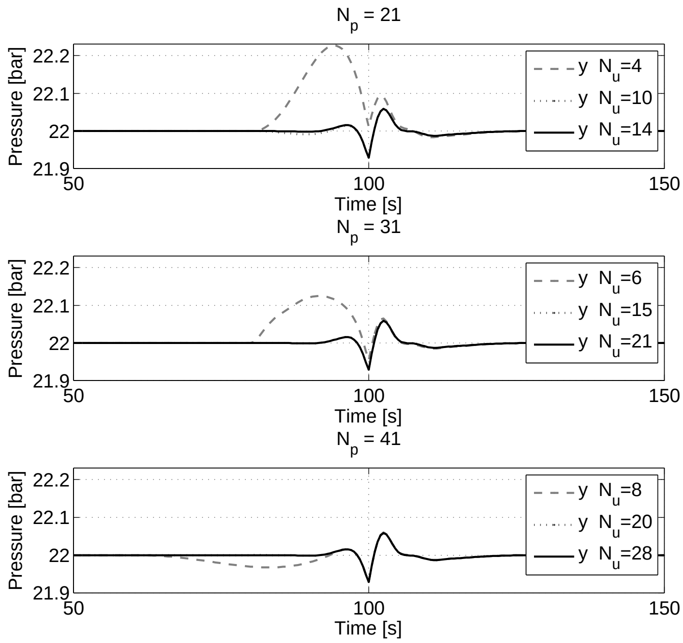
The layer of biased set-points and proportional-integrator control loops are tracking the pressure references in the pressure headers. In case of disturbances such as load changes on a MP and LP header the primary action taken by the controllers in this layer is to use the accumulator. This is why the pressure of the accumulator is useful for the MPC, because in this way it can easily measure the steam demand for the moment. If the pressure increases in the accumulator it could be needed to lower the fuel feed to the boiler.

The secondary action to be taken is to use the blow-out valve to vent the excessive steam and this is not preferable because energy is lost into the atmosphere. This is why the MPC tries to track a reference signal of almost zero for the steam flow through the blow-out valve. If the flow starts to increase it could be preferable to throttle down the controlled load boiler.

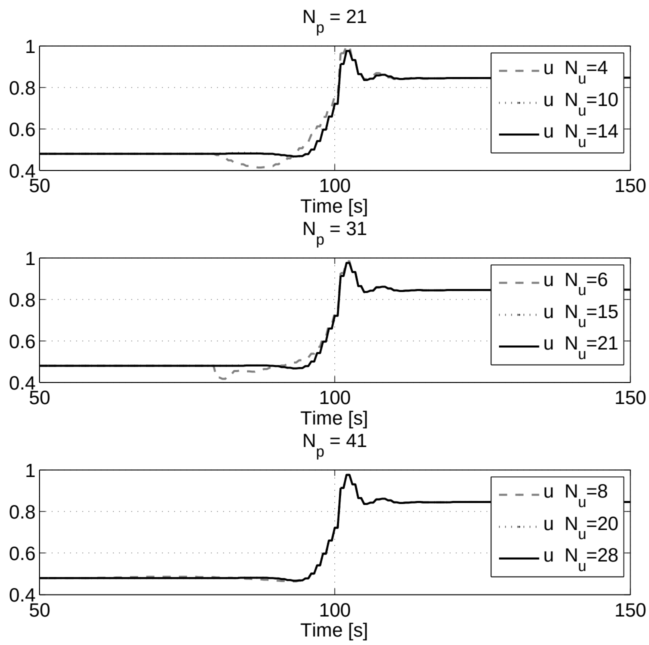
When the choosing of input and output signals of the process is done the signals need to be scaled such that they are normalized. This is because the control error will then be equally treated by the MPC when the normalized measurement outputs are moving in the same range. For instance, the pressure in the accumulator is preferably around 7 bar and can increase up to around 10 bar and decrease to about 3.5 bar. After the normalization the output signal will move around 0.5 instead of 70,000 Pa. Now the MPC can simply count with control errors with a quantity of 0 - 0.5 instead of 0 - 35,000.

For making the model detectable the unstable state variables need to be observable. This is done by measuring the plant where needed. As a first start the measured process output was chosen to be; the pressure in each pressure header; the valve positions of the outlet valves on the 58 bar and 85 bar pressure header, the inlet valves on the 22 bar and 10 bar pressure header; the steam pressure in the accumulator; the steam flow out of the blow-out valve; and at last the power output from the steam turbine. It would turn out that these measurements were not enough to make all unstable state variables observable and needs to be examined further.

After the measurement outputs have been set the model needs to be linearized at a chosen



(a) *MPC strives to keep the pressure at 22 bar*



(b) *Plot of the control signal from the MPC. Control signal one is fully open valve.*

Figure 5.2: *Results from MPC simulations with different values of the control parameters; prediction horizon (N_p) and control horizon (N_u). MPC is controlling the outlet valve of a tank to keep the pressure on a certain level. The measurable disturbance is the change of inflow of steam to the tank. This occurs at 100 s and the forecast is 20 s in advance.*

working point. The nonlinear model of the multilevel steam network and its first layer of control loops in Dymola is translated into a differential-algebraic equation which has around 4500 scalar unknowns and as many unknown equations. By again, using the built in function in Dymola the model is linearized at simulation time 1000 s when the variables has settled and steady state has been reached.

The result from linearization is a continuous time state-space object of 69 state variables, all of them with positive real part, 68 unobservable and 68 uncontrollable. The state variables are describing dynamics such as; pressure and enthalpy in the steam tanks; position of each valve actuator; and output from the integrators in the controllers. The state-space model order needs to be reduced to avoid numerical problem when predicting the future state variables. This is done with help of the Matlab command *minreal*() which takes a state-space object as an argument and eliminates the uncontrollable and unobservable states within a certain tolerance in the state-space model, according to [MAT11].

The disturbance forecast in the multilevel steam network is the intermittent changes in load on the 10 bar header. This forecast is modelled such that the first variable tells the estimated demand in 2 minutes from now, the next in 10 minutes and so on for every tenth minutes until 40 minutes. The disturbance in the simulations consists of a decrease of 6 kg/s in steam flow to the steam consumers on the 10 bar header and occurs at 4020 s.

The results from this simulation can be seen in Figure 5.3. The simulations are performed with the MPC taking the forecast into account and when it is not. Figure 5.3a is the amount of heat added to the boiler which is manipulated by the MPC. In the scenario when the MPC obey the disturbance forecast, it starts to throttle down the boiler earlier than the otherwise. This prepares the pressure in the accumulator for the upcoming disturbance. The second Figure 5.3b is the pressure in the accumulator and the third Figure 5.3c is the outflow through the blow-out valve.

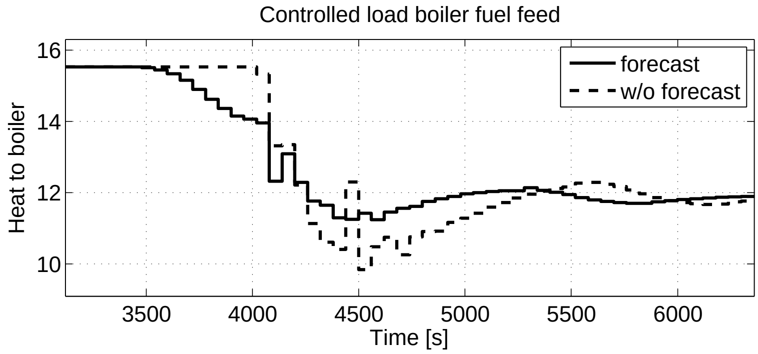
It can be seen from Figure 5.3b that the MPC controller manage to keep the pressure in the accumulator and the flow of steam through the blow-out valve, in Figure 5.3c, closer to the reference signals when taking the disturbance forecast into account.

6 Discussion

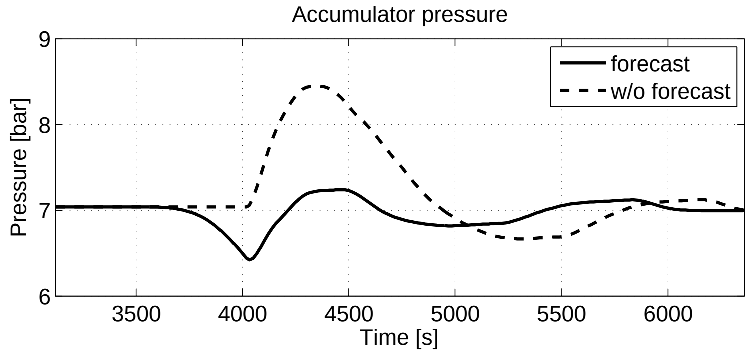
The MPC controller's placement in the control hierarchy, were in this thesis, above the layer with the 'one set-point, one control-loop' structure. In this first layer there are master pressure controllers, as referred to in Section 3.2.4, which could possibly be replaced by the MPC. By letting the MPC manipulate set-point pressure for each network it could possibly make use of these levels when acting preventative. For instance it could divers the pressure in the header prior to a major disturbance to prevent the pressure changes to populate to other headers in the steam network.

This needs some further testing and it evokes ideas on handing over more control tasks to be handled by the MPC instead of the lower layer in the hierarchy. Obviously the MPC is unable to handle control which requires more frequent actions without increasing its sample rate.

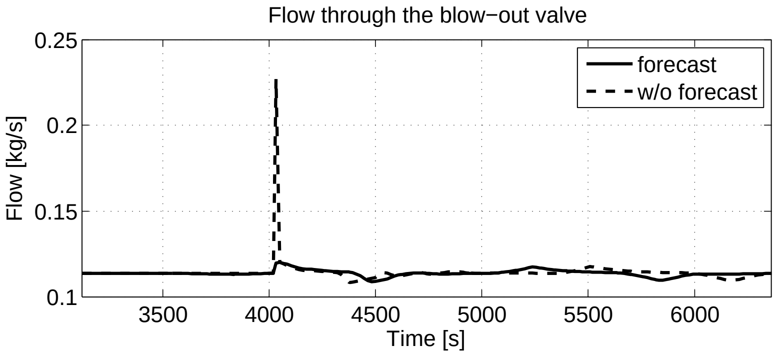
The capacity of the equipment that possibly could host the MPC controller in an eventual implementation is unknown for this work. These limitations on computing capacity would otherwise have been considered when choosing the sampling period and the two horizons.



(a) *MPC controlling the fuel feed to the bark boiler.*



(b) *MPC tracks a constant reference of 7 bar for the accumulator pressure.*



(c) *MPC tracks a constant reference of 0.11 kg/s steam flow through the blow-out valve.*

Figure 5.3: *MPC controlling the multilevel steam network with* $N_p = 11$ *and* $N_u = 8$. *Disturbance occurs at 4020 s and constitutes of decrease in load on the 10 bar pressure header.*

The designing of a MPC controller requires a mathematical model describing the process to be controlled. In case of changes in the controlled process, the adjustments need to be reflected in the internal model of the MPC. The easier this is to perform the better are the chances that the MPC will be kept in use after this. It is only a matter of updating the Dymola model; generate the state-space object and configuring the MPC with the new model. This procedure is easiest performed by an engineer who is familiar to Dymola and advanced control.

Comparing this update procedure with the steps needed to be taken to update the layer with all the *ad hoc* "fixes" and logic, which MPC is replacing, it could be more straight-forward to update the MPC.

Another possibility to make use of the MPC with feedforward could be to serve the operator the information about how the controller would have had controlled the process as suggestions of actions to be made. This solution is even applicable to plants having only manually operated boilers. The MPC could possibly be running on a computer next to the operator and based on samples of the plant present suggestions for the operator on a graphical display.

When the selling price for electricity is higher than the price of burning bark for producing steam then it is profitable to run the steam turbines on higher capacity even if the consequence is steam being vented out to the atmosphere. The MPC could be fed with the feedback signal from a comparison between the three different prices; selling price of electricity; price of boiler fuel; and venting steam, such that, if the signal is positive the MPC should priorities the higher power output of the turbines than keeping the flow through the blow-out valve low.

One possibility could be to make the MPC to track a reference signal for the power output. Tests have shown that this could be difficult because the loads on the MP and LP headers are changing and this affects the power output of the steam turbine. So instead of tracking a reference it would have been preferable if the MPC could strive to maximize the power output. This could possibly be done by choosing the objective function in such way but this has not been looked into further by this thesis.

The stability of the system needs to be examined further. The simulations done on the system where the MPC controls the large steam network were done with an inner model which would need some improvements. If closed-loop stability is wanted the states of the inner model needs to be stabilizable and detectable, according to [Wan09]. For definition of stabilizability and detectability terms, see [GL00]. These pre-requisites require the measurement outputs to be chosen properly and the model that is going to be linearized needs to be stripped so that it only describes dynamics that are of concern for the MPC.

The complexity of the simulation model tends to grow during the work of adding more and more models. This made the simulations performed in Dymola more and more time consuming. If the inner model is large, i.e. consists of 40 - 50 state variables and the prediction horizon is long as 10 - 15, then it takes several minutes only to translate the model and prepare it for the simulation in Dymola. This could slow down the work of designing a controller and evokes the idea of exporting the process model from Dymola for using it in other simulation software such as Matlab/Simulink. This would also have the advantage of easier handling of the large variables of matrices and arrays. Dymola has a complicated way of presenting the content in matrices.

7 Remaining work

The multilevel steam network model has been built based on a reference pulp and paper plant's multilevel steam network. The components in the model are simplifications and approximations of the components of the real network and so is the first level in the control hierarchy. These components need to be configured and tuned even more thoroughly to get more realistic character of the systems dynamics.

For instance the controlled load boiler has some lag between changes on the fuel feed until the flow of produced steam changes and this timing is too rapid in the current model. If the dynamics in the boiler would have been more realistic then the advantages from using the disturbance forecast would have been more obvious.

All the different control loops with its related bias term also needs more thorough tuning before the characteristics of the pressure control get close to the reference steam network's pressure control. These are parameters like thresholds for when to use the by-pass valves and when to use the accumulator.

The measured process output signals should have some measurement noise added in order to simulate a more realistic environment for the observer. This would put some extra requirements on the design of the observer to make use of statistical information of the noise on each measured signal. The generated disturbance forecast could also have some noise added to its output signals such that the prediction would be more realistic and less exact the further into the future the estimates will occur.

The choice of signals to be measured by the MPC also needs to be examined further. These signals provide information needed for the MPC to make its internal model to reflect the state of the process model. It is crucial for the MPC to be able to estimate all state variables in its internal model in order to compute a suitable control signal.

Also the process model which is linearized could be reduced in prior. The sample period of MPC controller is much longer than the time constants of the majority of the dynamics in the process model. Therefore these dynamics could be removed before linearizing and this would result in a more stable MPC controller when risk for numerical problems is avoided.

Some additional models for describing even more of the reference multilevel needs to be added. A Modelica model for the prices of sold electricity and for boiler fuel needs to be added such that the MPC can be designed to strive to make the system work in the most profitable state.

In order to use the accumulator to mitigate fast changes in pressure it needs to be given the possibility to build up and to lose pressure depending on the situation. This can be achieved even though the MPC is trying to track a reference pressure for the accumulator by using a dead-zone-function. In this way the MPC controller will let the pressure move to a certain level before taking action.

8 Conclusion

It has been shown that the advantages of using feedforward to take preventative actions before the impact of a disturbance is essential when mitigating pressure deviations. When combining the feedforward feature with a controller which has the capability of controlling multiple actuators based on multiple measurement signals, a suitable control solution is formed. To top it all, the controller is optimizing the control signal on-the-fly to eliminate the control error with the ability of considering constraints on both the control signal and the process outputs when optimizing.

These are the primary advantages from using the MPC and there are others, such as the relatively intuitive way of tuning of the control parameters. Especially when comparing with the *ad hoc* solutions discussed in Section 4.2, the tuning of the MPC appear intuitive.

Simulations with the MPC controlling the multilevel steam network and its first level of control system should be considered as a hint on how interesting it would be to take a deeper look into the problem. The results of this thesis needs more work before it can constitute a base for a future implementation of a MPC on a multilevel steam network.

References

[Dym] *Dymola Dynamic Modeling Laboratory. User Manual Volume 1.* Version 13. Dassault Systèmes AB, 2012.

[Eks08] F. Ekström. *2004 005-40-4.0 Reglerteknisk beskrivning av ångnätsreglering och överordnad optimerande reglering.* Tech. rep. Solvina AB, 2008.

[GL00] T. Glad and L. Ljung. *Control Theory Multivariable and Nonlinear Methods.* Taylor and Francis, 2000. ISBN: 0-7484-0878-9.

[Jal06] J.-P. Jalkanen. The world's northernmost paper mill relies on model predictive controls. *Automation* **1** (2006), 20–21. URL: http://www.metso.com/Automation/magazinebank.nsf/Resource/aut_1_06_pages20_21/$File/aut_1_06_pages20_21.pdf.

[Lue97] D. G. Luenberger. *Optimization by Vector Space Methods.* 1st. New York, NY, USA: John Wiley & Sons, Inc., 1997. ISBN: 047118117X.

[Mac02] J. M. Maciejowski. *Predictive Control with constraints.* Pearson Education Limited 2002, 2002. ISBN: 0-201-39823-0.

[Maj05] Y. Majanne. Model predictive pressure control of steam networks. *Control Engineering Practice* **13**.12 (2005). Special Section on Power Plants and Power Systems Control, 1499 –1505. ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/j.conengprac.2005.03.008. URL: http://www.sciencedirect.com/science/article/pii/S0967066105000870.

[MAT11] MATLAB. *version 7.12.0 (R2011a).* Natick, Massachusetts: The MathWorks Inc., 2011.

[Res10] C. W. Ressel. "Modelling and Control of Feed Water Systems in a Pressurized Water Reactor". EX074/2010. MA thesis. Chalmers University of Technology, 2010.

[SL27] A. Stodola and L. Loewenstein. *Steam and gas turbines: with a supplement on The prospects of the thermal prime mover.* Steam and Gas Turbines: With a Supplement on The Prospects of the Thermal Prime Mover. McGraw-Hill, 1927. URL: http://books.google.se/books?id=CMuEAAAAIAAJ.

[Wan09] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB.* Springer, 2009. ISBN: 1-848-82330-4.

[WC78] D. A. Wismer and R Cattergy. *Introduction to nonlinear optimization: A problem solving approach*. North-Holland (New York), 1978. ISBN: 0444002340.

[WCD+00] W. Wagner, J. R. Cooper, A. Dittmann, et al. The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. *Journal of Engineering for Gas Turbines and Power* **122** (2000).

A Appendix

A.1 Model development and simulation environment

The most useful views in Dymola are; the graphical model presentation as can be seen in Figure A.1; the Modelica code of the model as in Figure A.2; and the plotting view where the results can be displayed in graphs as in Figure A.3.

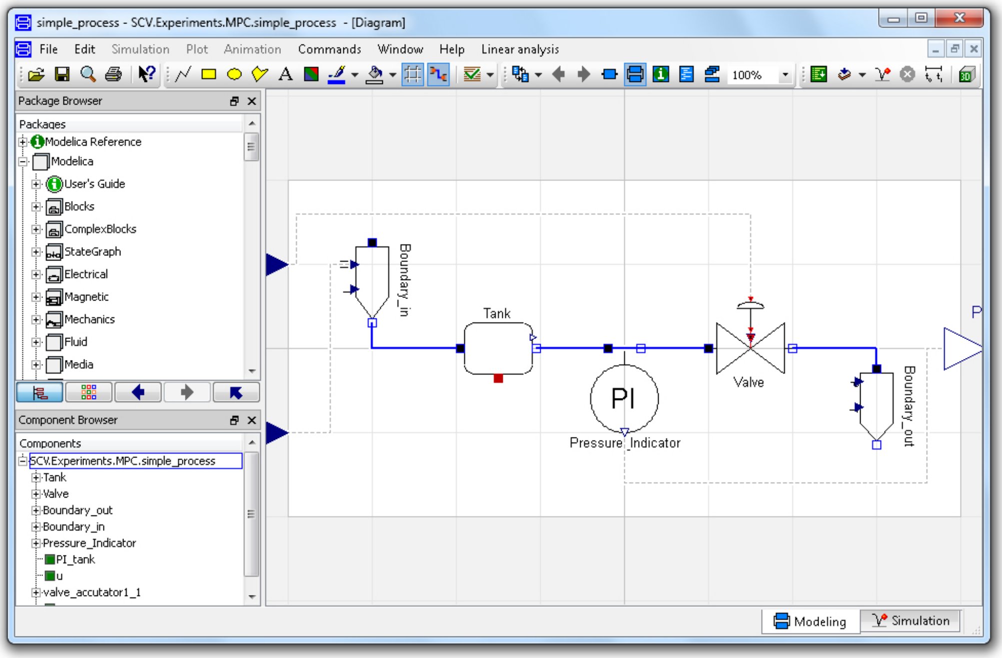


Figure A.1: *The figure is a screen shot from the graphical presentation of the Modelica model in Dymola.*

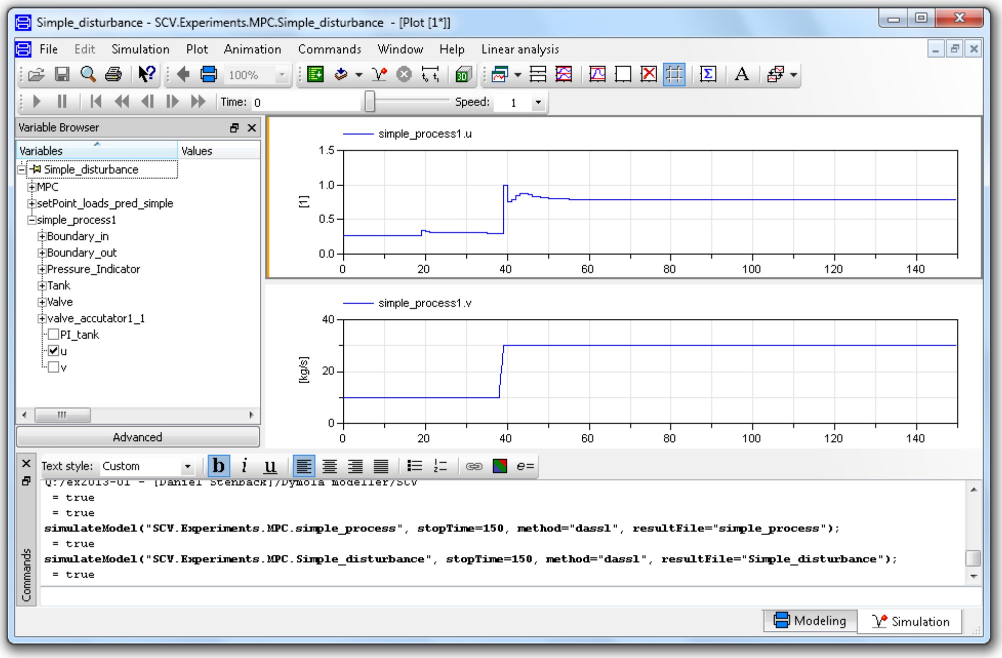


Figure A.2: *Screen shot of the plotting environment in Dymola. The variables of the simulation results are arranged in a tree structure in the column to the left.*

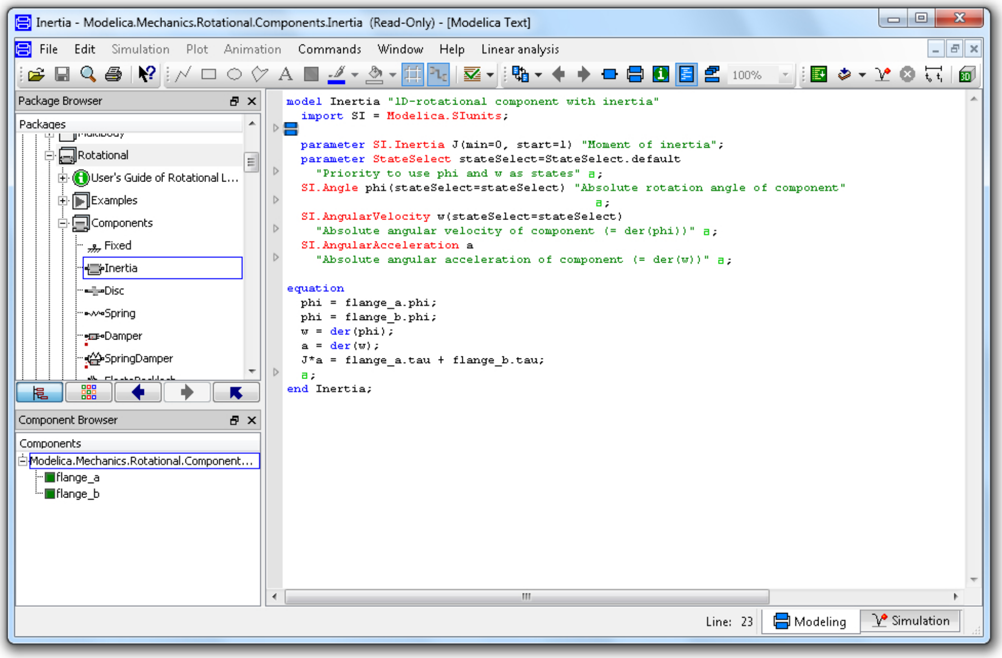


Figure A.3: *Screen shot of the Modelica code editor view in Dymola. The equations are describing the dynamics of a rotational component with inertia, e.g. the relation between the rotational angle and the torque.*

A.2 Modelica model interface for configuring the biases

In the dialog in Figure A.4 it is possible to edit the individual bias terms for the different controllers. The pressure controllers are arranged graphically as in Figure A.5. Based on which

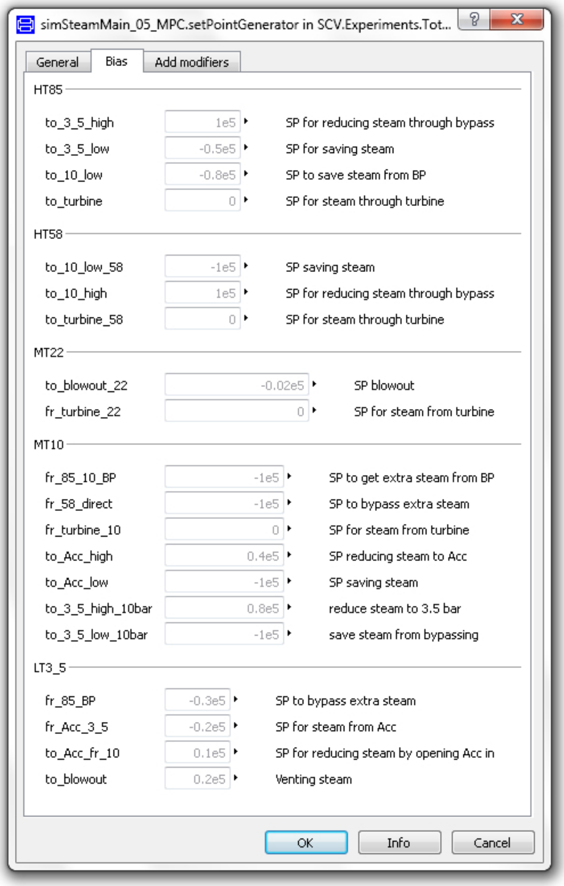


Figure A.4: *Interface for parameter configuration of the model constructing the reference signals.*

measurement signal that is used by the controller, it is placed vertically and its horizontal position determines of which actuator it is wired to.

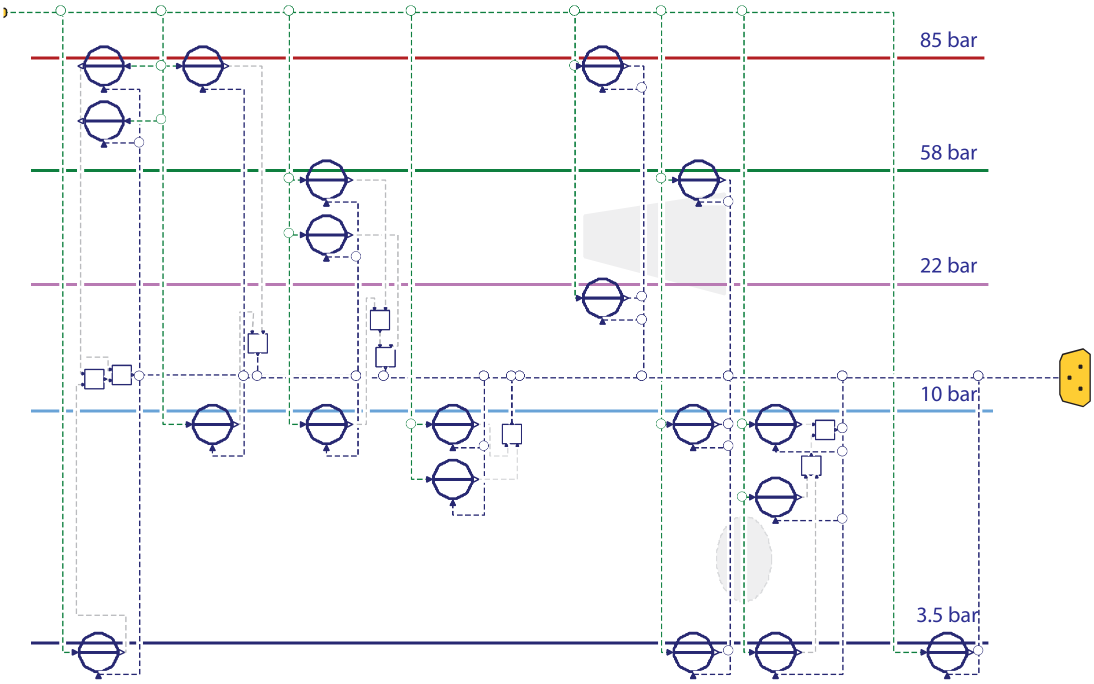


Figure A.5: *The graphical arrangement of the pressure controller models in the control container model. Each column contains every controller acting on the same valve. The horizontal coloured lines are representing the pressure header from which the controller gets its feedback signal from.*

A.3 Modelica code for Model Predictive Control

The input parameters of the MPC Modelica model is, among others, the state-space object, prediction horizon and control horizon. These parameters needs to be transformed into the matrices needed for the prediction and the optimizing of the control signal. This code is executed only once during the initialisation and can be viewed in Figure A.6.

The code for updating the variables needed at each sample and to prepare the input arguments to the quadratic programming solver can be seen in Figure A.7 on page 36. This code calls the function which declaration can be seen in Figure A.8 on page 37.

```
algorithm
  F := SCV.Experiments.MPC.Ctrl_utils.get_F(A,C,Np);
  Phi := SCV.Experiments.MPC.Ctrl_utils.BlockToeplitz_helper_Phi( A,B,C,Np,Nu);
  Psi_full := SCV.Experiments.MPC.Ctrl_utils.BlockToeplitz_helper_Phi_v2( A, V, C, Np, Np);

  // Object fcn equation (1.72) on page 35 chapter 1
  // J = (Rs – F*x_hat(ki))' * (Rs_bar*r(ki) – F*x_hat(ki)) – 2*DeltaU'*Phi' * (Rs – F * x_hat(ki)) + DeltaU' (Phi'*Phi + R_bar)*DeltaU,
  // R_bar weight matrix to pay any attention to how large the DeltaU are - according to page 9 in the book
  // Rs set point information for the hole optimization window
  // Rs_bar only the vector used to form Rs and is Np times the number of outputs long

  // R_bar & Rs
  R_bar := identity(Nu*size(B, 2));
  for i in 1:size(B, 2):Nu*size(B, 2) loop
    for j in 0:size(B, 2)-1 loop
      R_bar[i+j,i+j] := r_bar[1+j];
    end for;
  end for;

//  Rs_bar := ones(Np*size(C,1), 1);
  for i in 0:size(C,1):(Np*size(C,1)-1) loop
    for j in 1:size(C,1) loop
      Rs[i+j,1] := SP[j,1];// - y0[j];
    end for;
  end for;

  // E
  E := 2*(transpose(Phi)*Phi+R_bar);

  // F
  F1 := -2 *(transpose(Phi)*Rs);
  F2 := -2 *(transpose(Phi)*(-F));
  FV_full := -2*(transpose(Phi)*(-Psi_full));

  // Constraints matrix M
  M :=zeros(2*Nu*size(B, 2), Nu*size(B, 2));
  for i in 0:Nu-1 loop
    M[i*size(B,2)+1,1] :=1;
    M[i*size(B,2)+1+Nu*size(B,2),1] :=-1;
  end for;
  for r in 2:Nu*size(B,2) loop
    for c in 2:r loop
      M[r,c] :=M[r - 1, c - 1];
      M[r+Nu*size(B,2),c] :=M[r - 1 + Nu*size(B, 2), c - 1];
    end for;
  end for;
```

Figure A.6: *The Modelica code for the preparations of the matrices used when predicting future process outputs. This code runs only ones before the sampling starts.*

```
when sampleTrigger then
  // --- subtract linearization point ---
  v :=v_in - v0; y :=y_in - y0;
  Dv :=v[1] - pre_v[1];    pre_v[1] := v[1];

  // -- construct V_abs --
  pred_index := 2;
  if not startCtrl then // fill with holded values
    for ki in 1:Np*size(V,2) loop
      V_abs[ki,1] :=v[pred_index-1];
      if samplePeriod*(ki+1) >= v_pred_times[pred_index] and samplePeriod*(ki+1) <= v_pred_times[nv] then
        pred_index := pred_index + 1;
      end if;
    end for;
  else // not first time - prediction move forward one sample
    pred_index := 1;
    V_abs[1,1] :=v[pred_index];
    pred_index := pred_index + 1;
    for ki in 2:Np*size(V,2)-1 loop
      V_abs[ki,1] :=V_abs[ki+1, 1]; // copy one sample ahead
      if samplePeriod*(ki-1) >= v_pred_times[pred_index] and
         samplePeriod*(ki-1) <= v_pred_times[nv] then
        V_abs[ki,1] :=v[pred_index]; // overwright with the fresch forecast
        if pred_index < nv then
          pred_index := pred_index + 1;
        end if;
      elseif samplePeriod*(Np*size(V,2)) <= v_pred_times[pred_index] then
        V_abs[ki,1] :=v[pred_index-1]; // if there is no more forecast then fill with the latest
      end if;
    end for;
    if samplePeriod*(Np*size(V,2)-1) >= v_pred_times[pred_index] and
       samplePeriod*(Np*size(V,2)-1) <= v_pred_times[nv] then
      V_abs[Np*size(V,2),1] :=v[pred_index];
    else
      V_abs[Np*size(V,2),1] :=v[pred_index-1];
    end if;
  end if;

  // -- construct DeltaV --
  for ki in 2:Np*size(V,2) loop
    DeltaV[ki,1] := V_abs[ki,1] - V_abs[ki-1,1];
  end for;
  DeltaV[1,1] :=Dv;

  for i in 0:Nu-1 loop // calculate gamma matrix
    for k in 1:size(B,2) loop
      gamma[i*size(B,2)+k,1] :=umax[k] - u[k];
      gamma[i*size(B,2)+k+Nu*size(B,2),1] :=-umin[k] + u[k];
    end for;
  end for;

  // Update the model state variables
  x_hat :=vector(A*x_hat) + vector(B*Du) + vector(V*Dv_ob) + vector(Kob*(y - Z*x_hat));//
  Dv_ob :=v[1] - pre_v_ob[1];    pre_v_ob[1] := v[1];

  FF :=F1 + matrix(F2*x_hat) + matrix(FV_full*DeltaV);

  // Optimization of the control signal
  DU :=SCV.Experiments.MPC.Ctrl_utils.QuadProgramMIMO_simple(E, FF, M, gamma);
  Du :=DU[1:size(B, 2), 1];
  // Send out the new control signal
  u  :=u + Du;
  u_out := u + u0;
 end when;
```

Figure A.7: *The Modelica code that is executed each sample.*

```
function QuadProgramMIMO
  input Real E[:, size(E, 1)];
  input Real F[size(E, 1), 1];
  input Real M[:, size(E, 1)];
  input Real gamma[:,1];
  output Real DU[size(E, 1), 1];

protected
  Real H[size(gamma, 1), size(gamma, 1)];
  Real K[size(gamma, 1), 1];
  Real lambda[size(gamma, 1), 1];
  Real lambda_p[size(gamma, 1), 1];
  Real error;
  Integer k;
  Real w;
  Boolean ActiveConstraint=false;
  Boolean OptimNotDone=true;

algorithm
  DU := -matrix(Modelica.Math.Matrices.solve(E, vector(F)));
  for i in 1:size(gamma,1) loop
    if scalar(M[i,:]*DU) > gamma[i,1] then
      ActiveConstraint := true;
    end if;
  end for;

  if ActiveConstraint then
    H := M*Modelica.Math.Matrices.inv(E)*transpose(M);
    K := gamma + M*Modelica.Math.Matrices.inv(E)*F;
    lambda := zeros(size(gamma, 1), 1);

    for k in 1:40 loop
      if OptimNotDone then
        lambda_p := lambda;

        for i in 1:size(gamma, 1) loop
          w := (-1/H[i, i])*(K[i, 1] + scalar(H[i, :]*lambda) - H[i, i]*
            lambda[i, 1]);
          lambda[i, 1] := max(0, w);
        end for;

        error := scalar(transpose(lambda - lambda_p)*(lambda - lambda_p));
        if error <= 10^(-8) then
          OptimNotDone := false;
        end if;
      else
        break;
      end if;
    end for;
    assert(OptimNotDone, "Did not find a DU that minimizes @@@@@");

    //E * DU = -F-M'*lambda
    DU := -matrix(Modelica.Math.Matrices.solve(E,vector(F+transpose(M)*lambda)));
  end if;
end QuadProgramMIMO;
```

Figure A.8: *The Modelica code for the function that solves the quadratic optimization problem.*