CHALMERS GÖTEBORG UNIVERSITY

MASTER'S THESIS

Optimisation of fresh water consumption for Doggy AB using simulation

Maja Olsson Uppenberg Moa Ryberg

Department of Mathematics CHALMERS UNIVERSITY OF TECHNOLOGY GÖTEBORG UNIVERSITY Göteborg Sweden 2005 Thesis for the Degree of Master of Science

Optimisation of fresh water consumption for Doggy AB using simulation

Maja Olsson Uppenberg Moa Ryberg

CHALMERS | GÖTEBORG UNIVERSITY



Department of Mathematics Chalmers University of Technology and Göteborg University SE-412 96 Göteborg, Sweden Göteborg, December 2005

Matematiskt centrum Göteborg 2005

Abstract

Doggy AB is the only Swedish producer of tinned pet food. In the food process a large amount of cold water is needed at the stage when the boiled tins are cooled. Doggy's problem is to get sufficient amounts of cold water with a low enough temperature. The aim of the thesis is to minimise the usage of fresh water in the cooling process and to analyse the result.

We build a simulation model in Simulink, that describes the flows of water within the system of tanks supplying the cooling process with water. The optimisation problem is to find a set of parameters that gives the optimal outcome of the simulation. Our problems shows to be nonconvex and also nondifferentiable at some points.

For the optimisation, we use the built-in MATLAB function fmincon and TOMLAB with the solvers snopt, conSolvenlpSolve. The algorithms used in these solvers are all based on sequential quadratic programming but they use different methods to fulfill the constraints. The different methods for the constraints make the solvers give different results; some of the solvers prioritise feasibility, while some allow infeasible solutions.

Our conclusion is that if Doggy circulates the water in an optimal way, the only fresh water that needs to be added is the water lost in the system; i.e., no fresh water should be added for the sole reason of decreasing the temperature. The most important change we suggest is to use the cold fresh water as late as possible in the cooling process. With the above changes, the temperature of the cooling water can be lowered with up to $6.5 \,^{\circ}$ C.

Sammanfattning

Doggy AB är den enda svenska producenten av konserverat djurfoder. I konservmatsproduktionen behövs stora mängder kallt vatten då de kokta burkarna ska kylas. Doggys problem är att få tillräcklig mängd kallt vatten med tillräckligt låg temperatur. Målet med detta examensarbete är att minimera användningen av färskvatten och analysera resultatet.

Vi skapar en simulering i Simulink, som beskriver vattenflödena i det tanksystem som försörjer kylprocessen med kallvatten. Vårt optimeringsproblem är därmed att finna parametervärden som ger en optimal simulering. Vårt problem visar sig vara ickekonvext och icke-differentierbart i vissa punkter.

För att lösa optimeringsproblemet använder vi den inbyggda MATLAB-funktionen fmincon samt TOMLAB med olika lösare: snopt, conSolve och nlp-Solve. Algoritmerna som används i de olika lösarna bygger alla på sekventiell kvadratprogrammering, men använder olika metoder för att uppfylla bivillkoren. De olika metoderna för bivillkoren gör att lösarna ger olika resultat; vissa prioriterar tillåtenhet, medan andra tillåter otillåtenhet. Detta beror på skillnader i de straffunktioner som används, för bivillkoren, i algoritmerna.

Vår slutsats är att om Doggy cirkulerar vattnet på ett optimalt sätt, så kommer det enda färskvatten som behöver sättas in vara det som går förlorat i systemet; med andra ord ska inget färskvatten användas enbart för temperatursänkning. Den viktigaste förändringen vi föreslår är att tillsätta färskvattnet så sent som möjligt i kylprocessen. Med ovan nämnda förändringar kan temperaturen på kylvattnet sänkas med upp till 6.5 °C.

Acknowledgments

We would like to thank our supervisor Professor Michael Patriksson at the Department of Mathematical Sciences at Chalmers University of Technology and Göteborg University. You are always honest and cheer us up when we need it the most. You give us space to develop our own ideas and to use our knowledge, and, of course, thank you for the great pictures of the factory.

We would also like to thank Doggy AB, especially Torbjörn Pettersson, Göran Lundevall and Lars-Åke Larsson. You give us the information we need and have patience with us when our calculations seem to take forever.

We would also like to thank Anna Nyström for the cooperation with Simulink. Without your help to get started, we would probably still be working with the simulations.

We would also like to thank Inga Lasson at Alfa Laval for taking time to help us with heat exchanger calculations.

Finally we would like to thank John and Tony for being there for us when we need it the most. Thank you for taking care of us after long working days. Behind every successful woman, there is a supporting husband.

Contents

1	Introduction	1
	1.1 About Doggy AB	1
	1.2 The aim of the project	1
	1.3 Methods	2
	1.4 Outline	2
2	The production	2
	2.1 Tank 1	5
	2.2 Tank 2	6
	2.3 Tank 3	7
	2.4 Tank 4	7
	2.5 Tank C	8
	2.5.1 Dilution	9
	2.5.2 Night cooling	9
	2.5.3 Batch	9
3	The mathematical model	9
	3.1 General expressions	9
	3.2 Equations describing the system of tanks	10
	3.3 Additional constraints	13
	3.4 A summary of the problem	15
4	The simulation in Simulink	16
5	The optimisation	18
	5.1 Description of the problem	18
	5.2 The solvers EGO, rbfSolve and fminsearch	22
	5.3 SQP	22
	5.4 The built-in MATLAB function fmincon	23
	5.5 The TOMLAB solver snopt	24
	5.6 The TOMLAB solver conSolve	24
	5.7 The TOMLAB solver nlpSolve	24
6	Results	25
	6.1 Optimisation	26
	6.2 Improvements	33
7	Discussion	36
8	Further possible developments	37
Re	eferences	38
A	Heat balance	I
B	Overflow	П
С	Constants	ш

D Term descriptions

1 Introduction

1.1 About Doggy AB

Doggy AB is one of the main producers of pet food in Sweden. The factory is located in Vårgårda, 70 km north east of Gothenburg, and produces both pellets and tinned food. Doggy's factory makes about 70 % of the pellets and 100 % of the tinned pet food produced in Sweden. The primary market is Sweden, but Doggy also exports, mainly to England and Germany.^[Pet05] In the tin pro-



duction, one production cost stems from the input of extra water in the cooling of the boiled food. The aim of the project, which is made on behalf of Doggy, is to minimise the total amount of added water and to find its optimal use. The factory is divided into two sub factories: one for the pellets food and one for the tinned food. This thesis will only consider the tin production so from now on, we will focus on this part of the factory, which is split into two parts, one for the production of the original aluminium tins and one for a recently started Tetra Pak production line. The food is boiled in the tins (both the aluminium and Tetra Pak) and thereafter cooled for handling reasons. The cooling of the tins has to finish within 35 minutes, due to hygiene restrictions and time constraints. To achieve a sufficient cooling, large quantities of cold water are needed. The cooling water can be reused by going through a heat exchanger and a series of tanks. During the summer when the temperature in the system is generally higher, it is difficult to lower the temperature of the food sufficiently and efficiently. A simple but expensive solution to decrease the temperature of the cooling water is to add cold fresh water. Some fresh water must be inserted, since there is a loss of water in the production, but apart from that, excessive water will just be lost to a nearby narrow river. An optimisation of the circulation of the cooling water will bring down the usage of fresh water, which is the objective of this thesis.

The circulation of the water can be controlled by changing the parameters in the regulations of the system. The problem has several constraints: the cooling of the tins has to finish within 35 minutes (which can be achieved only if the temperature of the cooling water is below $20 \,^{\circ}$ C), the tanks are not allowed to run empty and there are pumps and pipes in the systems with limited capacities.

1.2 The aim of the project

The goal of this thesis is to minimise the waste of fresh water and to keep the cooling water at a low temperature, by finding an optimal circulation of the water in the tin production. Flows going into and out from the system can be rearranged as long as necessary constraints are fulfilled.

We will not consider very drastic solutions, for example significant rebuildings of the factory. Even very expensive solutions are ignored; even though some kind of electrical cooling system would work, the cost of it would vastly exceed the money saved from the decreased water usage.

1.3 Methods

In this project, a broad range of tasks has been necessary to perform: from measurements to advanced calculations. The first thing we did was getting an overview over the system by mapping it out. This part of the project was merely done at the factory where we measured flows, looked at blueprints over the tank system, and registered temperatures. We measured the flows with a flow measurement tool during a day with normal production.

The next part of the project is to model the system mathematically. The equations describing the problem are non-linear and time dependent and set up as a system of ordinary differential equations (ODEs). This system is very complex and it is too difficult, if not impossible, to solve it explicitly. Therefore, we perform a simulation of the problem to be used as a "black-box" in the optimisation. Our task is then to find the ingoing parameters to the simulation that result in the optimal circulation of the water. We get an optimisation problem where the constants and some variables are set by us, and other variables are obtained from the simulation. The modelling and simulation of the problem is made with the MATLAB toolbox Simulink and to solve the optimisation problem, MATLAB is used with different solvers. We have chosen to use the built-in MATLAB function fmincon, and also use three of TOMLAB's solvers: snopt, conSolve and nlpSolve.

1.4 Outline

In the next section a more detailed description of the production is presented, especially of the tanks and the cooling process. The equations describing the system are presented in Section 3, where we also discuss the properties of the problem. A description of the Simulink simulation is shown in Section 4. In Section 5 the optimisation problem is discussed and analysed and some solvers are presented. The differences between the solvers are also discussed. In Sections 6 and 7 the results of the optimisation are shown and discussed. Finally, in Section 8 further developments of the system are suggested. In Appendix A and B some derivations of mathematical expressions are shown; the constants used are found in Appendix C and in Appendix D some term descriptions are explained.

2 The production

Cat and dog food is made out of different kinds of meat and additives, for example vitamins. The ingredients are mixed and poured into tins before the boiling procedure takes place in an autoclave.^[Pet05] An autoclave is a closed system where physical reactions take place with a high temperature and under high pressure.^[NE05] To the left in Figure 1 below some autoclaves are shown. There are five autoclaves for the aluminium tins and three for the Tetra Pak packages. The picture to the right in Figure 1 shows the Tetra Paks on their way to distribution.



Figure 1: To the left, we can see the autoclaves for the aluminium tin production. The opened autoclaves are empty and the closed one is in use. To the right the boiled and cooled Tetra Pak packages are lined for distribution.

The tins are heated until the center of the tin has a temperature of about 130 °C (403 K), a process which takes approximately one and a half hours. The hot tins have to be cooled to about 30 °C for handling and packaging reasons. Post boiling, the water in the autoclave is lowered in temperature in a plate heat exchanger, called Heat Exchanger AC, see Figure 2.^[Pet05]



Figure 2: The dark boxes, lined on the floor, are the Heat Exchanger(s) AC.

The eight autoclaves have one such heat exchanger each, but we refer to them all as Heat Exchanger AC, since they are identical and assumed to run one at a time. A plate heat exchanger is a special type of heat exchanger where fluids of different temperatures are separated with plates. That way, the fluids never mix, but heat is transferred between them.

In one hour, up to two and a half boilings of aluminium tins and one boiling of Tetra Pak are carried out, and the production runs for 17 hours a day, five days

a week.

The production of pet food is subject to certain hygiene restrictions; therefore the cooling time is limited to 35 minutes, which can be achieved only when the temperature of the cold stream going into Heat Exchanger AC (CS_{in} in Figure 3) has a temperature of 20 °C or less.



Figure 3: The connection between the autoclaves, Heat Exchanger AC and the system of tanks. The arrows illustrate the water flow direction, where WS_{in} , WS_{out} , CS_{in} and CS_{out} is the warm stream in and out and the cold stream in and out respectively.

The stream CS_{in} is pumped from a tank, called Tank C, which belongs to a system of five water tanks with individual temperatures. When this stream has passed Heat Exchanger AC, it is substantially warmer and is led to one of the other tanks in the system. Figure 4 shows the connection between the autoclaves, the Heat Exchanger AC and the system of tanks.



Figure 4: A schematic illustration of the connection between the tanks and between the tank system and the autoclaves as the system is today. The arrowed lines show the direction of the water flows.

Since the water in the autoclaves gets cooler during the 35 minutes cooling process, the stream CS_{out} is warmer at the beginning of the process than towards the end. Hence, the temperature of this outgoing stream can be described as a function of time, plotted in Figure 5. One curve in the figure is plotted with experimental values on CS_{out} that we measured and the other curve is fitted by



Figure 5: The temperature curve of the water returning from Heat Exchanger AC to the system of tanks where the temperature is a function of time. The dashed curve shows the fitted curve.

regression. The curve fitting is made with the MATLAB function nlinfit. The equation for the fitted curve is represented as

$$T(t) = 303 + 86.1e^{\frac{-t}{370}},\tag{1}$$

where T is the temperature [K] and t is the time [s]. Which tank the outgoing water is led into is dependent on the temperature of the flow and therefore (according to the time dependence) on the time the cooling process has been running.

In the text, we have used $^{\circ}C$ as the unit for temperature, since it is easy to relate to, but in the calculations we use the SI-unit Kelvin.¹

2.1 Tank 1

In the first minutes of the cooling process, the water returning to the tank system, from Heat Exchanger AC, is very hot (up to $120 \degree$ C), see the left picture in Figure 6 below.

At this stage, the returning water goes into the first tank, Tank 1, until it has decreased in temperature to a certain preset temperature, T_{set_1} , when a valve opens to the next tank, Tank 2. It is easy to change the preset temperatures; they are connected to a digital regulator, shown to the right in Figure 6 below.

Some water from Tank 1 is used to produce steam, as shown in Figure 4. The steamer is also in use during the night and is at this time the only flow connected to this tank. This will cause a low water level in the morning. If the flow from Heat Exchanger AC exceeds the flow to the steamer, then the water level will increase and eventually water in Tank 1 will overflow into Tank 2.

 $^{^1}SI$ unit, with the designation K for thermo dynamic temperature (absolute temperature). 0 K is 273.15° C



Figure 6: To the right the opening of Tank 1. The water in this tank is hot; notice the steam coming out of the tank! To the right the display, where the temperature limits can be changed, is shown.

Tank 1 is not allowed to run empty since this could damage the pump to the steamer, so if the water level falls too low, a signal will activate an insertion of fresh water to fill the tank. Today, the net loss in the tank system is inserted in Tank 1. The ingoing and outgoing flows of Tank 1 are shown in Figure 7.



Figure 7: Tank 1 with its ingoing and outgoing flows.

2.2 Tank 2

The second warmest tank, Tank 2, has two incoming flows: water from Heat Exchanger AC and water overflowing from Tank 1. Valves switch the water from the autoclaves; it will be led into Tank 2 when its temperature is between T_{set_1} and T_{set_2} . Water is taken from Tank 2 to be used in the cleaning of the factory, even the part that produces pellets. The cleaning is also done during the night and is at this time the only flow connected to this tank. This will cause a low water level in the morning. This tank can also overflow, to Tank 3, when the water level is too high.

Tank 2 is not allowed to run empty (due to the pump for the cleaning) so if the water level is too low, a signal will activate an insertion of fresh water to fill the tank. The ingoing and outgoing flows of Tank 2 are shown in Figure 8.



Figure 8: Tank 2 with its ingoing and outgoing flows.

2.3 Tank 3

The water coming from Heat Exchanger AC, which has a temperature between T_{set_2} and the next limit, T_{set_3} , is led into the largest tank, Tank 3. The only flow leaving Tank 3 is the water overflowing to the next tank, Tank 4. Another flow of water going into Tank 3 is a flow coming from the air conditioning of the computer rooms. The ingoing and outgoing flows of Tank 3 are shown in Figure 9.



Figure 9: Tank 3 with its ingoing and outgoing flows.

2.4 Tank 4

At the end of the cooling process, when the temperature of the stream from Heat Exchanger AC is lower than T_{set_3} , the stream goes to Tank 4, which is the last tank in the series with incoming water from the cooling process.

The water from the tanks is reused in the cooling process, where the cooling water has to have a temperature below 20 °C. To achieve this, the water in Tank 4 is pumped through a heat exchanger, called Heat Exchanger TS. Note that Heat Exchanger TS has nothing to do with the heat exchangers connected to the autoclaves, Heat exchanger AC. The cooled water going out of Heat Exchanger TS goes into a tank, Tank C, from which water is taken to the cooling of the autoclaves; see Figure 4. The cold stream in Heat Exchanger TS is water from a nearby narrow river. One of the reasons that the cooling is less efficient in the summer is the relatively high temperature of the river at that time of the year. Heat Exchanger TS is shown in Figure 10 below.

Water flows into Tank 4 from Tank 3 and Tank C. When Tank 4 fills up to the brim, water overflows out to the river, and is thereby lost. Tank 4 has an incoming flow from a process in the pellets factory.

A tap is connected to Tank C to be opened whenever the level of Tank 4 is too low. The ingoing and outgoing flows of Tank 4 are shown in Figure 11.

A high temperature of the river gives a negative result in the cooling process in two ways: the temperature gradient in Heat Exchanger TS is small (because



Figure 10: To the left, we can see Heat Exchanger TS and to the right the plates are visible.



Figure 11: Tank 4 with its ingoing and outgoing flows.

the temperature difference between the two streams is small), which decreases the cooling effect of the water and at the same time algae create an isolating layer on the plates in the heat exchanger which decreases its efficiency.

2.5 Tank C

Ideally, the water going out of Heat Exchanger TS into Tank C is below 20 $^{\circ}$ C. As previously stated, this is not always the case, especially in the summer. There are several options to lower the temperature of the water going to the cooling process; dilution, night cooling and batch; see the next subsections.

A tap is connected to this tank and is opened whenever the level of Tank 4 is too low, since Tank C overflows into Tank 4. The ingoing and outgoing flows of Tank C are shown in Figure 12.



Figure 12: Tank C with its ingoing and outgoing flows.

2.5.1 Dilution

A fast option to cool the flow going from Tank C to the Heat Exchanger AC is to dilute it with cold fresh water. The fresh water is added directly into the pipe leading to the autoclaves. The result is an immediate lowering of the temperature, but it can be necessary to add large amounts of fresh water.

2.5.2 Night cooling

Since the tanks are connected to each other and overflow into each other, a low starting temperature in any tank is beneficial for the cooling. To achieve this, the water in Tank 3 can be cooled over night, when the production is not running. When this is done, a valve redirects the flow to Heat Exchanger TS so that water from Tank 3 is cooled against the water in the river, see Figure 13. Possibly, a continuous flow through the heat exchanger can also prevent the growth of algae.



Figure 13: When the factory uses night cooling, a valve switches the ingoing flow to Heat Exchanger TS so that the water in Tank 3 is cooled.

2.5.3 Batch

The tap opened at a low level of Tank 4 can also be used when the temperature in Tank C is high, to add cold fresh water. A timer is connected to a button, so when the button is pushed, water will be added into Tank C for 30 minutes. The button is only used for emergency purposes, when all other methods to cool the water are insufficient.

3 The mathematical model

3.1 General expressions

The properties of the system can be modelled as a system of time dependent equations. ODEs describing the changes in mass and temperature of the tanks are derived by writing mass and heat balances. The general balances over a tank are written as

$$\frac{dm}{dt} = \sum_{i} W_{in_i} - \sum_{j} W_{out_j},\tag{2}$$

$$\frac{dT}{dt} = \frac{\sum_{i} W_{in_i}(T_{in_i} - T)}{m},\tag{3}$$

where $\sum_{i} W_{in_i}$ and $\sum_{j} W_{out_j}$ are the sums of water flows going in and out of the tank, T_{in_i} is the temperature of the i:th ingoing flow and T is the temperature of the water in the tank. The derivation of the heat balance equation is shown in Appendix A; the derivation of the mass balance equation is analogous.

If equation (2) is divided by ρA it can be rewritten as

$$\frac{dh}{dt} = \frac{\sum_{i} W_{in_i} - \sum_{j} W_{out_j}}{\rho A},\tag{4}$$

where h is the height of the water level, ρ is the density of water [kg/m³] and A is the bottom area [m²] of the tank. The height of the water level is compared to the height of the pipe leading to the next tank, H. If h is larger than H, water will overflow, see Figure 14.



Figure 14: A tank viewed from the side. h is the current water level, H is the height of the pipe leading to the next tank and h_{min} is the minimum water level allowed.

In this case, the pressure of the water at H will be large when the difference h - H is large, and with increased pressure, the outgoing flow will increase:

$$W_{overflow} = w_{overflow} A_{pipe} \rho \mu, \tag{5}$$

where

$$w_{overflow} = \sqrt{2g(h-H)}.$$

The area of the pipe is A_{pipe} ([m²]) and μ is a dimensionless factor used to calculate the decrease in flow when the edges of the pipe are not smooth. The value of μ can be between 0 and 1, but is typically around 0.6 for sharp edges.^[EG94] A derivation of equation (5) is shown in Appendix B.

3.2 Equations describing the system of tanks

The objective function is to minimise the total amount of added fresh water $(m_{fresh} [kg])$:

minimise
$$m_{fresh}$$
. (6)

The sum of the water added into the pipe (W_{dilute} [kg/s]) and the water added directly into Tank 1, Tank 2 and Tank C (W_{extra1} , W_{extra2} and W_{extraC} [kg/s])

gives the total amount of fresh water added to the system from a given starting time, τ_0 , to some finite time τ :

$$m_{fresh} = \int_{t=\tau_0}^{\tau} (W_{dilute} + W_{extra1} + W_{extra2} + W_{extraC}) dt.$$
(7)

The necessary mass flow of water to add in the pipe at a too high temperature can be calculated as

$$W_{dilute} = \frac{W_{ac}T_{ac} - W_{C_{out}}T_C}{T_{fresh}},\tag{8}$$

where W_{ac} [kg/s] is the flow needed in the cooling process (illustrated as WS_{in} in Figure 3), T_{ac} [K] is the temperature of this flow, $W_{C_{out}}$ [kg/s] is the mass flow pumped from Tank C, T_C [K] is the temperature of Tank C, and T_{fresh} [K] is the temperature of the fresh water.

To get a fast enough cooling of the autoclaves, there is a certain demand of cooling water. The demand from the autoclaves for the aluminium tins $(D_{ac_{tin}} [kg/s])$ is different from the demand from the Tetra Pak autoclaves $(D_{ac_{tetra}} [kg/s])$. The demands are calculated assuming a fixed number of boilings per hour, and the flow to the autoclaves is assumed to be exactly the demand:

$$W_{ac} = D_{ac_{tin}} + D_{ac_{tetra}},\tag{9}$$

where

$$W_{ac} = W_{dilute} + W_{Cout}.$$
 (10)

Equations (2) and (3) are used to calculate the mass and heat balances for Tank C and give

$$\frac{dm_C}{dt} = W_{cool} - W_{C4} - W_{C_{out}} + W_{extra},$$
(11)

$$\frac{dT_C}{dt} = \frac{W_{cool}(T_{coolout} - T_C) + W_{extra}(T_{fresh} - T_C)}{m_C},$$
(12)

where W_{cool} [kg/s] is the flow of water going through Heat Exchanger TS and W_{C4} [kg/s] is the water flow overflowing from Tank C to Tank 4. $T_{coolout}$ [kg/s] is the temperature of the outgoing water from Heat Exchanger TS.

There are no exact relations between the in- and outgoing temperatures and flows for plate heat exchangers that can be used in our simulations. The model for Heat Exchanger TS is therefore derived from calculations done by Alfa Laval. How the simulations of the heat exchanger are done is unfortunately not known to us, since the simulation programs are confidential. Alfa Laval's results are calculated with variable values of the flow of the warm stream (which is called W_{cool} , since it is the water used to cool the autoclaves). The values of the flow of the cold stream and of the ingoing temperature are kept constant in each simulation, but some different cases, with different values on these parameters, are tested.^[Las05] The equation for the temperature of the outgoing cooling water, calculated via curve fitting, is

$$T_{coolout} = 287 + (0.0271T_{in} - 7.8128)W_{cool}.$$
(13)

In Figure 15 the temperature of the outgoing flow from the heat exchanger is plotted against the flow of the warm stream. The x-axis corresponds to the flow through the heat exchanger in the interval between the lower and upper bounds. The lower bound is the demand to Heat Exchanger AC from Tank C and the upper bound is given by the pump capacity. The y-axis is the temperature of the outgoing water when the water is exchanged against 15 °C cold water. (If the river would be at 17 °C, the temperature of the outgoing warm stream from Heat Exchanger TS would be about 1 °C higher for all x-values in the interval.)



Figure 15: The temperature of the outgoing flow from the heat exchanger plotted against the flow of the warm stream.

The calculation of the mass and temperature changes in Tank 4 are analogous to the ones in Tank C:

$$\frac{dm_4}{dt} = W_{4_{in}} + W_{34} + W_{C4} + W_{pellets} - W_{cool} - W_{4_{river}} - W_{computer}, \quad (14)$$

$$\frac{dT_4}{dt} = \frac{W_{4_{in}}(T_{4_{in}} - T_4) + W_{34}(T_3 - T_4) + W_{C4}(T_C - T_4) + W_{pellets}(T_{pellets} - T_4)}{m_4},$$
(15)

where W_{4in} [kg/s] is the flow of water coming from the autoclaves, T_{4in} [K] is the temperature of this flow, $W_{pellets}$ [kg/s] is the flow of water coming from the pellets factory, $W_{computer}$ is the flow of the water going to the air conditioning of the computer room, $T_{pellets}$ [K] is the temperature of the flow from the pellets factory, and $W_{4_{river}}$ [kg/s] is the water flowing out to the river. T_3 [K] and T_4 [K] are the temperatures of the water in Tank 3 and Tank 4, respectively.

The calculations of the mass and temperature changes in Tank 3, 2 and 1 are analogous to the above tanks. For Tank 3, we have that

$$\frac{dm_3}{dt} = W_{3_{in}} + W_{23} - W_{34} + W_{computer},\tag{16}$$

$$\frac{dT_3}{dt} = \frac{W_{3_{in}}(T_{3_{in}} - T_3) + W_{23}(T_2 - T_3) + W_{computer}(T_5 - T_3)}{m_3}.$$
 (17)

In the equations for Tank 2, the term describing the flow used in the cleaning of the factory is included (W_{clean} [kg/s]):

$$\frac{dm_2}{dt} = W_{2_{in}} + W_{12} + W_{extra2} - W_{clean} - W_{23},\tag{18}$$

$$\frac{dT_2}{dt} = \frac{W_{2_{in}}(T_{2_{in}} - T_2) + W_{12}(T_1 - T_2) + W_{extra2}(T_{fresh} - T_2)}{m_2}.$$
 (19)

The equations for Tank 1 are analogous to the above calculations, except that the hot water from Tank 1 is used for the steamer (W_{steam} [kg/s]):

$$\frac{dm_1}{dt} = W_{1_{in}} + W_{extra1} - W_{steam} - W_{12},$$
(20)

$$\frac{dT_1}{dt} = \frac{W_{1_{in}}(T_{1_{in}} - T_1) + W_{extra1}(T_{fresh} - T_1)}{m_1}.$$
(21)

The equations (1)–(21) represent the system as it is today. In this system, Tank 2 runs empty in the evening since a lot of water is used in the cleaning process. The low water level gives a low starting temperature in the tank since fresh water is added to increase the height, which also results in cold cleaning water. To solve this problem, we take the water for the cleaning process from Tank 3 instead, which will change equations (16) and (18) to:

$$\frac{dm_3}{dt} = W_{3_{in}} + W_{23} - W_{34} + W_{computer} - W_{clean} + W_{extra_3}, \qquad (16')$$

$$\frac{dm_2}{dt} = W_{2_{in}} + W_{12} + W_{extra2} - W_{23}.$$
(18)

Figure 16 illustrates where each equation belongs in the system. There are two equations for every tank, one for the change in mass and one for the change in temperature. Two systems are represented here: the system as it looks today and the system where the insertion of fresh water is changes as well as the outtake of the cleaning water.

3.3 Additional constraints

The water level in each tank $(h_i \text{ [m]})$ is not allowed to ever be below a certain level $(h_{i_{min}} \text{ [m]})$, because the pumps in the system must not run empty:

$$h_i \ge h_{i_{min}}, \ i = 1, ..., 4.$$
 (22)

The flows going through the pumps are limited by the capacities of the pumps $(C_{pump} \text{ [kg/s]})$, which gives upper bounds on W_{cool} and W_{fresh} :

$$W_{cool} \le C_{pump_{cool}},\tag{23}$$



Figure 16: The tank system with the equations marked out. equation for the mass and one equation for the temperature. Each tank has one

$$W_{fresh} \le C_{pump_{fresh}}.$$
(24)

If Tank 4 does not have a low water level, extra water will not be added to Tank C; therefore, to guarantee that Tank C is never emptied [see equation (11)], the following constraint is introduced:

$$W_{cool} \ge W_{Cout},$$

which in combination with equations (9) and (10) gives:

$$W_{dilute} + W_{cool} \ge D_{ac_{tin}} + D_{ac_{tetra}}.$$
(25)

All flows need to be positive:

$$W_i \geq 0, \ j \in J,$$

where J is the set containing all flows in the system.

The water going out from Heat Exchanger AC must go to the different tanks in order, i.e., the water from the cooling must go to Tank 1 first, then to Tank 2, then to Tank 3 and finally to Tank 4; this gives

$$T_{set_1} \ge T_{set_2} \ge T_{set_3}. \tag{26}$$

Finally, the cooling water must not have a temperature exceeding 293 K:

$$T_{ac} \le 293,\tag{27}$$

where

$$T_{ac} = \frac{W_{dilute}T_{fresh} + W_{C_{out}}T_C}{W_{ac}}$$

is calculated from equation (8).

For all values of the constants, see Table 7 in Appendix C.

3.4 A summary of the problem

The variables in the equations describing the problem are connected to each other in rather intriguing ways, due to the complexity of the real system. Hence the system of equations is probably impossible to solve explicitly.

Most of the flows are discontinuous; the discontinuities on the time dependence are easy to see (consider for example the flows going into the tanks from the autoclaves) but there are also discontinuities depending on other variables. To realise this, one can look at what happens when a tank overflows; this is not a direct function of time, but can still cause discontinuities in other flows. One possible for the discontinuities are the flows: they are either on or off and they will either overflow or not.

The properties of the objective function and the constraints will be discussed further in Section 5.1.

4 The simulation in Simulink

Since the system of equations is hard to solve, we build a simulation model with the MATLAB toolbox Simulink. Simulink uses a graphical user interface for building models and making simulations.

In Simulink, it is possible to build subsystems and connect them to each other to create a larger system. The system containing all subsystems is the root level. In the simulation only one tank at a time gets water from the autoclaves, and the temperature of that water is time dependent. To model this, we use a signal with four pulses, one for each tank, and a ramp signal. The pulse signal and the ramp signal can be seen in Figure 17 (the root level). The insertion of fresh water and the regulation of the water from Heat Exchanger AC are calculated in the root level.



Figure 17: The root level of the Simulink model.

Each tank is represented as a subsystem, where the change in mass, temperatures and heights are calculated with equations (2), (3) and (4) respectively. There is one subsystem for each tank, one for Heat Exchanger TS, and one for the insertion of fresh water into Tank C when Tank 4 has a low water level, see Figure 18.



Figure 18: The tank system modelled in Simulink.

At the root level, the subsystem containing the tanks is called Tanksystem. Figure 19 shows how Tank 4 is modelled; this is the subsystem called Tank4 in Figure 18.



Figure 19: The subsystem modelling Tank 4.

The subsystem, Overflow 4, in Tank 4 calculates the water level; see Figure 20.

In Figure 20, the first property calculated is h, which is compared to h_{min} . If $h \leq h_{min}$ a signal to Tank C will be activated to add fresh water. This way, we control the water level in the simulation and therefore we do not express it explicitly in the optimisation problem. If $h \geq h_{min}$ the water level is compared to H and the tank will overflow if $h \geq H$.

The arrows in the Simulink models are signals that carry information. In our



Figure 20: The calculation of the water level; the water can overflow to the river, nothing happens, or it will activate a signal to add fresh water in Tank 5.

system, the signals are mostly scalars or vectors (containing the values of flow, temperature and water levels) but can also be boolean.

The runtime of the simulation (which is τ in equation (7)) is an important parameter that has to be chosen with care. Interesting choices to consider are: one day, one hour or one boiling (35 minutes). One obvious drawback with long simulations is that the time to find an optimal value is going to be considerably longer. A long simulation time will also give longer time steps which will decrease the accuracy of the calculations. On the other hand, a short simulation time may lose important information about what happens when the system has been running for some time. In the optimisation, we have chosen to run the simulation corresponding to 10 boilings (21 000 s or 5 h 50 min) and compare with the results from one simulation with a run time of 17 hours (62 000 s) to see if the system is stabilised after 10 boilings. They are comparable and shown in Section 6.1. One simulation with a runtime of 21 000 s takes between 15 s and 15 minutes, depending on the steplengths.

Before Simulink is started, all the necessary parameters are defined in MAT-LAB. The outcome of the simulation is dependent on, and sensitive to, the variables and constants chosen. For a given set of parameters, the outcome of the simulation is unique. For some values of the parameters, the simulation does not work. The pulse signals, for example, must have positive width. For the values of the constants, see Appendix C.

5 The optimisation

5.1 Description of the problem

The simulation in Simulink is used as a "black-box" in the optimisation. As mentioned before, the objective function is to minimise the amount of inserted water into the cooling process. The problem is to find a set of parameters that gives optimal output values from Simulink. We will divide the variables into two different groups: the ones we can control and the ones obtained from the simulation.

The variables that we are optimising over are the ones we can control:

- W_{cool} the flow through Heat Exchanger TS;
- T_{set3} the temperature limit deciding at what temperature the water from Heat Exchanger AC is led into Tank 4; and
- W_{dilute} the flow of water added in the pipe to the autoclaves.

These variables are denoted by **x**. The dimension of **x** will either be two or three, since we sometimes have a fixed flow of W_{cool} . The variables, **x**, are constant in time. Another possibility is to have T_{set1} and T_{set2} as variables, but we have chosen not to, since this makes the problem too large. We have chosen to use the same temperatures for T_{set1} and T_{set2} as Doggy uses today: 353 K and 328 K respectively.

Some of the variables important to us are obtained through the simulation. One of these variables is the flow of water that overflows to the river, $W_{4_{river}}$, and the flow inserted to the tanks whenever the water level is too low, W_{extra_i} (i = 1, 2, C). The variables, which we cannot control, **y**, are defined by the set Sim(x). The variables, **x**, are continuous, but the mapping $\mathbf{x} \mapsto Sim(\mathbf{x})$ is discontinuous. Consider, for example, the case when W_{dilute} is small; then there will be a larger insertion of fresh water into Tank C, due to low water levels in Tank 4. Since the insertion of fresh water is either on or off, W_{extra} is discontinuous.

The optimisation problem can be written as that to

$$minimise f(\mathbf{x}, \mathbf{y}), \tag{28a}$$

subject to
$$c(\mathbf{x}, \mathbf{y}) \le 0$$
, (28b)

$$\mathbf{q}(\mathbf{x},\mathbf{y}) = \mathbf{0},\tag{28c}$$

$$\mathbf{a} \le \mathbf{x} \le \mathbf{b},\tag{28d}$$

The inequality constraint (28b) represents the temperature constraint (27). The temperature of the water flow to the autoclaves, T_{ac} , is given by the simulation; hence this temperature depends on the simulation results ($T_{ac} \in Sim(\mathbf{x})$). The temperature is not allowed to be over 20 °C on the entire simulation; the constraint is expressed as:

$$c(\mathbf{x}, \mathbf{y}) := \max_{a \in S} T_{a c_s} - 293 \le 0.$$
(27')

If T_{ac} exceeds 293 K, the cooling process will take longer time, but it will not cause any major damage. Therefore, constraint (27') is "soft".

The equality constraints (28c) represent the equations (11)–(21).

The lower and upper bounds on **x** are **a** and **b** (equation (28d)). These bounds are important to fulfill, because the mapping $\mathbf{x} \mapsto Sim(\mathbf{x})$ is undefined outside some of the bounds, and because some bounds are set by capacities that cannot be changed. Hence, the lower and upper bounds are "hard" constraints.

The system of equations (11)–(21) can be used to express y as a function of x. Due to the complexity of the system of equations, y is obtained through the simulation, which for every given x gives a unique y. Hence we can write

 $\mathbf{y} = Sim(\mathbf{x})$. Important is that the mapping $\mathbf{y} \mapsto Sim(\mathbf{x})$ is defined only when $\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$. In the optimisation, we do not handle equations (11)–(21) explicitly, since they will be calculated and (approximately) fulfilled in the simulation. Thus, we eliminate constraint (28c). When \mathbf{y} is substituted with $Sim(\mathbf{x})$, the optimisation problem can be rewritten as that to

$$\begin{array}{ll} \text{minimise} & f(\mathbf{x}, Sim(\mathbf{x})), \\ \text{subject to} & c(\mathbf{x}, Sim(\mathbf{x})) \leq 0, \\ & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}. \end{array}$$
(29)

We rewrite the problem so that it becomes dependent of **x** only:

$$\begin{array}{ll} \text{minimise} & f(\mathbf{x}), \\ \text{subject to} & \hat{c}(\mathbf{x}) \leq 0, \\ & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}, \end{array} \tag{30}$$

where $\hat{f}(\mathbf{x}) := f(\mathbf{x}, Sim(\mathbf{x}))$ and $\hat{c}(\mathbf{x}) := c(\mathbf{x}, Sim(\mathbf{x}))$.

We have chosen to include the integral of $W_{4_{river}}$ in our objective function because the water which is lost from the system must sooner or later be replaced. The modified expression for the objective function (6) becomes:

$$f(\mathbf{x}) = \int_{t=\tau_0}^{\tau} (W_{dilute} + W_{extra1} + W_{extra2} + W_{extraC}) dt + \int_{t=\tau_0}^{\tau} W_{4_{river}} dt.$$
(6')

The integrals in equation (6') is substituted with the sum over all time steps in the simulation:

$$\int_{t=\tau_0}^{\tau} W_{4river} dt \approx \sum_{s \in S} W_{4river_s},$$

and

$$m_{fresh} = \int_{t=\tau_0}^{\tau} (W_{dilute} + W_{extra1} + W_{extra2} + W_{extraC}) dt \approx \sum_{s \in S} (W_{dilute} + W_{extra1} + W_{extra2} + W_{extraC})$$

where S is the set of all time steps in the Simulink simulation.

In Simulink, we use a pulse generator that needs the pulses to have positive widths. In the calculations, we therefore need to have strict inequalities on constraint (26). T_{set_1} and T_{set_2} are constant in the simulation. We modify constraint (26) to:

$$T_{set_2} \ge T_{set_3} + 4.$$
 (26')

In some of the tanks, there are lower limits for the water levels. These constraints have been modelled so that water is added to the tanks when the water level sinks below these limits in the simulation. It is possible to model these constraints so that the algorithm only allows solutions where the water levels never sink too low. Unfortunately, to implement the level constraints in the optimisation has several drawbacks: the algorithm would be much slower, the emergency adding of water is a better model of how the system works in reality, and most importantly: we might want to see solutions where the level constraints are violated for short periods of time, if the rest of the solution is good. If a level constraint is violated, but the solution is good, it is easy to calculate how much extra water would be needed, and add it to the optimal solution; and maybe still have a good solution!

The simulation is used in the objective function; in each step of the iteration the algorithm evaluates the objective value by running the Simulink model. It is now obvious that the derivatives of the objective function and of the constraints cannot be calculated analytically; some kind of numerical procedure is necessary for the calculation of the gradients and Hessians.

The outcome of the simulation is very sensitive to the values of the constants and the choice of parameters, as well as the initial values for all the variables, both x and y. Hence, it is of great importance to use accurate initial values.

To test the convexity and the differentiability of the problem, we run the simulation with different values of T_{set3} and W_{dilute} . The result is shown to the left in Figure 21, where the objective function is plotted against W_{dilute} and T_{set3} .



Figure 21: The plot to the left, shows the non-convexity of the problem. The plot to the right shows that the problem is infeasible for low values of W_{dilute} .

It becomes quite clear that the system is non-convex and non-differentiable. From the figure, we can see that the region with low values of W_{dilute} looks differentiable and maybe even convex. To see which solutions are feasible (that is, where max $T_{ac_s} \leq 293$) we add a large penalty for the infeasible solutions, seen to the right in Figure 21. We can see that the region with low values of the objective function have infeasible temperatures, since this part of the plot changed when the penalty was added.

Infeasible solutions can still be interesting when the temperatures are moderately over the allowed limit, if the problem is stable and the value of the objective function is much better than in the feasible regions. Therefore, we will study this smaller region closer, which is shown in Figure 22. To the left the objective function without penalty is plotted, and to the right the objective function with a high penalty.

The function plotted in Figure 22 is convex everywhere except for at low values of T_{set3} and W_{dilute} . This is the region where we expect to find the optimal solution.



Figure 22: A plot showing that the problem has a region where it is both convex and differentiable.

For the optimisation, we use MATLAB together with different solvers. We have used the built-in MATLAB function fmincon and a software called TOM-LAB with the solvers snopt, conSolve and nlpSolve. The solver fmincon and many of the solvers in TOMLAB are based on the general optimisation method of sequential quadratic programming (SQP). SQP and the different solvers are described below. If we do not state anything else, the optimisation runs the system where the outtake of the water for the cleaning is moved from Tank 2 to Tank 3.

5.2 The solvers EGO, rbfSolve and fminsearch

The TOMLAB solver EGO is designed specially for problems that are costly to solve, for example those that involve simulations, and it does not use derivatives. It would therefore be very well suited for our problem. Another interesting solver is rbfSolve in TOMLAB. It is, as EGO, designed to find globally optimal solutions also for nonconvex optimisation problems and does not require derivatives. Unfortunately, we do not have licences for these two solvers so we are forced to use some SQP-based solvers.

The solver fminsearch uses a search method, akin to the classic Nelder-Mead algorithm, to find an optimal solution; new iteration points are tried through a procedure that generates trial points conforming to certain geometrical patterns. The objective value at a trial point is calculated to see if it is better than the previously generated ones. In contrast to SQP, it only uses the objective value to find the new iteration point. This method needs too many function evaluations for our problem, so the running time becomes too long.

5.3 SQP

For solving nonlinearly constrained problems, SQP is a popular method. The approach is to find a search direction by solving quadratic programs (problems with a quadratic function and linear constraints). It is a generalisation of Newton's method for unconstrained minimisation. Our problem (30) can be written

$$\begin{array}{ll} \text{minimise}_{\mathbf{x}} & \hat{f}(\mathbf{x}), \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{array} \tag{31}$$

where

as:

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} \hat{c}(\mathbf{x}) \\ -\mathbf{x} + \mathbf{a} \\ \mathbf{x} - \mathbf{b} \end{pmatrix}.$$
 (32)

Remember that $\hat{c}(\mathbf{x}) = \max_{s \in S} T_{ac_s} - 293.$

The Lagrangian to problem (31) is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \hat{f}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}), \qquad (33)$$

where $\lambda \in \mathbf{R}^{2n+1}_+$ (due to the number of constraints: 2n + 1) is the vector of Lagrange multipliers for the constraints (32).

The quadratic programming problem can be written as:

minimise_p
$$\frac{1}{2} \mathbf{p}^T \nabla_{xx}^2 L(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p} + \nabla_x L(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p},$$

subject to $\mathbf{g}(\mathbf{x}_k) + \nabla \mathbf{g}(\mathbf{x}_k)^T \mathbf{p} \leq \mathbf{0},$ (34)

where the Lagrange multipliers to the constraints of (34) are denoted by \mathbf{v}_k . This optimisation problem is a quadratic program, where the quadratic function is a Taylor series approximation of the Lagrangian (33) at $(\mathbf{x}_k, \lambda_k)$ and the constraints are a linear approximation to $\mathbf{g}(\mathbf{x}_k + \mathbf{p}) \leq \mathbf{0}$. At each iteration the problem (34) is solved to obtain $(\mathbf{p}_k, \mathbf{v}_k)$ (the search directions for \mathbf{x}_k and λ_k respectively), which are used to update $(\mathbf{x}_k, \lambda_k)$. SQP uses a first order Taylor expansion to replace the nonlinear constraints and uses a second order Taylor series approximation augmented by second order information from the constraints to replace the nonlinear objective. Our problem does not have analytical derivatives and is not differentiable at every point, so the information required for SQP is difficult to obtain. Because of this, SQP is not an ideal method for our problem. The reason that we use SQP anyway is that most solvers available to us are based on some SQP algorithm. In all our calculations, derivatives are calculated numerically if possible. The calculation of the derivatives are calculated automatically in TOMLAB.^{[AEP05][NS96]}

The advantage with SQP is that it uses more information about the objective function when it choses a new search direction. In contrast to fminsearch, to induce convergence, SQP updates the iteration step with a line search in a penalty function which is a linear combination of the objective function and some measure of the constraint violation.^[nlp05]

5.4 The built-in MATLAB function fmincon

One of the solvers we use is the built-in MATLAB function fmincon. The function fmincon uses an SQP algorithm with a penalty function for the constraints. At every iteration a QP subproblem is solved; the Hessian of the Lagrangian is estimated and updated at every iteration similarly to quasi-Newton methods.^[MaW05b] The penalty function used in fmincon is added to the objective function to form:

$$\boldsymbol{\phi}(\mathbf{x}) = \hat{f}(\mathbf{x}) + \sum_{i=1}^{2n+1} \varphi_i \max\{0, g_i(\mathbf{x})\},\$$

where φ_i is a penalty parameter, initially set to

$$\varphi_i = \frac{||\nabla f(\mathbf{x}_0)||}{||\nabla g_i(\mathbf{x}_0)||},$$

and which is updated in each step of the algorithm. The values in \mathbf{x}_0) are the starting points. The objective function $\hat{f}(\mathbf{x})$ and the constraint functions $g_i(\mathbf{x})$ are required to be continuous, which we cannot guarantee that our functions are. fmincon guarantees to find only stationary points, that is, KKT points. [Mat99]

5.5 The TOMLAB solver snopt

The solver snopt solves nonlinearly constrained problems like our problem (31). It uses an SQP algorithm that exploits the sparsity of the Jacobian of the nonlinear constraints. In our problem (30), the variables $\mathbf{y} = Sim(\mathbf{x})$ are nonlinearly dependent on \mathbf{x} and cause many nonlinear constraints. We cannot say with certainty that the Jacobian of (30) is sparse.

5.6 The TOMLAB solver conSolve

The TOMLAB solver conSolve also uses an SQP algorithm. The optimisation problem is written as problem (31) with the Lagrangian function (33). The penalty function for the problem with inequalities is:

$$\hat{f}(\mathbf{x}) - \sum_{i=1}^{2n+1} \lambda_i u_i \log(u_i - g_i(\mathbf{x})),$$

where **u** is a vector of barrier parameters.^[MaW05a]

We are interested in slightly infeasible solutions, and the barrier function used in conSolve is very strict on the constraints, since it uses an interior point penalty function.

5.7 The TOMLAB solver nlpSolve

Another solver in TOMLAB is nlpSolve, which also uses SQP, but with a different approach to fulfill the constraints: it uses a filter instead of a penalty function. Our optimisation problem (31) can be seen as two conflicting minimisation problems: one is to minimise the objective function and one is to minimise some function of the constraint violation. It is written as the multiobjective optimisation program:

$$\begin{array}{l} \text{minimise}_x \quad f(\mathbf{x}) \\ \text{and} \\ \text{minimise}_x \quad w(\mathbf{x}), \end{array} \tag{35}$$

where $w(\mathbf{x}) = \sum_{i=1}^{2n+1} \max\{0, g_i(\mathbf{x})\}$. The filter is a list of pairs of $\hat{f}(\mathbf{x}_k)$ and $w(\mathbf{x}_k)$, defined by iterates \mathbf{x}_k , such that for no pairs, $(\mathbf{x}_k, \mathbf{x}_l)$ the following is true: $\hat{f}(\mathbf{x}_k) \leq \hat{f}(\mathbf{x}_l), w(\mathbf{x}_k) \leq w(\mathbf{x}_l)$: if this is the case, we say that no pair dominates another pair.

At each iteration step of nlpSolve, the new point, \mathbf{x}_{k+1} , is accepted if and only if the corresponding pair $(\hat{f}(\mathbf{x}_{k+1}) \text{ and } w(\mathbf{x}_{k+1}))$ is not dominated by any pair in the filter. If the point is accepted it is included in the filter, and all points in the previous filter, that are dominated by the new point, will be rejected. If the point is rejected, the region where the algorithm searches for points is decreased.^[nlp05]

The advantage of nlpSolve for our problem is that the filter can allow some violation of the nonlinear constraint. An important drawback with nlpSolve for our problem is that it evaluates the objective function many times, which is extremely time consuming.

6 **Results**

In this section, the results for the various solvers are presented. The solvers are compared for three different cases: a constant value on the flow through the Heat Exchanger TS, W_{cool} , with and without night cooling, and a variable W_{cool} with night cooling.

Tank 1 and Tank 3 have low initial values, since water is used during the night for the cleaning and for the steamer. The low levels can be interpreted as a deficit of water in the system, which causes low levels in Tank 4. Thus, low starting levels in some tanks will inevitably cause water to be added into Tank C. An interesting part of the analysis of the results is that of the total amount of water missing in the system. The total mass of this "missing water" in the morning is:

$$m_{missing} = m_{steam} + m_{clean} = 24\ 500\ \mathrm{kg}$$

where

$$\begin{split} m_{steam} &= (W_{steam} \cdot 3\ 600 \cdot 7), \\ m_{clean} &= (W_{clean} \cdot 3\ 600 \cdot 7). \end{split}$$

If the tanks are assumed to be full at the end of the day (which they should be under normal circumstances), it is possible to calculate the initial water levels of Tank 1 and Tank 3 ($h_{1_{init}}$ and $h_{3_{init}}$):

$$h_{1_{init}} = H_1 - \frac{m_{steam}}{A_1\rho} = H_1 - 0.1,$$

and

$$h_{3_{init}} = H_3 - \frac{m_{clean}}{A_3\rho} = H_3 - 0.53.$$

These values of $h_{1_{init}}$ and $h_{3_{init}}$ are used as initial values of h_1 and h_3 .

During the day, W_{steam} and W_{clean} are taken out from the system. The total loss of water during the 10 first boilings is

$$m_{loss} = (W_{steam} + W_{clean}) \cdot 21\ 000 = 19\ 525\ \text{kg}.$$

Due to the water used in the cleaning and in the steamer, a total mass of $m_{missing} + m_{loss} = 44\,025$ kg can be regarded as necessary to add to the system. If some water is lost to the river ($W_{4_{river}} \neq 0$), more water has to be added. Hence, to get a low value on $W_{4_{river}}$ is important.

If nothing else is stated, we have used the following starting values on the variables:

 $\begin{array}{rcl} W_{cool} &=& 17.5 \ {\rm kg/s} & ({\rm only \ used \ when \ } W_{cool} \ {\rm is \ a \ variable}) \\ W_{dilute} &=& 1.11 \ {\rm kg/s} \\ T_{set3} &=& 308 \ {\rm K} \\ {\rm Simulink \ run \ time} &=& 21 \ 000 \ {\rm s} \end{array}$

When W_{cool} is a constant it is set to its upper bound, 20.3 kg/s. This value is chosen because it is the value Doggy uses today.

The initial values on the tanks are as follows:

Tank i	1	2	3		4	С
			with night	without night		
			cooling	cooling		
T_i [K]	363	343	295	328	313	295
h_i [m]	$H_1 - 1 = 1.7$	$H_2 = 2.6$	$H_3 - 0$	0.53 = 1.97	$H_4 = 2.7$	$H_C = 2.6$

When we solve the problem with fmincon, the built-in penalty function for non linear constraints do not work very well; the algorithm terminates before it had changed any values. We therefore implement a penalty function. To the objective function, we add the term

$$\sum_{s \in S} 500(\max\{0, T_{ac_s} - 293\})^2.$$
(36)

To be consistent, we try to implement the same penalty in the other solvers, but this do not work; the algorithms terminate without changing any values. The drawback with setting our own penalty is that the penalty parameter will not be updated by the algorithm.

We have chosen to present the results from fmincon together with the results from the other solvers. However, due to the differences in the penalty function between fmincon and the other solvers, great care should be taken when the results are analysed. Whether the built-in or our penalty is used is stated in the tables.

The value of the objective function is shown in the tables below, together with the values of the corresponding variables and the runtime of the algorithm.

6.1 **Optimisation**

The results from the optimisation of the system with fixed W_{cool} and with night cooling are shown in Table 1 for the solvers fmincon, snopt, conSolve and nlpSolve. Figure 23 shows the strong connection between T_{ac} [K], h_4 [m] and m_{extra} [kg] (plotted against time); when the water level in Tank 4 is low, extra fresh water is added and therefore T_{ac} is lowered. m_{extra} is the flow W_{extra} multiplied with the timestep. The results plotted in Figure 23 are obtained from fmincon.



Figure 23: The result from the optimisation shown in Table 1. The left figure shows T_{ac} , the middle figure the h_4 and the right figure m_{extra} .

The solvers fmincon, conSolve and nlpSolve give similar terminal values of T_{set3} and W_{dilute} , but conSolve solves the problem much faster and also runs less iterations. Both fmincon and nlpSolve terminate before an optimal solution is found, because too many function evaluations have been carried out. The solver snopt differs from the other three in several ways: the values of the variables are different and the terminal objective value is almost twice as high compared to the other solvers.

The solver snopt gives a result where $\sum W_{4_{river}}$ is high, around 7.5 m³ during the 10 boilings, while the other solvers give $\sum W_{4_{river}} = 0$.

To see how infeasible a solution is, we introduce the "Total infeasibility", which is the approximated area of the temperatures above 293 K. The unit of the total infeasibility is Ks. Even here, the result from snopt differs from the results from the other solvers: the total infeasibility is lower for snopt.

fmincon		snopt		
Our penalty		Built-in penalty		
Variable	Value	Variable	Value	
W _{cool}	fixed	W _{cool}	fixed	
W_{dilute}	1.1 kg/s	W_{dilute}	2.0 kg/s	
T_{set3}	307.9 K	T_{set3}	301.0 K	
$\max T_{ac}$	294.4 K	$\max T_{ac}$	293.4 K	
$\sum W_{extra}$	15 000 kg	$\sum W_{extra}$	10 000 kg	
$\sum W_{4_{river}}$	0 kg	$\sum W_{4_{river}}$	7 500 kg	
$\sum W_{dilute}$	23 000 kg	$\sum W_{dilute}$	43 000 kg	
$\min h_4$	0.69 m	$\min h_4$	0.56 m	
Obj. value	38 000	Obj. value	60 000 kg	
$Tot.\ infeas.$	6 300 Ks	Tot. infeas.	390 Ks	
Time elapsed	9h 41min	Time elapsed	16 h 39 min	
No. of iter.	32	No. of iter.	26	
Message	maximum number	Message	The current point	
	of function evalua-		cannot be im-	
	tions exceeded.		proved.	
cc	nSolve	nl	pSolve	
Built	-in penalty	Built	Built-in penalty	
Variable	Value	Variable	Value	
W _{cool}	fixed	W _{cool}	fixed	
W_{dilute}	1.0 kg/s	W_{dilute}	1.1 kg/s	
T_{set3}	309.3 K	T_{set3}	310.7 K	
$\max T_{ac}$	294.6 K	$\max T_{ac}$	294.4 K	
$\sum W_{extra}$	16 000 kg	$\sum W_{extra}$	16 000 kg	
$\sum W_{4_{river}}$	0 kg	$\sum W_{4_{river}}$	0 kg	
$\sum W_{dilute}$	21 000 kg	$\sum W_{dilute}$	22 000 kg	
$\min h_4$	0.48 m	$\min h_4$	0.41 m	
Obj. value	38 000 kg	Obj. value	38 000	
Tot. infeas.	6 800 Ks	Tot. infeas.	5 800 Ks	
Time elapsed	6 h 32 min	Time elapsed	38 h 17 min	
No. of iter.	8	No. of iter.	500	
Message	close iterations but	Message	Maximum num-	
	constraints not ful-		ber of iterations	
	filled. Too large		reached	
	penalty weights to			
	be able to con-			
	tinue.			

Table 1: The results from the different solvers when a fixed value of W_{cool} and night cooling has been used.

In Figure 24 we can see that the temperature of the outgoing water T_{ac} are nearly identical for the three solvers fmincon, conSolve and nlpSolve. The heights and temperatures in the different tanks are similar as well. The similar results are interesting since the penalty functions for the solvers are different.



Figure 24: The value of T_{ac} given from the different solvers.

The results from the optimisation (with fmincon, snopt, conSolve and nlpSolve) of the system with fixed W_{cool} , but without night cooling, are shown in Table 2. Here, fmincon's results differ from the other three solvers. The solution obtained with fmincon is feasible (T_{ac} is below 293 K) but the drawback is that water is lost to the river. The other solvers give a solution where fresh water is added, but no water is lost to the river.

fmincon		snopt	
Our penalty		Built-in penalty	
Variable	Value	Variable	Value
W_{cool}	fixed	W _{cool}	fixed
W_{dilute}	2.8 kg/s	W_{dilute}	1.4 kg/s
T_{set3}	322.9 K	T_{set3}	322.9 K
$\max T_{ac}$	292.3 K	$\max T_{ac}$	294.8 K
$\sum W_{extra}$	0 kg	$\sum W_{extra}$	12 000 kg
$\sum W_{4_{river}}$	14 000 kg	$\sum W_{4_{river}}$	0 kg
$\sum W_{dilute}$	59 000 kg	$\sum W_{dilute}$	29 000 kg
$\overline{\min h_4}$	2.0 m	$\overline{\min h_4}$	1.1 m
Obj. value	72 000 kg	Obj. value	40 000
Tot. infeas.	0 Ks	Tot. infeas.	8 000 Ks
Time elapsed	2h 38 min	Time elapsed	8 h 7 min
No. of iter.	10	No. of iter.	18
Message		Message	The current point
			cannot be im-
			proved
cc	onSolve	nl	pSolve
cc Buil	onSolve t-in penalty	nl Buil	_pSolve t-in penalty
CC Built Variable	onSolve t-in penalty Value	n I Built Variable	pSolve t-in penalty Value
$CC Built Variable W_{cool}$	bnSolve t-in penalty Value fixed	nl Buil Variable	-pSolve t-in penalty Value fixed
$\begin{tabular}{c} c c \\ \hline Built \\ \hline Variable \\ \hline W_{cool} \\ \hline W_{dilute} \\ \end{tabular}$	bnSolve t-in penalty Value fixed 1.1 kg/s	nl Buil Variable W _{cool} W _{dilute}	-pSolve t-in penalty Value fixed 1.2 kg/s
$\begin{tabular}{c} c c \\ \hline Built \\ \hline Variable \\ \hline W_{cool} \\ \hline W_{dilute} \\ \hline T_{set3} \\ \hline \end{tabular}$	bnSolve t-in penalty Value fixed 1.1 kg/s 317.8 K	$\begin{tabular}{ c c c c } \hline & & & & & & \\ \hline & & & & & \\ \hline & & & &$	rin penalty Value fixed 1.2 kg/s 312.3 K
$\begin{tabular}{c} & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & &$	value fixed 1.1 kg/s 317.8 K 295.5 K	$\begin{tabular}{ c c c c } \hline &n \end{tabular}\\ \hline & & & & \\ \hline & & & \\ \hline & & & & \\ \hline & & & &$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K
$\begin{tabular}{c} & & & & \\ \hline \textbf{Built} \\ \hline \textbf{Variable} \\ \hline W_{cool} \\ \hline W_{dilute} \\ \hline T_{set3} \\ \hline max T_{ac} \\ \hline \sum W_{extra} \\ \hline \end{tabular}$	value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg	$\begin{tabular}{ c c c c c } \hline &n \end{tabular}\\ \hline & & & & \\ \hline \hline & & & \\ \hline & & & \\ \hline \hline \\ \hline & & & \\ \hline \hline & & & \\ \hline \hline \hline \\ \hline \hline \\ \hline \hline \hline \\ \hline \hline \hline \\ \hline \hline \hline \hline \\ \hline \hline \hline \hline \hline \\ \hline \hline \hline \hline \hline \hline \hline \hline \hline \\ \hline \hline$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	bnSolve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg	$\begin{tabular}{ c c c c c } \hline n1\\ \hline Built\\ \hline Variable\\ \hline \hline W_{cool}\\ \hline W_{dilute}\\ \hline T_{set3}\\ \hline max T_{ac}\\ \hline \sum W_{extra}\\ \hline \sum W_{4river}\\ \hline \end{array}$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg
$\begin{array}{c} & & & \\ & & & \\ \hline \textbf{Built} \\ \hline \textbf{Variable} \\ \hline W_{cool} \\ \hline W_{dilute} \\ \hline T_{set3} \\ \hline max T_{ac} \\ \hline \sum W_{extra} \\ \hline \sum W_{4river} \\ \hline \sum W_{dilute} \\ \hline \end{array}$	bnSolve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg	$\begin{tabular}{ c c c c c } \hline & & & & & & \\ \hline & & & & & & \\ \hline & & & &$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	bit bit t-in penalty trin penalty Value fixed fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 0 kg 23 000 kg 0.80 m m	$\begin{tabular}{ c c c c c } \hline &n \end{tabular}\\ \hline &Built\\ \hline &Variable\\ \hline &W_{cool}\\ \hline &W_{dilute}\\ \hline &T_{set3}\\ \hline &max T_{ac}\\ \hline &\sum W_{extra}\\ \hline &\sum W_{4river}\\ \hline &\sum W_{4river}\\ \hline &\sum W_{dilute}\\ \hline &min h_4 \end{tabular}$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	on Solve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	on Solve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg 5 200 Ks	$\begin{tabular}{ c c c c c } \hline n \end{tabular} \\ \hline Built \\ \hline Wariable \\ \hline W_{cool} \\ \hline W_{dilute} \\ \hline T_{set3} \\ \hline max T_{ac} \\ \hline \sum W_{extra} \\ \hline \sum W_{4river} \\ \hline \sum W_{4river} \\ \hline \sum W_{dilute} \\ \hline min h_4 \\ \hline Obj. value \\ \hline Tot. infeas. \\ \hline \end{tabular}$	-pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000 3 500 Ks
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	on Solve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg 5 200 Ks 6h 56min	$\begin{tabular}{ c c c c c } \hline & & & & & & & \\ \hline & & & & & & \\ \hline & & & &$.pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000 3 500 Ks 33 h 43 min
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	onSolve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg 5 200 Ks 6h 56min 9	$\begin{tabular}{ c c c c c } \hline & & & & & & & \\ \hline & & & & & & \\ \hline & & & &$.pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000 3 500 Ks 33 h 43 min 500
Col BuiltVariable W_{cool} W_{dilute} T_{set3} $\max T_{ac}$ $\sum W_{extra}$ $\sum W_{4river}$ $\sum W_{dilute}$ $\min h_4$ $Obj. value$ $Tot. infeas.$ Time elapsedNo. of iter.Message	on Solve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg 5 200 Ks 6h 56min 9	$\begin{tabular}{ c c c c c } \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & &$.pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000 3 500 Ks 33 h 43 min 500 Maximum num-
Col BuiltVariable W_{cool} W_{dilute} T_{set3} $\max T_{ac}$ $\sum W_{extra}$ $\sum W_{4river}$ $\sum W_{dilute}$ $\min h_4$ $Obj. value$ $Tot. infeas.$ Time elapsedNo. of iter.Message	on Solve t-in penalty Value fixed 1.1 kg/s 317.8 K 295.5 K 15 000 kg 0 kg 23 000 kg 0.80 m 38 000 kg 5 200 Ks 6h 56min 9	$\begin{tabular}{ c c c c } \hline n \end{tabular} \\ \hline Built \\ \hline Built \\ \hline Variable \\ \hline \hline W_{cool} \\ \hline W_{dilute} \\ \hline T_{set3} \\ \hline max T_{ac} \\ \hline \sum W_{extra} \\ \hline \sum W_{extra} \\ \hline \sum W_{dilute} \\ \hline min h_4 \\ \hline Obj. value \\ \hline Tot. infeas. \\ \hline Time elapsed \\ \hline No. of iter. \\ \hline Message \\ \hline \end{tabular}$	pSolve t-in penalty Value fixed 1.2 kg/s 312.3 K 295.7 K 15 000 kg 0 kg 26 000 kg 0.49 m 40 000 3 500 Ks 33 h 43 min 500 Maximum number of iterations

 Table 2: The results for the three different solvers when no night cooling has been used.

 T_{ac} is plotted in Figure 25 for the different solvers; fmincon gives a better result, with respect to T_{ac} , than the other two solvers.



Figure 25: The value of T_{ac} from the different solvers.

Because of the similarities between snopt, conSolve and nlpSolve we will not make any more optimisations using snopt and nlpSolve since their run times are significantly longer and nlpSolve terminates before an optimal solution is found.

The results from the optimisation of the system with W_{cool} as a variable and with night cooling (using fmincon and conSolve) are shown in Table 3. The individual solvers' results differ very much; fmincon maximises the values of T_{set3} , W_{dilute} and W_{cool} , while conSolve uses neither the maximum nor the minimum allowed values. The advantage with fmincon's result is that the water level in Tank 4 has a less fluctuation, as seen to the left in Figure 26. To the right in Figure 26, T_{ac} is plotted.



Figure 26: The value of h_4 to the left and T_{ac} to the right, from the two solvers fmincon and conSolve.

The fact that fmincon uses only one iteration however gives us reason to be critical to the results.

fmincon Our penalty			conSolve Built-in penalty	
Variable	Value		Variable	Value
W _{cool}	20.3		W _{cool}	16.2 kg/s
W_{dilute}	3.1 kg/s		W_{dilute}	1.6 kg/s
T_{set3}	322.9 K		T_{set3}	301.0 K
$\max T_{ac}$	292.9 K		$\max T_{ac}$	293.1 K
$\sum W_{extra}$	0 kg		$\sum W_{extra}$	12 000 kg
$\sum W_{4_{river}}$	22 000 kg		$\sum W_{4_{river}}$	2 000 kg
$\sum W_{dilute}$	64 000 kg		$\sum W_{dilute}$	34 000 kg
$\min h_4$	2.0 m		$\min h_4$	0.46 m
Obj. value	84 000 kg		Obj. value	48 000 kg
Tot. infeas.	3 500 Ks		Tot. infeas.	4 000 Ks
Time elapsed	20 min		Time elapsed	8 h 5 min
No. of iter.	1		No. of iter.	6
Message	Optimal found.	solution	Message	Too high penalty values.

Table 3: The results where night cooling has been used. W_{cool} is a variable.

Figure 27 shows T_{ac} as obtained with conSolve; the line curve is one simulation with 10 boilings (21 000 s) and the dotted curve is one simulation over a day (62 000 s). It uses the optimal values presented in Table 3. The temperature has not stabilised during the 10 boilings; it becomes about 1 K larger at the end of the day.



Figure 27: T_{ac} using conSolve; the line curve is one simulation for 10 boilings (21 000 s) and the dotted curve is one simulation over a day (62 000 s).

The different kinds of penalty functions in the solvers cause the solvers to

work differently. At the first sight, the solvers in TOMLAB seem to allow infeasible solutions while fmincon prioritise feasibility. However, it is possible that the TOMLAB solvers are so sensitive to infeasible solutions that they terminate before they have found an optimal solution. fmincon on the other hand, gives a penalty on infeasible solutions, but does not terminate.

We do not get an obvious answer on which T_{set3} should be used. The solvers give varying results that seem to depend highly on the other variables. However, there is one trend concerning T_{set3} : when the level of the water in Tank 4 is low, T_{set3} tends to be low, which will result in a larger input of water into Tank 4 from the autoclaves.

Most of our solutions are infeasible; the solvers could not find feasible solutions. There exist feasible solutions though, since more water can be added, i.e W_{dilute} can be increased. It is of great interest to see the solutions we have obtained, since sometimes it can be acceptable to have a too large temperature for a short period of time, when it is very expensive to lower it with fresh water.

Most solvers terminate before an optimal solution is found. Therefore the there might be better solutions for the problem, than the ones stated above.

Because Tank 1 and Tank 3 have low initial levels, Tank 4 is emptied faster than the extra water is added. Thus, if the limit where Tank 4 starts to be filled with extra water is set too low, Tank 4 is emptied. An easy solution to this problem is to set this limit higher. Therefore, we have changed the original setting on $h_{4_{min}}$ from 0.8 meters to 1.5 meters.

The results depend highly on the initial values of the water levels: if Tank 1 and Tank 2 have low levels, Tank 4 can be emptied, which will cause a large flow on W_{extraC} . Important to notice is that the extra water added, due to low levels in Tank 4, is not always unnecessary. The levels of Tank 1 and Tank 3 are low initially because water has been removed from the tanks during the night for cleaning and for the steamer. Since this is a loss in the production, replaceing lost water from the system is necessary. One could even think of a way to include the low levels of the tanks in the objective function.

6.2 Improvements

To see if the changes made of the system have improved it, we run simulations on the "old" and the "new" system and compare them. The old system is the system as it is today; the insertion of water is to Tank 1 and water to the cleaning is taken from Tank 2. The new system is the system we used in our calculations: the cleaning water is taken from Tank 3 and the fresh water W_{dilute} is added in the pipe.

The results from the simulations of the old and the new system, when night cooling has been used, are shown in Table 4. The values for T_{set3} and W_{dilute} are taken from the results from fmincon with night cooling (Table 1). The new system has two important advantages: the loss to the river is small or zero and T_{ac} is lower (that is the total infeasibility is lower).

Old system		New system	
Variable	Value	Variable	Value
$\sum W_{extra}$	17 000 kg	$\sum W_{extra}$	16 000 kg
$\sum W_{4_{river}}$	0 kg	$\sum W_{4_{river}}$	0 kg
Tot. infeas.	32 000 Ks	Tot. infeas.	5 900 Ks

Table 4: A comparison between the old and the new system. With night cooling.

The values of T_{ac} and h_4 (with night cooling) are shown in Figure 28, where the values for old system is plotted with the dashed line.



Figure 28: Comparisons between the old and the new system. Night cooling has been used.

The results from the simulations of the two systems, when no night cooling has been used, are shown in Table 5. The values for T_{set3} and W_{dilute} are taken from the results from fmincon without night cooling (Table 2). When the night cooling has not been used, the differences between the systems are even more obvious. Notice the value of W_{4river} and the total infeasibility for the old system!

Old system		New system	
Variable	Value	Variable	Value
$\sum W_{extra}$	6 600 kg	$\sum W_{extra}$	0 kg
$\sum W_{4_{river}}$	35 000 kg	$\sum W_{4_{river}}$	13 000 kg
Tot. infeas.	85 000 Ks	Tot. infeas.	0 Ks

Table 5: A comparison between the old and the new system. Without night cooling.

When no night cooling is used, the difference in T_{ac} is remarkably large. The values of T_{ac} and h_4 without night cooling are shown in Figure 29. The values for the old system are plotted with the dashed line.

The value of T_{ac} is up to 6.5 K lower in the new system compared to the old one.



Figure 29: Comparisons between the old and the new system. No night cooling has been used.

The results from the simulations of the two systems, when night cooling has been used, are shown in Table 6. The values for T_{set3} and W_{dilute} are taken from the results from fmincon with night cooling (Table 3). When W_{cool} is a variable, the differences between the systems are even more obvious.

Table 6: A comparison between the old and the new system. With night cooling and variable W_{cool} .

Old system		New system	
Variable	Value	Variable	Value
$\sum W_{extra}$	5 800 kg	$\sum W_{extra}$	0 kg
$\sum W_{4_{river}}$	47 000 kg	$\sum W_{4_{river}}$	21 000 kg
Tot. infeas.	88 000 Ks	Tot. infeas.	0 Ks

The values of T_{ac} and h_4 without night cooling are shown in Figure 30. The values for the old system are plotted with the dashed line.

The value of T_{ac} is up to 8 K lower, and the water level in Tank 4 is clearly more stable, in the new system compared to the old one.



Figure 30: Comparisons between the old and the new system. Night cooling has been used.

The results from this section show that it is better to insert the fresh water in the pipe instead of into Tank 1. To do so will lower the temperature 3–8 K. The water level in Tank 4 is also more stable in the new system.

To show the effect of the night cooling, the optimal values obtained from fmincon with fixed W_{cool} and night cooling (Table 1) are used to run two simulations: one with and one without night cooling. The values of T_{ac} from the two simulations are plotted in Figure 31. Night cooling (plotted with the solid line) lowers the temperature with up to 3.5 K.



Figure 31: The value of T_{ac} , with and without night cooling.

7 Discussion

Tank 1, 2 and 4 are not allowed to run empty since there are pumps there. If the water level is too low, a signal will start a pump (or, in Tank 4, a tap) that adds water to the tank. When the water reaches a certain level, another signal will close the tap. In this project, a flow is going to fill the tank as long as the level is on or below the minimum, but it will stop as soon as the water level goes over this limit. If the water level is around this height, this is going to make the system less robust than in reality.

When the water level in a tank reaches a certain level, it will overflow into the next tank through a pipe in the tank wall. The height of the pipes are different for each tank and the water will therefore mostly go to the next tank in line. However, if a tank is full, and the previous tank has a low level, the water could flow backwards in the system, which would cause the temperature differences between the tanks to decrease. In the simulation this possibility has been removed, simply because the calculations become too large and slow. With the large amounts of water in the system during our simulations, this effect can be neglected, but in future simulation it may have to be taken into account.

We have assumed that the tanks are perfectly mixed and therefore have the same temperature throughout the tank. This is not an unreasonable approximation for Tank 1 and Tank 2 because the high temperatures in these tanks cause turbulence. Tank 4 and Tank C can also be considered mixed, since there is a large exchange of water in these tanks. The tank where the assumption, of perfectly mixed tanks, is not always correct is Tank 3 where layers of water with different temperatures can occur.

A difficulty in this project was that the system includes several plate heat exchangers, both in the tank system and in connection with the autoclaves. The principle of plate heat exchangers (run two streams of water on each side of a plate) may be quite simple, but the calculations are far from trivial. In fact, there is hardly any literature on the subject and many of the relations used in calculations on plate heat exchangers are based on experiments. Since our model for Heat Exchanger TS is derived using such relations and regression, it is only an approximation, and even a rather crude one.

Another interesting problem is how to establish the demand of water needed in the cooling process. A first thought was to calculate the demand from the properties of the autoclaves and the tins. It soon showed that this problem was very complex. For example, the properties of plate heat exchangers are not very well known, and the calculation of the heat transfer from the tins to the water in the autoclaves is difficult. An important thing to notice is that the demand of cooling water is not a constant value. When the temperature of the cooling water is low, a smaller flow is sufficient for the cooling, while a higher flow is needed at higher temperatures. The heat exchangers in the autoclaves also have an effect since they work better at certain flows. Altogether, the calculations of the demand of cooling water is actually an optimisation problem of its own, and would need more study than is possible in this thesis. Therefore we have measured the present flows going to the cooling process and used them in our calculations. The production of the Tetra Pak is limited to one boiling per hour while there can be about 1.7 boiling of aluminium tins. These limits depend on the time it takes to fill the autoclaves and boil the tins. In our calculations, we have assumed the demand of cooling water to the autoclaves to be constant, but this is only true for the Tetra Pak autoclaves. The cooling processes of the original tins occasionally overlap, and then the mass flow of cooling water per autoclave is decreased.

To change the outtake of the cleaning water is beneficial for the system in several ways; the temperature of the cleaning water will have a more appropriate temperature and will not run out. The change of the cleaning water will affect the system mainly at night; when cleaning water is used even when the production is not running. Hence, this change will not make a big difference in our calculations, since we do not study the behaviour of the system at night.

8 Further possible developments

A future change to make a better cooling of the water can be to consider the size of Tank 4. This tank is small and has very large flows which makes it sensitive. One possible improvement is to run the heat exchanger through Tank 3 instead of Tank 4 and to put the outlet of water to the narrow river in Tank 3. Another possibility is to remove the wall between Tank 3 and Tank 4. This will make the tank larger and more stable, but the temperatures in it would be more difficult to control. Another option to solve the problem with the instability of Tank 4 is to make one large tank out of Tank 4 and Tank 5. A drawback with making a large tank out of the two, is that the water led to the autoclaves could have a higher temperature.

References

- [AEP05] N. Andréasson, A. Evgrafov, and M. Patriksson. *An Introduction to Continuous Optimization*. Studentlitteratur, Lund, 2005.
- [con05] Tomlab solver consolve, 07112005. http://tomlab.biz/products/base/solvers% textup/conSolve.php.
- [EG94] I. Ekroth and E. Granryd. *Tillämpad termodynamik*. AB Grafiska Gruppen, Stockholm, 1994.
- [GMS05] P. Gill, W. Murray, and M. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. SIAM Review, 47(1):99–131, 2005.
- [Las05] I. Lasson, 2005. Product Manager, Thermal equipment, Alfa Laval. Written communication.
- [Mat99] The MathWorks. Optimizatioon toolbox, for use with matlab. 1999.
- [MaW05a] The math works, optimization toolbox help, 0312005. http://www.mathworks.com/access/helpdesk/help/toolbox/gads/bqf9jvg-1.html.
- [MaW05b] The math works, optimization toolbox help, 07112005. http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/.
- [NE05] National encyclopedin, 10102005. http://www.ne.se/jsp/search/article.jsp? i_art_id=120909&i_word=autoklav.
- [nlp05] Tomlab solver nlpsolve, 04112005. http://www.math.dundee.ac.uk/ftp/na-reportsup/NA171_RF_SL.ps.
- [NS96] S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, Singapore, 1996.
- [Pet05] T. Pettersson, 2005. Produktionsteknisk chef, Doggy AB. Oral communication.

A Heat balance

To get an expression for the temperatures in the tanks, a heat balance needs to be solved. We look over a small period of time to get the derivative of the temperature. The change of temperature in a given tank can be calculated as

$$\Delta T = T_n - T_o, \tag{37}$$

where T_n is the new temperature in the tank and T_o is the temperature in the tank before the change. T_n is calculated with a heat balance as

$$T_n = \frac{T_o m_o + \sum_{all \ inflows} \left(W_{in} \Delta t T_{in} \right) - \sum_{all \ outflows} \left(W_{out} \Delta t T_{out} \right)}{m_o + \sum_{all \ inflows} W_{in} \Delta t - \sum_{all \ outflows} W_{out} \Delta t} =$$

$$= \frac{T_o(m_o - \sum_{all \ outflows} W_{out} \Delta t) + \sum_{all \ inflows} W_{in} \Delta t T_{in}}{m_o + \sum_{all \ inflows} W_{in} \Delta t - \sum_{all \ outflows} W_{out} \Delta t}$$

which gives us an expression for ΔT :

$$\Delta T = \frac{T_o(m_o - \sum_{all \ outflows} W_{out} \Delta t) + \sum_{all \ inflows} W_{in} \Delta t T_{in}}{m_o + \sum_{all \ inflows} W_{in} \Delta t - \sum_{all \ outflows} W_{out} \Delta t} - T_o =$$

$$=\frac{T_o(m_o-\sum_{all\ outflows}W_{out}\Delta t)+\sum_{all\ inflows}W_{in}\Delta tT_{in}}{m_o+\sum_{all\ inflows}W_{in}\Delta t-\sum_{all\ outflows}W_{out}\Delta t}-$$

$$\frac{T_o(m_o + \sum_{all \ inflows} W_{in}\Delta t - \sum_{all \ outflows} W_{out}\Delta t)}{m_o + \sum_{all \ inflows} W_{in}\Delta t - \sum_{all \ outflows} W_{out}\Delta t} =$$

$$= \frac{\sum_{all \ inflows} (W_{in} \Delta t T_{in}) - T_o \sum_{all \ inflows} W_{in} \Delta t}{m_o + \sum_{all \ inflows} W_{in} \Delta t - \sum_{all \ outflows} W_{out} \Delta t} =$$

$$=\frac{\sum_{all\ inflows}W_{in}\Delta t(T_{in}-T_o)}{m_o+\sum_{all\ inflows}W_{in}\Delta t-\sum_{all\ outflows}W_{out}\Delta t}.$$

The change in temperature on a small time step becomes:

$$\frac{\Delta T}{\Delta t} = \frac{\sum_{all \ inflows} W_{in}(T_{in} - T_o)}{m_o + \sum_{all \ inflows} W_{in}\Delta t - \sum_{all \ outflows} W_{out}\Delta t}.$$

When $\Delta t \rightarrow 0$ we obtain an expression for the time derivative of the temperature:

$$\frac{\Delta T}{\Delta t}\Big|_{\Delta t \to 0} \Rightarrow \frac{dT}{dt} = \frac{\sum_{all \ inflows} W_{in}(T_{in} - T_o)}{m_o} \tag{38}$$

The derivation is analogous for the mass balance of each tank.

B Overflow

The overflow from one tank to another, with the points 1 and 2 in Figure 32 is calculated with Bernoulli's equation ^[EG94]:

$$\frac{p_1}{\rho} + \frac{w_1^2}{2} + gz_1 = \frac{p_2}{\rho} + \frac{w_2^2}{2} + gz_2,$$
(39)



Figure 32: Bernoulli's equation is calculated on the two points 1 and 2.

where p is the pressure, w is the velocity and z is the water level, which with $p_1 = p_2$, $w_1 = \frac{dh}{dt}$, $z_1 = h$, $w_2 = w_2$, and $z_2 = H$ gives

$$\frac{\left(\frac{dh}{dt}\right)^2}{2} + gh = \frac{w_2^2}{2} + gH.$$
 (40)

If we rewrite this, we have:

$$w_2^2 = 2g(h - H) + \left(\frac{dh}{dt}\right)^2,$$
 (41)

which gives us an expression for the velocity going out of the hole:

$$w_2 = \sqrt{2g(h-H) + \left(\frac{dh}{dt}\right)^2},$$

where w_2 is the velocity of the water going through the hole and $\frac{dh}{dt}$ is the change of the water level in Tank 1, see Figure 32. Since $\left(\frac{dh}{dt}\right)^2 << w_2$ we neglect it, and the equation becomes:

$$w_2 = \sqrt{2g(h-H)}$$

To get the mass flow instead of the velocity we multiply with the area of the hole, A, the density of water, ρ and a resistance factor, μ .

$$W_2 = w_2 A \rho \mu. \tag{42}$$

 μ is a dimensionless factor used to calculate the decrease in flow when the edges of the pipe are not smooth $^{\rm [EG94]}.$

C Constants

Constant	Value	Dimension
$C_{pump_{river}}$	11.9	kg/s
$C_{pump_{fresh}}$	3.6	kg/s
$C_{pump_{cool}}$	20.3	kg/s
$D_{ac_{tin}}$	10.6	kg/s
$D_{ac_{tetra}}$	5	kg/s
$h_{1_{min}}$	0.80	m
$h_{2_{min}}$	0.80	m
$h_{3_{min}}$	0.80	m
$h_{4_{min}}$	1.5	m
$h_{C_{min}}$	1.5	m
H_1	2.7	m
H_2	2.6	m
H_3	2.5	m
H_4	2.7	m
H_C	2.6	m
Wclean	0.74	kg/s
$W_{pellets}$	0.56	kg/s
Wriver	11.9	kg/s
Wsteam	0.83	kg/s
Wcomputer	0.28	kg/s
Wac	15.6	kg/s
$T_{pellets}$	280	K
T_{fresh}	277	K
T_{river}	288	K
A_1	7	m^2
A_2	10.5	m^2
A_3	35	m^2
A_4	7.5	m^2
A_C	7.5	m^2
A_{pipe}	0.044	m^2
ρ	1000	kg/m ³
μ	0.6	
n_{tin}	1.7	boilings/h
n _{tetra}	1	boilings/h
Tset1	353	K
T_{set2}	328	K

Table 7: Constants used in the simulation and the optimisation.

D Term descriptions

- m
- Mass [kg] Mass flow [kg/s] W
- C
- Capacity [kg/s] Demand of water required in a process [kg/s] D
- A
- Area [m²] Water level [m] h
- Η
- Pipe height [m] Temperature [K] T