Adaptive PI-control for an air sterilization module

Henrik Stålbom

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6112
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2020

Abstract

The goal of this master thesis has been to implement adaptive PI-control of two air flow heaters in an air sterilization module. This is done by modeling the module and using the information from the model to estimate the dynamics of the heaters.

The idea of this thesis is to implement adaptive control with as simple control structures and tuning algorithms as possible, to keep it as generalized and accessible as possible. This way it will be easier to apply the same framework to other components. The implementation uses a range of models to express the thermodynamic behaviour of two heaters interconnected by a heat exchanger and some piping with heat losses. Using the models, a recursive least squares algorithm estimates the parameters of each heater, and finally the heaters are tuned using the lambda-method and the AMIGO-method. The models are created both from physical models and from non-physical models based purely on measurement data using the best available order.

The implementation was done using MATLAB Simulink from which the code was then ported over to BECKHOFF TwinCat and integrated into the machine code. Due to time constraints the implementation only got to the simulation stage in TwinCat, but it showed promise and worked as expected. Simulations were also done in Simulink using measurement data which also showed promise for a functional framework.

Acknowledgements

I would like to thank Tetra Pak for giving me the opportunity to work on this master thesis and giving me the resources I have required. I would also like to extend my gratitude to my supervisor Mattias Darmell at Tetra Pak, for providing me with anything that I have asked for and helped me along the journey.

Lastly I wish to thank my academic supervisor, Prof. Tore Hägglund, for all his help and guidance.

Henrik Stålbom
June 2020

Contents

1

Introduction

This chapter will give some introductory information and relevant background information for this thesis.

1.1 About Tetra Pak

Tetra Pak has many business areas, but the main areas are processing and packaging solutions for food. Every day Tetra Pak provides products that meet the needs of hundreds of millions of people across the globe, with their vision "to make food safe and available, everywhere" [*Tetra Pak* 2020].

The A1 packaging machine for TFA [*Tetra Pak* 2020] is developed and sold by Tetra Pak. It uses aseptic technology to ensure safe packaging, and is the focus of this thesis.

1.2 Background

Aseptic packaging

Aseptic packaging is the result of aseptic processing which is a process where sterilized products are packaged in sterile packages under sterile conditions [*Tetra Pak* 2020]. Today it is the main commercial packaging technique for liquid foods such as milk, juices and yoghurts and many other liquids.

The packaging material is sterilized by passing it through a bath of hydrogen peroxide. The hydrogen peroxide is then removed through evaporation by passing the material through a heating chamber with hot air. The product is sterilized by quickly heating it up for a short duration before it is packaged. To keep everything under sterile conditions the surrounding air and the air used in the heating chamber must also be sterile. This is done by heating up the air to 360 degrees Celsius after which it is considered sterile.

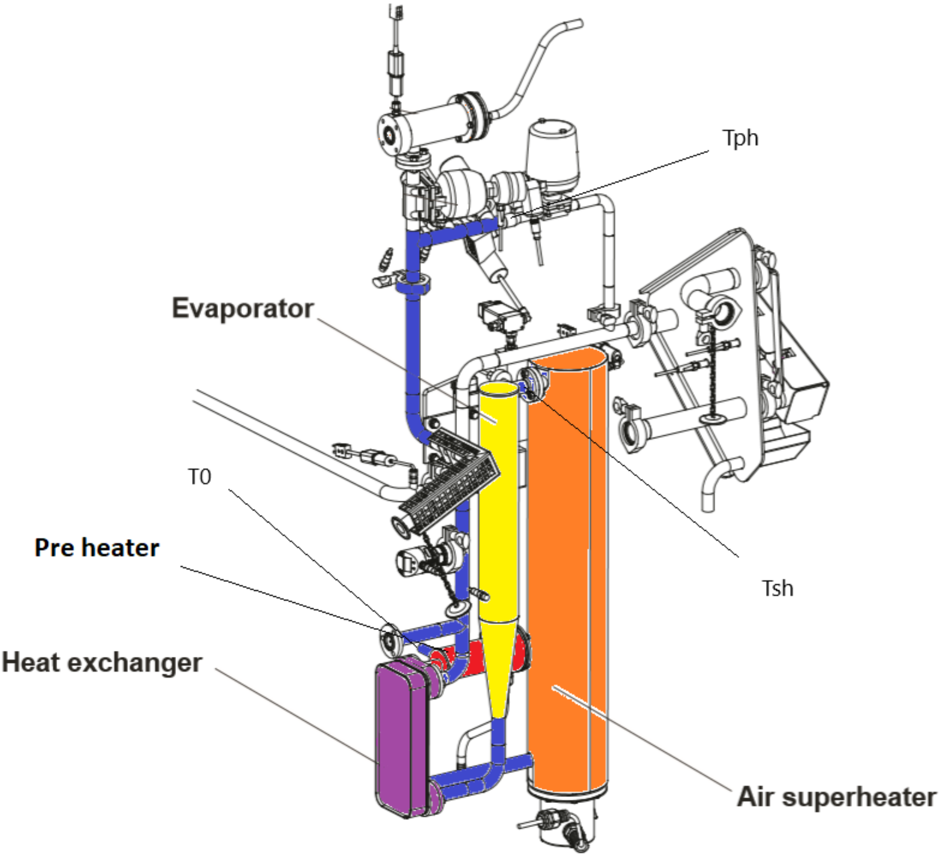


Figure 1.1: Air sterilization system overview.

1.3 Problem description

In figure 1.1 the overview of the air sterilization system can be seen. The system takes pressurized room temperature air T_0 as input flow into the air pre heater. The air then flows into the heat exchanger and out into the air super heater, where it is heated to 360 degrees Celsius at the exhaust T_{sh}. The air then continues through the evaporator and into the heat exchanger. The second output of the heat exchanger then splits into different branches leading to different parts of the machine.

There are two temperatures that are being measured. The temperature of the outlet of the super heater T_{sh}, which is being measured and controlled by the super heater to be 360 degrees with a margin of ± 10 degrees. The second temperature T_{ph} is measured on one of the branches at the end of the system, and is being controlled at different temperatures depending on the machine state. This temperature is controlled by the pre heater.

The heat exchanger is used as a form of regenerating heat energy. The temperature of the air only has to reach a peak of 360 degrees a short time to be considered sterile. The final temperature at the end of the system lies in the range (85, 120) degrees and as such the heat exchanger helps to both heat up and cool down the air. The evaporator is only used when the machine is staging into production. During this time hydrogen peroxide is injected into the evaporator and will evaporate because of the heat. This hydrogen peroxide vapor is then sprayed through the machine in order to sterilise the interior.

1.4 Goals and limitations

The purpose of this thesis is to investigate the possibility of implementing adaptive control of the air sterilization sub system of the A1 packaging machine for TFA. The aim of the adaptive control implementation is to increase the degree of automation of the A1 machine and will serve as an initial study of the possibility of adaptive control in many other parts of the machine. To achieve this the work has been divided into three sub-goals.

Goal 1: Modelling

Modelling of all components of the air sterilization system. This will include machine testing and different modelling approaches such as grey box modelling and black box modelling.

Goal 2: Parameter estimation

Online parameter estimation based on input and output data. Model structure and the number of parameters will be decided based on the result of the modelling.

Goal 3: Adaptive control

Finally an adaptive control law will be implemented with the use of the models and their estimated parameters.

When the machine is staging into production there are some non linear behaviour occurring due to valves changing and the hydrogen peroxide spraying. This causes major changes to the thermodynamic behaviour of the air and is not being treated in this thesis. Instead the thesis is limited to dealing with the behaviour of the machine during production.

2

Modeling

The following chapter will cover the modeling of the system components. The estimated model parameter values will be presented later in the Results chapter.

2.1 System overview

An overview of the air sterilization system can be seen in figure 2.1. The temperature designations in the image will designate the specific air flow, i.e. where in the system it is, and the temperature at that point. The system consists of the following components:

- **Pre heater:** The small heater that heats pressurized air T_0 which is at room temperature, using the applied power P_{ph}. The outflow of air is T_1.
- **Heat exchanger:** A counter flow heat exchanger with inflows T_1 and T_3, and outflows T_2 and T_4.

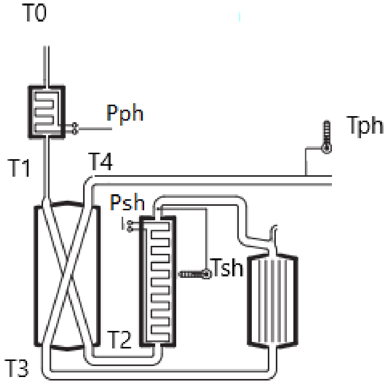


Figure 2.1: Air heater concept model

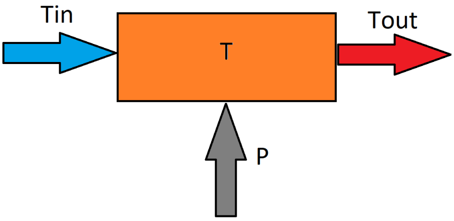


Figure 2.2: Air heater concept model

- **Super heater:** The large heater with inflow T_2, applied power P_{sh} and outflow T_{sh}
- **Evaporator:** The container in between T_{sh} and T_3 is the evaporator which is a passive component in regards to this thesis. However it still has an effect on the temperatures as there is a lot of heat loss to the surrounding air.

Temperature measurements are given from T_{sh} and T_{ph}. They are named as such because the pre heater controller is using T_{ph} as feedback and similarly the super heater is controlled based on T_{sh}. A final thing to mention is the distance of the pipe between T_4 and T_{ph} which is long enough to affect the temperature difference due to heat losses.

With the goal of implementing an online parameter estimation of the heaters, modelling has to be performed to be able to estimate the input and output signals to the heaters. The following signals must be known: T_0, P_{ph}, T_1 for the pre heater and T_2, T_{sh}, P_{sh} for the super heater. By assuming that the model for the passive components, i.e. the heat exchanger, the evaporator and the pipe between T_4 and T_{ph} are time invariant, the few measurements in the system should be adequate.

2.2 Heaters

The pre heater and the super heater are both modeled as first order processes as in figure 2.2. In the figure T_{in} stands for the inlet temperature, P stands for the applied power, T stands for the air temperature inside the heater and T_{out} stands for the outlet temperature. Some assumptions are made in this model. The first being that the temperature of the air inside the heater is homogeneous. The second assumption is that there is no energy loss from heat transfer to the surrounding air.

Starting with an equation for the heat energy of the internal air T gives:

$$E = mC_p(T - T_{ref}) = mC_pT \tag{2.1}$$

where E is the heat energy of the internal air, m is the mass of the internal air, C_p is the specific heat capacity of the air and $T_{ref} = 0$ is a reference temperature in Celsius to simply the expression.

With this a heat balance equation can be set up as:

$$\frac{d}{dt}E = w_{in}C_pT_{in} - w_{out}C_pT_{out} + kP = wC_pT_{in} - wC_pT + kP$$

where w is the mass flow of the air which is assumed to be the same for both the inlet and the outlet, P is the applied power in percentage, k is a conversion parameter from percentage to watts, and where it is also assumed that $T_{out} = T$. From equation 2.1 and with the assumption that the mass is constant, the following can be derived:

$$\begin{aligned} \frac{d}{dt}E &= \frac{d}{dt}(mC_pT) = mC_p\frac{d}{dt}T \\ \implies \frac{d}{dt}T &= \frac{w}{m}(T_{in} - T) + \frac{1}{mC_p}kP \end{aligned}$$

This can be rewritten as a first order system with two inputs and one output:

$$\frac{d}{dt}T = -\frac{w}{m}T + \begin{bmatrix} \frac{w}{m} & \frac{k}{mC_p} \end{bmatrix} \begin{bmatrix} T_{in} \\ P \end{bmatrix}$$

And in discrete form it turns into:

$$\begin{aligned} T(t+h) &= T(t) + h(-\frac{w}{m}T(t) + \begin{bmatrix} \frac{w}{m} & \frac{k}{mC_p} \end{bmatrix} \begin{bmatrix} T_{in}(t) \\ P(t) \end{bmatrix}) \\ &= (1 - \frac{hw}{m})T(t) + \begin{bmatrix} \frac{hw}{m} & \frac{hk}{mC_p} \end{bmatrix} \begin{bmatrix} T_{in}(t) \\ P(t) \end{bmatrix} \\ &= aT(t) + \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} T_{in}(t) \\ P(t) \end{bmatrix} \qquad (2.2) \end{aligned}$$

where h is the sampling time and the parameters have been concatenated into three parameters that express the dynamics of a general heater with two inputs and one output. The model transfer functions can be seen in equation 2.3.

$$H(z^{-1}) = \frac{1}{1 - az^{-1}} \begin{bmatrix} b_1 & b_2 \end{bmatrix} \qquad (2.3)$$

Worth noting here is that the first transfer function can be written as a standard IIR low pass filter.

$$H_1(z^{-1}) = \frac{\frac{hw}{m}}{1 - (1 - \frac{hw}{m})z^{-1}} = \frac{b_1}{1 - (1 - b_1)z^{-1}}$$

Since there will be unmodelled nonlinearities and heat losses involved in reality, it can be advantageous to have two different denominator polynomials. So the final form of the transfer functions is given in equation 2.4

$$H(z^{-1}) = \begin{bmatrix} \frac{b_1}{1-a_1z^{-1}} & \frac{b_2}{1-a_2z^{-1}} \end{bmatrix} \qquad (2.4)$$

2.3 Heat exchanger

Modeling of dynamic temperature profiles in a heat exchanger is quite complicated and the result would likely be an overly complex model. For the purpose of this thesis it was decided that estimating four transfer functions using black-box modeling with data from machine testing is the best course of action. As such the model order and structure is chosen based on trial and error in search for the best transfer functions. The resulting transfer functions are shown later in the results.

2.4 Evaporator and pipes

The evaporator will be modeled as a standard discrete low pass filter with some static attenuation. There will also be a temperature difference between T_4 and T_{ph} in figure 2.1, which is the result of heat loss. So this will also be modeled as a low pass filter with static attenuation.

3

Parameter Estimation

This chapter will go over the theory of the online parameter estimation and how it is implemented.

3.1 Least squares estimation

A simple way of estimating parameters is a linear regression method such as least squares estimation [Johansson, 2019]. The equation is seen in 3.1

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N \tag{3.1}$$

where $\hat{\theta}$ is the parameter estimation vector, Φ_N is the regressor matrix and Y_N is the output vector. This method is good for offline parameter estimation with measurement data. However for online estimation a recursive version i recommended as large matrix algebra is inefficient to perform for each iteration.

3.2 Recursive least squares estimation

The recursive equivalent of equation 3.1 with a forgetting factor is the recursive least squares (RLS) algorithm [Johansson, 2019] in 3.2

$$\begin{aligned}
\hat{\theta}_k &= \hat{\theta}_{k-1} + P_k \phi_k \varepsilon_k \\
\varepsilon_k &= y_k - \phi_k^T \hat{\theta}_{k-1} \\
P_k &= \frac{1}{\lambda}(P_{k-1} - \frac{P_{k-1}\phi_k \phi_k^T P_{k-1}}{\lambda + \phi_k^T P_{k-1} \phi_k})
\end{aligned} \tag{3.2}$$

where indexing shows iteration, ε is the estimation error, P is a parameter covariance estimate, ϕ is the regressor vector and λ is a forgetting factor which is chosen offline.

Things to note are that P_0 and θ_0 are chosen offline, and the significance of the forgetting factor λ. The purpose of the RLS is to track time varying parameters, so it is reasonable to care more about recent measurements than old measurements. By using the forgetting factor the algorithm will put more emphasis on more recent measurements, where a rough estimate of the amount of data points that are used can be calculated in equation 3.3

$$\frac{1}{1-\lambda} \tag{3.3}$$

3.3 Implementation

The purpose of the RLS is to estimate the parameters of the two heaters in the system. The idea being that over time the heaters might be affected by degrading efficiency or applied power to the heater might change. This could either slow it down or make it faster. The models for the heaters 2.4 have been repeated below for convenience. Since H_1 only describes the effect that the inlet temperature has on the outlet, it is reasonable to assume that the first transfer function will stay constant over time. It can also be argued that the pole of H_2 should stay constant through time, since the system dynamics should not change, only the efficiency of the system. With this in mind the model can be separated into two as in equation 3.4

$$H(z^{-1}) = \frac{1}{1-az^{-1}}\begin{bmatrix} b_1 & b_2 \end{bmatrix} \tag{2.4}$$

$$\begin{aligned} H(z^{-1}) &= H_1(z^{-1}) + H_2(z^{-1}) \\ H_1(z^{-1}) &= \frac{b_1}{1-a_1z^{-1}} \\ H_2(z^{-1}) &= \frac{b_2}{1-a_2z^{-1}} \end{aligned} \tag{3.4}$$

By separating them the output can also be separated into two outputs

$$\begin{aligned} Y_{out} &= Y_1 + Y_2 \\ Y_1 &= \frac{b_1}{1-a_1z^{-1}}T_{in} \\ Y_2 &= \frac{b_2}{1-a_2z^{-1}}U \end{aligned}$$

With H_1 and T_{in} known Y_{out1} can be calculated. And with that the system can be rewritten on linear regression form as in equation 3.5

$$
\begin{aligned}
Y_2 &= Y_{out} - \frac{b_1}{1 - a_1 z^{-1}} T_{in} \\
Y_2 &= \frac{b_2}{1 - a_2 z^{-1}} U \\
Y_2^k &- a_2 Y_2^{k-1} = U^k b_2 \\
&= y_k = \phi_k \theta
\end{aligned}
\tag{3.5}
$$

So in the end the parameter estimation can get away with just estimating one parameter for each heater. This is desirable as less parameters means better estimation of each parameter.

4

Controller design

This chapter will look at the controller design and the tuning methods used.

4.1 Control

The current control of the two heaters is implemented as two PI-controllers. Given the slow nature of the system, which will be more apparent later, it is reasonable to keep such a simple control design. As such the adaptive control will be a problem of optimizing PI-controllers based on estimated system parameters. The method for choosing the control parameters will be based on the lambda-method and the AMIGO-method. These are two different methods for choosing a PI-controller for a given system based on a step-response test. However they can also be used for a system with a known model.

Lambda-method

In a general case the lambda-method is used to tune a PI or PID-controller based on a step-response test of the system. This test will give the static gain K_p, the time delay L and the time constant T of a first-order representation of the system. With a known model of the system these variables can all be calculated instead and be used in the lambda-equation for a PI-controller 4.1 [Hägglund, 2017].

$$\begin{aligned} K &= \frac{1}{K_p}\frac{T}{L+\lambda} \\ T_i &= T \end{aligned} \tag{4.1}$$

In this equation K is the proportional gain and T_i is the integration time of the controller. It is up to the user to define λ but common practise is to choose it as $\lambda = T$, which will result in a closed loop system with about the same time constant as the process.

AMIGO-method

Like the lambda-method, the AMIGO-method is also based on a step-response test where K_p, L and T are given. The difference is that there is no user defined variable, instead the control is chosen based solely on the AMIGO-equation 4.2 [Åström and Hägglund, 2006].

$$\begin{aligned} K &= \frac{0.15}{K_p} + (0.35 - \frac{LT}{(L+T)^2})\frac{T}{K_pL} \\ T_i &= 0.35L + \frac{13LT^2}{T^2+12LT+7L^2} \end{aligned} \tag{4.2}$$

Robustness

Both the lambda-method and the AMIGO-method should ensure a stable closed loop system given that the parameter estimator can properly estimate the parameters. However to ensure stability of the closed loop system the proportional gain K will be saturated with both an upper and a lower limit. This will also handle initial transients from the parameter estimator. Note that with constant denominators of the heaters and a constant time delay, the integration time of the controllers will remain constant. Only the proportional gain will vary over time so there is no need to limit T_i.

First-order approximation

The above tuning methods assume a system model with first-order characteristics with no zeros. To approximate a higher order system as first-order system a general rule of thumb is to sum all the time constants of the poles and subtract all the time constants of the zeros. The time delay and the static gain stays the same.

5

Implementation

This chapter will look at the implementation of the thesis.

5.1 Simulink

The simulations in MATLAB were done using Simulink where the final simulation model can be seen below in figure 5.1.

The actual implementation is the three subsystems to the bottom left. They are called upon in order starting with the **Temperature estimator**, followed by the **Parameter estimator** and lastly the **Adaptive control tuning**. The **Controller** subsystem consists of two controllers for the two heaters and the **Sterilization system** contains all the models made for the whole system.

Temperature estimator

The temperature estimator takes the two measured temperatures T_{ph} and T_{sh} and estimates the temperatures T_1 and T_2. It also has T_0, U_{ph} and U_{sh} as inputs to synchronize the iterations after the estimations.

Estimation of T_1 The first step is to estimate T_4 and T_3 by using the model for the evaporator and the inverse model of the pipes. After that T_1 can be estimated using equation 5.1.

$$\hat{T}_4(k) = \frac{1}{pipe(z)} T_{ph}(k)$$

$$\hat{T}_3(k) = evaporator(z) T_{sh}(k)$$

$$\begin{aligned} T_4(k) &= G_{21}(z)T_1(k) + G_{22}(z)T_3(k) \\ \Leftrightarrow \hat{T}_1(k) &= \frac{1}{G_{21}(z)}(\hat{T}_4(k) - G_{22}(z)\hat{T}_3(k)) \end{aligned} \tag{5.1}$$

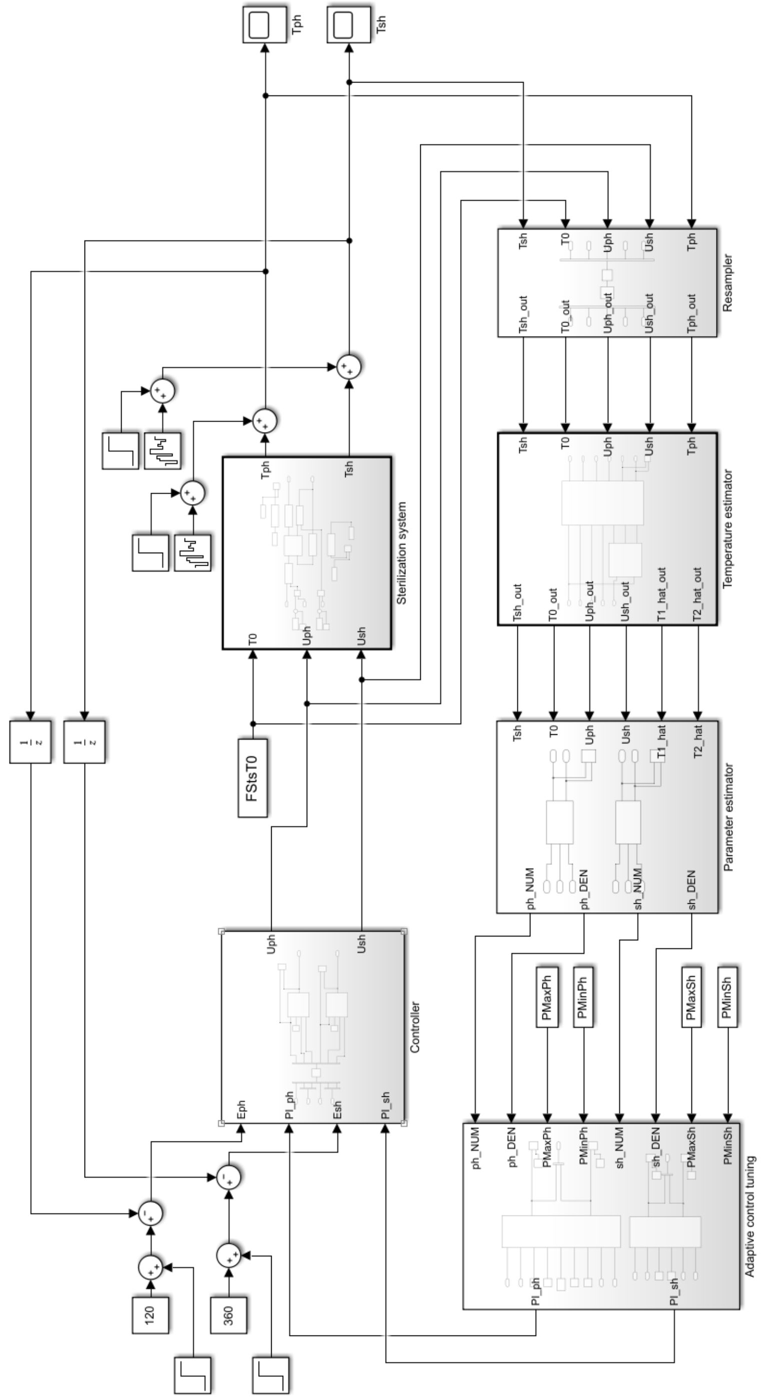


Figure 5.1: Full simulink simulation model

Estimation of T_2 With T_1 and T_3 known the estimation of T_2 is done as in equation 5.2

$$\hat{T}_2(k) = G_{11}(z)\hat{T}_1(k) + G_{12}(z)\hat{T}_3(k) \tag{5.2}$$

It can be noted that non-causality and time delays are no issues here as these calculations can be done some iterations later. The end result is to update the adaptive control which can handle some delayed calculations. However some focus has to be put on making sure that all the output values are of the same iteration so some delays will also have to be added.

Parameter estimation

The **Parameter estimator** now has access to all inputs and outputs of both the heaters. By running two RLS algorithms in parallel with the adjustments that were described in chapter 3, the numerator of both the heaters are estimated. In figure 5.1 the output also includes the denominators in case they would be adjusted to also estimate these. However for the moment these are simply constants.

Adaptive control tuning

Lastly the **Adaptive control tuning** is run using the estimated heater models. For the super heater this is straight forward as the transfer function from U_{sh} to T_{sh} is only the heater model. For the pre heater it is slightly more tricky as the transfer function from U_{ph} to T_{ph} also includes the heat exchanger, the pipes and the evaporator. Since the system contains feedback and the transfer functions contains time delays a proper transfer function cannot be derived for the full system. Instead a transfer function from T_1 to T_{ph} can be approximated and then used instead when computing the control parameters.

In figure 5.2 an overview of how U_{ph} affects T_{ph} can be seen. To tune the pre heater using one of the methods require that this whole system is approximated as a first order system. The input signals T_0 and U_{sh} can be seen as disturbances so they can be discarded in this case. By rewriting figure 5.2 to figure 5.3 and estimating a transfer function for H we have a system that can be used for the tuning of the pre heater.

5.2 TwinCat

The implementation in Beckhoff TwinCat is essentially exactly the same as in simulink. Using the PLC Code Generator in Simulink the above three subsystems were ported over to TwinCat where they are called upon in the same manner as in Simulink.

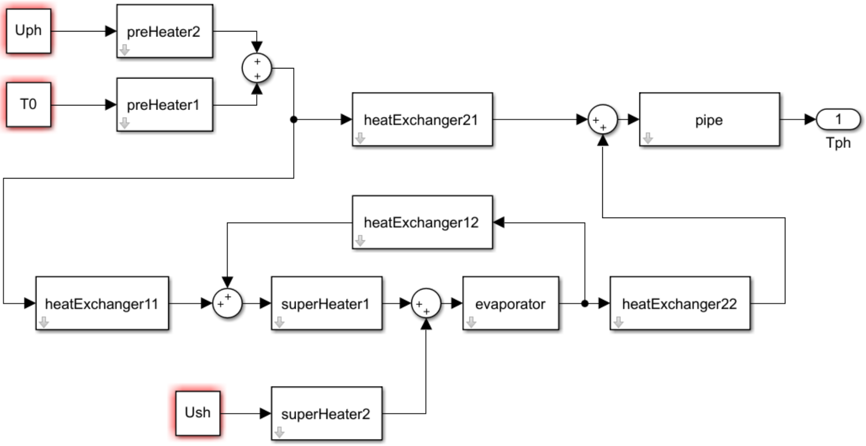


Figure 5.2: Model overview for the pre heater.

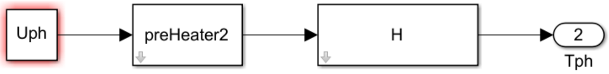


Figure 5.3: Simplified overview for the pre heater.

6

Results

This chapter will cover the result of this thesis.

6.1 Modeling

During the modeling of the system some issues arose regarding the simplicity of the models. Since none of the models take the surrounding room temperature into consideration, it can be said that they are essentially linearized around the room temperature of the model estimation data. This became apparent when the same models were used with new measurement data from about one month after the initial modelling was done.

In this section the model output for all the models will be compared to the measurement data for both the first and the second measurements. All the components will be tested individually, meaning that they will use measurement data as inputs and the output will be compared to the measurement data output. During the second measurements there is a setpoint change at around t = 1750 s. The models are linearized around the second setpoint, which is the setpoint that is kept during production. This is why the modeling generally performs better after the change. However due to the difference in room temperature they tend to have some static deviation from the measurements. Before this setpoint change there are some steps taken in the machine that affect air flow and other things that affect the system, which is why there are some spikes and other weird behaviour going on.

Pre heater

The final pre heater model was estimated as 6.1 where both the discrete and the continuous transfer functions are shown.

$$\begin{aligned} H_{ph}(z^{-1}) &= \begin{bmatrix} \frac{0.02748}{1-0.9725z^{-1}} & z^{-2}\frac{0.05145}{1-0.9725z^{-1}} \end{bmatrix} \\ G_{ph}(s) &= \begin{bmatrix} \frac{0.02748s+0.1393}{s+0.1393} & e^{-0.4s}\frac{0.05145s+0.2608}{s+0.1393} \end{bmatrix} \end{aligned} \tag{6.1}$$

with the sampling time $h = 0.2$ and with the inputs T_0 and U_{ph}. The time varying parameter is expected to be the second discrete numerator $b_{2ph} = 0.05145$. In figure 6.1 the model output is compared to the measurement data in simulink. From this comparison it is quite clear that the model seems to cover the dynamics of the heater quite well and should be sufficient. It is also very apparent from the setpoint change that the model is quite heavily linearized around the heat losses of the second setpoint. After the setpoint change the static offset is likely because the second measurements were taken one month later during the spring, where the room temperature could very well be higher which would result in this kind of behaviour. This would not only affect the heat losses to the surrounding air, but also the inflow of air to the pre heater which is room temperature. So the reason for the static offset is most likely because of the room temperature.

Super heater

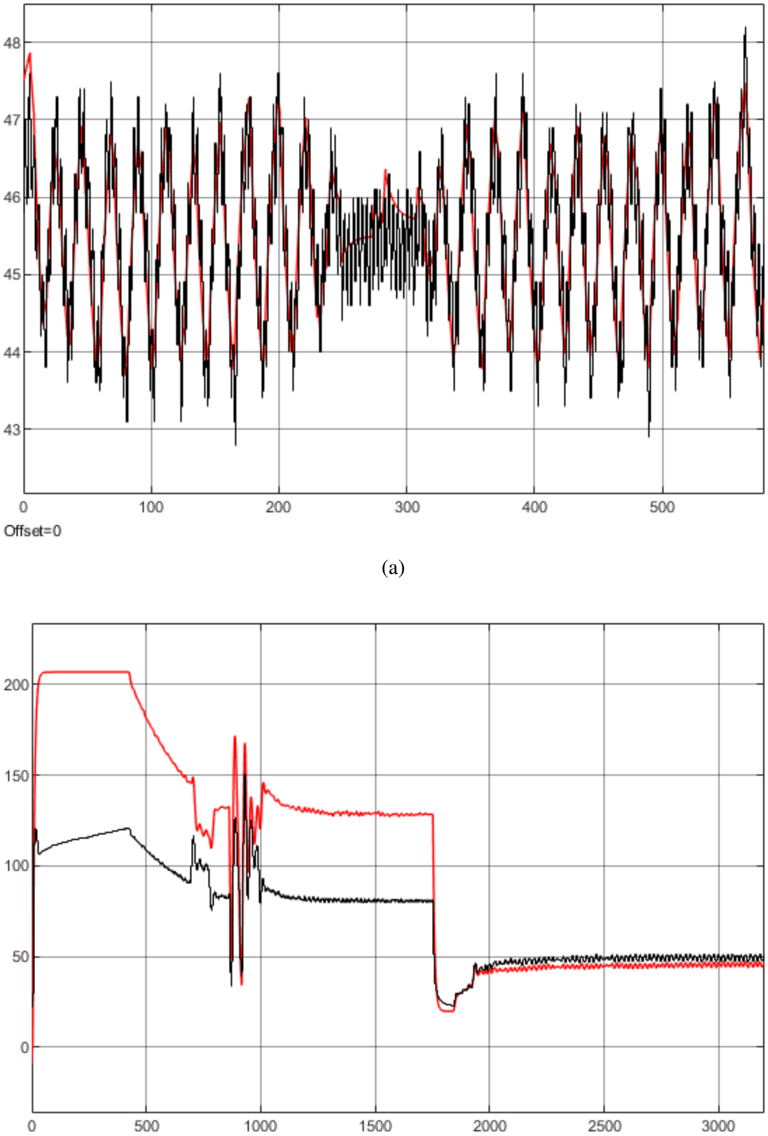
The final super heater model was estimated as 6.2

$$
\begin{aligned}
H_{sh}(z^{-1}) &= \begin{bmatrix} \frac{0.0005}{1-0.9995z^{-1}} & z^{-40}\frac{0.001672}{1-0.9993z^{-1}} \end{bmatrix} \\
G_{sh}(s) &= \begin{bmatrix} \frac{0.0005s+0.002501}{s+0.002501} & e^{-8s}\frac{0.001672s+0.008363}{s+0.00336} \end{bmatrix}
\end{aligned}
\tag{6.2}
$$

with the sampling time $h = 0.2$ and with the inputs T_2 and U_{sh}. The time varying parameter is expected to be the second discrete numerator $b_{2sh} = 0.001672$. In figure 6.2 the model output is compared to the measurement data in simulink. This model is not quite as good as the pre heater model even for the first measurements, which is likely because the super heater is quite large. During the modeling there was quite a large payoff between simplicity and performance. Since a RLS is going to try to estimate the parameters it was decided that a simple model was the best approach. However the simplified model that assumes homogeneous temperature inside the heater probably fails here. As with the pre heater there seems to be a static offset that is slightly below the data in the end of the second measurements, which again is probably because of room temperature difference and heat losses associated with it.

Heat exchanger

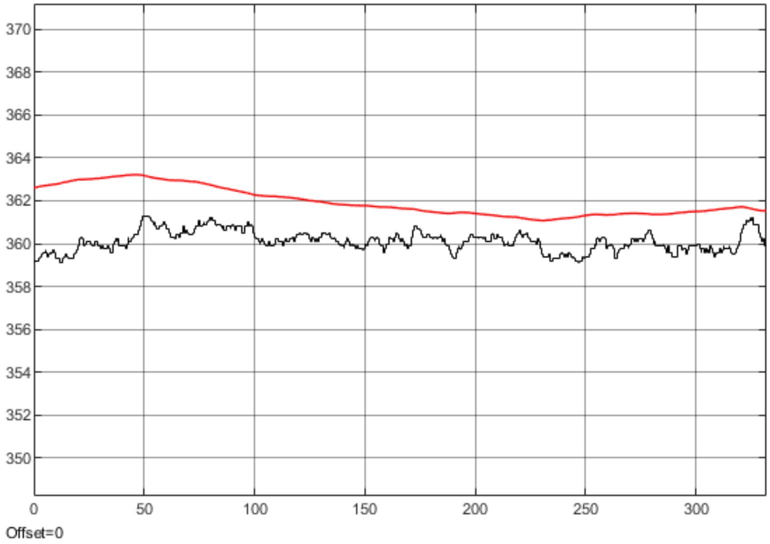
The final heat exchanger model was estimated to eq. 6.3



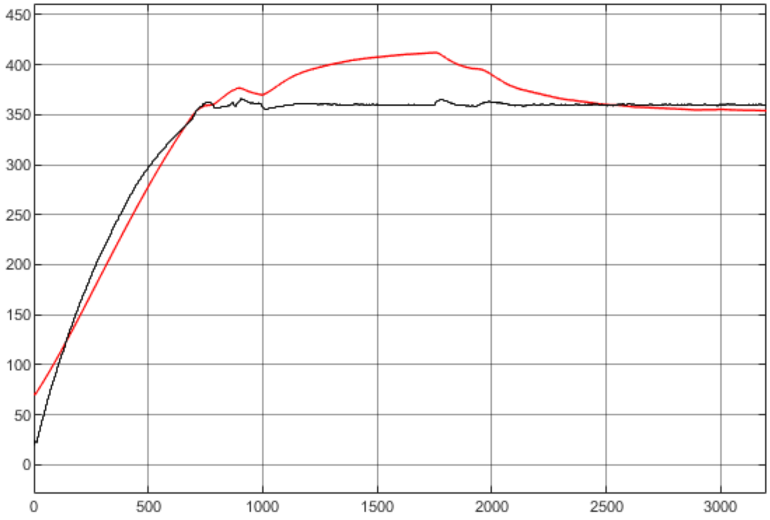
(a)

(b)

Figure 6.1: Pre heater model output using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model output and the black graph is the measurement output. The x-axis is in seconds.



(a)



(b)

Figure 6.2: Super heater model output using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model output and the black graph is the measurement output. The x-axis is in seconds.

$$
\begin{aligned}
H_{11he}(z^{-1}) &= z^{-2}\frac{-0.003102+0.006345z^{-1}-0.003242z^{-2}}{1-1.994z^{-1}+0.9942z^{-1}}\\
H_{21he}(z^{-1}) &= z^{-1}\frac{0.01466+0.005184z^{-1}-0.007031z^{-2}}{1-1.724z^{-1}+0.7274z^{-2}-0.001071z^{-3}}\\
H_{12he}(z^{-1}) &= z^{-2}\frac{0.004195-0.004782z^{-1}+0.0005879z^{-2}}{1-1.994z^{-1}-0.9942z^{-2}}\\
H_{22he}(z^{-1}) &= z^{-1}\frac{0.02728-0.05867z^{-1}+0.03173z^{-2}}{1-1.724z^{-1}+0.7274z^{-2}-0.001071z^{-3}}\\
H_{he}(z^{-1}) &= \begin{bmatrix} H_{11he}(z^{-1}) & H_{12he}(z^{-1}) \\ H_{21he}(z^{-1}) & H_{22he}(z^{-1}) \end{bmatrix}
\end{aligned} \tag{6.3}
$$

with the sampling time h = 0.004 s. In figure 6.3 the model output is compared to the measurement data in simulink. The higher frequency of the measurement data in 6.3 (a) is currently unexplained and has not shown up again since, so it is likely not a phenomenon to consider important. Besides that it looks like the model is working better for T_4 than for T_2, but the static gain is still almost the same and with the slow dynamical nature of T_2, it should be fine. What is interesting here is that there is a positive static offset for T_4 (d) instead of a negative offset for the earlier models. This is quite surprising since the heat exchanger in the machine is not thermally isolated and gives off quite a lot of heat energy to the surrounding air. This has not been understood as of yet.

Evaporator and pipes

The transfer functions for the Evaporator and the Pipes are given in equation 6.4

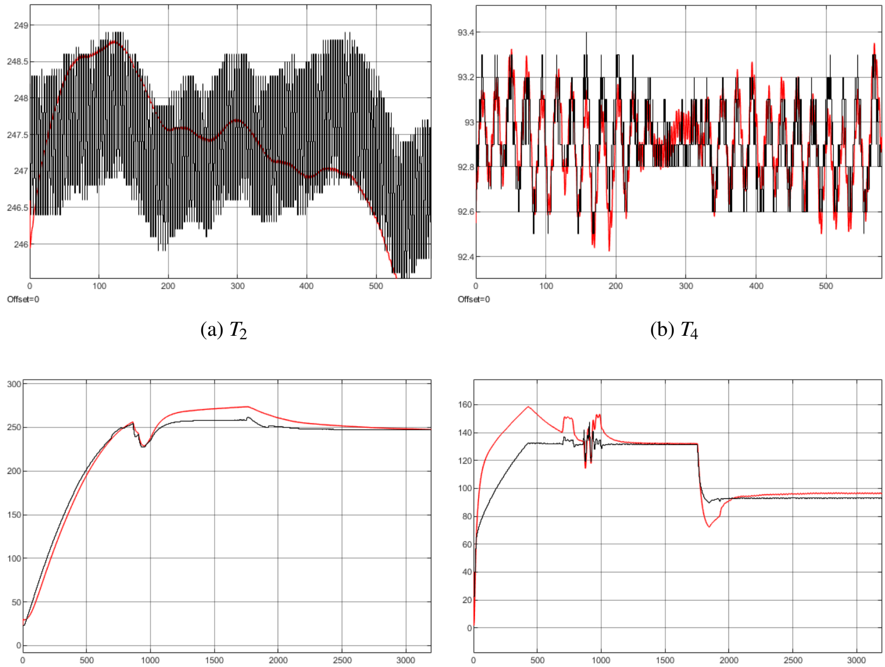
$$
\begin{aligned}
H_{Evaporator}(z^{-1}) &= \frac{0.000035705}{1-0.999958z^{-1}}\\
H_{Pipes}(z^{-1}) &= \frac{0.2801-0.2797z^{-1}}{1-0.9995z^{-1}}
\end{aligned} \tag{6.4}
$$

where the sampling time is h = 0.004. Note here that the transfer function for the Pipes is a low-pass filter with finite attenuation for infinite frequency. However when looking at the bode plot in figure 6.4 it still attenuates higher frequencies.

In figures 6.5 and 6.6 the model outputs are compared to the measurement data in simulink. The disturbance frequency in fig 6.5 (a) can again be ignored.

Full system

With everything put together the full system can be simulated and compared to measurement data. In figure 6.7 the measurement of T_{ph} can be seen. For the first



(a) T_2

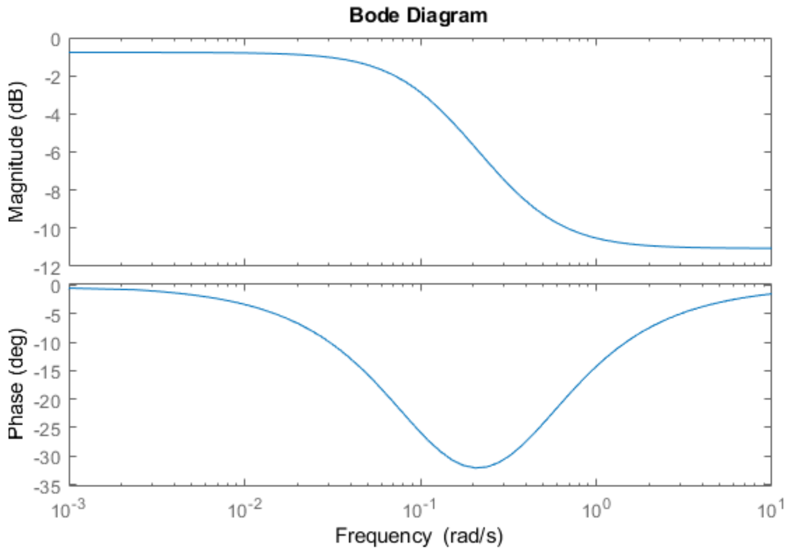
(b) T_4

(c) T_2

(d) T_4

Figure 6.3: Heat exhanger model output using: (a) and (b) the first measurement data, and (c) and (d) the second measurement data. The red graph is the model output and the black graph is the measurement output. The x-axis is in seconds.

measurements it cannot quite follow the behaviour of the data but the gain is still very close, so the modeling could not be much better realistically. For the second measurements the behaviour looks similar to the behaviour of the pre heater, which makes sense. The positive offset can be explained by the large overshoot of the super heater, which can be seen in figure 6.8 (b). This temperature feeds back into the heat exchanger through the evaporator. The overshoot itself is because of the unmodeled behaviour of the system when air flows and other things affect it before it goes into production. There is unfortunately not enough data to see if figure 6.8 (b) will converge to the measurements, but since figure 6.2 (b) is close it should be somewhat close at least. In figure 6.8 (a) there is some manual setpoint changes to the system to see how the super heater reacts with more excitation. While it seems to handle it fairly well it looks like the time delay is a bit off. However it should not matter in the end as the setpoint for the super heater is always set to 360 degrees, so it should not be particularly excited under normal circumstances.



(a)

Figure 6.4: Bode plot for the transfer function of the pipes

Temperature estimator

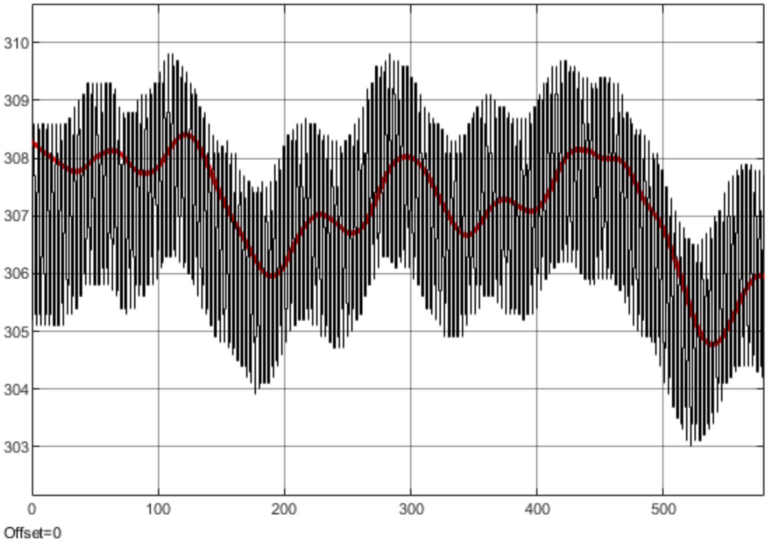
The end result of the modeling is the temperature estimator that estimates T_1 and T_2. In figures 6.9 and 6.10 these estimates can be seen compared to measurement data. The estimate $\hat{T}_1$ seems to be working very well and with only a slight static offset in the end in figure 6.9 (b). The estimate $\hat{T}_2$ is not quite as good, however when keeping in mind that T_2 changes very slowly and that the static gain is still close, it should be good enough. Note also that in 6.10 (a) there are some initial conditions that are causing some issues with the plot. For some reason MATLAB had a difficult time trying to estimate these conditions.

6.2 Parameter Estimation

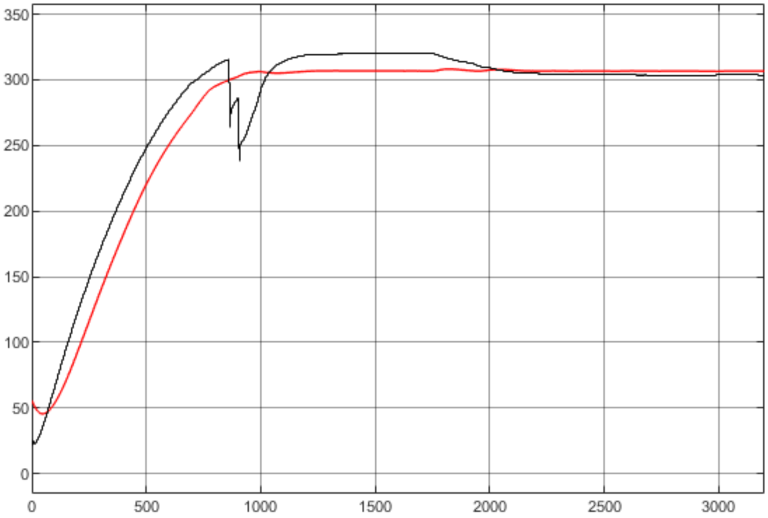
In this section the parameter estimation results will be covered. Both of the RLS estimators are run with the sample time h = 0.2 seconds so they will estimate the same parameters as in equations 6.1 and 6.2.

Pre Heater

For the pre heater the RLS is run with $P_{init} = 1000, \lambda = 0.999$. With this λ the RLS will be very slow to pick up changes in the parameters which is fine for this system. There is nothing that depends on fast changes to the system so for that reason the



(a)



(b)

Figure 6.5: Evaporator model output using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model output and the black graph is the measurement output. The x-axis is in seconds.

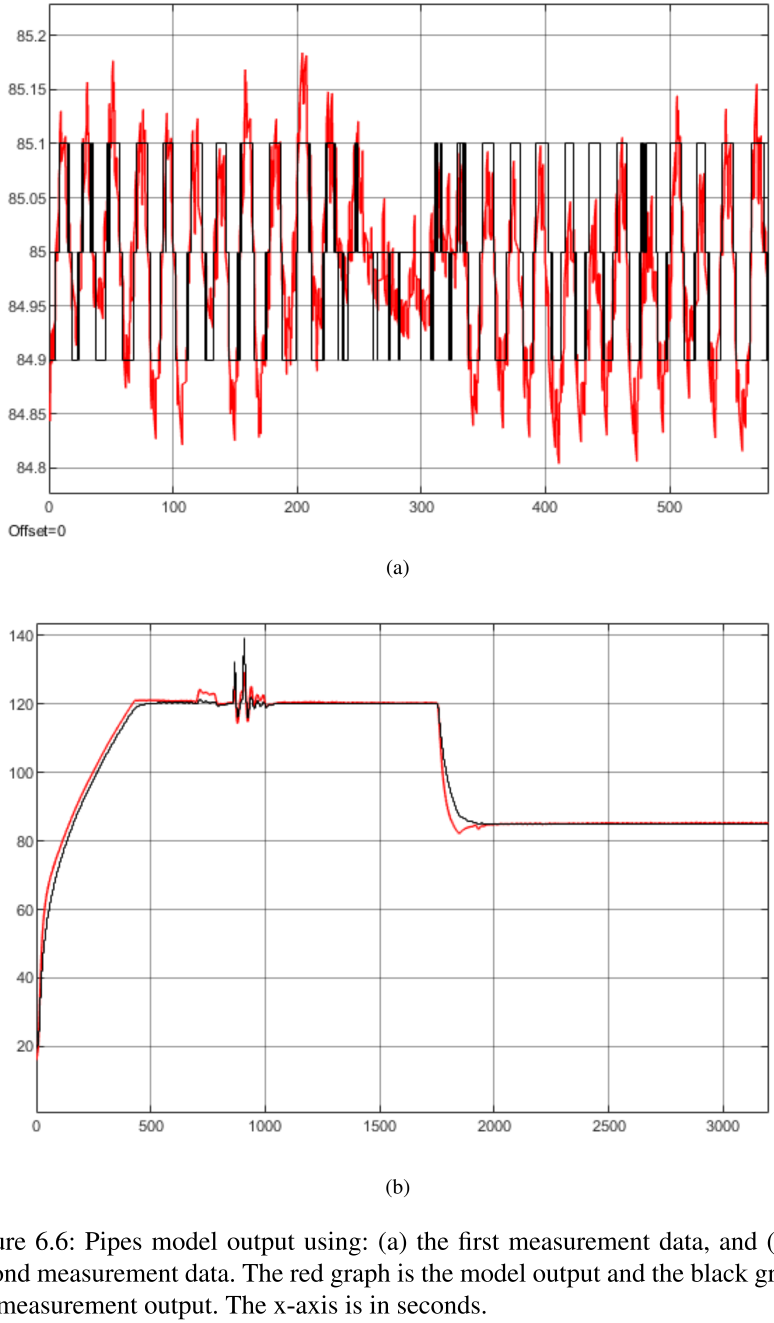


Figure 6.6: Pipes model output using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model output and the black graph is the measurement output. The x-axis is in seconds.

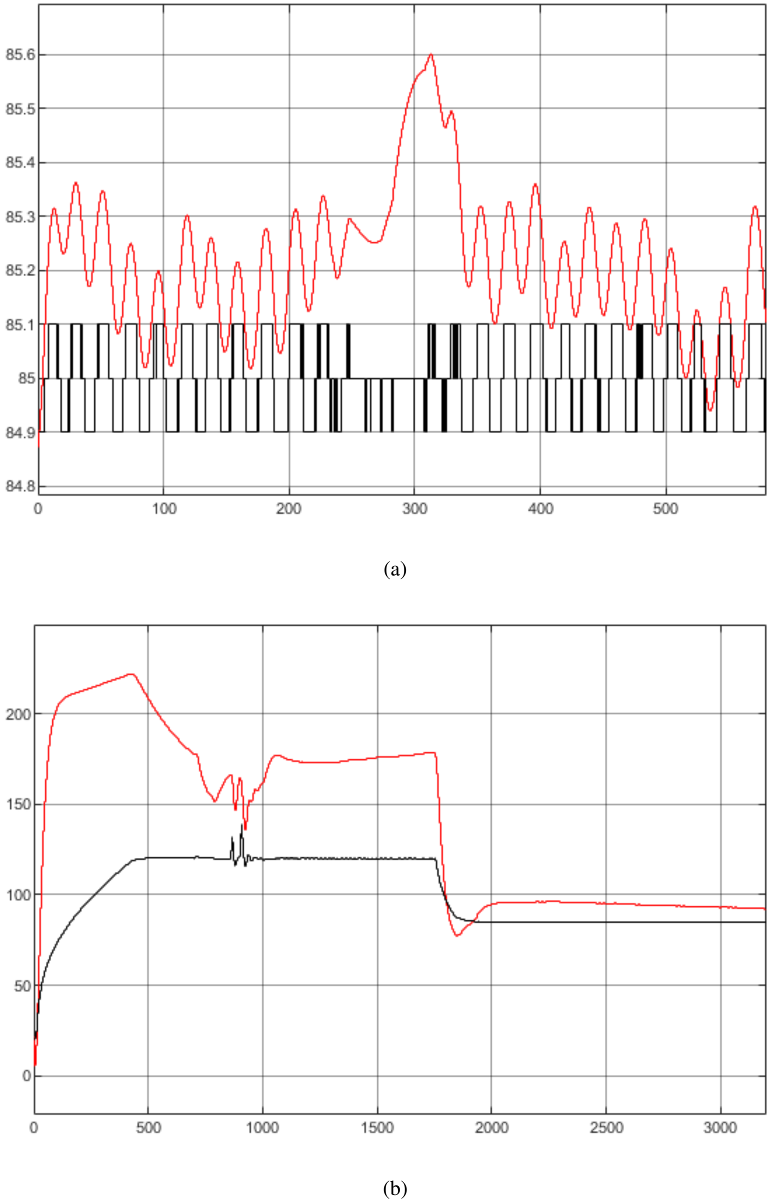
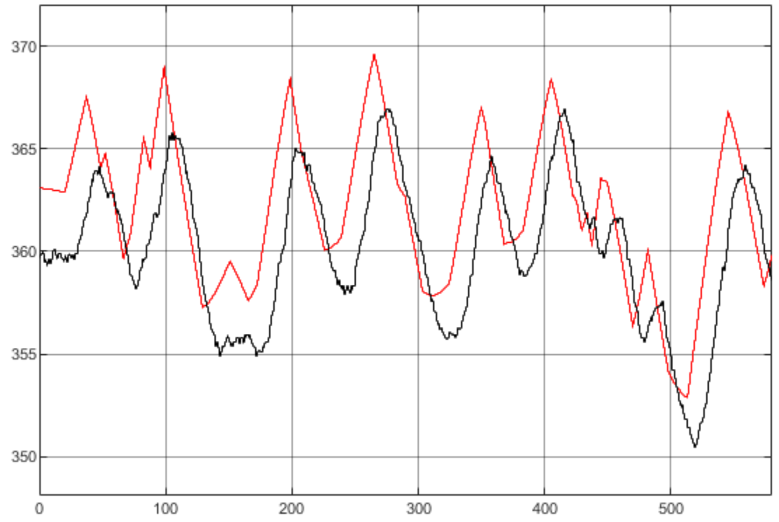
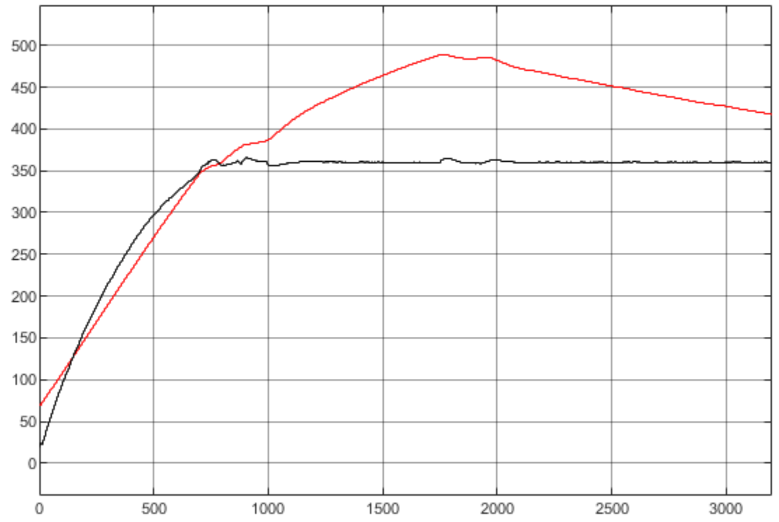


Figure 6.7: Full system model output T_{ph} using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model temperature and the black graph is the measurement output. The x-axis is in seconds.

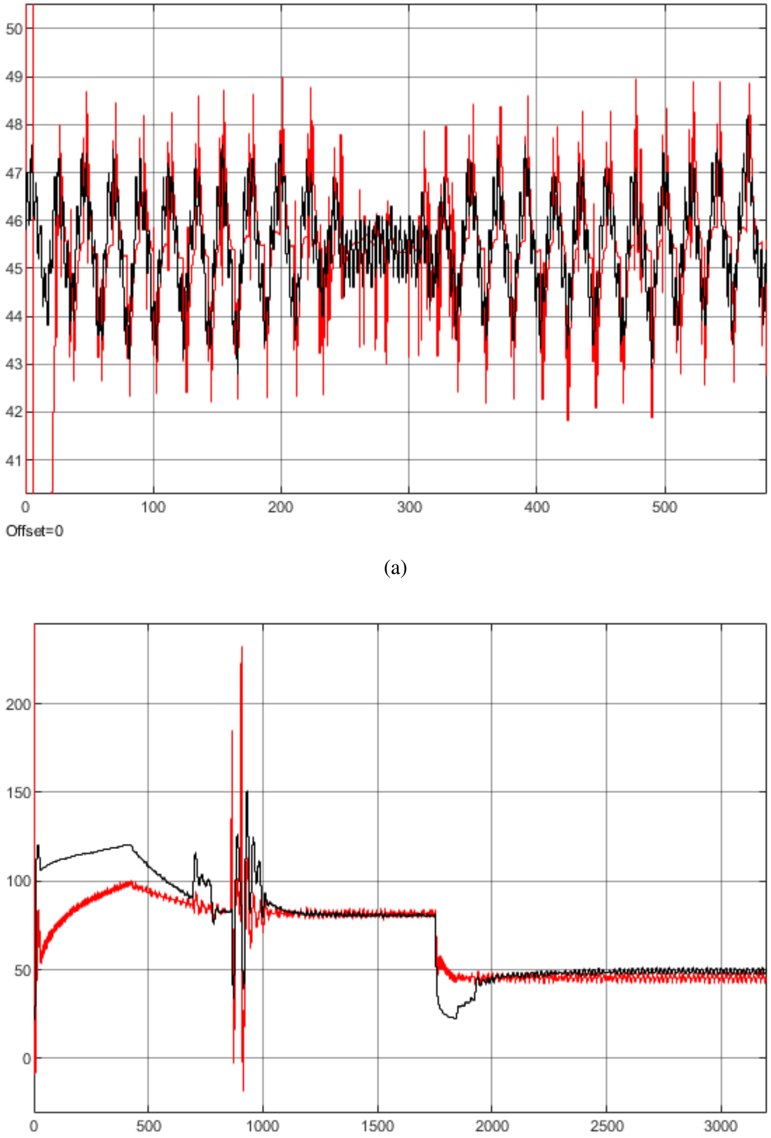


(a)



(b)

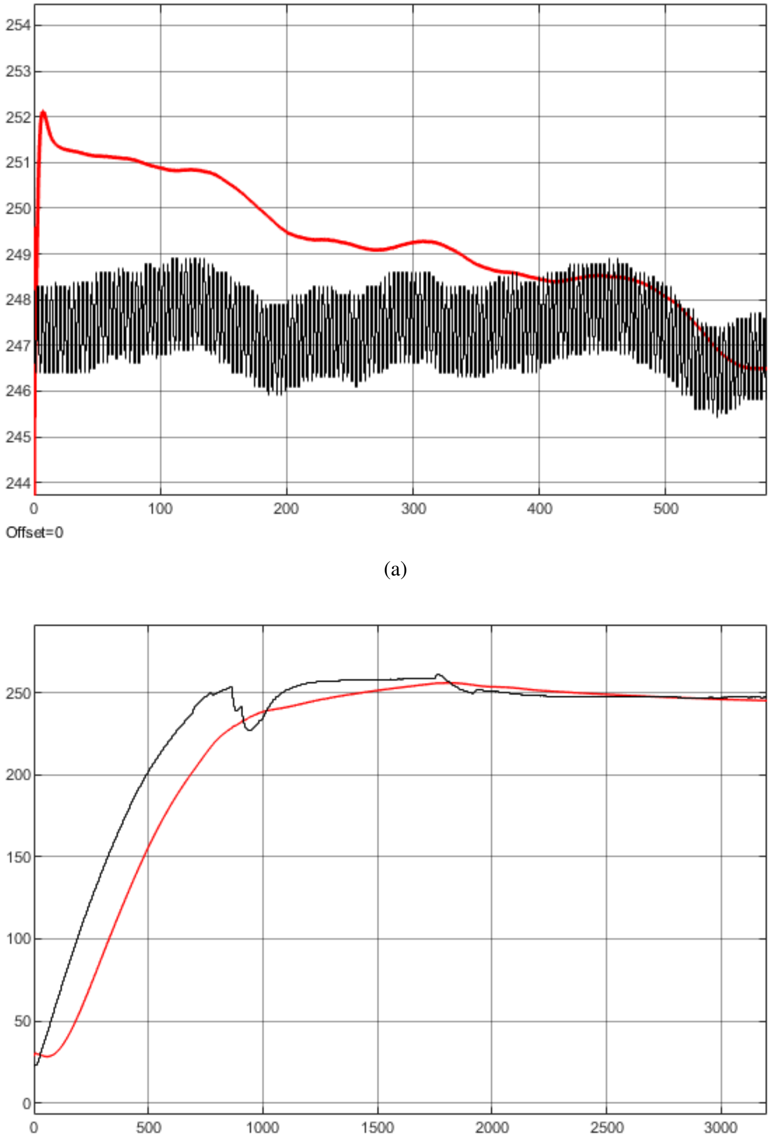
Figure 6.8: Full system model output T_{sh} using: (a) the first measurement data, and (b) the second measurement data. The red graph is the model temperature and the black graph is the measurement output. The x-axis is in seconds.



(a)

(b)

Figure 6.9: Temperature estimator output $\hat{T}_1$ using: (a) the first measurement data, and (b) the second measurement data. The red graph is the estimated temperature and the black graph is the measurement output. The x-axis is in seconds.



(a)

(b)

Figure 6.10: Temperature estimator output $\hat{T}_2$ using: (a) the first measurement data, and (b) the second measurement data. The red graph is the estimated temperature and the black graph is the measurement output. The x-axis is in seconds.

slower the better, as the slow estimation should give better estimates. In figure 6.11 it is run with measurement data as input. It is not surprising that the estimate is way off at the beginning in (b) compared to the offline model, as the system is not in the linear region of the model. The offset in the end is again likely because of room temperature difference. It can be interpreted as the gain of the system with some scaling done by the denominator of the transfer function. So what it estimates is a system with a higher gain than the offline model. This makes sense as with a higher room temperature the system does not suffer as much heat loss, which means that it can input less effect to get the same result. So it is the same as having a higher gain.

Super Heater

For the Super heater the RLS is run with $P_{init} = 1000, \lambda = 0.999$. The same reasoning regarding the value of λ is done here as for the pre heater. In figure 6.12 it is run with measurement data as input. It seems to converge to a value slightly higher than the offline model, which again makes sense. It does however seem to be quite sensitive to noise.

6.3 Control tuning

This section will look at the control tuning of the two controllers. To begin with the simplified model estimation for the pre heater will be shown. Then the two methods will be shown in an offline tuning followed up by an online tuning.

Pre heater approximation

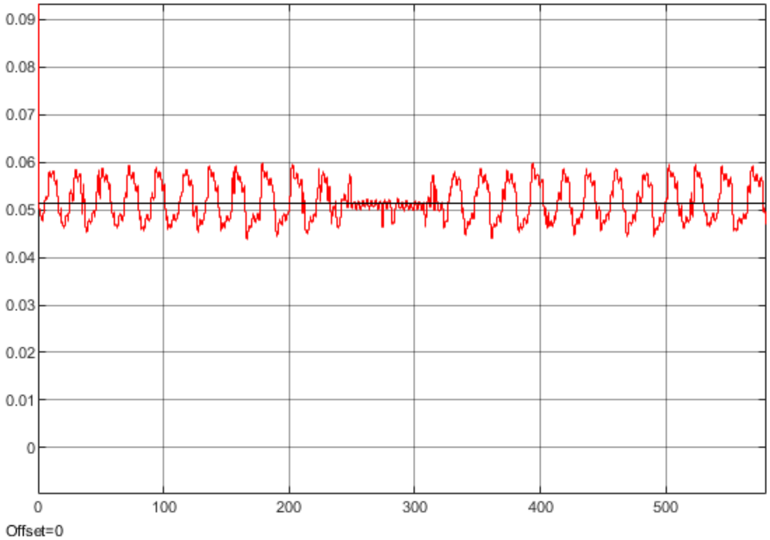
In figure 6.13 the approximation of H in figure 5.3 is shown, along with equation 6.5 for the continuous transfer function. The step responses look very similar and should be good enough. However the bode diagram is giving some more information about the static gain. The behaviour is the result of a system with a mix of both fast and very slow poles where there is a magnitude difference of up to 4. It was decided that this approximation was good enough, however more time could have been spent on a better approximation.

$$H_{approximation}(s) = e^{-0.8s}\frac{0.97}{25s+1} \tag{6.5}$$

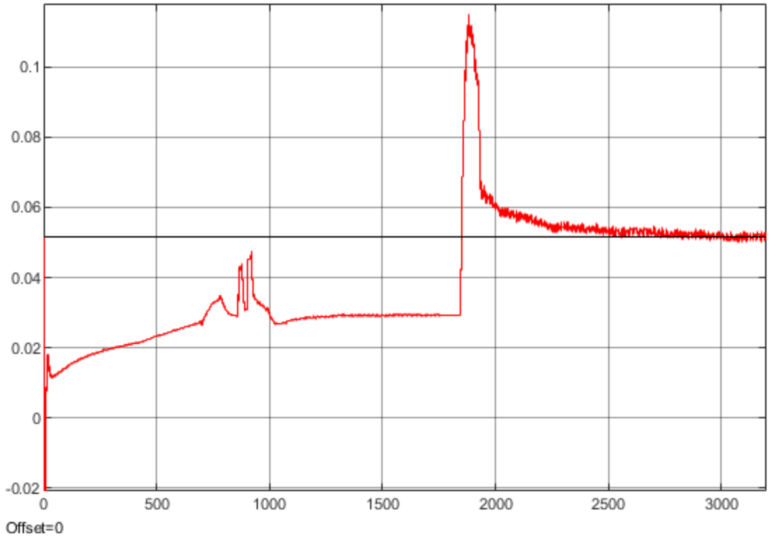
Offline control tuning

An initial assessment of the viability of the methods can be gained from looking at the tuning offline first.

The first order parameters for the two heaters are given in table 6.1, using the rule of thumb for the approximation of the time constant. In table 6.2 the offline PI-control parameters are calculated using the model parameters in table 6.1. In figure 6.14 the step response of the closed loop system can be seen with these controllers.

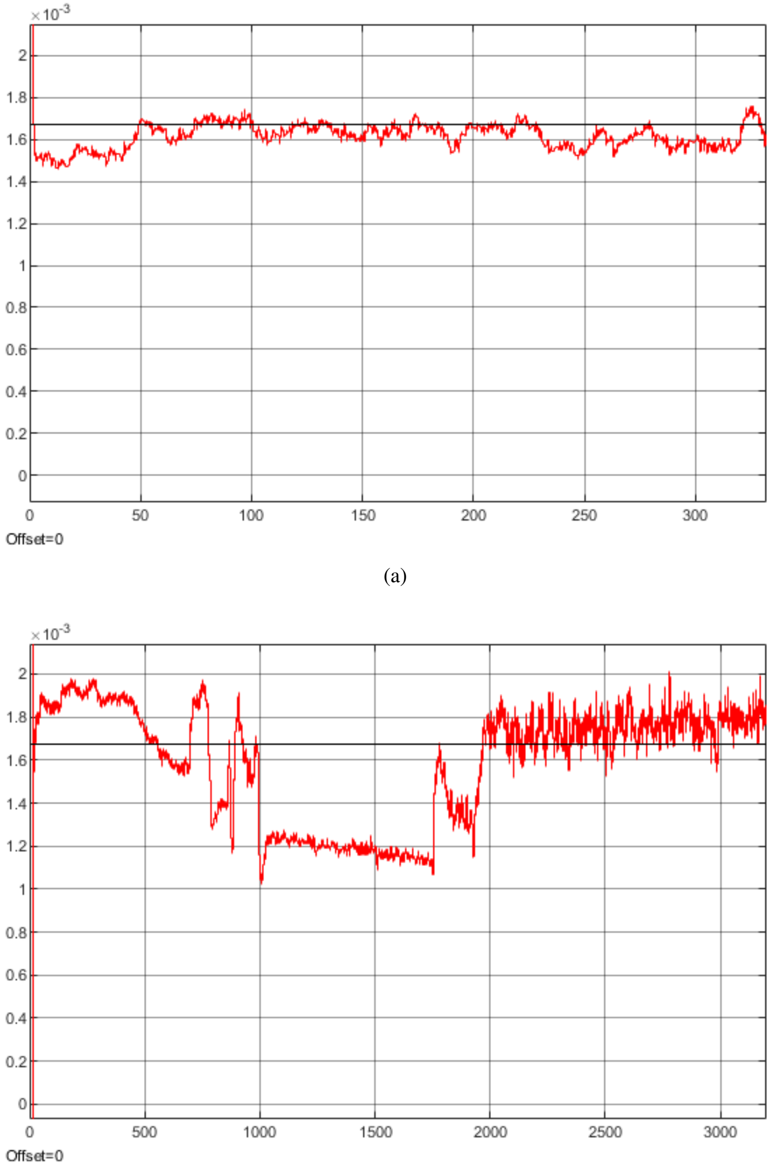


(a)



(b)

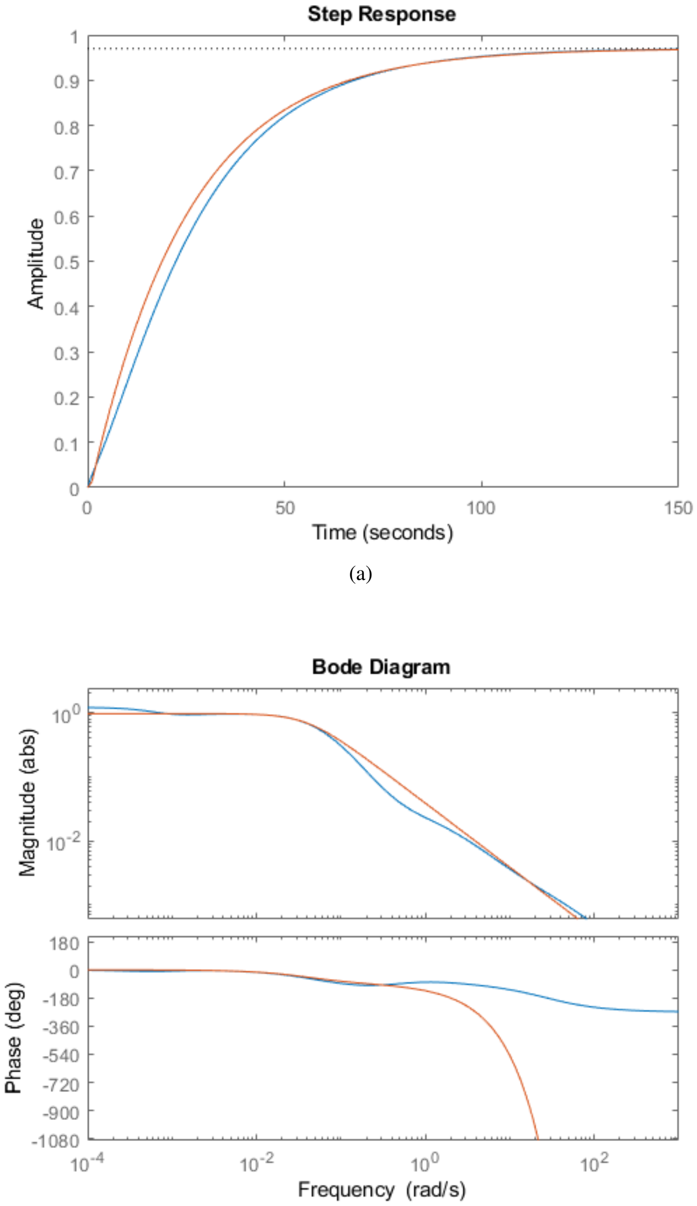
Figure 6.11: Parameter estimator estimate of b_{1ph} using: (a) the first measurement data, and (b) the second measurement data. The red graph is the estimated parameter and the black graph is the offline estimate. The x-axis is in seconds.



(a)

(b)

Figure 6.12: Parameter estimator estimate of b_{1sh} using: (a) the first measurement data, and (b) the second measurement data. The red graph is the estimated parameter and the black graph is the offline estimate. The x-axis is in seconds.



(a)

(b)

Figure 6.13: Approximation of the system following the pre heater. The approximation in orange and the modelled system in blue.

Model parameters		
	pre heater	super heater
K_p	1.82	2.49
L	1.2	8
T	32.18	297

Table 6.1: Model parameters for the first order approximation of the heaters.

Control parameters				
	pre heater		super heater	
	K	T_i	K	T_i
Current	14	40	3	50
Lambda-method	0.53	31.98	0.3912	297.42
AMIGO-method	4.707	11.12	4.907	81.39

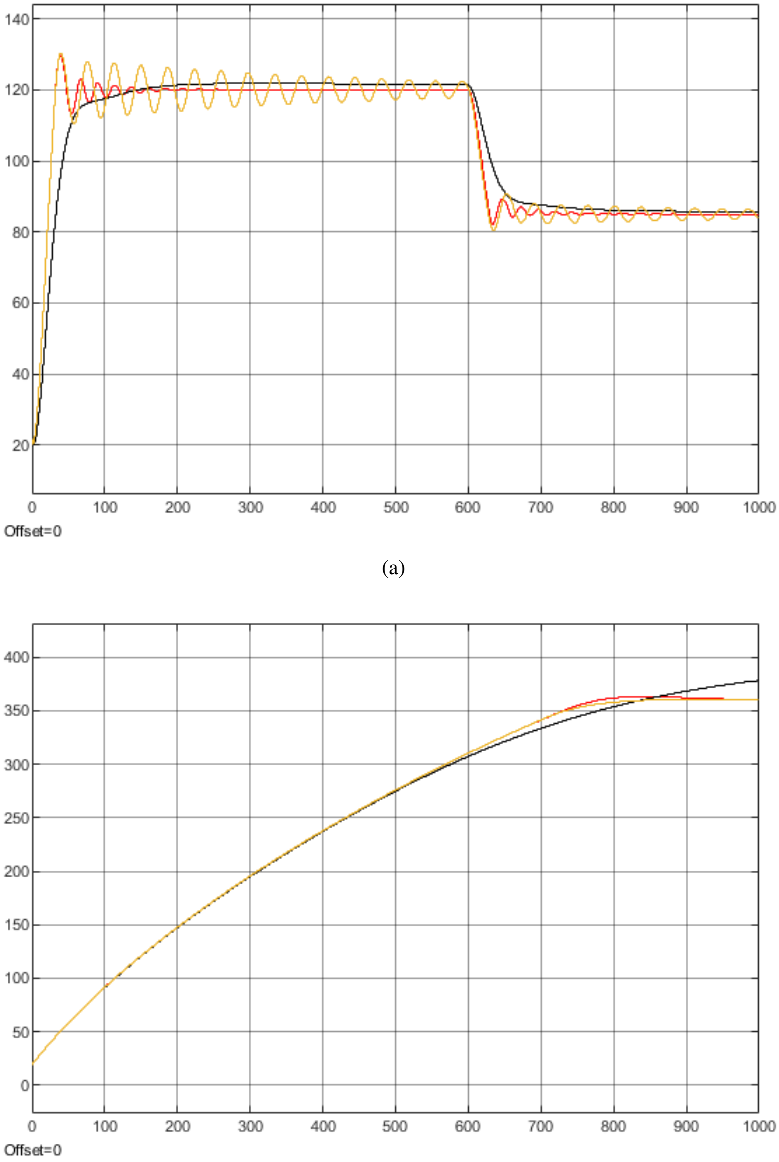
Table 6.2: Control parameters for the offline tuning of the PI-controllers using the lambda-method and the AMIGO-method. The current control parameters in the machine are also included.

Looking first at the current control of the pre heater it is clear that there are some oscillations caused by an aggressive control. The lambda-method avoids this but is considerably slower. There is also some overshoot for the lambda control, however this is because the super heater can be seen as a ramp-like disturbance on the system, which is difficult for a PI-controller to handle. Finally the AMIGO-method seems to be even more aggressive than the current control which results in heavy oscillations. The reason for this is described in [Åström and Hägglund, 2006], which states that the ratio of the dead time and the time constant for the pre heater is difficult to handle for the AMIGO-method.

For the super heater the current control has a very slight overshoot and behaves very similar to the AMIGO-method that has zero overshoot. The lambda-method however has an extreme overshoot and the control is much too slow for the system. This is because the system has a very large time constant and the lambda-method used here is using $\lambda = T_i$. There are better ways of choosing the value for λ in this case, but since the AMIGO-method looks good and the lambda-method is best for the pre heater it was decided to proceed with the AMIGO-method for the super heater.

Online control tuning

The online control tuning using measurement data can be seen in figure 6.15 using the second measurements. Since the inputs to the control tuner are the estimated parameters in figures 6.11 and 6.12, they are repeated in 6.15. It looks like the proportional gain is in some sense mirrored by the parameter estimates around the of-



(a)

(b)

Figure 6.14: Step response for the tuning of the pre heater (a) and super heater (b). In the figures the red graph is the current control in the machine, the black graph is the lambda tuning and the yellow graph is the AMIGO tuning. The x-axis is in seconds.

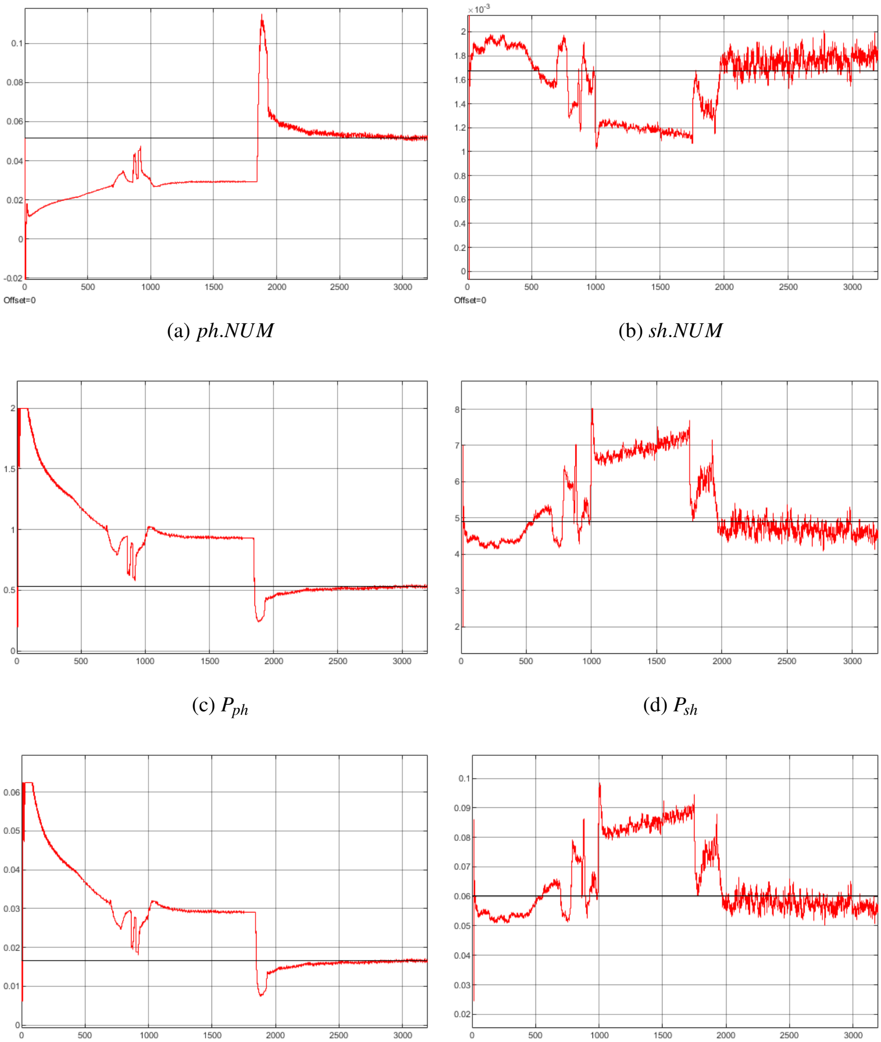
fline parameters of the models and the controllers. This makes sense since a smaller estimated model parameter would in this case give a lower model gain, which a controller would want to make up for by increasing the proportional gain. Notice also that the initial transient of the pre heater causes the proportional gain to saturate at 2, which is the saturation limit given to the controller to ensure reasonable control parameters. The integral gain of the controllers have the same appearance as the proportional gain except for some scaling. In another application with time varying time constants these could also be estimated which would cause more variations in the integral time of the controllers. But in this case, since the time constants are constant, the integral time is also constant.

6.4 Simulations in Simulink

Finally with the full system implemented in Simulink the end result can be seen in figure 6.16. The parameter estimation converges nicely after the parameter change but the noise affects the estimation and the tuning quite heavily, given that the noise itself is rather small. However the applied noise is still much more severe than what is realistically encountered in the real machine.

6.5 Simulations in TwinCat

After porting over the simulink code to TwinCat a similar simulation was done using the simulation environment with the actual code for the machine. In figure 6.17 the result of this simulation can be seen for each of the heaters.



(a) *ph.NUM*

(b) *sh.NUM*

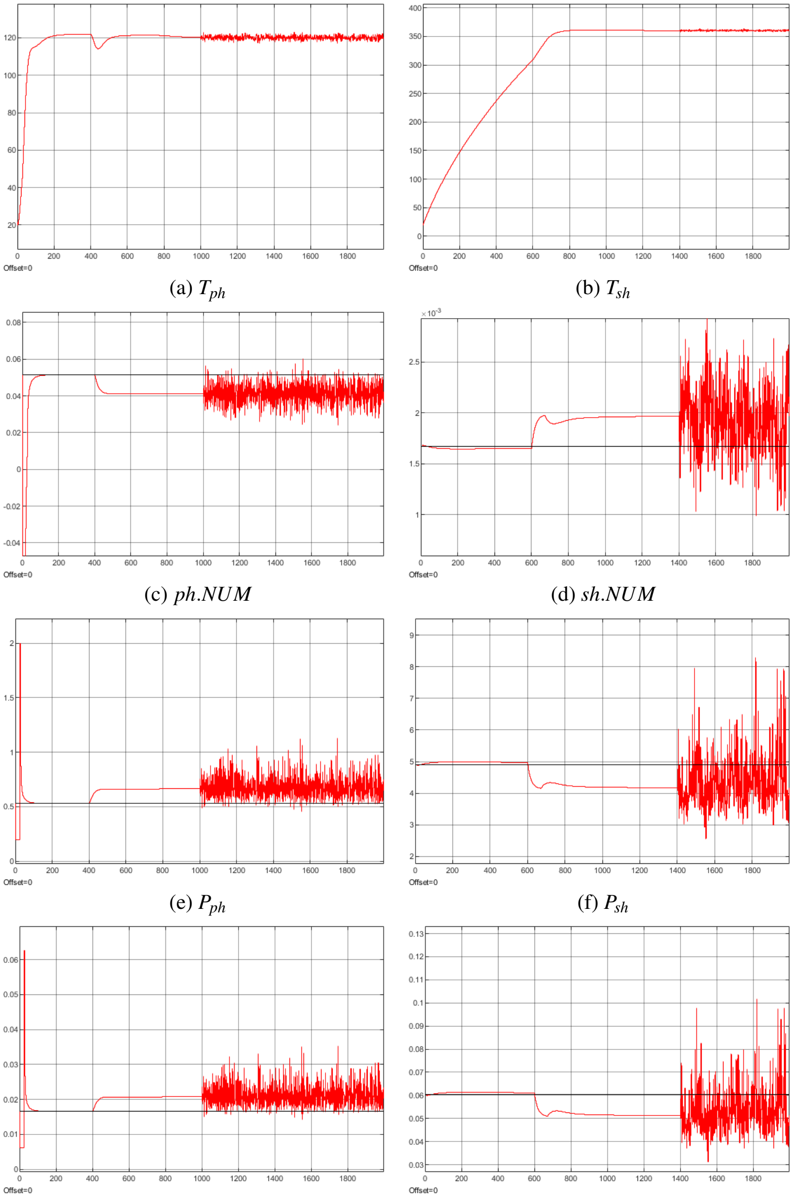
(c) P_{ph}

(d) P_{sh}

(e) I_{ph}

(f) I_{sh}

Figure 6.15: Online control tuning for the pre heater (left) and for the super heater (right). The red graphs are the online control parameters and the black graphs are the offline control parameters. Note that P stands for the proportional gain and $I = P/T_i$ for the integral gain. The x-axis is in seconds.



(a) T_{ph}

(b) T_{sh}

(c) $ph.NUM$

(d) $sh.NUM$

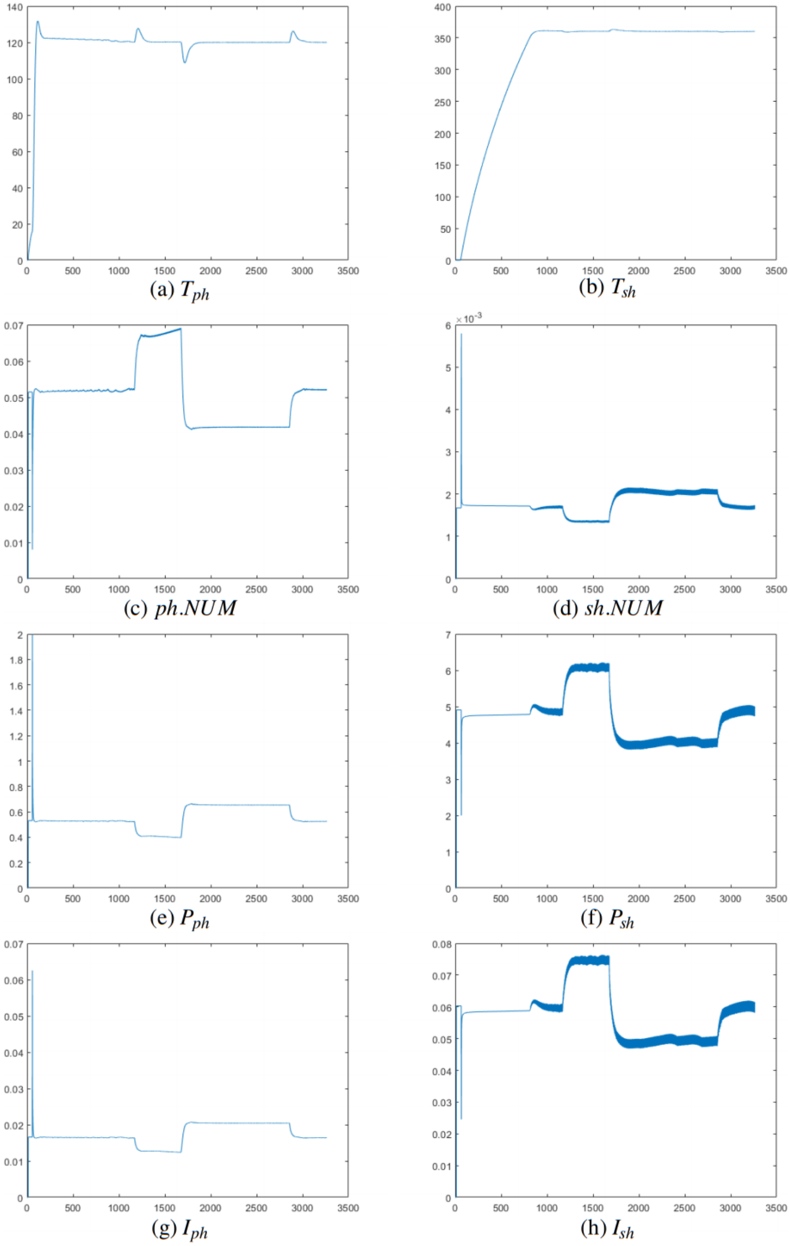
(e) P_{ph}

(f) P_{sh}

(g) I_{ph}

(h) I_{sh}

Figure 6.16: Online control tuning for the pre heater (left) and the super heater (right). At t = 400 the efficiency of the pre heater is reduced by 0.2 and at t = 600 the efficiency of the super heater is increased by 0.2. Measurement noise is added at t = 1000 and t = 1400. The x-axis is in seconds.



(a) T_{ph}

(b) T_{sh}

(c) $ph.NUM$

(d) $sh.NUM$

(e) P_{ph}

(f) P_{sh}

(g) I_{ph}

(h) I_{sh}

Figure 6.17: Online control tuning for the pre heater (left) and the super heater (right) in TwinCat. The efficiency of the heaters are changed at t = 1200, 1700 and 1900. The x-axis is in seconds.

7
Discussion

This chapter will cover the discussion of the results of this thesis.

Due to time constraints the implementation was unfortunately never used on an actual machine. However looking at the modeling it looks promising in the linearized area so it would not be too surprising if it ended up working in the end. Everything seems to be converging to their expected values in terms of model temperatures, parameter estimates and control parameters. However the noise sensitivity in figure 6.16 is a bit worrisome especially for the super heater. The noise is barely visible and still the estimation is very noisy. For this specific process this is likely not an issue given how noise free the measurements are. However for a different process where noise is an issue this might be concerning. Based on the nature of the time variance of the system dynamics it should not be an issue to low-pass filter either the measurements or the parameter estimates such that the time variance is below the break frequency of the filter. In the case of the system in this thesis it would be a valid approach as any changes to the system would happen very slowly. When looking at the TwinCat implementation the system seems to perform about the same, however there is some unexpected frequency showing up in the parameter estimate of the super heater that is currently unexplained.

For this system the robustness was ensured by setting saturation limits on the proportional gain to keep the controller within a stable region. These limits were chosen somewhat arbitrary mostly because this system is robust enough to essentially never become unstable. For a less robust system more thought could go into these limits. The limits can also serve another purpose, where it would be easy to implement some kind of alarm that triggers if the control parameters are saturated for too long. This could show that the component might be damaged. Of course more intelligent predictive maintenance algorithms can be used in conjunction with the parameter estimates.

Improvements and further work

There are plenty of potential improvements and further work that can be done. The modeling could have been improved by taking the room temperature into consideration. There are two ways of going about this which would both require the addition of a new temperature sensor for the room temperature. One way to do this would be to simply remake the models with an additional input being the room temperature. This might not be so easy as adding another input into the models increases the complexity. The second way would instead be to use a "lookup table" of linearized models for different room temperatures. These models would essentially be the same as the models that were developed in this thesis but with different gains to compensate for smaller or larger heat losses. The issue with both of these improvements is that they require testing using different room temperatures which is cumbersome. Either the machine would have to be put in some kind of heating chamber, or data would have to be gathered from different parts of the world where there are different temperatures.

Another thing that might need to be looked at is the implementation of the parameter estimators. Currently they are run constantly which might cause a problem. If there is not enough excitation on the system the RLS algorithm will try to estimate the system based on constant signals, which may result in undesirable behaviour. A better way could be to try to estimate the covariance of the signals and only run the RLS when there is sufficient excitation. Another way would be to every once in a while run the RLS along with some added virtual noise to the heaters to get excitation into the system. What approach to use depends on the system. In this case there will likely never be enough excitation once it has entered production and converged to the more or less stationary values. So in this case some added noise to the control output might be necessary.

The use of the lambda-method in the adaptive tuning can also be improved. Currently the method is using $\lambda = T_i$, however by choosing another value of λ the performance can be improved. A better value for λ was not developed as the main focus of this thesis was more so on investigating the viability of the framework, than it was to optimize it. Finding other tuning methods to expand the toolbox of the framework is also another way to potentially improve the result, but again this was not done as the provided methods were enough for this thesis.

A way of reducing the amount of work needed to implement this system would be to add additional temperature sensors into the system. If measurements for T_1 and T_2 were available, the temperature estimator could have been removed completely from the implementation, leaving only the parameter estimator and the control tuner. This would also yield more reliable measurements, but of course it would be at a higher expense.

The implementation should have no problem dealing with a PID-controller instead of a PI-controller as long as there is a PID tuning method available, which both the AMIGO-method and the lambda-method has variations of.

8

Conclusion

This thesis investigated and provided a framework for how a simple automatic PI-control tuning can be done, using an air sterilization module as an example. Using modeling, parameter estimation and tuning methods such as the lambda-method and the AMIGO-method, the PI-control can adapt to time varying system changes. While a final conclusion on the viability of the framework cannot be made the results are promising.

There are many improvements that can be done, different tuning methods to use, and practical improvements depending on the process. More details of suggested improvements can be seen in the discussion chapter.

Bibliography

Åström, K. J. and T. Hägglund (2006). *Advanced PID control*. ISA - The Instrumentation, Systems and Automation Society.

Hägglund, T. (2017). *AUTOMATIC CONTROL - Lecture notes*. Department of automatic control, Lund University.

Johansson, R. (2019). *SYSTEM MODELING AND IDENTIFICATION*. Prentice Hall.

Tetra Pak (2020). URL: `https://www.tetrapak.com/about/tetra-pak-in-brief`. (visited on 04.06.2020).

Tetra Pak (2020). URL: `https://www.tetrapak.com/packaging/tetra-pak-a1-for-tfa`. (visited on 16.06.2020).

Tetra Pak (2020). URL: `https://www.tetrapak.com/packaging/aseptic-solutions`. (visited on 04.06.2020).

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER’S THESIS

Date of issue
July 2020

Document Number
TFRT-6112

Author(s)
Henrik Stålbom

Supervisor
Mattias Darmell, Tetra Pak, Sweden
Tore Hägglund, Dept. of Automatic Control, Lund University, Sweden
Carlotta Johnsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle

Adaptive PI-control for an air sterilization module

Abstract

The goal of this master thesis has been to implement adaptive PI-control of two air flow heaters in an air sterilization module. This is done by modeling the module and using the information from the model to estimate the dynamics of the heaters.

The idea of this thesis is to implement adaptive control with as simple control structures and tuning algorithms as possible, to keep it as generalized and accessible as possible. This way it will be easier to apply the same framework to other components. The implementation uses a range of models to express the thermodynamic behaviour of two heaters interconnected by a heat exchanger and some piping with heat losses. Using the models, a recursive least squares algorithm estimates the parameters of each heater, and finally the heaters are tuned using the lambda-method and the AMIGO-method. The models are created both from physical models and from non-physical models based purely on measurement data using the best available order.

The implementation was done using MATLAB Simulink from which the code was then ported over to BECKHOFF TwinCat and integrated into the machine code. Due to time constraints the implementation only got to the simulation stage in Twin-Cat, but it showed promise and worked as expected. Simulations were also done in Simulink using measurement data which also showed promise for a functional framework.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-51

Recipient’s notes

Security classification

http://www.control.lth.se/publications/