



Structural Optimization and Self-tuning of Marine Boilers



Master Thesis, 2009
Intelligent Autonomous Systems

Mads Kronborg Agesen and Hans Jørgen Uggerhøj

Aalborg University
Automation and Control
Institute of Electronic Systems



Titel:

Structural optimization and self-tuning
of marine boiler control

Tema:

Autonomous and Reliable Systems

Specialeretning:

Intelligente Autonome Systemer

Projektperiode:

P3 & P4 Kandidat Uddannelsen 2008/
2009

Projektgruppe:

1032

Deltagere:

Hans Jørgen Uggerhøj
Mads Kronborg Agesen

Vejledere:

Tom S. Pedersen
Palle Andersen

Oplagstal: 6

Sideantal: 143

Heraf appendiks: 15

Bilagsantal og -art: 1 stk. DVD

Afsluttet den 03-06-2009

Synopsis:

Dette projekt omhandler udvikling af et proof-of-concept self-tuning system til et Mission OSTM kedelsystem. Ydermere er der udviklet en metode, der sikrer ren forbrænding, selv ved saturering af aktuatorerne i brændstofssystemet. Mission OSTM kedelsystemet er modelleret som tre separate ulineære modeller; en model af Mission OSTM kedlen, fødevandssystemet og brændstofssystemet. I forbindelse med design af SISO regulatorer, er de komplekse ulineære modeller simplificeret til integratormodeller. De komplekse ulineære modeller og de simple integratormodeller er verificeret mod måledata fra kedelsystemet. Regulatorerne er designet med henblik på anvendelse i det udviklede self-tuning system, som er udviklet til at identificere modelparametre under opstart af kedelsystemet. De identificerede parametre omfatter karakteristisk af servomotorer, fødevandsventil, damptrykbygning samt et kurvebånd, der beskriver olie/luft forhold, anvendt til iltregulering. På baggrund af disse parametre udregnes regulatorparametre, inden kedelsystemet sættes i normal drift. Regulering af brændstofssystemet til Mission OSTM kedelsystemet inkluderer pulsbreddemoduleret (PWM) positionsregulering af olieventil og luftspjæld. Opgaven at følge et kurvebånd ved aktuator saturering, er beskrevet som et minimeringsproblem. Minimeringsproblemet er løst ved brug af et modelprædiktivt referencekorrektions-system, der sikrer konstant iltniveau ved overholdelse af kurvebåndet. Endeligt er en Kalman estimator designet til at estimere middelvandstand i kedlen og det umålbare dampforbrug, med henblik på at vurdere mulig forbedring af vandstandsreguleringen. De udviklede metoder er verificeret gennem simulering og viser tilfredsstillende resultater.

Title:

Structural optimization and self-tuning
of marine boiler control

Theme:

Autonomous and Reliable Systems

Specialization

Intelligent Autonomous Systems

Project period:

P3 & P4 Master Program 2008/2009

Project group:

1032

Participants:

Hans Jørgen Uggerhøj
Mads Kronborg Agesen

Supervisors:

Tom S. Pedersen
Palle Andersen

Copies: 6

Number of pages: 143

Appendices hereof: 15

Attachment: 1 DVD

Finished 03-06-2009

Abstract:

This project concerns development of a proof of concept self-tuning system for a Mission OS™ boiler system. Furthermore, a method to ensure clean combustion, in case of fuel actuator saturation, has been developed.

The Mission OS™ boiler system is modelled as three separate non-linear models; a model of the Mission OS™ boiler, the feed water actuator system and the fuel actuator system. For design of SISO controllers, the complex non-linear models are simplified to integrator models. Both the complex non-linear models and the simple integrator models are verified using measurement data from the boiler system. Controllers are designed to comply with the developed self-tuning scheme, capable of identifying model parameters during the start-up phase of the boiler system. The identified parameters includes fuel actuator servo characteristics, feed water valve characteristics, a curveband, describing the oil/air ratio, used as oxygen level control and pressure build-up characteristics. Based on these parameters, controller parameters are calculated prior to operating the boiler.

Control of the fuel actuator system for the Mission OS™ boiler system includes Pulse Width Modulation (PWM) position control of oil valve and air damper. To comply with the curveband, in case of PWM saturation, a constrained minimisation problem has been derived and solved by a model predictive reference correction scheme, ensuring steady oxygen level by compliance of the curveband.

Finally, to assess potential improvements of the water level control, a Kalman estimator is designed to estimate the mean water level and the unmeasurable steam consumption.

The developed methods are verified by simulation and shows satisfactory performance.

Preface

This report is written as a documentary proof of a long-term 9th and 10th master project made by project group 1032 at Department of Electronic Systems, Section of Automation and Control at Aalborg University in the period from the 1st of September 2008 through the 3rd of June 2009.

The theme of the project is "Autonomous and Reliable Systems" and has its origin in a project proposal specified by Aalborg Industries (AI), and Associate Professor Tom S. Pedersen and Associate Professor Palle Andersen.

The target audience of the report is mainly the project supervisor, censor, fellow students and others with interest in design of self-tuning controllers, estimation and modelling of boiler systems. It is furthermore expected that the reader of this report has background knowledge corresponding to the syllabus for a M.Sc.EE student at Department of Electronic Systems at Aalborg University.

When referring to figures, tables and equations in the report, the reference carries numbers. The standards ISO 31 and ISO 1000 are applied to number typography together with notation of scientific units. The used units are all SI-units, unless other units are stated.

The bibliography is placed last in the main report. References to literature is made to maintain the Harvard method. For instance "page 23" in "Feedback Control of Dynamic Systems" is referred to using [Franklin et al., 2006, p. 23]. In the bibliography the authors full name, the title of the literature, ISBN/URL, year of publication and note, if any, are shown. Literature that is available online, relevant Matlab files and written S-function C-code, are enclosed on a DVD-ROM.

Appendices are located after the main report and are named A, B, C etc. Furthermore each appendix has its own pagination, e.g. pages in the first appendix are numbered as A1, A2 etc.

In the following pages the nomenclature of this report is outlined to ease the reading.

Hans Jørgen Uggerhøj

hansu@es.aau.dk

Mads Kronborg Agesen

madage@es.aau.dk

Nomenclature

Roman

Symbol	Unit	Description
<i>a</i>		Polynomial coefficient
<i>b</i>		Polynomial coefficient
<i>c</i>	$\frac{\text{J}}{\text{kg}\cdot\text{K}}$	Specific heat capacity
<i>e</i>		Error
<i>g</i>	$\frac{\text{m}}{\text{s}^2}$	Gravity
<i>h</i>	$\frac{\text{J}}{\text{kg}}$	Specific enthalpy
<i>m</i>	kg	Mass
<i>n</i>		Number of mole
<i>p</i>	Pa	Pressure
<i>q</i>	$\frac{\text{J}}{\text{s}}$	Energy flow
<i>r</i>	m	Radius
<i>t</i>	s	Time
<i>w</i>	$\frac{\text{kg}}{\text{s}}$	Mass flow
<i>x</i>		Mole fraction
<i>y</i>		Mass fraction
<i>z</i>	$\frac{\text{mole}}{\text{s}}$	Mole flow
<i>A</i>	m^2	Area
<i>E</i>	J	Internal energy
<i>H</i>	$\frac{\text{J}}{\text{kg}}$	Calorific value
<i>L</i>	m	Level
<i>Q</i>	J	Thermal energy transfer
<i>R</i>	$\frac{\text{Pa}\cdot\text{m}^3}{\text{mole}\cdot\text{K}}$	Boltzmann constant
<i>T</i>	$^{\circ}\text{C}$	Temperature
<i>V</i>	m^3	Volume

Greek

Symbol	Unit	Description
θ	%	Valve/Damper position
ϑ	$\frac{\text{W}}{\text{m}^2 \cdot \text{K}}$	Heat transfer coefficient
ρ	$\frac{\text{kg}}{\text{m}^3}$	Mass density
τ	s	Time constant
η		Efficiency
γ_{p_s}		Steam pressure integrator gain

Subscripts

Symbol	Description
a	Air
b	Boiler
f	Furnace
fl	Flue gas
fu	Fuel
fw	Feed water
i	Input
m	Metal
o	Output
s	Steam
sat	Saturated
sb	Steam bubbles
w	Water
C	Carbon
H	Hydrogen
N	Nitrogen
O_2	Oxygen
0 (zero)	Ambient

Miscellaneous

Symbol	Usage	Description
\checkmark	$\checkmark w_{fw}$	<i>Measured</i> feed water flow
\hat{w}	\hat{w}_{fw}	<i>Estimated</i> feed water flow
\bar{p}	\bar{p}_s	<i>Operating point</i> for steam pressure
Δ	Δp_s	<i>Small signal gain</i> for steam pressure
\dot{w}	\dot{w}_{fw}	The <i>time derivative</i> of the feed water flow
$\tilde{\theta}$	$\tilde{\theta}_{fw}$	<i>Normed value</i> of the feed water valve position
\rightarrow	$q_{w \rightarrow sb}$	Energy flow <i>from</i> water <i>to</i> steam bubbles

Contents

1	Introduction	1
2	System Description	3
2.1	The Mission OS TM Boiler	3
2.2	Feed Water Actuator System	5
2.3	Fuel Actuator System	6
2.4	Sensors on the Mission OS TM Boiler System	7
2.5	The Morpheus Control Platform	10
2.6	Start-up Sequence for the Boiler System	11
2.7	Problem Description	13
3	Modelling of Boiler System	15
3.1	Boiler Modelling	16
3.1.1	Boiler Decomposition	16
3.1.2	Preliminary System Assumptions	17
3.1.3	Furnace and Flue Gas Pipes	20
3.1.4	Water and Steam	25
3.1.5	Non-linear Model of the Boiler	28
3.1.6	Parameter Estimation and Verification of the Non-linear Boiler Model	29
3.2	Feed Water Actuator Modelling	34
3.2.1	Valve Positioner	34
3.2.2	Piping System	34
3.2.3	Flow Sensor	36

3.2.4	Parameter Estimation and Verification of the Feed Water Actuator Model	37
3.3	Fuel Actuator Modelling	38
3.3.1	Servo Motor	39
3.3.2	Piping System	39
3.3.3	Flow Sensors	39
3.3.4	Parameter Estimation and Verification of the Fuel Actuator Model	40
3.4	Modelling Resume	42
4	Controller Design	43
4.1	Inner-loop Controller Design	44
4.1.1	Feed Water Actuator Control	44
4.1.2	Oil Valve Position Control	47
4.1.3	Air Damper Position Control	49
4.1.4	PWM Saturation	49
4.1.5	Parameter Estimation and Verification of Inner-loop Controllers	52
4.2	Outer-loop Controller Design	54
4.2.1	Steam Pressure Control	54
4.2.2	Water Level Control	57
4.2.3	Oxygen Level Control	61
4.2.4	Simulation of Outer-loop Controllers	64
4.3	Controller Resume	69
5	Self-tuning	71
5.1	Oil Valve/Air Damper Characteristics Identification	73
5.2	Feed Water Actuator Identification	75
5.3	Oil/Air Curveband Identification	79
5.4	Steam Pressure Integrator Gain Identification	83
5.5	Self-tuning Evaluation	85
5.6	Limitations of the Developed Self-tuning Scheme	88

6	Constrained Control of Fuel Valve and Air Damper	89
6.1	Fuel/Air Ratio Mismatch	89
6.2	Model Predictive Control Problem Formulation	91
6.3	Model Predictive Control Scheme for Ratio-Constrained PWM Servo Positioning	93
7	Kalman Estimation	99
7.1	Linear Kalman Estimation	99
7.1.1	Linear Kalman Estimator Algorithm	101
7.1.2	Simulation of Linear Kalman Estimator	102
7.1.3	Evaluation of the Linear Kalman Estimator	102
8	Conclusion	105
9	Discussion	109
	Bibliography	111
A	Boiler Dimensions	A1
B	Constants	B1
C	Parameter Estimation using Senstools	C1
D	Control Model Validation	D1
E	Linearisation using Taylor Series Expansion	E1
F	Derivation of Linear State Space Model	F1

Introduction

Aalborg Industries A/S is the world's leading supplier of marine boilers. They have a wide range of different types of boilers used for different purposes in both the industrial and marine world. The boilers can i.a. be steam boilers, heat exchangers or hot-water boilers and they are typical oil fired. The boilers are used by many different consumers and for different purposes. A steam boiler is used to produce steam for various services as generating electricity and heating. Common for all consumers is to acquire a safe and reliable solution, which is the objective for Aalborg Industries.

For this specific project the focus is on an oil fired steam boiler called a Mission OSTM boiler, primary used on minor vessels. When a new boiler is installed on a vessel, there is a need to use time and manpower to properly adjust the control of the boiler, in order to obtain satisfactory performance. This is required because the piping system, the type of valves and other mechanics varies from installation to installation.

The main objective of this project is to develop a proof of concept self-tuning system, capable of analysing the newly installed boiler system and from this analysis is able to automatically adjust the control of the boiler to perform satisfactory. Thereby Aalborg Industries can save time and money when installing a new boiler. Another advantage of this self-tuning system might be, that after many working hours, the boiler system slightly wears down. This could change the characteristics of the valves and pumps, which affects the control of the boiler system. If the self-tuning system in run regular, the boiler system always performs satisfactory.

Hence, the self-tuning system is developed to perform analytical tasks during the start-up sequence of the boiler system, in order to determine model parameters of a simplified model, allowing for simple automatic controller tuning. The overall principle of the developed self-tuning scheme is inherited from previous projects, carried out in cooperation between Aalborg University and Aalborg Industries and further developed through this project.

An objective for Aalborg Industries is to maintain a clean combustion to minimize the energy consumption and thereby save money and minimize CO₂ emissions. Thus, a secondary objective of this project is to develop a solution to cope with the challenge of ensuring a clean combustion.

Lastly, if the water level control of the Mission OSTM boiler is improved, a boiler of smaller dimensions can perform equally to the existing boilers. Hence, the final objective for the project is to investigate the benefits of introducing classic estimation techniques, in order to optimise the control of the water level of the Mission OSTM boiler.

The following Chapter features a description of the Mission OSTM boiler system in order to outline a specific problem description for this project.

System Description

This chapter contains a description of the Mission OSTM boiler system used for the project. The Mission OSTM boiler system consists of the Mission OSTM boiler, a fuel actuator system and a feed water actuator system. Subsequently the sensors on the Mission OSTM boiler system are listed and the Morpheus control platform is described. Finally the start-up procedure for the Mission OSTM boiler system is explained before a problem description for this project is stated.

2.1 The Mission OSTM Boiler

The Mission OSTM boiler is a side fired boiler capable of producing up to 3 tons steam per hour. Basically, the Mission OSTM boiler consists of two separate parts; the water/steam part and the furnace/flue gas pipes part, as illustrated in Figure 2.1. The figure shows the most important elements of the boiler, to understand the functionality. In appendix A a more detailed drawing of the Mission OSTM boiler is given, along with some of the dimensions. The dimensions listed in the appendix are used later, when modelling the boiler. The purpose of the boiler is to produce steam, which is the output from the boiler and the inputs to the boiler are the feed water, fuel and air.

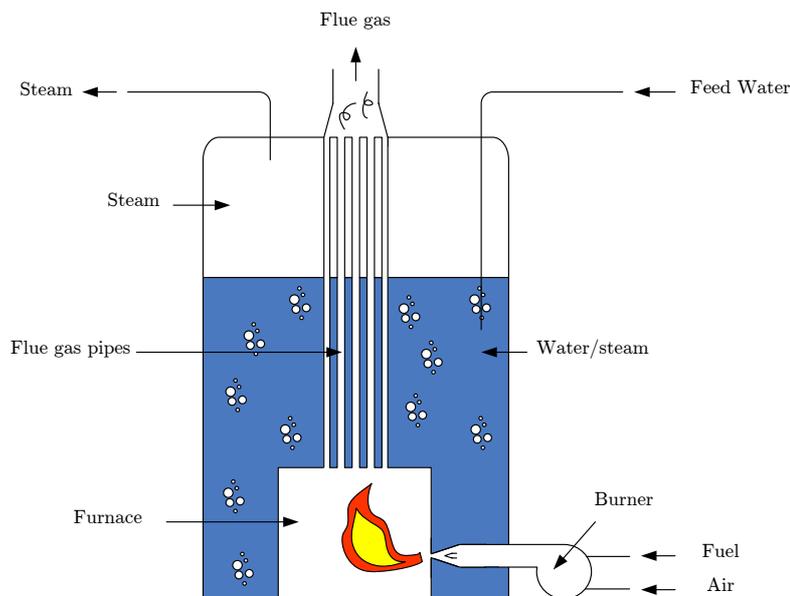


Figure 2.1: Illustration of the Mission OSTM boiler. Inspired by [Andersen and Jørgensen, 2007].

The objective of the boiler system is to maintain a steam pressure of 8 bar and a steady water

level. The upper and lower limits for the water level are defined as a deviation of 21 cm from the normal water level, as illustrated in Figure A.1 in Appendix A. If the water level exceeds the upper limit, feed water will flow into the steam pipes, which must be avoided. If the water level exceeds the lower limit, the furnace/flue gas pipes part cannot dispose the heat to the water/steam part, which eventually causes the furnace and flue gas pipes to deform. Thus, if the steam pressure or water level deviates to much, the consequences can be fatal. Consequently there are a number of safety valves on the boiler, e.g. to let out steam if the steam pressure is too high.

To produce steam, a mixture of air and fuel is injected into the furnace through the burner. By burning the mixture of air and fuel, heat is transferred from the furnace/flue gas pipes part to the water/steam part of the Mission OSTM boiler via metal surrounding the furnace and flue gas pipes. Transferring heat to the water leads to evaporation of the water and thereby steam production to the consumer.

When steam is wanted, water is required. The feed water is led into the Mission OSTM boiler simply by opening a feed water valve. When changing the feed water inflow or steam outflow leads to a phenomenon called shrink and swell, which challenges the design of a water level controller. Thus, the phenomenon is described in the following.

Shrink and Swell

The swell phenomenon is e.g. seen when the steam outflow is abruptly increased. Intuitively the water level should drop. However, the pressure in the water/steam part decreases due to the increased steam outflow. Thereby causing the boiling temperature of the water to decrease and consequently a short-term increase of the water level arises, due to expansion of the steam bubbles in the water. The long-term water level will however decrease due to the increase of steam outflow. Abruptly decreasing the steam outflow leads to the reverse, namely shrink; a short-term decrease of water level followed by a long-term increase.

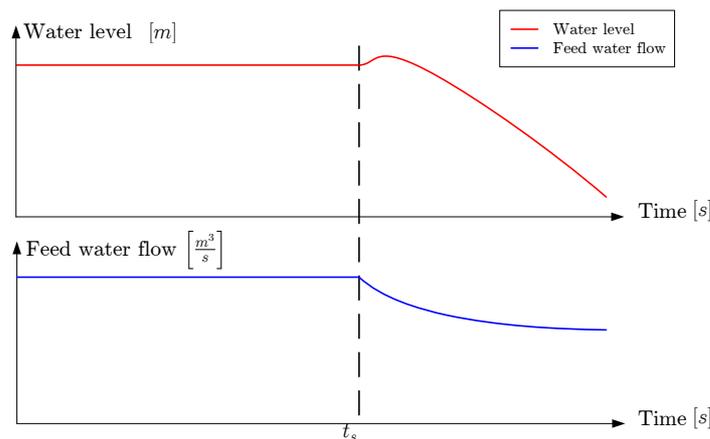


Figure 2.2: Illustration of swell phenomenon caused by abruptly decrease of feed water flow. Red curve shows the water level and the blue shows the feed water flow into the boiler.

Abruptly increasing the feed water inflow also leads to the shrink phenomenon. The feed water is cold compared to the water in the boiler and thereby causing the water temperature to decrease.

Consequently the steam bubbles will shrink and the water level decrease. However, the long-term water level will of course increase. Swell caused by decrease in feed water inflow is illustrated in Figure 2.2. The decrease in feed water inflow is applied at time $t = t_s$.

The boilers produced by AI, are generally resistant to the abruptly changes in the feed water flow, but for abruptly changes in the steam outlet flow the shrink and swell phenomenon exists.

In order to operate the boiler, feed water, fuel and air is a necessity. The feed water system is described in the following section.

2.2 Feed Water Actuator System

The feed water actuator system supplies feed water to the boiler, through some pipes and valves. Figure 2.3 illustrates a simplified feed water actuator system. The water pump delivers a constant feed water flow, $w_{fw,f}$, at high pressure around 15 bar. The feed water pressure, $p_{fw,i}$, must be greater than the pressure inside the boiler, p_s , in order to obtain a feed water flow into the boiler.

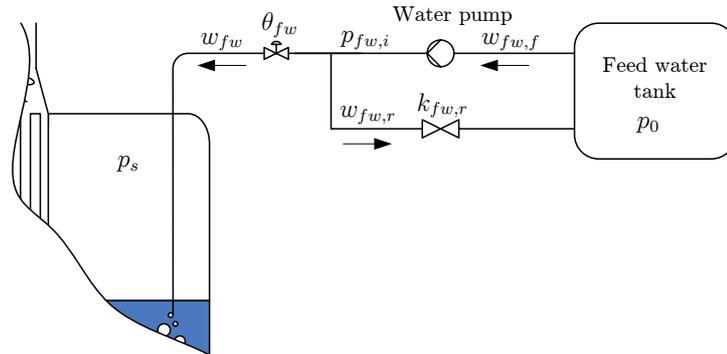


Figure 2.3: Illustration of Mission OSTM feed water actuator system.

There are two important valves in the feed water actuator system; θ_{fw} and $k_{fw,r}$. The valve θ_{fw} is the feed water valve, used to control the feed water flow, w_{fw} , into the boiler. This is the controllable actuator in the feed water actuator system.

The feed water not fed into the boiler is returned to the water tank, through a return valve $k_{fw,r}$. The return valve is during operation adjusted to a fixed valve stroke, but by changing the valve stroke, the range of the feed water flow, w_{fw} , is changed due to a change in the pressure, $p_{fw,i}$.

In addition to the feed water actuator system, providing water to the boiler, a fuel actuator system is required to supply the fuel and air to the boiler. The fuel actuator system is described in the following.

2.3 Fuel Actuator System

The fuel pump in the fuel actuator system delivers the fuel from a fuel tank, through a piping system and some valves into a fuel atomiser inside the burner. The fuel atomiser injects the fuel through a nozzle into the furnace. Furthermore an air fan is blowing an amount of air around the fuel atomiser into the furnace. The amount of air is changed by an air damper and should be adjusted to the fuel flow, in order to obtain an optimal combustion in the furnace.

A simplified diagram of the piping system, valves and burner is shown in Figure 2.4.

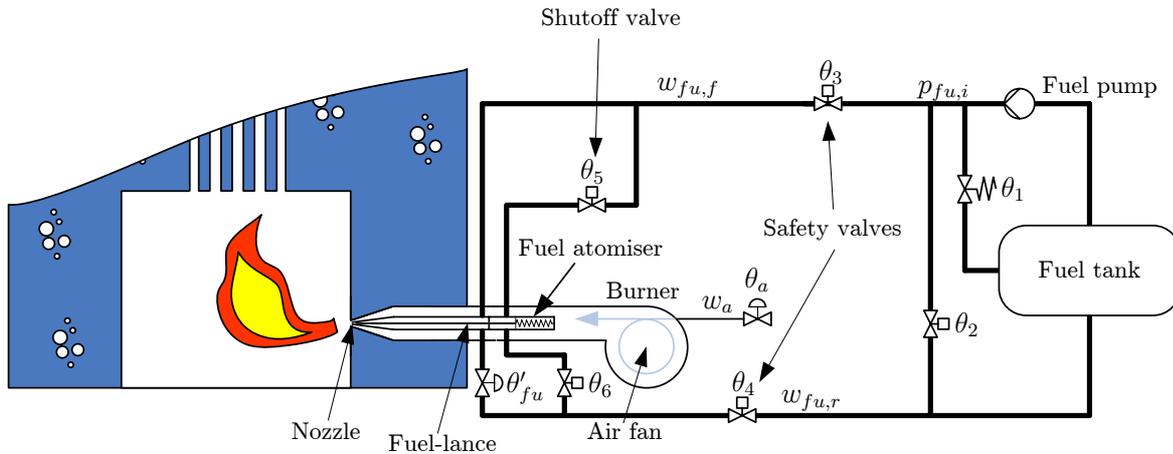


Figure 2.4: Illustration of Mission OSTM fuel actuator system.

Fuel piping system and valves

The bold lines in the figure represent the piping system in which the fuel flows. Besides the piping system there are 7 valves in total, used for different purposes in the fuel piping system. The valves θ_2 , θ_3 , θ_4 , θ_5 and θ_6 are magnet valves, implying they are on/off valves. The valve, θ'_{fu} , is an adjustable valve controlled by a servo motor and the valve, θ_1 , is a spring valve, also adjustable. However, the spring valve, θ_1 , is not of interest, as it is not a part of the burner system.

The fuel pump delivers a fuel pressure of approximately 28 bar, but in order to maintain a constant fuel pressure, $p_{fu,i}$, a spring valve θ_1 is mounted. If the fuel flow changes in the piping system, the spring valve stroke is automatically adjusted to obtain a stable fuel pressure, $p_{fu,i}$.

The valves θ_3 and θ_4 are safety valves, implying that when these are closed, no fuel is led into the burner. If the safety valves are closed, the fuel flows through the return valve θ_2 , to maintain circulation of the fuel.

Valve θ_5 is a shutoff valve and enables the fuel flow into the furnace through the fuel atomiser in the burner, and thereby combustion in the furnace. For simplicity this is further illustrated in Figure 2.5, where the orange colour illustrates the pipes where the fuel flows. When the shutoff valve is closed, the fuel flows through the fuel atomiser but not into the furnace, as the fuel-lance blocks the nozzle. When the shutoff valve is open, the valve θ_6 is closed and thereby causing the pressure to increase inside the back of the fuel atomiser, forcing the fuel-lance on the spring

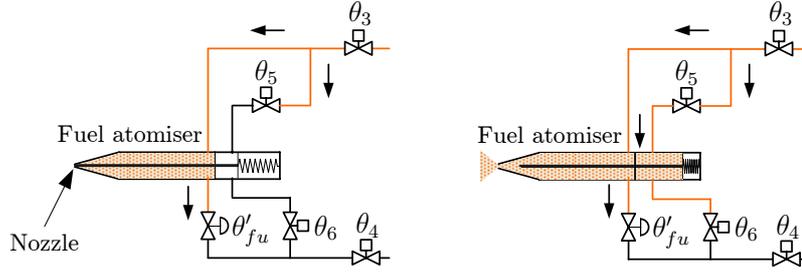


Figure 2.5: Illustration of how the fuel is injected into the furnace, by changing the valve stroke of the shutoff valve, θ_5 .

backwards and the fuel flows through the nozzle into the furnace. The valve stroke of valve θ_5 and θ_6 is always opposite.

Looking at Figure 2.4, the fuel flow, w_{fu} , into the furnace is controlled by the fuel valve θ'_{fu} . When the fuel valve opens, the fuel is led back to the fuel tank and reduces the fuel flow into the furnace. Hence for convenience, the fuel valve position, θ_{fu} is defined as:

$$\theta_{fu} = 1 - \theta'_{fu}, \quad (2.1)$$

and will be used from now on when referring to the fuel valve position. The fuel valve is the actuator, seen from a control point of view, to adjust the combustion and thereby the steam pressure.

Air damper

Figure 2.4 also shows an air damper, θ_a , to adjust the air flow, w_a , into the burner. When the air damper opens, the air flow into the furnace increases. To obtain a sensible combustion, the oil and air flow must be adjusted to obtain a constant oxygen level in the flue gas of approximately 5 % [Aalborg Industries, 2008].

The fuel and feed water actuator systems facilitates control of the steam production and control of the water level in the boiler, but to obtain feedback control, various sensors are needed. The sensors mounted on the Mission OSTM boiler are described in the following section.

2.4 Sensors on the Mission OSTM Boiler System

The Mission OSTM boiler at AI's test center is used for test purposes only, and is for that reason equipped with several and different sensors, i.e. not all sensors are used for control of the boiler. The commercial available Mission OSTM boilers sold by AI is however not equipped by all these sensors, but only the sensors needed to maintain the performance level of the specific boiler. Thereby AI aims to use as few default inexpensive sensors as possible. The following contains a description of used sensors in the boiler system.

There are various types of sensors on the boiler system, i.a. flow transducers (FT) and temperature transducers (TT). Figure 2.6 illustrates all the sensors mounted on the Mission OSTM boiler system in AI's test center. Furthermore a list of the mounted sensors is provided in Table 2.1 to clarify the measured quantities. Table 2.1 also reveals which sensors there are mounted

Water/steam part				
\check{w}_{fw}	Flow Transducer	Measures the feed water flow into the boiler		T
\check{w}_s	Flow Transducer	Measures the steam flow out of the boiler		T
\check{p}_s	Pressure Transducer	Measures the steam pressure inside the boiler		T C
\check{L}_w	Level Transducer	Measures the water level inside the boiler		T C
Furnace/flue gas pipes part				
$\check{w}_{fu,f}$	Flow Transducer	Measures the fuel flow into the burner		T(C)
$\check{w}_{fu,r}$	Flow Transducer	Measures the fuel flow out from the burner		T(C)
$\check{\theta}_a$	Attenuator Transducer	Measures the air valve position		T C
$\check{\theta}_{fu}$	Attenuator Transducer	Measures the fuel valve position		T C
\check{p}_{fu}	Pressure Transducer	Measures the fuel pressure in the fuel actuator system		T
\check{T}_{fu}	Temperature Transducer	Measures the fuel temperature in the fuel tank		T C
\check{x}_{o,O_2}	Oxygen Transducer	Measures the oxygen level in the flue gas		T(C)
\check{T}_{fl}	Temperature Transducer	Measures the flue gas temperature		T C

Table 2.1: Table showing the sensors mounted on the Mission OSTM boiler. The right column indicates if the sensor is mounted on the boiler in the AI test center "T" and/or on the commercial boilers "C". Optionally mounting is marked by parenthesis.

Water Level Sensor Measurements

When the water reaches the boiling temperature and steam bubbles are produced, it leads to fluctuating water level measurements due to the steam bubbles breaking the water surface.

Furthermore there is a larger production of steam bubbles near the flue gas pipes, which gives an uneven water surface and forms small waves. Altogether this results in noisy steady state measurements of the fluctuating water level, which challenges the control of the water level.

Figure 2.7 illustrates the water tending towards the flue gas pipes.

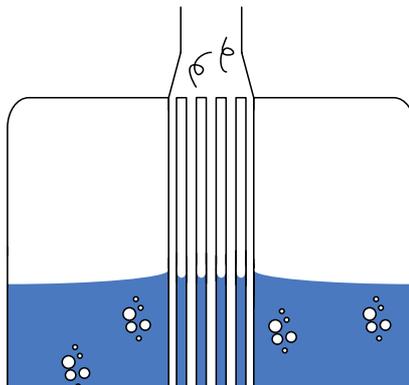


Figure 2.7: Illustration of the fluctuating water level near the flue gas pipes.

[Solberg, 2008] showed that by use of model predictive control, it was possible to improve the control of the water level, but the shrink and swell phenomenon is unavoidable.

In addition to the actual Mission OSTM boiler system, a control platform at AI's test center has been developed, to ease the development and test of new control systems for the boilers. The Morpheus platform is described in the following.

2.5 The Morpheus Control Platform

To perform control of a boiler as the Mission OSTM a control platform, Morpheus, is developed by AI in association with CISS Aalborg University, IO Technologies A/S and with contributions from [Andersen and Jørgensen, 2006]. The Morpheus control platform is shortly presented in this section. The structure of the Morpheus control platform is shown in Figure 2.8.

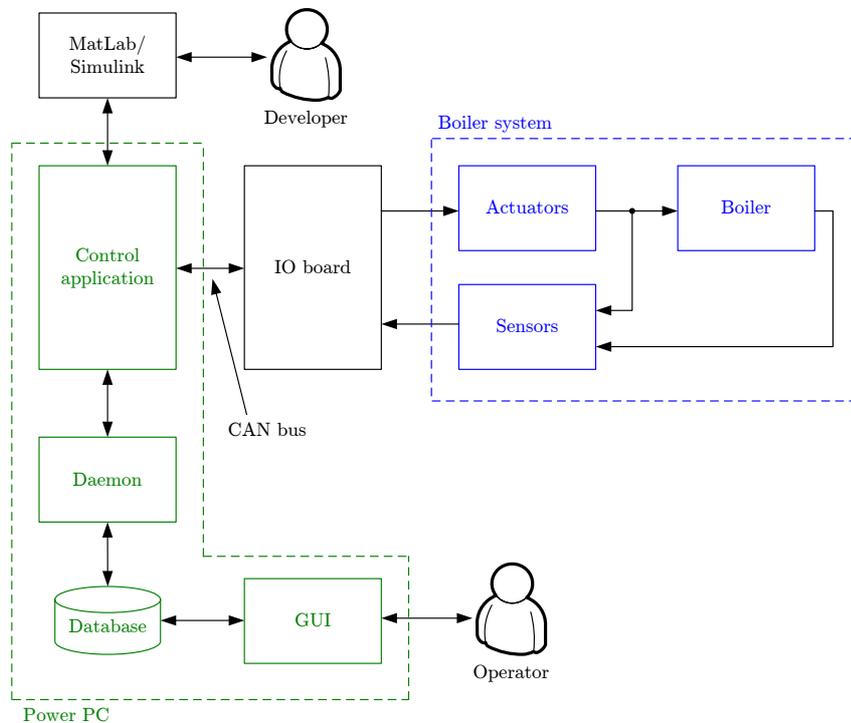


Figure 2.8: Structural overview of the Morpheus control platform. Inspired by [Andersen and Jørgensen, 2006].

As seen, the platform mainly consists of three elements; a Power PC (PPC), an IO board and a boiler system.

The software on the PPC is mainly divided into three parts; a control application, a database and a graphical user interface, GUI, used as interface for the daily operator of the boiler system. All communication between control application and the database passes through a daemon taking care of database write locks etc.

The PPC is based on a linux kernel with the necessary packages to run the target application. The target application is compiled to a binary file using MATLABTM Real-Time Workshop (RTW), and primary features the control application.

The PPC interacts with the boiler system through an IO board, developed by Prevas, using

a CAN bus. An advantage gained using the Morpheus control platform, is the possibility to connect different types of boiler systems; but only able to control one at a time.

The developer interfaces the boiler system through a MATLABTM/Simulink application from a PC. Using Simulink as interface for the developer makes it easy to implement controllers developed using MATLABTM/Simulink. Furthermore use of Simulink RTW allows the developer to monitor the controller and set parameters in run time.

For the project of developing controllers, the parts of main interest is reduced to the control application, the IO board and the boiler system of the Morpheus platform. The database will be used when testing the boiler system, in order to obtain measurement data.

Apart from giving an introduction of the Mission OSTM boiler system and the Morpheus control platform setup, it is relevant to understand the start-up sequence of the boiler system.

2.6 Start-up Sequence for the Boiler System

The start-up sequence of the boiler system is implemented as a state machine, by using a Simulink Toolbox called StateFlow. In StateFlow it is possible to define the different states, transitions between states, entry actions etc. Figure 2.9 illustrates the different states in the start-up sequence for the Mission OSTM boiler system. Each state is associated to a status label describing the progress of the start-up sequence. When the boiler system is powered up, it always starts in the safety interlock state.

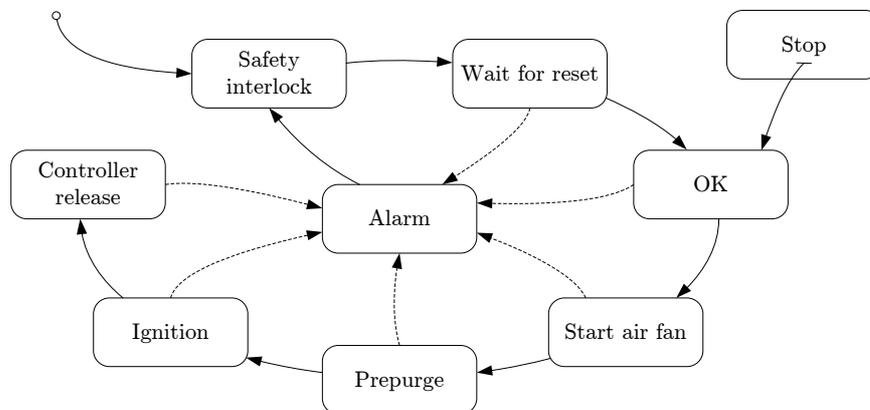


Figure 2.9: Flow chart of the start-up sequence for the Mission OSTM boiler system at AI's test center.

The states seen in Figure 2.9 are described below. The numbers written in parenthesis represents the status label of the specific states.

Safety interlock (1) - The system checks whether any of the alarms are activated. If there is an active alarm this must be taken care of. If no alarms are activated, a transition is performed to the next state; wait for reset.

Wait for reset (2) - In this state the system waits for a reset from the operator, before entering normal operation, the OK state.

OK (3) - In this state the system waits for the operator to push the start button. When the operator push start, a transition is performed to start air fan.

Start air fan (4) - Here the valves in the fuel actuator system are set to prepurge mode, the air fan is turned on and the air damper is adjusted to 100 % open. When the air fan is at 98 % open, a transition is performed to prepurge.

Prepurge (5) - In prepurge mode the burner and the piping near the burner are cleaned by the air flow from the air fan. If there for some reason should be fuel remains in the burner or pipes before igniting, an impending danger of blowing up the burner is present. After prepurging for 30 seconds the next state is entered.

Ignition (6) - First the oil and air valves are set to ignition position. Then an ignition-transformer is turned on to create the sparks for ignition and the shutoff valve is opened to inject fuel into the furnace. After ignition the flame stabilises for 10 seconds before the system enters the final state.

Controller release (7) - In this state the oil and air flow is held constant until the steam pressure reaches 1 bar. From 1 to 7 bar the steam controller takes over the control of oil and air flow by increasing the steam pressure reference slowly toward 7 bar. When 7 bar is reached, the controller is fully released, and the boiler system stays in this state unless the operator stops the boiler system.

Stop - At any time the operator can stop the boiler system and the system goes into the stop state, where the shutoff valve closes to put out the flame. Then a postpurge takes place before the system performs a transition to the OK state; waiting for the operator to start up the boiler system again.

Alarm - In all of the states it is appropriate to have a quality or stability check, in form of an alarm system. If the boiler system enters a deadlock or livelock in some of the states, typically a timer is used to remedy the lock and the system goes into the alarm state. From here the system performs a transition to the safety interlock state and the alarm must be dealt with.

After giving a description of the Mission OSTM boiler system, the Morpheus control platform and an explanation of the start-up sequence for the boiler system, the problem description for this project is outlined.

2.7 Problem Description

As mentioned in the introduction in Chapter 1, the main objective of this project is to develop a self-tuning system, in order to reduce the costs when installing new boilers. Secondly, the possibilities for developing a method to ensure clean combustion together with the benefits of applying classic estimation techniques, in order to improve the control of the water level, are to be investigated. In all this leads to a specific problem description, consisting of four objectives.

1. *Is it possible to simplify a complex system model in order to design a model-based self-tuning system, capable of determining model and controller parameters during the start-up sequence of the Mission OSTM boiler system?*
2. *Is it possible to ensure a clean combustion by mixing fuel and air according to a specific mixture proportion, by adjusting the oil valve and the air damper using a curveband?*
3. *How does the actuator characteristics influence the ability to follow a specified curveband, and is it possible to develop controllers ensuring that the curveband is obeyed at any time?*
4. *Is it possible to gain improved water level control by applying classic estimation techniques?*

Limitation

Aalborg Industries produces various types of boilers for various usage. However, to lower the complexity of a developed self-tuning system and to ensure compatibility between developed methods and the specific boiler system, a set of limitations are made for this specific project.

- The boiler system structure must be similar of that described in Chapter 2
- The boiler type must be similar to the Mission OSTM, entailing the behaviour of the boiler can be described by the same model structure
- The surface area of the water inside the boiler must be known beforehand, e.g. specified using the inner dimensions of boiler and the flue gas pipes.
- The oil valve, air damper and the feed water valve must be controlable
- The position of oil valve and the air damper and also the feed water flow must be measurable
- Tests are carried out with diesel as fuel type¹
- The boiler system utilises SISO control

With the above stated problems and limitations, initially a model of the boiler system must be derived, in order to design a model based controller and a self-tuning system to find the controller parameters.

¹The system is designed for use of both diesel oil and heavy fuel.

Modelling of Boiler System

This chapter describes the entire modelling of the Mission OSTM boiler system consisting of the boiler, the feed water actuator system and fuel actuator system. The models derived in this chapter forms the basis for controller design, estimation as well as non-linear simulation of the boiler system.

The content of this chapter covers initial assumptions to simplify the modelling of the boiler. These assumptions together with an overall decomposition of the boiler, results in separation of the boiler into a "furnace/flue gas pipes part" and a "water/steam part". The "furnace/flue gas pipes part" is modelled as two separate models; a dynamic oxygen model based on the chemical reaction of the combustion and a static efficiency model for the energy transferred from the furnace to the water/steam part. The water/steam part model is derived, based on energy and mass balances for three control volumes; water, steam bubbles and steam. The oxygen model, the efficiency function and the model of the water/steam part together constitutes the non-linear boiler model, which is represented as a state space model on descriptor form. Prior to verification of the non-linear boiler model, some model parameters are estimated. Subsequently the feed water actuator and the fuel actuator systems are modelled, having linear dynamics and non-linear flow characteristics. As for the non-linear boiler model, the required model parameters for the actuator systems are estimated and the feed water and fuel actuator models are verified. Finally the chapter is ended by a short resume of the modelling of the boiler system.

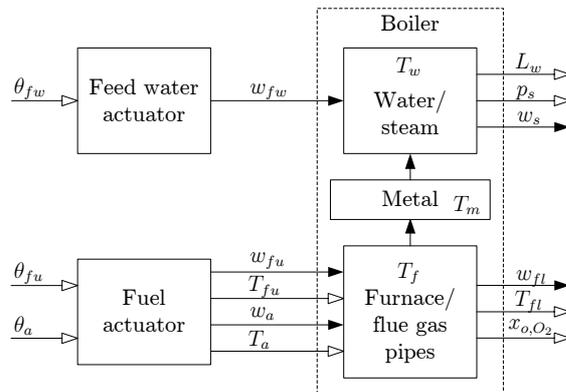


Figure 3.1: Block diagram of the boiler system.

Figure 3.1 illustrates a block diagram of the boiler system, which consists of the boiler, the feed water actuator system and the fuel actuator system, as described in Chapter 2. In Figure 3.1, the fuel and feed water actuators are simplified to not include sensor dynamics and control input signals. The block diagram also shows the inputs and outputs on the different blocks. The black arrows indicate actual flows while the white arrows indicate measurable quantities. In the following sections a model is derived for the boiler, the feed water actuator and the fuel actuator, which together constitutes the model of the boiler system.

3.1 Boiler Modelling

In this section a non-linear model of the boiler is derived. To ease the modelling, the boiler is decomposed into control volumes and based on previous projects and knowledge, assumptions are made to further simplify the modelling. This results in a reduced number of control volumes, on the basis of which a number of equations are derived to describe the dynamics of the boiler.

Figure 3.2 illustrates the inputs and outputs of the boiler. The main purpose of the boiler model is to describe the water level, L_w , and steam pressure, p_s , in terms of the feed water flow, w_{fw} , the fuel flow, w_{fu} , and the air flow, w_a . Furthermore, it is desirable to express the oxygen level, x_{o,O_2} , in the flue gas, from the fuel flow and air flow into the furnace of the boiler.

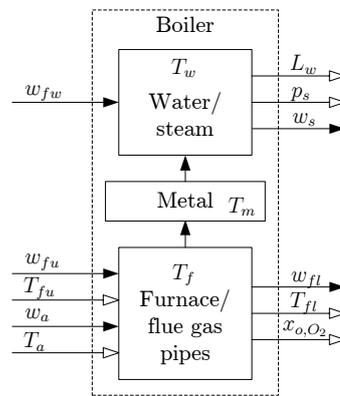


Figure 3.2: Block diagram of the boiler with the furnace/flue gas pipes part, the water/steam part, the metal between the two parts and the different inputs and outputs.

3.1.1 Boiler Decomposition

Prior to deriving the boiler model, the boiler is decomposed into control volumes to describe the mass and energy flow through the boiler and thus to simplify the modelling of the boiler. In Figure 3.3 a block diagram illustrates the control volumes of the boiler; furnace, flue gas, metal, water, steam bubbles and steam. According to [Andersen and Jørgensen, 2007] this decomposition has proved to be useful.

Furnace

The furnace converts the energy from combustion of fuel and air, to heat transferred to the furnace jacket of metal and to the flue gas.

Flue gas

The energy in the flue gas is partly transferred to the flue gas pipes of metal. The rest of the energy is lost to the surroundings, primary as air with higher temperature than the ambient temperature. In the output from the flue gas pipes, the oxygen level, x_{o,O_2} , is measured, and used as feedback to obtain a more effective combustion in the furnace.

Metal

The energy in the metal, from the furnace and flue gas, is emitted to the water/steam part.

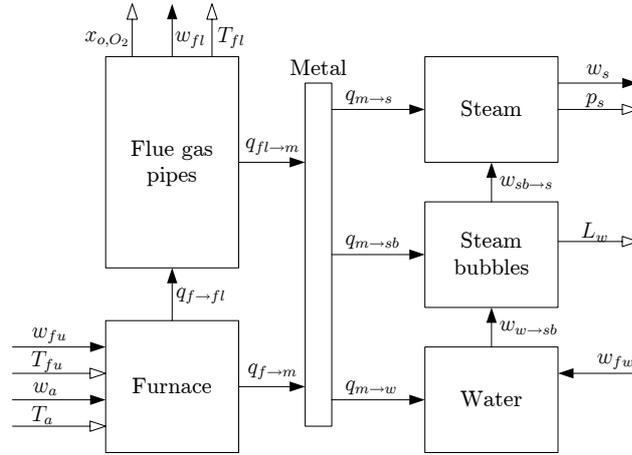


Figure 3.3: Illustrates how the boiler is decomposed into control volumes.

Water

The energy from the feed water and the metal to the water, is transferred to the steam bubbles, when the temperature reaches the boiling point.

Steam bubbles

The steam bubbles is a significant control volume, as this is the part where the shrink and swell phenomenon exists. Furthermore this volume, together with the volume of water, defines the water level in the boiler. The steam bubbles is the transition volume between water and steam.

Steam

The last of the five control volumes contains the output of the boiler system, i.e. the produced steam.

3.1.2 Preliminary System Assumptions

After decomposing the boiler into control volumes, some assumptions are made to simplify the modelling, without compromising the dominating dynamics of the system.

- The volume of the boiler is constant, $\dot{V}_{tot} = 0$, and the boiler is filled with water, steam bubbles and steam; $V_{tot} = V_w + V_{sb} + V_s$. Furthermore this assumption implies stringent divisioning between the control volumes, thereby neglecting potential water in the steam part of the boiler and steam outlet.
- Due to the presence of both water and steam in the boiler, the water and steam is considered saturated with uniform temperature and pressure, with respect to place and not time. The temperature of water, T_w , and steam, T_s , is dependent of the steam pressure, p_s , which depends on time:

$$T_{sat}(p_s(t)) = T_w(p_s(t)) = T_s(p_s(t)) . \quad (3.1)$$

Hence the steam pressure, p_s , is chosen as a state variable to reduce the complexity of a system output equation.

The assumption of uniform pressure for water and steam can be verified by calculating the pressure difference in the boiler. In the top of the boiler the pressure is 8 bar in steady state. The pressure in the bottom of the boiler is calculated by adding the pressure of the water amount as:

$$p_{bottom} = p_{top} + \rho_w L_w g \cdot 1 \cdot 10^{-5} = 8 + 997 \cdot 4 \cdot 9.82 \cdot 1 \cdot 10^{-5} = 8.39 \text{ [bar]} , \quad (3.2)$$

where L_w [m] is the height of the boiler, g [$\frac{m}{s^2}$] is the gravity and the factor $1 \cdot 10^{-5}$ [$\frac{\text{bar}}{\text{Pa}}$] is the conversion from [Pa] to [bar].

This results in a difference of approximately 2.5 % which is smaller in practice, as the water level will never reach 4 m.

- The energy transfer between furnace and metal consists of both convective and radiative energy transfer. Modelling convective and radiative energy transfer, requires a higher order transfer function [Andersen and Pedersen, 2007]. However, the temperature of the metal, T_m , is assumed uniform and equal to the water temperature, T_w . This assumption seems reasonable as the convective heat transfer coefficient from metal to water is very large [Andersen and Jørgensen, 2007]. Hence the metal control volume in Figure 3.3 can be neglected.
- All energy transferred from the combustion to the water/steam part is assumed to be transferred to the water. Thereby neglecting the presence of super heated steam. This assumption seems reasonable, as the amount of super heated steam is relatively small, when present, compared to the total amount of steam.
- The energy loss of the boiler is only through the flue gas. To prove this, an estimate of the energy loss through the metal jacket of the boiler is calculated:

$$q_{loss} = A_b \vartheta_b \Delta T , \quad (3.3)$$

where A_b [m^2] is the surface area of the boiler, ϑ_b [$\frac{W}{m^2 \cdot ^\circ C}$] is the heat transfer coefficient and ΔT [$^\circ C$] is the temperature difference between the inside and outside of the boiler. The surface area of the boiler is estimated as:

$$A_b = 2\pi r_b L_b + 2\pi r_b^2 = 2\pi(0.75 \cdot 4 + 0.75^2) \approx 22 \text{ [m}^2\text{]} , \quad (3.4)$$

where r_b is the radius of the boiler and L_b is the height of the boiler.

Due to good lagging of the boiler, the heat transfer coefficient is approximated to $\vartheta_b = 0.1$. The temperature difference ΔT is found as the saturation temperature of water at 8 bar and subtracting the ambient temperature:

$$\Delta T = 170.41 - 20 \approx 150 \text{ [}^\circ C\text{]} , \quad (3.5)$$

which gives an estimated energy loss of:

$$q_{loss} \approx 22 \cdot 0.1 \cdot 150 \approx 330 \text{ [W]} . \quad (3.6)$$

The main part of the energy inside the boiler originates from combustion of fuel and is expressed by:

$$q_{fu} = w_{fu} H_{fu} , \quad (3.7)$$

where w_{fu} [$\frac{kg}{s}$] is the fuel flow and H_{fu} [$\frac{J}{kg}$] is the calorific value.

The fuel flow is approximately 0.033 and the calorific value of the fuel is $40 \cdot 10^6$, see Appendix B. This gives an energy of:

$$q_{fu} \approx 0.033 \cdot 40 \cdot 10^6 \approx 1.32 \cdot 10^6 [W] . \quad (3.8)$$

The energy loss in percent is by use of Equation (3.6) and (3.8) calculated to 0.025 %.

- The temperature of the inlet air, T_a , is constant and approximately equal to the ambient temperature, T_0 . Also the feed water temperature, T_{fw} , and the fuel temperature, T_{fu} , are considered constant.
- The steam bubbles are assumed to have physical properties similar to steam, i.e. same mass density, $\rho_{sb} = \rho_s$ and enthalpy $h_{sb} = h_s$.

Based on previous work by [Solberg and Hvistendahl, 2004] and [Andersen and Jørgensen, 2007], the dynamics of the furnace and flue gas pipes can be neglected due to fast time response compared to the closed loop boiler dynamics. They derived balance equations for the furnace and obtained a time constant of 0.25 s. This is considerable faster than the time constant for the water level and steam pressure control loops, which was found to be in the interval between 50 s and 150 s. Thus the furnace and flue gas pipes control volumes can be modelled as one control volume, and the energy flow from the furnace/flue gas pipes to the water/steam part, $q_{f \rightarrow w}$, can be modelled statically in form of an efficiency function.

This reduces the previous decomposition of the boiler to the control volumes shown in Figure 3.4.

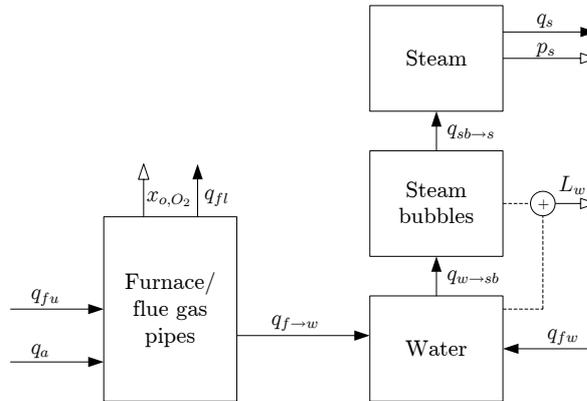


Figure 3.4: Illustrates the control volumes and the energy flows in the final decomposition of the boiler.

As illustrated in Figure 3.4 the water level, L_w , is defined as the height of both the water and steam bubbles volumes, as previous described.

Some of the presented assumptions might not seem relevant yet. However, this will be clear during the modelling. The models describing the dynamics of the boiler are in the next two subsections derived, using the control volumes defined in Figure 3.4. In Subsection 3.1.3 the furnace/flue gas pipes part is modelled and in Subsection 3.1.4 the water/steam part is modelled.

The overall aim is, as mentioned, to derive one model expressing the steam pressure, p_s , the water level, L_w , and the oxygen level, x_{o,O_2} , from the inputs to the boiler, w_{fw} , w_a and w_{fu} . The final non-linear boiler model is presented in Subsection 3.1.5.

3.1.3 Furnace and Flue Gas Pipes

For the control volume "furnace and flue gas pipes" two models are derived. One describing the dynamics of the combustion, in form of an oxygen model, and subsequently a static model describing the energy transfer from the combustion to the water/steam part.

Oxygen Model

The oxygen model describes the oxygen level in the flue gas on the basis of the input mass flows of fuel, w_{fu} , and air, w_a , assuming immediate combustion. Figure 3.5 illustrates a block diagram of the oxygen model, which consists of two parts. The combustion describes the chemical reaction and thereby the excess of oxygen, while the oxygen dynamics describes the change of oxygen content over time. The output from the combustion, x_{i,O_2} , denotes the mole fraction of oxygen.

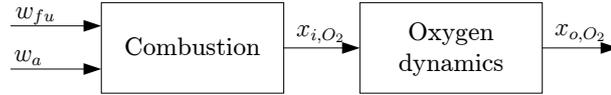


Figure 3.5: Block diagram of the oxygen model structure.

Combustion

The applicable fuel is diesel, which is assumed to have the chemical formula $C_{15}H_{32}$, as the rest of the substances in diesel are poorly represented and thus neglected. It is assumed that the combustion is complete, implying that all carbon, C, is transmuted into carbon dioxide, CO_2 , and all hydrogen, H, is transmuted into di-hydrogen oxide, H_2O , during the combustion process.

The input mole flows, $z_{(\cdot)}$, from the diesel and air are described by:

$$z_{fu,C} = \frac{w_{fu}y_C}{M_C} \quad (3.9)$$

$$z_{fu,H} = \frac{w_{fu}y_H}{M_H} \quad (3.10)$$

$$z_{a,O_2} = \frac{w_a y_{O_2}}{M_{O_2}} \quad (3.11)$$

$$z_{a,N} = \frac{w_a y_N}{M_N}, \quad (3.12)$$

where y_C , y_H , y_{O_2} and y_N denotes the mass fractions for carbon, hydrogen, oxygen and nitrogen respectively, and are given by:

$$y_C = x_C \frac{M_C}{M_{fu}}, \quad y_H = x_H \frac{M_H}{M_{fu}} \quad (3.13)$$

$$y_{O_2} = x_{O_2} \frac{M_{O_2}}{M_a}, \quad y_N = x_N \frac{M_N}{M_a}. \quad (3.14)$$

M_C and M_H are the molar masses and \bar{M}_{fu} is the average molar mass of diesel, given by the sum:

$$\bar{M}_{fu} = x_C M_C + x_H M_H , \quad (3.15)$$

where the mole fractions are given by the composition of the chemical formula for diesel, C15H32:

$$x_C = \frac{15}{15 + 32} , \quad x_H = \frac{32}{15 + 32} . \quad (3.16)$$

Similar M_{O_2} and M_N are molar masses and \bar{M}_a is the average molar mass of air, given by the sum:

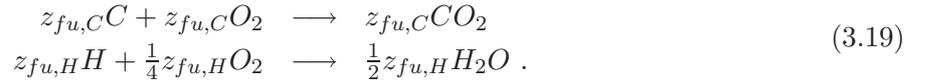
$$\bar{M}_a = x_{O_2} M_{O_2} + x_N M_N , \quad (3.17)$$

where the mole fractions are given by:

$$x_{O_2} = \frac{21}{21 + 79} , \quad x_N = \frac{79}{21 + 79} , \quad (3.18)$$

assuming that the air consists of 21 % oxygen and 79 % nitrogen.

After defining the input mole flows, reaction schemes for the combustion are set up, to define the amount of oxygen needed to obtain a complete combustion:



From Equation (3.19) the required amount of oxygen for a complete combustion is defined as the mole flow:

$$z_{fu,C} + \frac{1}{4} z_{fu,H} . \quad (3.20)$$

An expression for the oxygen mole flow leaving the combustion, z_{i,O_2} , is found, by subtracting the amount of oxygen consumed in the combustion, from the input mole flow, z_{a,O_2} :

$$z_{i,O_2} = z_{a,O_2} - \left(z_{fu,C} + \frac{1}{4} z_{fu,H} \right) . \quad (3.21)$$

The input to the oxygen dynamics is according to Figure 3.5 the mole fraction, x_{i,O_2} , which is given by the oxygen mole flow, z_{i,O_2} , relative to the total mole flow, z_i , into the oxygen dynamics:

$$x_{i,O_2} = \frac{z_{i,O_2}}{z_i} = \frac{z_{a,O_2} - \left(z_{fu,C} + \frac{1}{4} z_{fu,H} \right)}{z_{i,O_2} + z_{a,N} + z_{fu,C} + \frac{1}{2} z_{fu,H}} . \quad (3.22)$$

In stead of expressing the mole fraction, x_{i,O_2} , by mole flows, Equations (3.9)–(3.14) are inserted into Equation (3.22) to express the mole fraction by the mass flows w_{fu} and w_a . After rearranging, the mole fraction is derived as:

$$\begin{aligned} x_{i,O_2} &= \frac{\frac{x_{O_2}}{M_a} w_a - \frac{x_C + \frac{1}{4} x_H}{M_{fu}} w_{fu}}{\frac{x_{O_2}}{M_a} w_a - \frac{x_C + \frac{1}{4} x_H}{M_{fu}} w_{fu} + \frac{x_N}{M_a} w_a + \frac{x_C + \frac{1}{2} x_H}{M_{fu}} w_{fu}} \\ &= \frac{\frac{x_{O_2}}{M_a} w_a - \frac{x_C + \frac{1}{4} x_H}{M_{fu}} w_{fu}}{\frac{x_{O_2} + x_N}{M_a} w_a + \frac{\frac{1}{4} x_H}{M_{fu}} w_{fu}} . \end{aligned} \quad (3.23)$$

Thereby Equation (3.23) is the model derived for the combustion in the furnace. In the following, the oxygen dynamics is expressed from the oxygen mole fraction.

Oxygen dynamics

The change of the oxygen level can be expressed using a mole balance. The general mole balance equation is given as:

$$\dot{n} = z_i - z_o , \quad (3.24)$$

where n is the number of moles and z_i and z_o are the input and output mole flows respectively.

Using Equation (3.24), a mole balance for the oxygen is expressed using a backward difference place discretisation, assuming complete and uniform mixing:

$$\frac{d(n \cdot x_{o,O_2})}{dt} = z_i \cdot x_{i,O_2} - z_o \cdot x_{o,O_2} , \quad (3.25)$$

where x_{i,O_2} and x_{o,O_2} are the oxygen fractions of the input and output mole flows respectively. Differentiating the left hand side gives:

$$n\dot{x}_{o,O_2} + x_{o,O_2}\dot{n} = z_i \cdot x_{i,O_2} - z_o \cdot x_{o,O_2} . \quad (3.26)$$

Inserting Equation (3.24) and rearranging, reduces the expression to a first order system:

$$\dot{x}_{o,O_2} = \frac{1}{\tau_{x_{o,O_2}}} (x_{i,O_2} - x_{o,O_2}) , \quad (3.27)$$

where the time constant is given by: $\tau_{x_{o,O_2}} = \frac{n}{z_i}$. This indicates the dynamics of the oxygen model is varying in proportion to the input mole flow. To calculate an estimate of the time constant, the number of mole, n , can be found using the equation for ideal gas: $pV = nRT$, assuming that the pressure and volume in the furnace remains constant

$$n = \frac{pV}{RT} = \frac{1 \cdot 10^5 \cdot \pi \cdot 0.6^2 \cdot 2.1}{8.314 \cdot 443.56} \approx 64.4 \text{ [mol]} , \quad (3.28)$$

where p is the pressure in steady state, V is the volume of the furnace (calculated from the dimensions in Appendix A), R is the Boltzmann constant and T is the absolute temperature. Also the input mole flow can be estimated, using the fuel flow in steady state and a constant ratio between fuel flow and air flow, $\kappa_{fu} = \frac{w_a}{w_{fu}}$:

$$z_i = \bar{M}_{fu}w_{fu} + \bar{M}_aw_a = 4.51 \cdot 0.033 + 24.5 \cdot 0.033 \cdot 16.9 = 13.8 \text{ [mol/s]} , \quad (3.29)$$

where κ_{fu} is calculated using Equation (3.44). This gives an estimated time constant in steady state of:

$$\tau_{x_{o,O_2}} = \frac{64.4}{13.8} \approx 4.7 \text{ [s]} . \quad (3.30)$$

The oxygen sensor on the Mission OSTM boiler is a Lambda probe, for which the response time is given in the datasheet to maximum 2 s. To simplify the oxygen model to a first order system, the sensor dynamics are neglected as they are faster than the oxygen dynamics. This entails that the oxygen model consists only of the combustion model and the oxygen dynamics.

By inserting Equation (3.23) into Equation (3.27) and rearranging, the oxygen model can be expressed as:

$$\underbrace{1}_{j_{44}} \dot{x}_{o,O_2} = \frac{1}{\tau_{x_o,O_2}} \underbrace{\left(\frac{\frac{x_{O_2} w_a}{M_a w_{fu}} - \frac{x_C + \frac{1}{4} x_H}{M_{fu}}}{\frac{x_{O_2} + x_N}{M_a} w_a + \frac{\frac{1}{4} x_H}{M_{fu}} w_{fu}} \right)}_{l_{44}} \underbrace{w_{fu}}_{l_{45}} - \frac{1}{\tau_{x_o,O_2}} x_{o,O_2} . \quad (3.31)$$

Having derived the oxygen model, the efficiency of the furnace is examined.

Efficiency Function for Furnace and Flue Gas Pipes

The second model of the furnace/flue gas pipes control volume describes the amount of energy, from the combustion of fuel and air, transferred to the water/steam part and thereby the efficiency of the boiler. Looking at Figure 3.4 on page 19, an equation of the steady state energy balance for the furnace/flue gas pipes volume can be derived as:

$$q_{f \rightarrow w} = q_{fu} + q_a - q_{fl} . \quad (3.32)$$

The efficiency, denoted η_{fu} , can be described as the energy transferred to the water, relative to the energy from the fuel and air as:

$$\eta_{fu} = \frac{q_{f \rightarrow w}}{q_{fu} + q_a} . \quad (3.33)$$

Inserting Equation (3.32) implies:

$$\eta_{fu} = \frac{q_{fu} + q_a - q_{fl}}{q_{fu} + q_a} = 1 - \frac{q_{fl}}{q_{fu} + q_a} , \quad (3.34)$$

where the energy flows are given by:

$$q_{fl} = (w_{fu} + w_a) c_{fl} (T_{fl} - T_0) \quad (3.35)$$

$$q_{fu} = w_{fu} (H_{fu} + c_{fu} (T_{fu} - T_0)) \quad (3.36)$$

$$q_a = w_a c_a (T_a - T_0) , \quad (3.37)$$

where T_0 is the reference temperature for the enthalpy.

The energy flows given by Equations (3.35)-(3.37) can be simplified by using some of the assumptions described in Subsection 3.1.2.

- The ambient temperature, T_0 , should not affect the efficiency function and can be neglected by inserting the energy flows in Equation (3.32):

$$\begin{aligned} q_{f \rightarrow w} &= w_{fu} (H_{fu} + c_{fu} (T_{fu} - T_0)) + w_a c_a (T_a - T_0) - (w_{fu} + w_a) c_{fl} (T_{fl} - T_0) \\ &= w_{fu} H_{fu} + w_{fu} c_{fu} T_{fu} + w_a c_a T_a - (w_{fu} + w_a) c_{fl} T_{fl} \\ &\quad - (w_{fu} c_{fu} + w_a c_a - (w_{fu} + w_a) c_{fl}) T_0 . \end{aligned} \quad (3.38)$$

The last term dependent on T_0 is neglected, assuming that the total heat capacity of fuel and air is equal to heat capacity of the flue gas.

- The specific heat capacity for the flue gas is approximately equal to the specific heat capacity for air $c_{fl} \approx c_a$.
- The fuel temperature, T_{fu} , is approximately constant, entailing that the fuel energy flow can be approximated as a constant multiplied with the fuel flow:

$$q_{fu} = \beta_{fu} w_{fu} , \quad (3.39)$$

where $\beta_{fu} = H_{fu} + c_{fu} T_{fu}$.

- Since the temperature of the air is approximately equal to the ambient temperature, the energy from the air flow is much smaller compared to the energy from the fuel, $q_a \ll q_{fu}$, and thus neglectable, $q_a = 0$.

When neglecting the energy from the air, the air flow, w_a , is approximated as a constant factor of the fuel flow, w_{fu} :

$$w_a = \kappa_{fu} w_{fu} . \quad (3.40)$$

Inserting the reduced energy flows into Equation (3.34) gives a simplified expression for the efficiency:

$$\eta_{fu} = 1 - \frac{(1 + \kappa_{fu})c_a T_{fl}}{\beta_{fu}} . \quad (3.41)$$

To express $q_{f \rightarrow w}$ relative to the energy input flow, which now is proportional to w_{fu} , Equation (3.39) and (3.41) is inserted in Equation (3.33) and rearranging gives:

$$\begin{aligned} q_{f \rightarrow w} &= \eta_{fu} q_{fu} \\ &= \eta_{fu} \beta_{fu} w_{fu} \\ &= [H_{fu} + c_{fu} T_{fu} - (1 + \kappa_{fu})c_a T_{fl}] w_{fu} . \end{aligned} \quad (3.42)$$

In this expression κ_{fu} and T_{fl} are the only unknown. κ_{fu} is the relation between the fuel flow and air flow and can thus be calculated from a desired oxygen content in the flue gas, using the oxygen model derived previously in this subsection.

The flue gas temperature, T_{fl} , can be approximated by a first order linear function of the fuel flow, $T_{fl} = a_{fl} w_{fu} + b_{fl}$ [Andersen and Jørgensen, 2007, p. 54]. This gives the final efficiency function:

$$q_{f \rightarrow w} = [H_{fu} + c_{fu} T_{fu} - (1 + \kappa_{fu})c_a (a_{fl} w_{fu} + b_{fl})] w_{fu} . \quad (3.43)$$

The relationship between the fuel and air flow, κ_{fu} , is calculated using Equation (3.23) and the constants in Appendix B. Given a desired oxygen level, x_{o,O_2} , of 5 % , κ_{fu} is calculated to:

$$\kappa_{fu}|_{5\%} = \frac{w_a}{w_{fu}} = 16.9. \quad (3.44)$$

To further simplify the efficiency function, the linear function describing the flue gas temperature is examined. [Andersen and Jørgensen, 2007] approximated the linear function of the fuel flow in an interval of [50; 200] [$\frac{kg}{h}$]. In this interval the flue gas temperature approximately changes

100 °C. By inserting the constants given in Appendix B into Equation (3.43) and changing the flue gas temperature from 300 to 400 °C, the efficiency factor, η_{fu} , changes in the interval:

$$\eta_{fu} = [0.820; 0.865] , \quad (3.45)$$

which is approximately a 5 % change of efficiency. By choosing the flue gas temperature, T_{ft} , as the mean temperature in the given interval, the efficiency function reduces to an efficiency factor:

$$q_{f \rightarrow w} = [H_{fu} + c_{fu}T_{fu} - (1 + \kappa_{fu})c_a350]w_{fu} . \quad (3.46)$$

Hereby the models of the furnace/flue gas pipes part are derived. The oxygen model is given by Equation (3.31) and the efficiency factor is given by Equation (3.46). Remaining is to derive a model for the water/steam part.

3.1.4 Water and Steam

The water and steam part consist of the last three control volumes; water, steam bubbles and steam. To describe the dynamic behavior of the water and steam part, a number of balance equations are derived and used in the final non-linear model of the boiler.

Energy Balance for the Water and Steam Part

The energy balance is used to express the change of energy in a control volume, and is given by [Andersen and Pedersen, 2007]:

$$\begin{aligned} \dot{E}_{tot} &= q_i - q_o + Q \\ &= w_i \cdot h_i - w_o \cdot h_o + Q , \end{aligned} \quad (3.47)$$

where E_{tot} is the total internal energy, q_i and q_o are the input and output energy flows respectively, h_i and h_o are the enthalpy for the input and output energy flows respectively and Q is the provided thermal energy.

An equation describing the energy balance for the boiler is derived by looking at Figure 3.6 as:

$$\dot{E}_{tot} = w_{fw}h_{fw} - w_s h_s(p_s) + q_{f \rightarrow w} , \quad (3.48)$$

where $q_{f \rightarrow w}$ is the effect transferred from the furnace/flue gas pipes and $h_{(\cdot)}$ is the enthalpy. The total energy, E_{tot} , in the control volume is contained in the water, steam bubbles, steam and metal:

$$E_{tot} = m_w h_w(p_s) + m_{sb} h_s(p_s) + m_s h_s(p_s) + m_m c_m T_{sat}(p_s) , \quad (3.49)$$

where $m_{(\cdot)}$ is the mass and $T_{sat} = T_w = T_m$ according to the boiler assumptions in Subsection

3.1.2. Differentiating E_{tot} , according to Equation (3.48), by use of the chain rule gives:

$$\begin{aligned}
 \dot{E}_{tot} &= \frac{d(m_w h_w(p_s))}{dt} + \frac{d(m_{sb} h_s(p_s))}{dt} + \frac{d(m_s h_s(p_s))}{dt} + \frac{d(m_m c_m T_{sat}(p_s))}{dt} \\
 &= \dot{m}_w h_w(p_s) + m_w \frac{\partial h_w}{\partial p_s} \dot{p}_s \\
 &+ \dot{m}_{sb} h_s(p_s) + m_{sb} \frac{\partial h_s}{\partial p_s} \dot{p}_s \\
 &+ \dot{m}_s h_s(p_s) + m_s \frac{\partial h_s}{\partial p_s} \dot{p}_s \\
 &+ m_m c_m \frac{\partial T_{sat}}{\partial p_s} \dot{p}_s .
 \end{aligned} \tag{3.50}$$

Inserting Equation (3.50) into Equation (3.48), using the assumptions $\dot{V}_{tot} = 0$ and $\rho_s h_s = \rho_{sb} h_{sb}$, and rearranging with respect to the state variables p_s and V_w , gives:

$$\begin{aligned}
 \underbrace{(\rho_w h_w - \rho_s h_s)}_{j_{12}} \dot{V}_w &+ \underbrace{\left(m_w \frac{\partial h_w}{\partial p_s} + m_{sb} \frac{\partial h_s}{\partial p_s} + m_s \frac{\partial h_s}{\partial p_s} + m_m c_m \frac{\partial T_{sat}}{\partial p_s} \right)}_{j_{11}} \dot{p}_s \\
 &= \underbrace{h_{fw}}_{l_{12}} w_{fw} - \underbrace{h_s p_s}_{l_{13}} w_s + \underbrace{\eta_{fu} \beta_{fu}}_{l_{14}} w_{fu} ,
 \end{aligned} \tag{3.51}$$

where $\eta_{fu} \beta_{fu}$ is the furnace efficiency factor given by Equation (3.46).

The unknown variables \dot{m}_w , \dot{m}_{sb} and \dot{m}_s in Equation (3.50) can be described by use of mass balance equations.

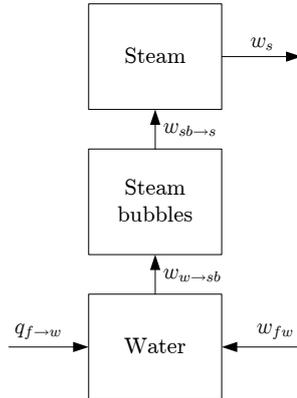


Figure 3.6: Block diagram of the boiler with terms to describe the energy and mass balances for the water and steam part.

Mass Balance for the Water and Steam Part

The mass balance is used to express the change of mass in a control volume, and is given by [Andersen and Pedersen, 2007]:

$$\begin{aligned}\dot{m} &= \frac{d(\rho \cdot V)}{dt} \\ &= \sum w_i - \sum w_o \\ &= w_i - w_o ,\end{aligned}\tag{3.52}$$

where ρ is the mass density, V is the control volume and $w_{(\cdot)}$ is the mass flow. A mass balance can be expressed for each of the control volumes by use of Figure 3.6, and by use of the chain rule, as the mass density $\rho(p_s)$ is dependent on the pressure.

Mass balance for the water volume:

$$\dot{m}_w = V_w \frac{\partial \rho_w}{\partial p_s} \dot{p}_s + \rho_w(p_s) \dot{V}_w = w_{fw} - w_{w \rightarrow sb} .\tag{3.53}$$

Mass balance for the steam bubbles volume:

$$\dot{m}_{sb} = V_{sb} \frac{\partial \rho_s}{\partial p_s} \dot{p}_s + \rho_s(p_s) \dot{V}_{sb} = w_{w \rightarrow sb} - w_{sb \rightarrow s} .\tag{3.54}$$

Mass balance for the steam volume:

$$\dot{m}_s = V_s \frac{\partial \rho_s}{\partial p_s} \dot{p}_s + \rho_s(p_s) \dot{V}_s = w_{sb \rightarrow s} - w_s .\tag{3.55}$$

At this point an energy balance equation and three mass balance equations are derived. However, the number of unknown variables must be reduced.

Reducing the number of unknown variables

To eliminate some of the unknown variables, the mass flow $w_{sb \rightarrow s}$ in Equation (3.54) and (3.55) can be approximated by using the rise time, t_{sb} , of air bubbles in fluids:

$$w_{sb \rightarrow s} \approx \frac{1}{t_{sb}} \dot{m}_{sb} = \frac{1}{t_{sb}} \rho_s V_{sb} .\tag{3.56}$$

Equation (3.56) is derived by [Andersen and Jørgensen, 2007] based on [Davies and Taylor, 1949].

Furthermore by combining Equation (3.53) and (3.54), the mass flow $w_{w \rightarrow sb}$ can be eliminated:

$$V_w \frac{\partial \rho_w}{\partial p_s} \dot{p}_s + \rho_w \dot{V}_w + V_{sb} \frac{\partial \rho_s}{\partial p_s} \dot{p}_s + \rho_s \dot{V}_{sb} = w_{fw} - \cancel{w_{w \rightarrow sb}} + \cancel{w_{w \rightarrow sb}} - \frac{1}{t_{sb}} \rho_s V_{sb} ,\tag{3.57}$$

which by rearranging reduces to:

$$\underbrace{\left(V_w \frac{\partial \rho_w}{\partial p_s} + V_{sb} \frac{\partial \rho_s}{\partial p_s} \right)}_{j_{21}} \dot{p}_s + \underbrace{\rho_w}_{j_{22}} \dot{V}_w + \underbrace{\rho_s}_{j_{23}} \dot{V}_{sb} = \underbrace{1}_{l_{22}} w_{fw} - \underbrace{\frac{1}{t_{sb}}}_{l_{21}} \rho_s V_{sb} .\tag{3.58}$$

According to the assumptions in Subsection 3.1.2, the volume of the boiler is constant, entailing:

$$V_{tot} = V_w + V_{sb} + V_s \Rightarrow V_s = V_{tot} - V_w - V_{sb} . \quad (3.59)$$

Combining Equation (3.59), (3.56) and mass balance Equation (3.55) and rearranging gives:

$$\underbrace{\left((V_{tot} - V_w - V_{sb}) \frac{\partial \rho_s}{\partial p_s} \right)}_{j_{31}} \underbrace{\dot{p}_s}_{j_{32}} \underbrace{-\rho_s}_{j_{33}} \underbrace{\dot{V}_w}_{j_{33}} \underbrace{-\rho_s}_{j_{33}} \underbrace{\dot{V}_{sb}}_{j_{33}} = \underbrace{\frac{1}{t_{sb}}}_{l_{31}} \underbrace{\rho_s}_{l_{31}} \underbrace{V_{sb}}_{l_{31}} \underbrace{-1}_{l_{33}} w_s . \quad (3.60)$$

Finally, all the necessary equations, for the different control volumes, to describe the non-linear boiler model are derived, and the non-linear model is presented in the following.

3.1.5 Non-linear Model of the Boiler

The non-linear boiler model is in the following presented as a state space model on descriptor form, given by the general expression:

$$\mathbf{J}\dot{\mathbf{x}} = \mathbf{L}\mathbf{b} .$$

Using Equation (3.31) for the oxygen model, Equation (3.51) for the energy balance equation, where the efficiency factor is included, and finally Equation (3.58) and (3.60) from the mass balance equation, the non-linear model is given by:

$$\begin{bmatrix} j_{11} & j_{12} & 0 & 0 \\ j_{21} & j_{22} & j_{23} & 0 \\ j_{31} & j_{32} & j_{33} & 0 \\ 0 & 0 & 0 & j_{44} \end{bmatrix} \cdot \begin{bmatrix} \dot{p}_s \\ \dot{V}_w \\ \dot{V}_{sb} \\ \dot{x}_{o,O_2} \end{bmatrix} = \begin{bmatrix} 0 & l_{12} & l_{13} & l_{14} & 0 \\ l_{21} & l_{22} & 0 & 0 & 0 \\ l_{31} & 0 & l_{33} & 0 & 0 \\ 0 & 0 & 0 & l_{44} & l_{45} \end{bmatrix} \cdot \begin{bmatrix} V_{sb} \\ w_{fw} \\ w_s \\ w_{fu} \\ x_{o,O_2} \end{bmatrix} . \quad (3.61)$$

The output equation is given by the general form:

$$\mathbf{y} = \mathbf{C}\mathbf{x} ,$$

where the output matrix is chosen, so that the water level, L_w , the steam pressure, p_s , and the oxygen level, x_{o,O_2} , are outputs:

$$\begin{bmatrix} p_s \\ L_w \\ x_{o,O_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{A_w} & \frac{1}{A_w} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \\ V_{sb} \\ x_{o,O_2} \end{bmatrix} . \quad (3.62)$$

By thoroughly examination of the expressions in Equation (3.61), the changes in steam pressure and water and steam bubble volumes are described by an integrator effect. Even though this might not seem obvious here, it is easy to see in Chapter 4, where control models are derived for controller design.

3.1.6 Parameter Estimation and Verification of the Non-linear Boiler Model

To verify the non-linear model of the boiler, the parameters κ_{fu} and t_{sb} , expressing the relation between fuel and air flow for a given oxygen level and the average rise time for steam bubbles respectively, must be estimated. These two parameters in the boiler model are the only ones, that are not determined by physical dimensions or possible to look up in thermodynamical tables.

In addition to κ_{fu} and t_{sb} , the time constant, τ_{x_o, O_2} , in the oxygen model could be estimated, as the time constant represents both the oxygen dynamics and the sensor dynamics. However, as the time constant in the oxygen dynamics is varying in proportion to the input mass flows, w_a and w_{fu} , this might not give an accurate estimate. Furthermore the air flow is unmeasurable on the Mission OSTM boiler and thus it is practically impossible to estimate the time constant of the oxygen model. Consequently the theoretical value of τ_{x_o, O_2} in steady state is used.

One method to estimate the parameters, κ_{fu} and t_{sb} , is by using Senstools by [Knudsen, 2004], which is a toolbox for MATLABTM. This estimation method is described in Appendix C.

The result of the estimation is shown in Table 3.1, where the theoretical and estimated values are listed. The estimated values are found by calculating the mean values from eight measurement series, with different input signals on steam flow, feed water flow and fuel flow.

Parameter	Theoretical value	Estimated value	Unit
κ_{fu}	16.9	24.4	
t_{sb}	5	0.8	s
τ_{x_o, O_2}	4.7		s

Table 3.1: The theoretical and estimated parameters for the non-linear boiler model.

The relation between fuel flow and air flow, κ_{fu} , is from [Aalborg Industries, 2008] given as approximately 20, which is close to both the theoretical and estimated value. The estimated value for the rise time for steam bubbles, t_{sb} , is somewhat smaller than the theoretical value. However, this seems reasonable as the theoretical value is for bubbles with larger diameter and under lower pressure and thus the steam bubbles in the boiler moves faster.

Verification of the Non-linear Boiler Model

To verify the boiler model, a new series of measurement data is collected and the non-linear boiler model is implemented in Simulink as an S-function `watersteam.c`. Some measurement data is available from previous projects, so not all measurement data are found necessary to be collected again. In the following, a comparison of the measurement data and the simulated output from the non-linear boiler model is outlined. The measurement data are step responses for inputs in steam flow, feed water flow and fuel flow.

Step in steam flow

The steam flow step response is obtained by maintaining a steady steam flow and adjusting the feed water flow and fuel flow to obtain steady state, with a water level of approximately 1.23 m

and a steam pressure of approximately 8 bar. The step response reveals the dynamics in the boiler and the result is shown in Figure 3.7.

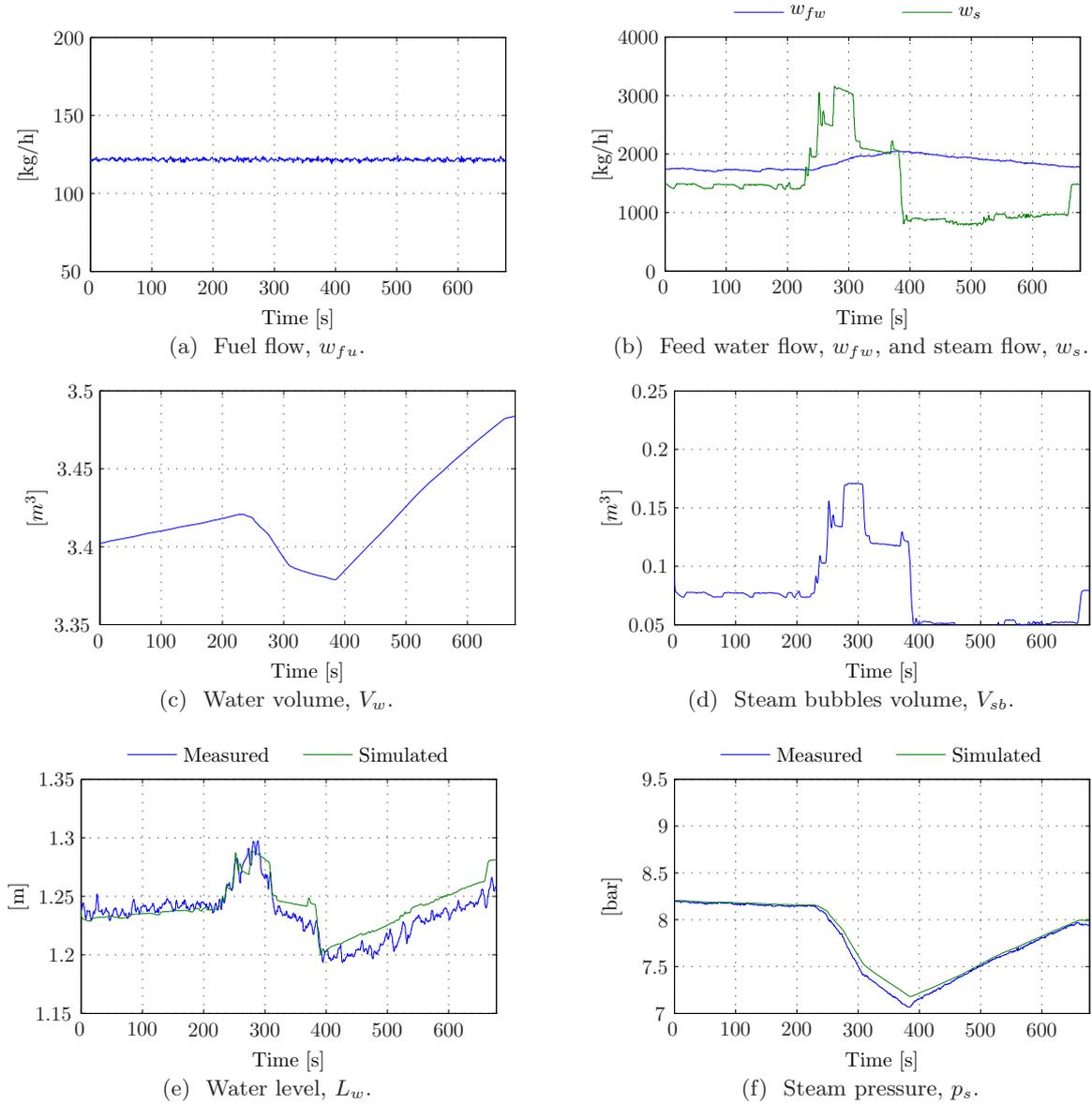


Figure 3.7: Non-linear boiler model response, from a step in steam flow.

When the step in steam flow is applied, Figure 3.7b shows a slightly increase in the feed water flow. This is caused by the decreasing steam pressure and thereby a larger pressure difference over the feed water valve. Figure 3.7f shows how the steam pressure decreases, and the model output is satisfactory similar to the boiler response. The change in water level is shown in Figure 3.7e, where it is seen that the step in steam flow causes the water level to increase. This is the swell phenomenon caused by the expanding steam bubbles volume, shown in Figure 3.7d, as the steam pressure drops. Also the water level from the model output is satisfactory similar to the boiler response, and thus the model is considered acceptable for a step in steam flow.

Step in feed water flow

Like the steam flow step response, the feed water step response is obtained by maintaining a steady feed water flow and adjusting the steam flow and fuel flow to obtain steady state, with a water level of approximately 1.23 m and a steam pressure of approximately 8 bar. The result is shown in Figure 3.8.

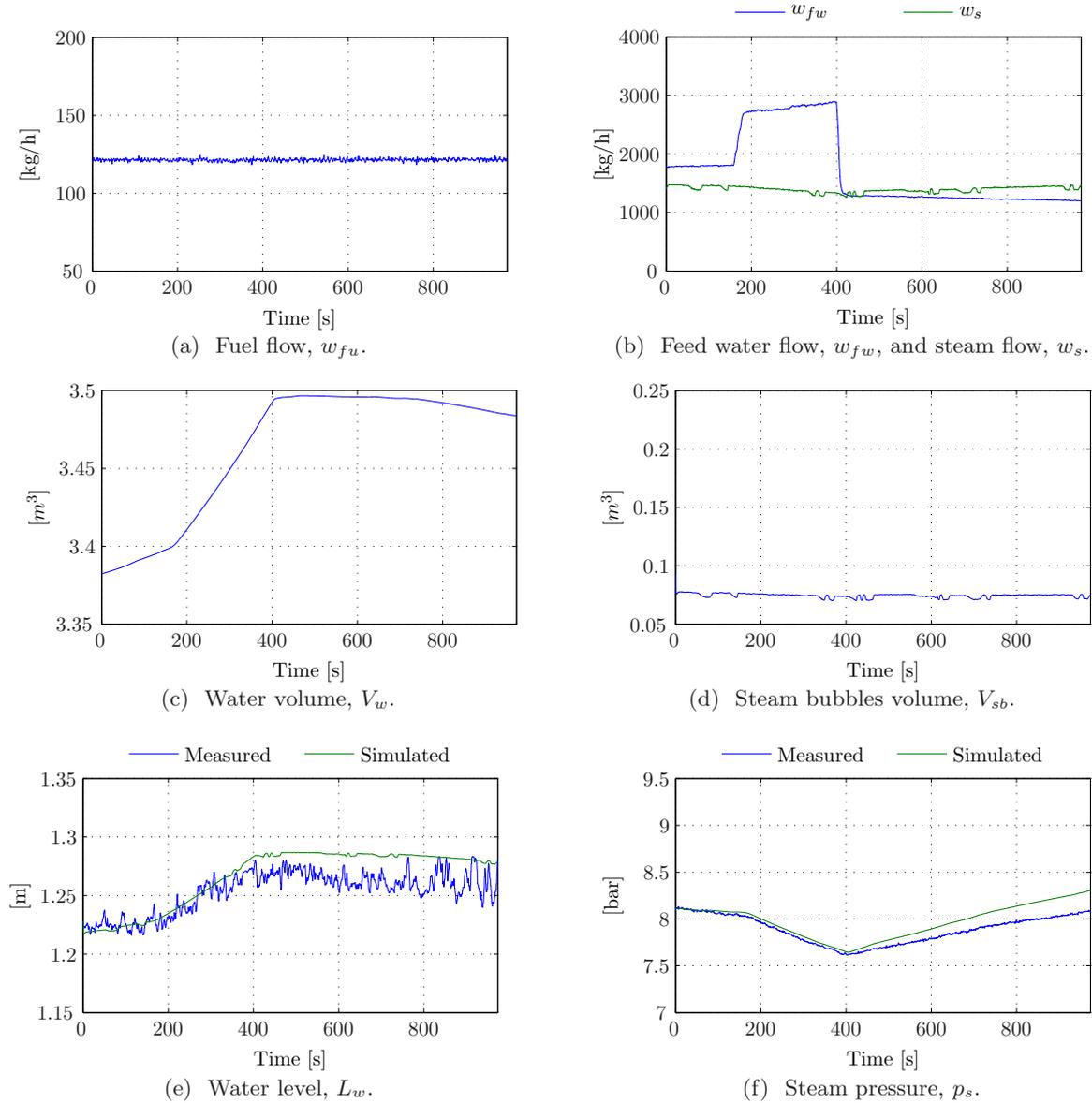


Figure 3.8: Non-linear boiler model response, from a step in feed water flow.

When the step in feed water flow is applied, the water level increases as shown in Figure 3.8e. Here the shrink phenomenon is not particularly visible, which is consistent to the description of the shrink and swell phenomenon in Chapter 2. However, the model output is similar to the boiler response. Because the feed water temperature is lower than the temperature inside the boiler, the increased feed water flow causes the steam pressure to decrease, as shown in Figure 3.8f. Also here the model output is sufficient similar to the boiler response. The decreased

steam pressure also induces a slightly decrease in steam flow, as shown in Figure 3.8b. The model output from a step in feed water flow is similar to the boiler response, and thus the model is considered acceptable.

Step in fuel flow

Finally the fuel flow step response is obtained by maintaining a steady fuel flow and adjusting the steam flow and feed water flow to obtain steady state, with a water level of approximately 1.23 m and a steam pressure of approximately 8 bar. In Figure 3.9b the changing levels during the first 150 s, concerns the stabilising of the steady state. The results from the step response is shown in Figure 3.9.

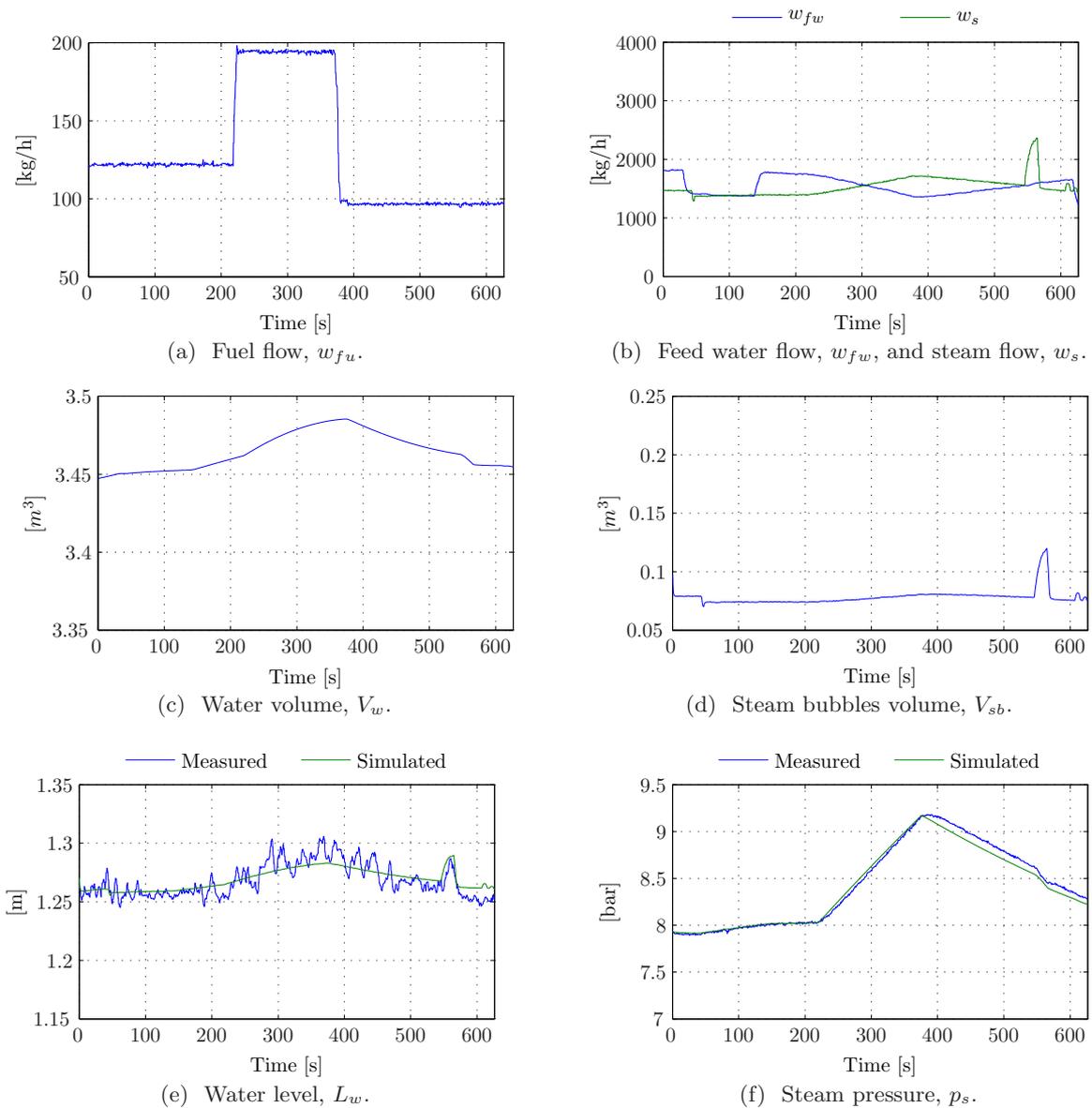


Figure 3.9: Non-linear boiler model response, from a step in fuel flow.

A step in fuel flow leads to an increase in steam pressure, shown in Figure 3.9f, as more water

is converted into steam. The increased steam pressure leads to a slightly increase in steam flow and a slightly decrease in feed water flow, as shown in Figure 3.9b. As more water is converted into steam, the volume of steam bubbles should increase, but only slightly as the increased steam pressure reduces the increase of steam bubble volume. The model output is very similar to the boiler response for both the water level and the steam pressure, in Figure 3.9e and 3.9f respectively.

Oxygen model verification

The derived oxygen model consists of a static part and a dynamic part. The static part is based on the chemical composition of fuel and air and is thus less uncertain compared to the dynamics of the oxygen model. In attempt to verify the model, measurements of fuel and air flow must be available. However, as the air flow is unmeasurable, the air damper position is used to estimate the air flow. By assuming a linear ratio between air damper position and air flow, this ratio can be estimated using Senstools and the result is shown in Figure 3.10.

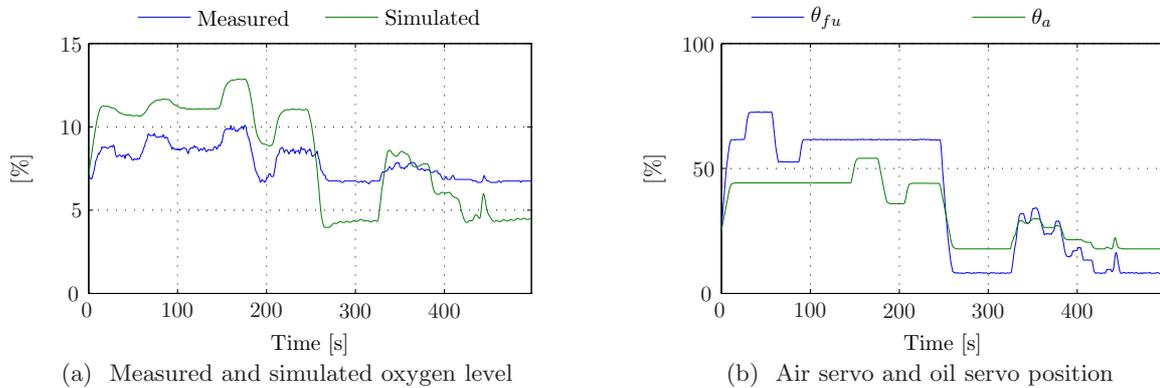


Figure 3.10: Estimation of linear ratio between air damper position and air flow. Oxygen levels (a) and servo positions (b).

Looking at the graphs, the dynamics of the oxygen model output is similar to the measured response, except from a gain variation. This implies that the ratio between the air damper position and the air flow is not linear; assuming the oxygen model is adequate. Looking at Figure 3.10a and Figure 3.10b, the simulated oxygen level is too high at high fuel and air flows and too low at low fuel and air flows. This indicates that instead of describing the air damper position to flow ratio as linear, an affine ratio, $w_a = a_a \cdot \theta_a + b_a$, might improve the estimate and thereby the ability to verify the oxygen model. The two parameters, a_a and b_a , are estimated and the result is shown in Figure 3.11.

This gives a more accurate estimate, and the oxygen model is considered acceptable, even though the inaccuracies potentially can be in both the oxygen model or in the estimated ratio between air damper position and air flow.

On the basis of the unmeasurable air flow, the oxygen model is considered acceptable and thereby the entire non-linear model of the boiler is verified, and can later be used for i.a. controller design and non-linear simulation.

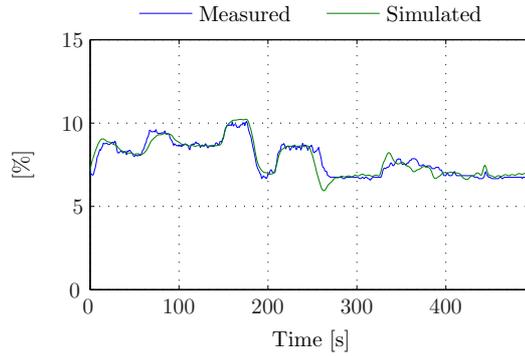


Figure 3.11: Estimation of affine ratio between air damper position and air flow.

3.2 Feed Water Actuator Modelling

This section describes how a model for the feed water actuator system is derived. The derived feed water actuator model is based on [Andersen and Jørgensen, 2007], as the structure of the actuator system is similar to the majority of boiler installations. Thus the focus in this section is mainly on how the model is derived and not on the detailed derivation.

On the basis of the system description in Chapter 2, the feed water actuator model consists of a valve positioner, a piping system and a flow sensor. A block diagram of this is illustrated in Figure 3.12.

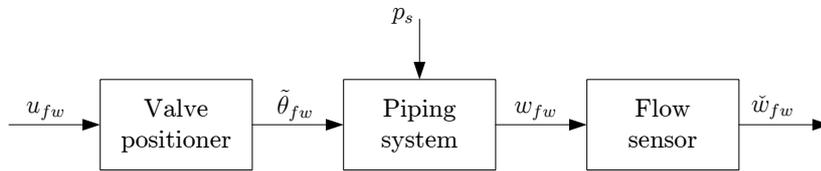


Figure 3.12: Block diagram of the feed water actuator system.

Due to the valve characteristics, the derived feed water actuator model will be non-linear.

3.2.1 Valve Positioner

The valve positioner is used to adjust the feed water valve stroke, and thereby indirectly the feed water flow. The valve positioner is controlled by pneumatic and has fast dynamics, compared to the flow sensor. Thus the dynamics of the valve positioner is ignored and the valve positioner consists only of a factor, that scales the control input signal, u_{fw} , to a normed feed water valve position: $\tilde{\theta}_{fw} \in [0; 1]$.

3.2.2 Piping System

The piping system is somewhat complex, although it mainly consists of two valves and a feed water pump. The feed water flow through the feed water valve is, according to Figure 2.3 on

page 5, dependent on the following parameters:

- The position of the feed water valve, θ_{fw}
- The position of the return valve, $k_{fw,r}$
- The pressure in the boiler, p_s
- The pressure in the feed water tank, p_0
- The pressure delivered by the feed water pump, $p_{fw,i}$

To limit the number of dependent parameters, the pressure from the feed water pump is assumed constant during operation. Also the position of the return valve is not changed during operation and the pressure in the feed water tank is equal to the ambient pressure. Thus the feed water flow can be described as a function of the remaining parameters; the feed water valve position and the steam pressure:

$$w_{fw} = f(\tilde{\theta}_{fw}, p_s) . \quad (3.63)$$

To derive an expression for the non-linear function, the feed water flow through the feed water valve can be expressed as:

$$w_{fw} = k_{fw} f(\tilde{\theta}_{fw}) \sqrt{p_{fw,i} - p_s} , \quad (3.64)$$

where k_{fw} is the feed water valve gain and $f(\tilde{\theta}_{fw})$ and $p_{fw,i}$ are unknown quantities. The non-linearity of the feed water valve, $f(\tilde{\theta}_{fw})$, can be expressed as an equal percentage characteristic, given by:

$$\begin{aligned} f_1(\tilde{\theta}_{fw}) &= a_{fw} \left(1 - e^{-b_{fw} \tilde{\theta}_{fw}} \right) & \text{for } 0 \leq \tilde{\theta}_{fw} \leq 0.1 \\ f_2(\tilde{\theta}_{fw}) &= R \tilde{\theta}_{fw}^{-1} & \text{for } 0.1 \leq \tilde{\theta}_{fw} \leq 1 , \end{aligned} \quad (3.65)$$

where R is a valve specific constant.

From practical experiments it appears, that the equal percentage characteristic can be well described by a second order polynomial [Andersen and Jørgensen, 2007].

An expression for the feed water pressure, $p_{fw,i}$, can be derived from examining the feed water pump and the return valve. Looking at Figure 2.3 on page 5, the pressure rise from the feed water pump can be expressed as:

$$\Delta p_{fw,i} = p_{fw,i} - p_0 . \quad (3.66)$$

The pressure rise, $\Delta p_{fw,i}$, is generally described by the lift height of the pump and is defined as:

$$\Delta p_{fw,i} = \rho_w g \Delta L_{fw,p} , \quad (3.67)$$

where ρ_w is the mass density, g is the gravity and $\Delta L_{fw,p}$ is the lifting height, dependent on the speed and flow capacity of the pump. Furthermore by using an expression for the flow through the linear return valve:

$$w_{fw,r} = k_{fw,r} \sqrt{p_{fw,i} - p_0} . \quad (3.68)$$

An expression for $\Delta p_{fw,i}$, can be derived as:

$$\Delta p_{fw,i} = \frac{-a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2}, \quad (3.69)$$

where

$$\begin{aligned} a_2 &= \left(1 + \frac{p_{fw,i,max}}{w_{fw,f,max}^2} \left(k_{fw}^2 f^2(\tilde{\theta}_{fw}) + k_{fw,r}^2 \right) \right)^2 - 4 \frac{p_{fw,i,max}^2}{w_{fw,f,max}^4} k_{fw}^2 f^2(\tilde{\theta}_{fw}) k_{fw,r}^2 \\ a_1 &= 2 \left(1 + \frac{p_{fw,i,max}}{w_{fw,f,max}^2} \left(k_{fw}^2 f^2(\tilde{\theta}_{fw}) + k_{fw,r}^2 \right) \right) \cdot \left(\frac{p_{fw,i,max}}{w_{fw,f,max}^2} k_{fw}^2 f^2(\tilde{\theta}_{fw}) (p_0 - p_s) - p_{fw,i,max} \right) \\ &\quad - 4 \frac{p_{fw,i,max}^2}{w_{fw,f,max}^4} k_{fw}^2 f^2(\tilde{\theta}_{fw}) k_{fw,r}^2 (p_0 - p_s) \\ a_0 &= \left(\frac{p_{fw,i,max}}{w_{fw,f,max}^2} k_{fw}^2 f^2(\tilde{\theta}_{fw}) (p_0 - p_s) - p_{fw,i,max} \right). \end{aligned}$$

For a detailed derivation of $\Delta p_{fw,i}$, see [Andersen and Jørgensen, 2007, p. 28–29].

Substituting Equation (3.66) into Equation (3.64), the final equation describing the feed water flow from the valve position and steam pressure, is given by:

$$w_{fw} = k_{fw} f(\tilde{\theta}_{fw}) \sqrt{\Delta p_{fw,i} + p_0 - p_s}, \quad (3.70)$$

where $f(\tilde{\theta}_{fw})$ and $\Delta p_{fw,i}$ is given by Equation (3.65) and Equation (3.69) respectively.

3.2.3 Flow Sensor

The feed water flow sensor can be described by a first order transfer function:

$$\frac{\check{w}_{fw}}{w_{fw}} = \frac{1}{1 + \tau_{fw} s}, \quad (3.71)$$

and has a time constant, τ_{fw} , of approximately 3 seconds according to the datasheet.

From these equations describing the feed water actuator system, a more detailed block diagram is illustrated in Figure 3.13.

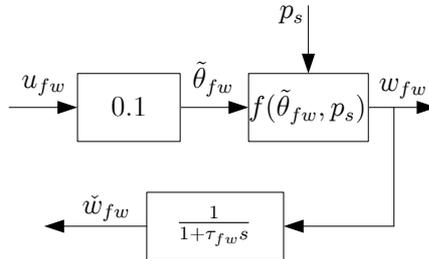


Figure 3.13: Detailed block diagram of the feed water actuator system.

Compared to the previous block diagram of the feed water actuator system in Figure 3.12, the sensor dynamics are placed in the feed back path. This implies that except from the non-linearities, no dynamics are present from control input, u_{fw} , to output flow, w_{fw} . This is not

exactly true as there is dynamics in the feed water valve, causing the feed water flow not to be immediate. However, relatively to the measured feed water flow from the sensor, the valve dynamics is neglectable as just described.

For the feed water actuator model it is necessary to determine the model parameters prior to verifying the model. Some of the parameters are calculated and others are estimated.

3.2.4 Parameter Estimation and Verification of the Feed Water Actuator Model

To verify the model of the feed water actuator system, the parameters $k_{fw,r}$ and τ_{fw} , expressing the valve position of the return valve and the time constant for the feed water actuator respectively, must be estimated. Prior to estimating the parameters, other constants in the feed water actuator model are calculated. These are listed in Table 3.2.

Parameter	Value	Unit
a_{fw}	17.47	
b_{fw}	0.044	
$\Delta p_{fw,i,max}$	$1.17 \cdot 10^6$	Pa
$w_{fw,f,max}$	2.63	$\frac{\text{kg}}{\text{s}}$
k_{fw}	0.0141	$\sqrt{\text{m} \cdot \text{kg}}$
R	40	

Table 3.2: Constants for the feed water actuator model [Andersen and Jørgensen, 2007].

The parameters, $k_{fw,r}$ and τ_{fw} , are estimated using Senstools in MATLABTM. The estimation method is described in Appendix C, and the estimated values of the parameters are listed in Table 3.3.

Parameter	Estimated value	Unit
$k_{fw,r}$	$0.62 \cdot 10^{-3}$	$\sqrt{\text{m} \cdot \text{kg}}$
τ_{fw}	3.69	s

Table 3.3: Estimated parameters for the feed water actuator model.

Remaining is to verify the feed water actuator model with the estimated parameters.

Verification of feed water actuator model

New measurement data is obtained from AI's test center. They are used in comparison to the feed water actuator model output, and the responses are shown in Figure 3.14. Before the measurements are retrieved, the boiler is brought into steady state, to obtain a steam pressure of 8 bar.

Figure 3.14a shows the applied control input signal, u_{fw} , to the feed water actuator, and Figure 3.14b shows the measured and simulated response. From this result it is easy to see the simulated

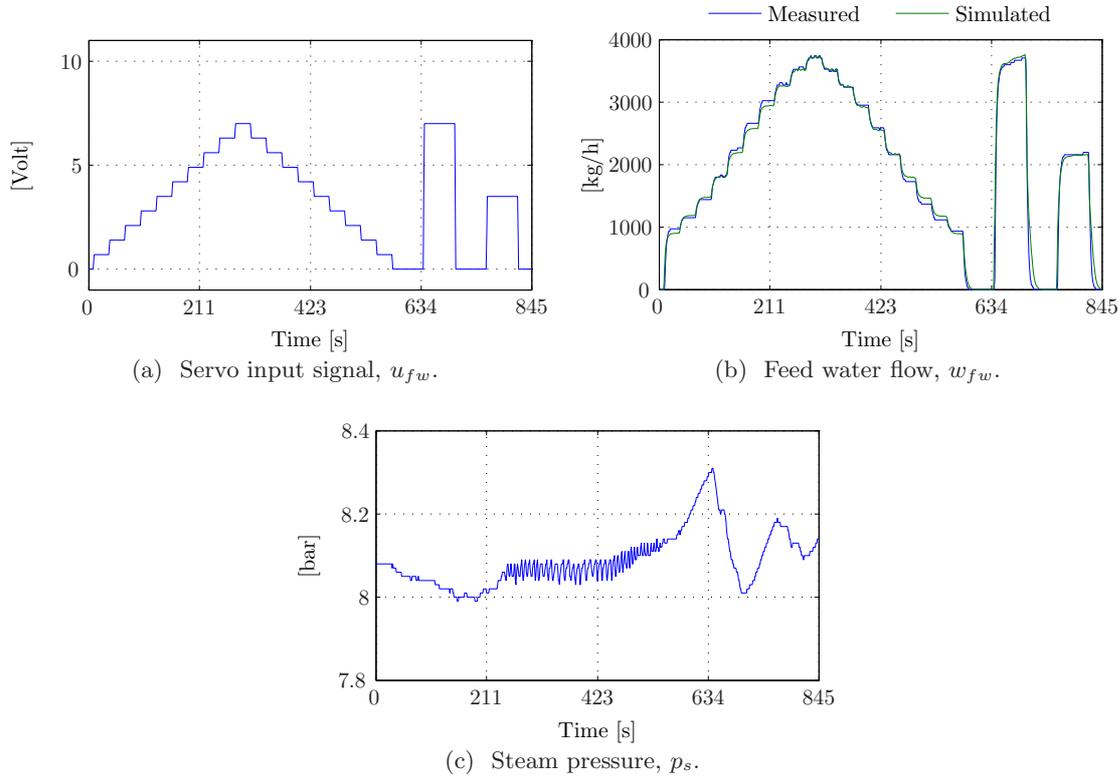


Figure 3.14: Verification of feed water actuator model.

response is similar to the measured response. In spite of small gain variations, the feed water actuator model with the estimated parameters is considered satisfactory. Figure 3.14c shows the steam pressure, p_s , which is around the operating point of 8 bar.

3.3 Fuel Actuator Modelling

This section describes the derivation of the model for the fuel actuator system, based on the description of the fuel actuator system in Chapter 2. The derived fuel actuator model is based on [Andersen and Jørgensen, 2007], as the structure of the actuator system is similar to the majority of boiler installations. Thus the focus in this section is mainly on how the model is derived and not on the detailed derivation.

As the structure of the oil and air actuators are similar, only a model for the oil actuator is derived. However, the differences between the two actuators are subsequently pointed out.

The oil actuator consists of a servo motor, the piping system including the oil valve and the fuel flow sensors. The servo motor, to adjust the valve stroke of the fuel valve, θ_{fu} , is controlled by an input signal denoted u_{fu} . The oil valve position results in a fuel flow, w_{fu} , in the piping system and the oil flow is measured by two flow sensors. A block diagram of the model structure is illustrated in Figure 3.15.

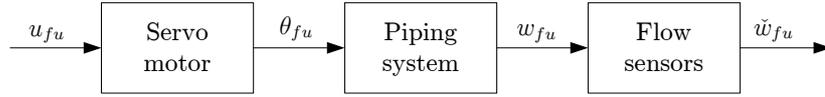


Figure 3.15: Block diagram of the oil actuator system.

As the piping system has non-linear characteristics, the model of the oil actuator will be non-linear.

3.3.1 Servo Motor

The servo motor has two terminals, where an input at one of the terminals makes the servo motor run clockwise (CW) and the other counter clockwise (CCW). When a control signal is applied to any of the terminals, the servo motor runs at constant speed, making the change in valve position constant. Thus the servo motor can be modelled as an integrator.

In stead of using a control signal for each of the terminals, the control input, u_{fu} , is normed to $\tilde{u}_{fu} \in \{-1, 0, +1\}$, where -1 corresponds to a change in the CCW direction and +1 corresponds to a change in the CW direction. Thus the sign of the normed input signal indicates the sign of integration. The traveling speed of the servo motor and thereby rate of change in valve stroke, is determined by an integrator gain defined by a constant, K_{fu} . All together this gives the model for the servo motor:

$$\theta_{fu}(t) = \theta_{fu}(t_0) + K_{fu} \int_{t_0}^t \tilde{u}_{fu}(\tau) d\tau \quad \text{for} \quad \tilde{u}_{fu} \in \{-1, 0, 1\}. \quad (3.72)$$

3.3.2 Piping System

The significant parts in the piping system are the valves and the fuel atomiser. As these parts are purchased from sub suppliers, the knowledge to the characteristics are limited. Assuming that the dynamics of the piping is fast, the piping system can be described by a static model. Hence, a third order polynomial on the form:

$$w_{fu} = f(\theta_{fu}) = a_{fu,3}\theta_{fu}^3 + a_{fu,2}\theta_{fu}^2 + a_{fu,1}\theta_{fu}^1 + a_{fu,0}, \quad (3.73)$$

is sufficient to describe the piping system.

3.3.3 Flow Sensors

As previously mentioned, two flow sensors are used to estimate the actual oil flow, w_{fu} . Each of the sensors constitute a first order system, resulting in a second order flow sensor model. Due to the small delay between the measurements of the fuel flow, the sensors are best modelled as a second order nonminimum phase system:

$$\frac{\check{w}_{fu}}{w_{fu}} = \frac{1 - \tau_{fu,2}s}{(1 + \tau_{fu,1}s)^2}. \quad (3.74)$$

Thereby the system has a zero in the RHP, causing a larger variation in the phase response.

From the outlined equations of the oil actuator, a more detailed block diagram is illustrated in Figure 3.16.

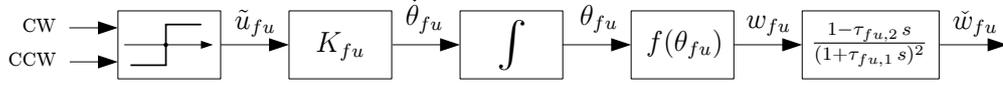


Figure 3.16: Detailed block diagram of the oil actuator system.

As stated earlier, the air actuator is similar in structure. Hence a block diagram of the air actuator system is illustrated in Figure 3.17.

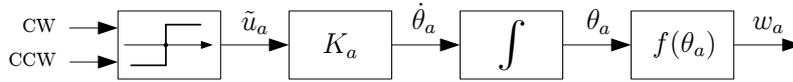


Figure 3.17: Detailed block diagram of the air actuator system.

As seen, the two systems are very similar, except from no flow sensor is present in the air actuator system and thereby no sensor dynamics in the model. Also, the non-linearity and gain of the oil valve, $f(\theta_{fu})$ and K_{fu} are replaced by their counterpart for an air damper, namely $f(\theta_a)$ and K_a .

The model of the fuel actuator system, including both oil and air actuators, is implemented in Simulink and verified in the following subsection.

3.3.4 Parameter Estimation and Verification of the Fuel Actuator Model

The parameter estimation and verification for the fuel actuator model is divided in two; the oil actuator model and air damper model.

Oil actuator model verification

For the oil actuator model there are many unknown parameters that must be estimated. This includes the non-linear valve characteristic coefficients, the valve integrator gain and the two time constants in the flow sensors.

Parameter	Estimated value	Unit
$a_{fu,3}$	$-2.79 \cdot 10^{-7}$	$\frac{\text{kg}}{\text{s} \cdot \%^3}$
$a_{fu,2}$	$5.37 \cdot 10^{-5}$	$\frac{\text{kg}}{\text{s} \cdot \%^2}$
$a_{fu,1}$	$-2.61 \cdot 10^{-3}$	$\frac{\text{kg}}{\text{s} \cdot \%}$
$a_{fu,0}$	0.045	$\frac{\text{kg}}{\text{s}}$
K_{fu}	15.17	$\frac{\%}{\text{s}}$
$\tau_{fu,1}$	3.56	s
$\tau_{fu,2}$	1.96	s

Table 3.4: The estimated parameters for the oil actuator model.

In Table 3.4 the estimated parameters for the oil valve actuator model are listed.

Given these estimated parameters, the oil actuator model is verified against measurement data, and the result is shown in Figure 3.18.

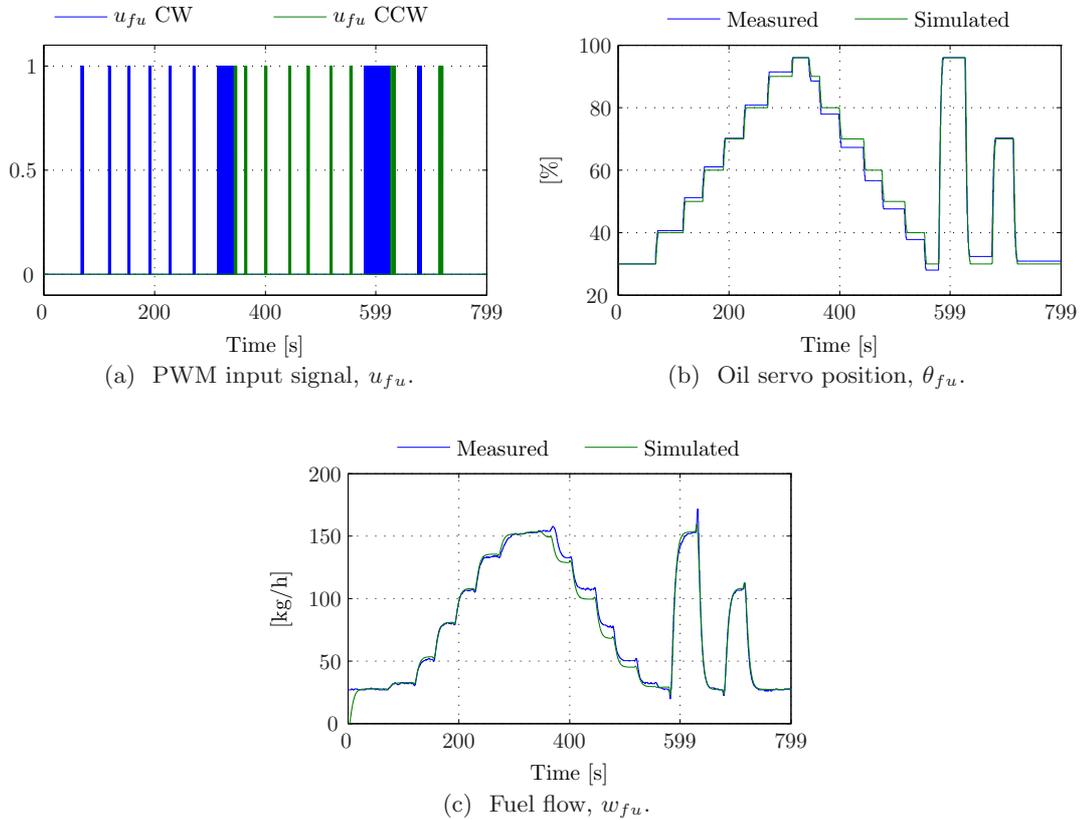


Figure 3.18: Verification of oil actuator model.

Figure 3.18a shows the input signal applied to the PWM servo motor. The resulting servo position is shown in Figure 3.18b, where the model output and measured response is similar. Furthermore the measured fuel flow is compared to the model output, and Figure 3.18c shows that also this result is satisfactory. Hereby the oil servo actuator model is approved with the estimated parameters.

Air actuator model

Also for the air actuator model there are parameters that needs to be estimated. The coefficients of the non-linear valve characteristic and the air damper integrator gain. The integrator gain is relative simple to estimate, but as it is not possible to measure the air flow in AI's test center, it is difficult to estimate the valve characteristic. However, relying on the oxygen model, the flow characteristics can be estimated as described in the verification of the oxygen model in Section 3.1.6 on Page 33. Thereby the resulting flow characteristic is defined by an affine function on the form:

$$w_a = a_a \cdot \theta_a + b_a , \quad (3.75)$$

and the estimated parameters are listed in Table 3.5.

Parameter	Estimated value	Unit
a_a	0.009	
a_b	0.168	
K_a	15.17	$\frac{\%}{s}$

Table 3.5: The estimated parameters for the air damper model.

At this point the entire Mission OSTM boiler system is modelled and the models are verified. The following section contains a short resume of the derived models.

3.4 Modelling Resume

The boiler system has been modelled as three separate non-linear models; a boiler model, a feed water actuator model and a fuel actuator model.

The non-linear boiler model was derived from a number of control volumes, where balance equations were used to describe the boiler dynamics. To simplify the model some assumptions were made and knowledge from previously projects was taking into consideration. The non-linear boiler model was derived and presented as a state space model on descriptor form. Furthermore the non-linear boiler model was implemented in Simulink as an S-function and a few parameters were estimated, before the model was successfully verified.

The feed water actuator model was modelled as three parts; a valve positioner, the piping system and a flow sensor. This resulted in a non-linear model, due to the non-linear valve characteristic. The model parameters were estimated and the feed water actuator model verification showed a satisfactory result.

The fuel actuator model consists of an oil actuator model and an air actuator model. The oil actuator was modelled as a servo motor, a piping system including valves and a flow sensor. As the servo motor is PWM controlled with constant angular velocity, it was modelled as an integrator. The piping system and valves have non-linear characteristics which was approximated by a third order polynomial. The flow sensors was modelled as a second order nonminimum phase system, due to a delay between the measurements of the forward and return flows. After estimating the unknown model parameters, the oil actuator model was verified against measurement data, and the model has similar dynamic behaviour as the physical response.

The air actuator model was modelled in the same way as the oil actuator model, except from no flow sensor is included. Thus the verification of the actuator model was conducted relying on the oxygen model.

Controller Design

This chapter contains a description of the designed controllers for the Mission OSTM boiler system. The overall task is to design and implement controllers to control the water level, the steam pressure and the oxygen level. However, to optimize the performance, actuator controllers are designed as well. The controllers are designed based on the models derived in Chapter 3, and the controllers are all SISO controllers. The choice of designing SISO controllers facilitates a simplified underlying basis for development and implementation of self-tuning of the Mission OSTM boiler system, described in Chapter 5.

First, an overview of the existing and desired controller structure, consisting of inner and outer-loop controllers, is given. Next the inner-loop controllers are designed and verified by simulation, followed by a derivation of control models and design of the outer-loop controllers. Finally the outer-loop controllers are verified by simulation and a resume of this chapter is given.

Figure 4.1 illustrates the controller structure for the Mission OSTM boiler system. In total, six SISO controllers are to be designed. Three inner-loop controllers must be designed, to control the feed water actuator, the oil valve and the air damper. In addition three outer-loop controllers must be designed in order to control the water level, the steam pressure and the oxygen level. The inner-loops are in Figure 4.1 marked by grey.

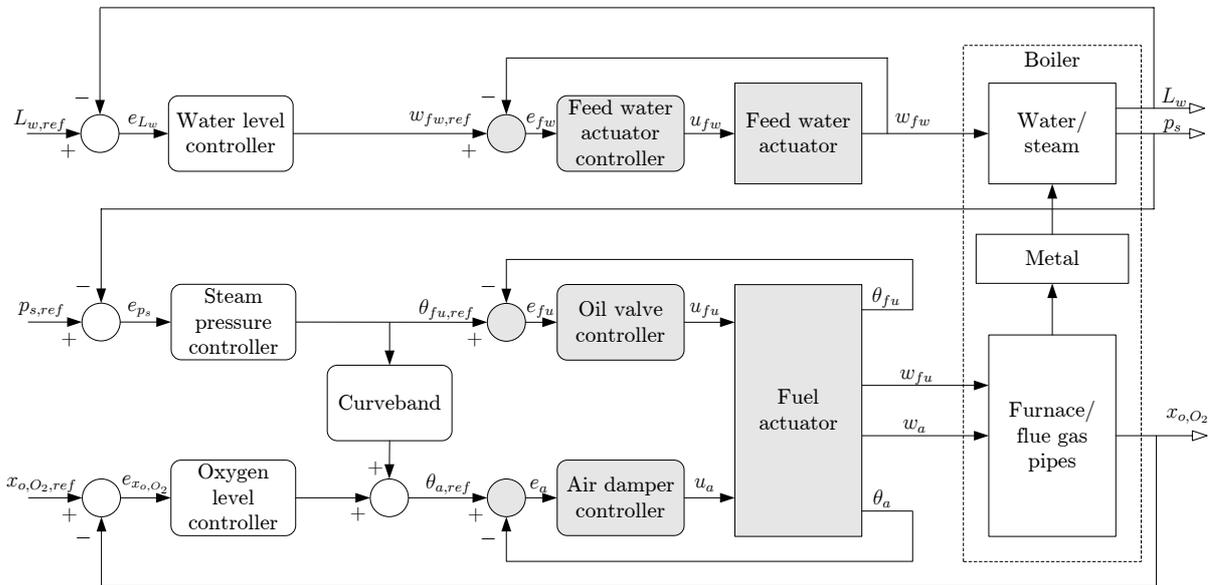


Figure 4.1: Illustrates the controller structure for the Mission OSTM boiler system.

The controller structure is based on the existing control structure used by at AI, except from the oxygen level controller and feed water actuator controller. Alternative to the oxygen level controller, AI only uses a curveband, fitted for a specified oxygen level, as feed forward from the

air damper position to oil valve position, resulting in an approximated relation between fuel flow and air flow. In this controller structure the curveband is used opposite, as feed-forward from oil valve position to air damper position. Furthermore, to fine-adjust the oxygen level, it is desired to add the oxygen level feedback controller as depicted in Figure 4.1. In attempt to improve the water level controller, a feed water actuator controller must be designed, to control the feed water flow into the boiler. The existing water level control is designed somewhat simpler, without an inner loop.

The remaining of this chapter first contains a description of how the inner-loop controllers are designed, and subsequently how the outer-loop controllers are designed before the controllers finally are verified.

4.1 Inner-loop Controller Design

The Mission OSTM boiler system features two actuator systems; the feed water actuator system and the fuel actuator system. The control of each of these is described in the following subsections. The inner-loop controller for the feed water actuator is designed to compensate for the non-linearities in the system, whereas the controllers for the oil valve and air damper are designed to facilitate position control of the oil valve and air damper. The position controllers for the oil valve and the air damper are designed to achieve fast fuel actuator response.

4.1.1 Feed Water Actuator Control

To compensate for the non-linearities, $f(\theta_{fw}, p_s)$, in the feed water actuator system, these non-linearities must first be identified. It seems obvious that the valve characteristic is non-linear, but the feed water flow also depends on the steam pressure in the boiler. [Andersen and Jørgensen, 2007] analysed the non-linearities of the feed water actuator system and concluded two things. In a steam pressure interval between 7.6 and 8.4 bar, around the operating point, the non-linearities only slightly varies and thus just slightly affects the feed water flow, w_{fw} . Thereby the non-linearity from a change in steam pressure is neglectable. To determine the non-linear effect of the valve characteristic, the small signal gain, $\frac{\Delta \dot{w}_{fw}}{\Delta \theta_{fw}}$, is examined. The result of this examination shows that the small signal gain varies a factor 14 and thus the non-linearity can not be neglected and must be included in the controller design.

There are various known methods to control non-linear systems, like gain scheduling or back-stepping, but a control strategy as illustrated in Figure 4.2 is advantageous to use. The idea is to identify the non-linear characteristic of the valve and use the inverse characteristic as feed forward. This entails a linearisation and thus a linear feedback controller, C_{fw} , can be used to control the feed water flow.

Inverse non-linear function

Firstly by ignoring the feedback loop, the inverse non-linearity, $f^{-1}(\tilde{\theta}_{fw})$, is derived. Based on the analysis of the non-linearity, a second order polynomial is sufficient to approximate the

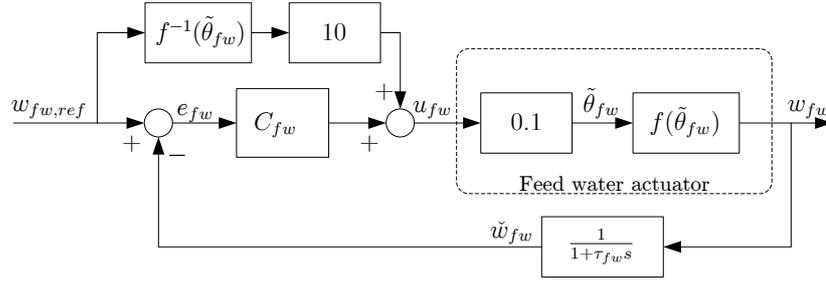


Figure 4.2: Control strategy for the non-linear feed water actuator system, featuring feed-forward and feedback.

non-linearity of the valve. A general second order polynomial is given as:

$$f(\tilde{\theta}_{fw}) = a_{fw,2}\tilde{\theta}_{fw}^2 + a_{fw,1}\tilde{\theta}_{fw} + a_{fw,0} . \quad (4.1)$$

From Figure 4.2 it is seen that $w_{fw} = f(\tilde{\theta}_{fw})$. By inserting this into Equation (4.1), the solution to $\tilde{\theta}_{fw}$ is given by:

$$\tilde{\theta}_{fw} = \frac{-a_{fw,1} + \sqrt{a_{fw,1}^2 - 4a_{fw,2}(a_{fw,0} - w_{fw})}}{2a_{fw,2}} , \quad (4.2)$$

where $\tilde{\theta}_{fw}$ is the input and w_{fw} is the output.

To find the inverse of the solution, the input and output are interchanged and by looking at Figure 4.2, the input and output to the inverse non-linearity is given by $w_{fw,ref}$ and $u_{fw} = f^{-1}(\tilde{\theta}_{fw})$ respectively. This gives the solution of the inverse non-linear function:

$$f^{-1}(w_{fw,ref}) = \frac{-a_{fw,1} + \sqrt{a_{fw,1}^2 - 4a_{fw,2}(a_{fw,0} - w_{fw,ref})}}{2a_{fw,2}} , \quad (4.3)$$

which is used as feed forward to eliminate the non-linearity. Furthermore to compensate for the norming factor in the feed water actuator, the inverse non-linear function is multiplied by a factor 10. Remaining is to design the feedback controller for the feed water actuator system.

Feedback control

Assuming the inverse non-linear function completely cancel out the non-linearity in the feed water actuator, the open loop system response is, according to Figure 4.2, given by the gain in the feed water actuator and the sensor dynamics:

$$G_{OL}(s) = C_{fw} \frac{w_{fw}}{u_{fw}} \frac{1}{1 + \tau_{fw} s} . \quad (4.4)$$

Using the expression for the open loop response, the closed loop response is defined as:

$$G_{CL}(s) = \frac{w_{fw}}{w_{fw,ref}} = \frac{C_{fw}(s) \frac{w_{fw}}{u_{fw}}}{1 + C_{fw}(s) G_{OL}(s)} . \quad (4.5)$$

Using a Skogestad Internal Model Control scheme [Skogestad, 2002], a desired closed loop response can be obtained:

$$\left(\frac{w_{fw}}{w_{fw,ref}} \right)_{desired} = 1 . \quad (4.6)$$

This indicates, that the feed water flow will have an immediate response from the reference signal, and the measured feedback signal contains the dynamics, as discussed in the modelling in Section 3.2 on Page 36.

From the closed loop response, Equation (4.5), it is possible to derive an expression for the feedback controller:

$$C_{fw}(s) = \frac{1}{\frac{\frac{w_{fw}}{u_{fw}}}{\left(\frac{w_{fw}}{w_{fw,ref}}\right)_{desired}} - G_{OL}(s)} . \quad (4.7)$$

Using the desired closed response from Equation (4.6) and rearranging, the resulting feed back controller reduces to:

$$C_{fw}(s) = \frac{u_{fw}}{w_{fw}} \left(1 + \frac{1}{\tau_{fw} s} \right) , \quad (4.8)$$

which is a PI-controller with proportional gain, $k_{fw,p} = \frac{u_{fw}}{w_{fw}}$, and integrator time constant, $T_{fw,i} = \tau_{fw}$. The proportional gain can be calculated if the maximum feed water flow is known and remembering the maximum value of the control signal is 10 V. Thus the only controller parameter needed to find is the time constant for the feed water actuator, τ_{fw} .

Anti wind-up

When designing and implementing a PI-controller, it is necessary to consider the integrator wind-up effect. If the actuator system introduces saturation, the PI-controller will still try to eliminate the error, causing integrator wind-up. Thus it is relevant to design an anti wind-up scheme, along with the design of the PI-controller. A tracking anti wind-up scheme has a good trade off between complex implementation and efficiency [Bohn and Atherton, 1995], and is thus chosen to implement. The tracking anti wind-up scheme is shown in Figure 4.3, and consists of the PI-controller together with the tracking anti wind-up.

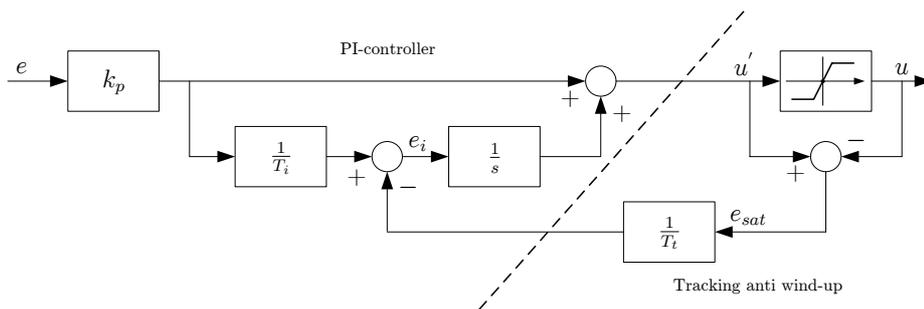


Figure 4.3: Generalised tracking anti wind-up scheme [Franklin et al., 2006, p. 671].

The parameter T_t is a free parameter and can be used as tuning parameter, to adjust the performance of the tracking anti wind-up.

When $u = u'$ no saturation is present, and the saturation error is zero, $e_{sat} = 0$. Thereby the tracking anti wind-up is passive, resulting in an ordinary PI-controller. Opposite when $e_{sat} \neq 0$, the wind-up value is subtracted from the integrator part, and thereby integrator wind-up is avoided. This can be expressed mathematically from Figure 4.3, where two equations can be

derived:

$$e_i = e \frac{k_p}{T_i} - e_{sat} \frac{1}{T_t} \quad (4.9)$$

$$u' = e \cdot k_p + e_i \frac{1}{s}. \quad (4.10)$$

Assuming the system is saturated, the small signal gain of e_{sat} is equal to the small signal gain of u' . Inserting $e_{sat} = u'$ into Equation (4.9) the transfer function from e to u' is derived as:

$$\frac{u'}{e} = k_p \frac{s + \frac{1}{T_t}}{s + \frac{1}{T_i}}. \quad (4.11)$$

It is obvious that if the tuning parameter, T_t , is chosen to be equal to the integrator parameter, T_i , the tracking anti wind-up scheme reduces to a proportional controller, and integrator wind-up is avoided whenever saturation is present.

At this point, a feed water flow controller is designed, to depend on the non-linear characteristic of the valve, with parameters $\psi_{fw} = [a_{fw,2} \ a_{fw,1} \ a_{fw,0}]$, the measurable feed water actuator time constant, τ_{fw} and gain, $k_{fw,p}$. The polynomial coefficients, the time constant and gain can be found using well known methods as parameter estimation or as a part of a self-tuning algorithm.

4.1.2 Oil Valve Position Control

The fuel actuator system, described in Chapter 2, facilitates an oil valve controlled by a PWM controller with position feedback. The controller is designed as a proportional controller and is chosen as the system in itself facilitates an integrator.

A block diagram of the controlled oil valve is shown in Figure 4.4.

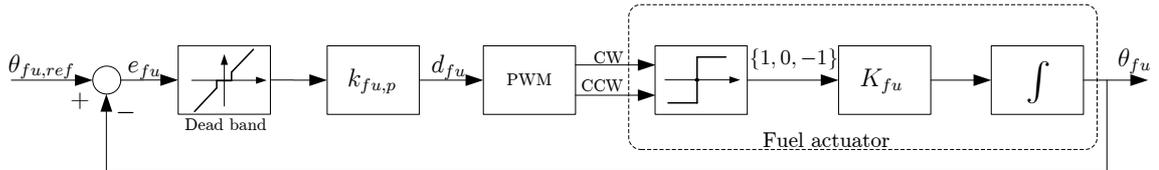


Figure 4.4: Oil valve control structure.

The PWM block features a normed periodic ramp signal with the period T_{PWM} , which is compared to the input signal, d_{fu} , to generate an PWM output signal. This is illustrated in Figure 4.5, where u_{fu} is the PWM signal applied to either the CW or CCW terminal. The duty cycle of the PWM output signal is calculated from the normed input, $|d_{fu}|$ whereas the output terminal is determined based on the sign of the input, i.e. the oil valve is opened by a clock-wise movement, when the input is greater than zero and vice versa.

Looking at the controller structure in Figure 4.4, the valve movement, $\Delta\theta_{fu}$, during the time T_{PWM} can be expressed as an integral as follows:

$$\Delta\theta_{fu} = \int_0^{T_{PWM}d_{fu}} K_{fu} dt \quad \text{for } d_{fu} \in [0, 1]. \quad (4.12)$$

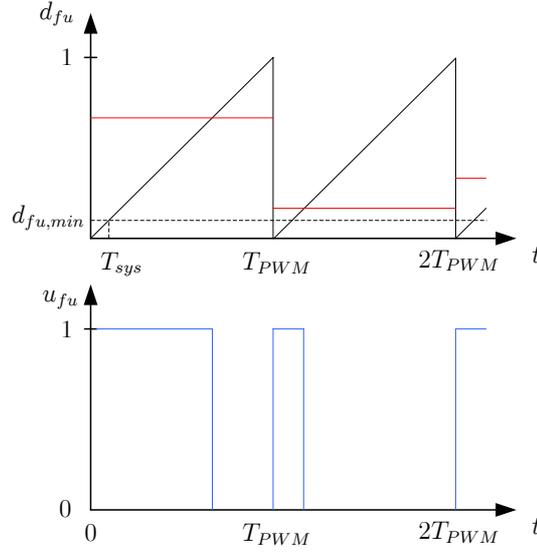


Figure 4.5: Illustration of PWM output generation.

To find the proportional gain, $k_{fu,p}$, for the controller, the valve movement, $\Delta\theta_{fu}$, is expressed by the valve position error, e_{fu} :

$$\Delta\theta_{fu} = K_{fu}T_{PWM}k_{fu,p}e_{fu} . \quad (4.13)$$

To achieve fast positioning of the valve, a dead beat controller is designed to ensure that the valve movement during T_{PWM} equals the error, e_{fu} . Thus the proportional gain, $k_{fu,p}$, can be found by inserting $\Delta\theta_{fu} = e_{fu}$:

$$k_{fu,p} = \frac{1}{T_{PWM}K_{fu}} . \quad (4.14)$$

A time period of $T_{PWM} = 1$ is found to be a good compromise between pulse with resolution and high effective sample rate, hence the proportional gain is only dependent on the traveling time of the oil actuator.

Dead band

To avoid limit cycles a dead band is introduced on the position error, e_{fu} . Looking at Figure 4.5, the minimum duty cycle is given by $d_{fu,min} = \frac{T_{sys}}{T_{PWM}}$, where T_{sys} is the system sampling time. Hence, the minimum representable error is given by:

$$|e_{fu,min}| = \frac{d_{fu,min}}{k_{fu,p}} = K_{fu}T_{sys} . \quad (4.15)$$

Finally, to ensure that limit cycles are avoided, a margin of 25 % is added to the minimum representable error. Thus the dead band is given by:

$$|e_{fu,db}| = 1.25K_{fu}T_{sys} . \quad (4.16)$$

At this point, an oil valve position controller is designed, to depend only on the measurable oil valve integrator gain, K_{fu} , and this integrator gain is also the only unknown parameter to specify the dead band. The integrator gain can be found using well known methods as parameter estimation or as a part of a self-tuning algorithm.

4.1.3 Air Damper Position Control

The air damper is controllable using the same principles as for the oil valve. The only difference is that the oil valve is replaced by an air damper. In this way the controller parameters for the air damper controller can be summarised as follows.

The proportional gain for the air damper controller, $k_{a,p}$, is given by:

$$k_{a,p} = \frac{1}{T_{PWM}K_a} . \quad (4.17)$$

Dead band

As for the designed oil valve position controller, a dead band to avoid limit cycles is given by:

$$|e_{a,db}| = 1.25K_aT_{sys} . \quad (4.18)$$

At this point, an air damper position controller is designed, to depend only on the measurable air damper integrator gain, K_a .

The controllers designed for the oil valve and the air damper leads to equal responses of the two systems, when the PWM is not saturated. The equal responses for the oil valve and air damper is desirable, to avoid large variation of the oxygen level. However, the use of PWM introduces saturation issues which are described in the following section.

4.1.4 PWM Saturation

The nature of PWM introduces saturation when the duty cycle reaches 100 %, i.e. a value of 1. The designed position controllers introduced a proportional control gain, inverse proportional to the valve gain, $K_{(\cdot)}$, and the PWM time period, T_{PWM} :

$$k_{(\cdot)} = \frac{1}{T_{PWM}K_{(\cdot)}} . \quad (4.19)$$

With the introduced proportional gain the PWM saturation emerges when:

$$e_{(\cdot)} \geq T_{PWM}K_{(\cdot)} . \quad (4.20)$$

For a valve gain of e.g. 0.05 and $T_{PWM} = 1$, the PWM saturates when the error is greater than 5 %, implying that the valve travels at full speed when the error is greater than 5 %. Hence the output is only modulated with an error less than 5 %.

A simulation of a step response with valve gains, $k_{fu} = 0.05$ and $k_a = 0.02$ is depicted in Figure 4.6.

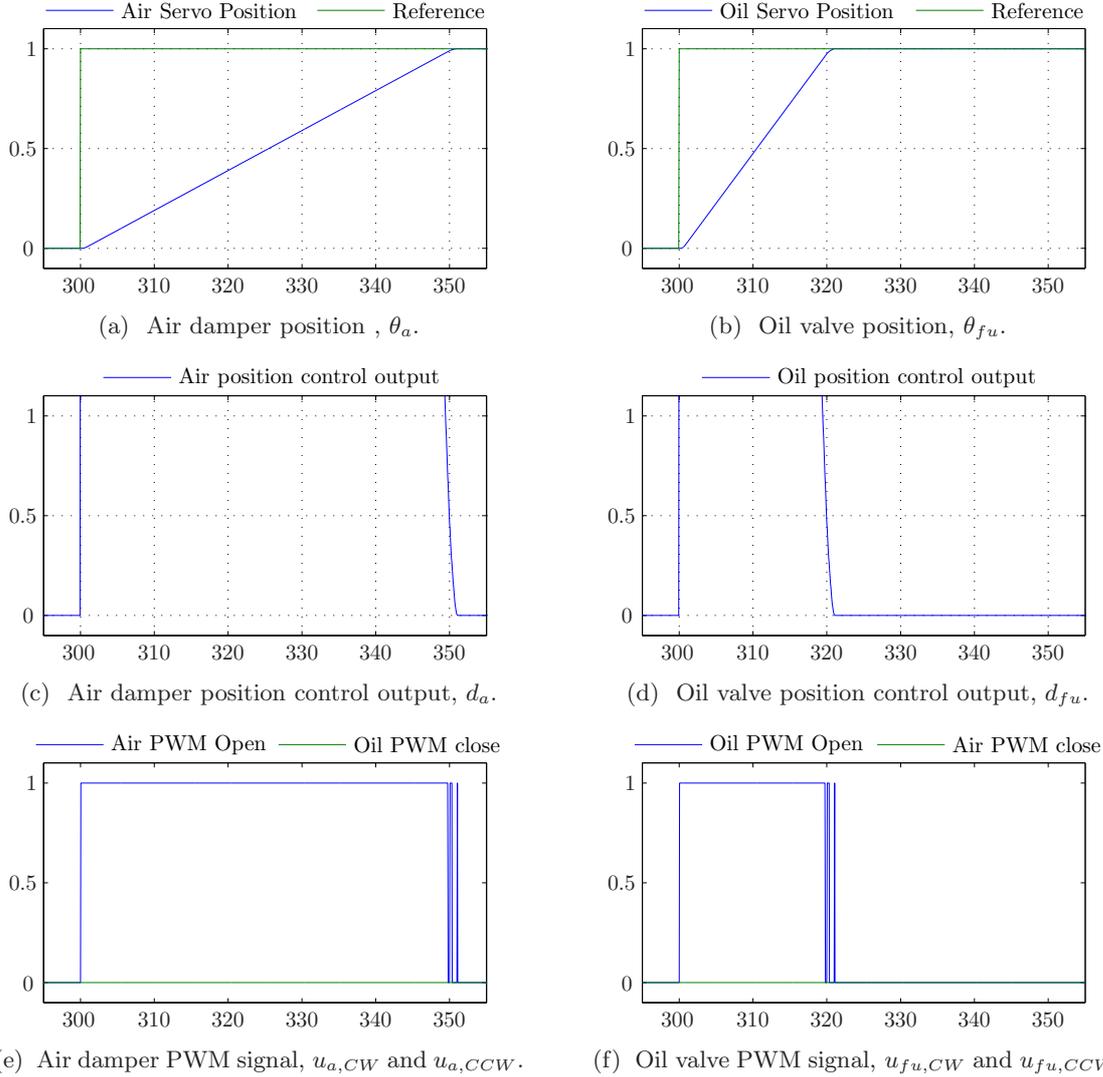


Figure 4.6: Oil valve and air damper response with step in reference at time, $t = 300$, without PWM saturation compensation.

From the simulation results, it is obvious, that the PWM saturation causes the air damper to reach the reference faster than the oil valve, simply because the air damper travels faster than the oil valve, when saturated.

The PWM saturation and the fact that the air damper and oil valve can have unequal valve gains, makes it less straightforward to design the position controllers for the oil valve and the air damper to have equal responses.

For the fast valve to become as slow as the slow valve, the control output for the fast valve must be limited. The saturation of the output of the proportional controller is for the fast valve determined as the relation between the valve gains and for the slow valve equal to 1:

$$|d(\cdot)| \leq \begin{cases} 1 & , \text{ for the slow valve} \\ \frac{MIN(K_a, K_{fu})}{MAX(K_a, K_{fu})} & , \text{ for the fast valve.} \end{cases} \quad (4.21)$$

With the duty cycle saturation compensation described above, the response for the air damper and oil valve is depicted in Figure 4.7. Compared to Figure 4.6, the valves now have equal

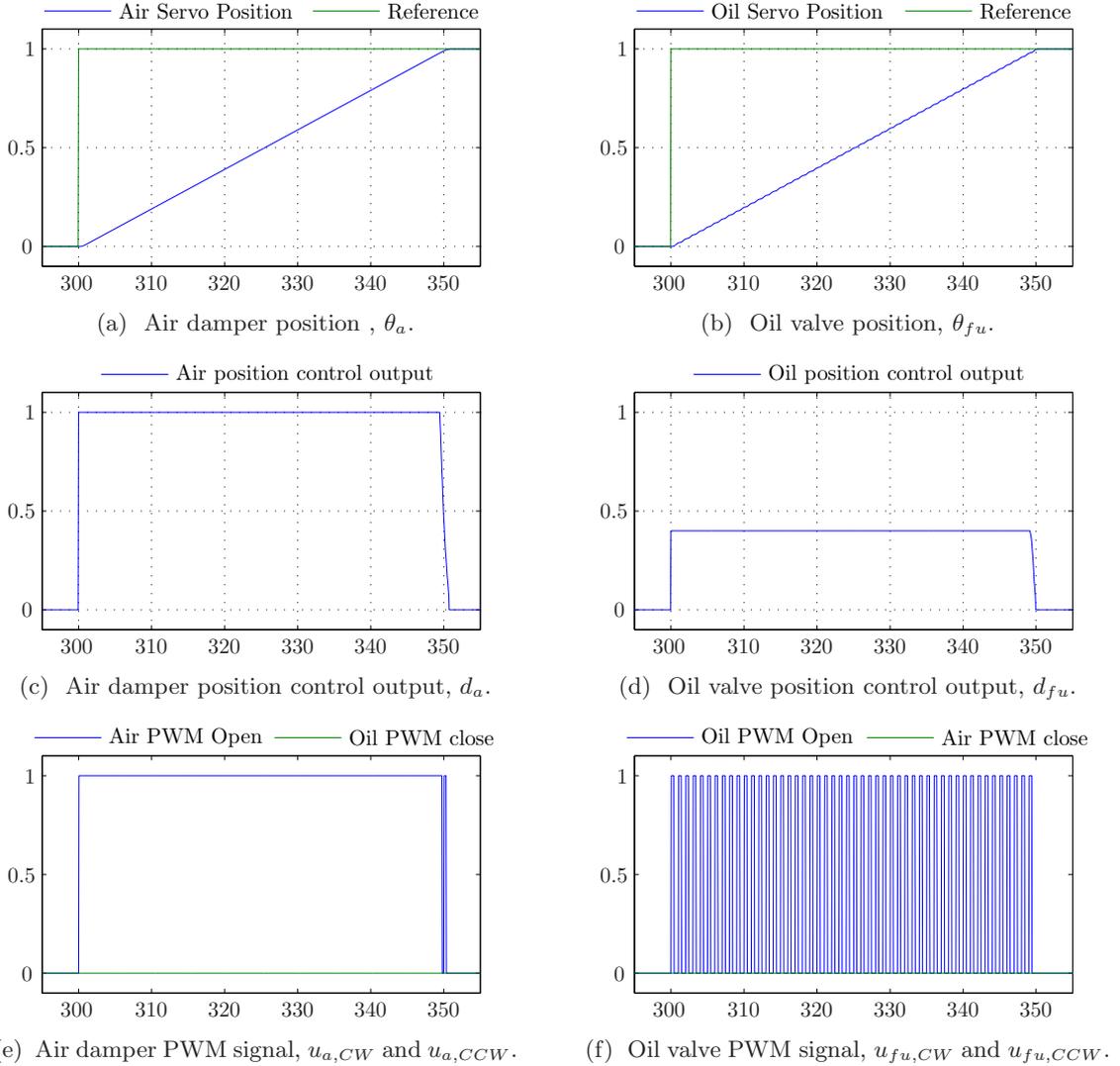


Figure 4.7: Oil valve and air damper response with step in reference at time, $t = 300$, with PWM saturation compensation.

responses, caused by the reduced duty cycle of the fast valve. This solution will of course result in a new problem if the curveband is non-linear, because the one valve then must travel longer than the other, within the same time span. This is not possible with this solution because the valves have equal responses. Later in Chapter 6 a solution to this issue will be presented.

All the inner-loop controllers are now designed, but prior to designing the outer-loop controllers, the inner-loop controller designs are verified by simulation in Simulink in MATLABTM.

4.1.5 Parameter Estimation and Verification of Inner-loop Controllers

In this subsection the designed inner-loop controllers are verified using simulations. The required model parameters are found prior to implementing the models in Simulink together with the designed inner-loop controllers.

Throughout the previous controller designs, only a few model parameters is required to design the controllers and most of them are estimated in the respective modelling sections. However, the derived models are verified against measurement data from a previous boiler setup, with a different burner unit and other valves. Thus the parameters of the present boiler setup are different and the present parameters are applied during the controller verification. The previous estimated parameters and the present model parameters are listed in Table 4.1. The three

Parameter	Estimated Value	Present Value	Unit
τ_{fw}	3.69	4	s
ψ_{fw}	N/A	[0.00024 0.009952 0.310675]	-
K_{fu}	15.17	5.52	$\frac{\%}{s}$
K_a	15.17	1.82	$\frac{\%}{s}$

Table 4.1: Estimated and present model parameters used for inner-loop controller verification.

inner-loop controllers are verified in the following.

Feed Water Actuator Controller Verification

A simulation of the feed water actuator controller is shown in Figure 4.8 and includes both the feed-forward and feedback controllers.

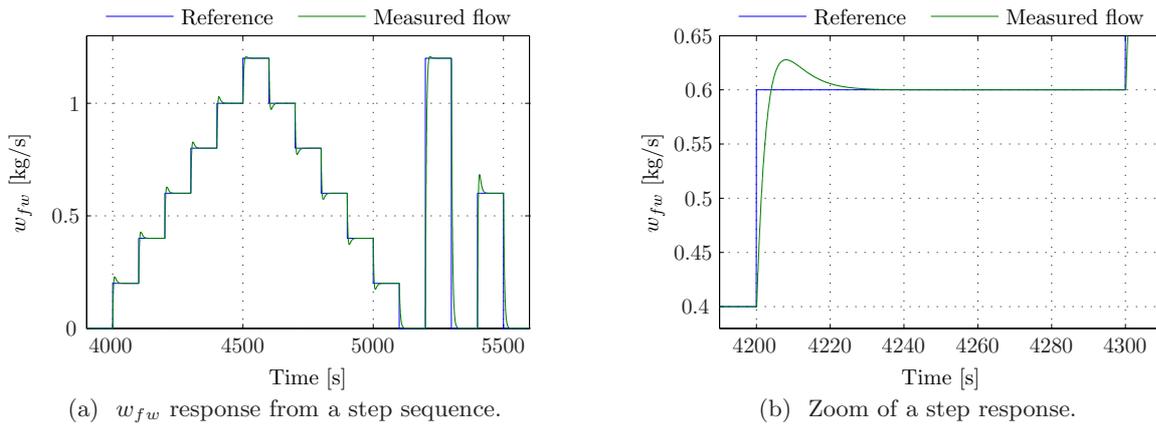


Figure 4.8: Simulation of feed water actuator controllers.

Figure 4.8a shows how the controllers perform from small and large changes in the reference, and the result is satisfactory. To further examine the controller performance, Figure 4.8b shows a zoom of a step response. The designed controllers leads to an overshoot, caused by the integrator

in the PI-controller, because the dominating dynamics are in the sensor. It is the error from the measured response and not the actual feed water flow, with a fast response, which is integrated, and thus the overshoot is considered acceptable.

Besides the overshoot, the desired fast closed loop response is obtained with a time constant of approximately 4 s as designed according to the sensor dynamics.

Oil Valve and Air Damper Controller Verification

A simulation of the oil valve and air damper controllers is shown in Figure 4.9.

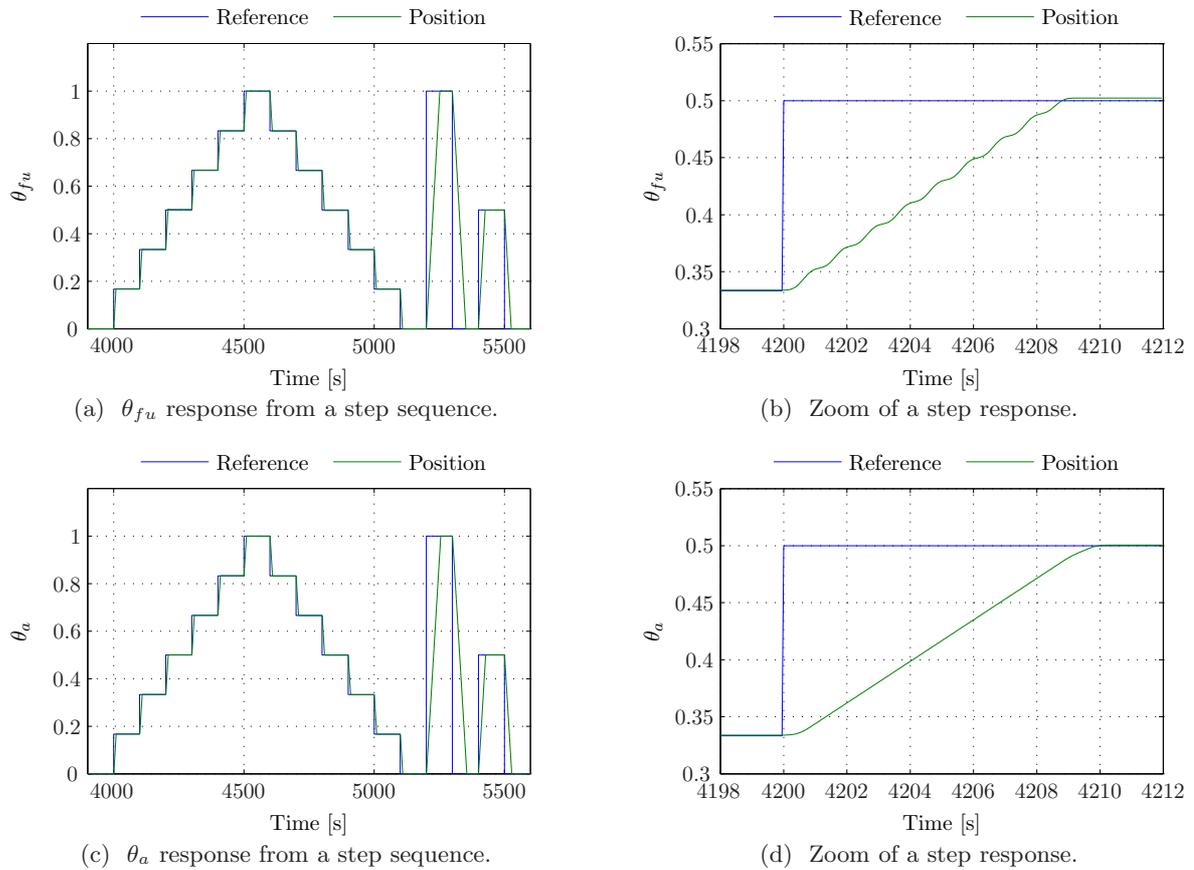


Figure 4.9: Verification of oil valve and air damper position controllers.

As expected the two actuator loops have almost identical responses both for small and large reference changes, in spite of the oil valve is three times faster as the air damper, as given in Table 4.1. Looking at Figure 4.9b and 4.9d it is obvious that there is two minor deviations in the two actuator responses.

Comparing the two figures, the slope of the controlled oil actuator position is slightly steeper than the controlled air actuator position. This is the result of the chosen quantization in the PWM and the normed constant duty cycle for the fastest valve, when saturated. If a higher quantization in the PWM had been designed, a more accurate PWM signal would be generated,

and the slopes of the oil valve and air damper positions would be identical.

The second deviation of the two actuator loops is the waving position of the oil valve. This is again caused by the normed and constant duty cycle of the fast oil valve when saturated, as opposed to the air valve duty cycle of 100 %.

All the inner-loop controllers show satisfactory performance according to the design. By use of the feed water flow controller together with an oil valve and air damper position controller, the inputs to the outer-loop controllers are better defined, which should improve the performance of the outer-loop controllers. In the following section, the outer-loop controllers are designed and verified.

4.2 Outer-loop Controller Design

This section describes the outer-loop controller design of the Mission OSTM boiler system. According to [Solberg, 2008], the Mission OSTM boiler system is essentially linear. Thereby for simplicity, a control model is derived as a linearized version of the non-linear model described in Chapter 3. SISO control is applied to simplify self-tuning of the controllers. Furthermore, the structure for steam pressure and water level control is chosen as proposed by [Andersen and Jørgensen, 2007] and thus only shortly described in this section.

The control application is divided into the following; steam pressure, water level and oxygen level control. Common for the steam pressure and water level controller designs is an analytical linearisation of the non-linear models. The oxygen level controller design is based on a mathematical linearised oxygen model.

4.2.1 Steam Pressure Control

This subsection contains a derivation of a linear control model, followed by the steam pressure controller design and finally the closed loop system response in shortly analysed.

Steam pressure control model

The steam pressure control model is derived from the energy balance, Equation (3.51) on page 26, and is repeated here:

$$\begin{aligned} (\rho_w h_w - \rho_s h_s) \dot{V}_w + \left(m_w \frac{\partial h_w}{\partial p_s} + m_{sb} \frac{\partial h_s}{\partial p_s} + m_s \frac{\partial h_s}{\partial p_s} + m_m c_m \frac{\partial T_{sat}}{\partial p_s} \right) \dot{p}_s \\ = h_{fw} w_{fw} - h_s(p_s) w_s + \eta_{fu} \beta_{fu} w_{fu} . \end{aligned} \quad (4.22)$$

The energy balance equation is simplified to a linearised steam pressure control model as follows. From examining the first term $(\rho_w h_w - \rho_s h_s) \dot{V}_w$, the ratio $\frac{\rho_w h_w}{\rho_s h_s}$ is calculated, using the values in Table B.2 in Appendix B. The ratio is calculated to 56, implying that the term $\rho_s h_s$ is 56 times smaller than $\rho_w h_w$ and thus the term $\rho_s h_s \dot{V}_w$ is neglected. The remaining part $\rho_w h_w \dot{V}_w$ can be substituted using the overall mass balance equation:

$$\frac{d(m_w + m_{sb} + m_s)}{dt} = w_{fw} - w_s . \quad (4.23)$$

In the mass balance the term m_w is greater than $m_{sb} + m_s$ and thus Equation (4.23), multiplied by h_w , reduces to:

$$\dot{m}_w h_w = w_{fw} h_w - w_s h_w . \quad (4.24)$$

Substituting this into Equation (4.22) and rearranging with respect to the output \dot{p}_s and the inputs \dot{w}_{fw} , \dot{w}_s and \dot{w}_{fu} gives:

$$\dot{p}_s = \frac{h_s - h_w}{\left(m_w \frac{\partial h_w}{\partial p_s} + m_{sb} \frac{\partial h_s}{\partial p_s} + m_s \frac{\partial h_s}{\partial p_s} + m_m c_m \frac{\partial T_{sat}}{\partial p_s} \right)} \left(\frac{h_{fw} - h_w}{h_s - h_w} w_{fw} + \frac{\eta_{fu} \beta_{fu}}{h_s - h_w} w_{fu} - w_s \right) . \quad (4.25)$$

In this expression the cross coupling from the feed water flow is examined. Again using the values from Appendix B, the ratio $\frac{h_{fw} - h_w}{h_s - h_w}$ is calculated to approximately -0.3. Even though this is only a factor 3 smaller than the contribution from the steam flow, the cross coupling is ignored and the term dependent on w_{fw} is disregarded.

Finally the fuel flow is assumed to be linear dependent on the fuel valve position, and thereby the final steam pressure control model is given as:

$$\begin{aligned} \dot{p}_s &= \frac{h_s - h_w}{\underbrace{\left(m_w \frac{\partial h_w}{\partial p_s} + m_{sb} \frac{\partial h_s}{\partial p_s} + m_s \frac{\partial h_s}{\partial p_s} + m_m c_m \frac{\partial T_{sat}}{\partial p_s} \right)}_{\beta_{ps}}} \left(\underbrace{\frac{\eta_{fu} \beta_{fu} k_{fu}}{h_s - h_w}}_{\alpha_{ps}} \theta_{fu} - w_s \right) \\ \dot{p}_s &= \beta_{ps} (\alpha_{ps} \theta_{fu} - w_s) . \end{aligned} \quad (4.26)$$

Equation (4.26) is shown as a block diagram in Figure 4.10.

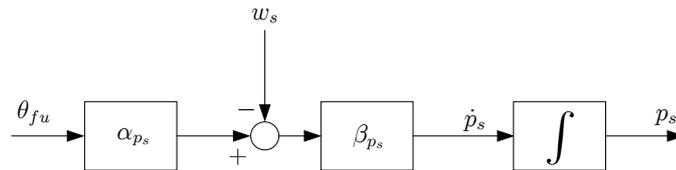


Figure 4.10: Steam pressure control model.

By considering the steam flow, w_s , to be an unmeasurable disturbance it can be set to zero. Thus, the steam pressure control loop can be represented by the block diagram in Figure 4.11. As the inner control loop is neglected, the steam controller is designed adequately slow, to ensure that the oil valve position reference, determined by the steam pressure controller, can be obtained by the inner-loop.

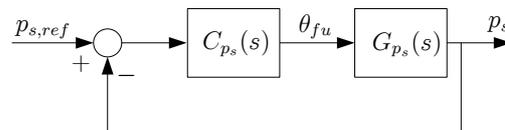


Figure 4.11: Block diagram of the steam pressure control loop.

The system transfer function for the linear steam pressure control model, $G_{p_s}(s)$, equals:

$$G_{p_s}(s) = \frac{\gamma_{p_s}}{s} , \quad (4.27)$$

where γ_{p_s} according to Figure 4.10 is given by:

$$\gamma_{p_s} = \alpha_{p_s} \beta_{p_s} . \quad (4.28)$$

The derived linear steam pressure control model, given by Equation (4.26), is verified prior to using the control model for controller design. Appendix D contains a description of parameter estimation and a verification, that approves the steam pressure control model for controller design.

Steam pressure controller design

The steam pressure controller is designed as a PI-controller on the form:

$$C_{p_s}(s) = k_{p_s,p} \frac{s + \frac{1}{T_{p_s,i}}}{s} . \quad (4.29)$$

This gives a second order open loop system, $OL(s) = C_{p_s}(s)G_{p_s}(s)$, for which a stability criterion can be expressed from magnitude and phase as [Franklin et al., 2006, p. 353]:

$$|OL(j\omega_{p_s})| = 1 \quad (4.30)$$

$$\angle OL(j\omega_{p_s}) = -\pi + \phi_{p_s,m} . \quad (4.31)$$

Based on the stability criterion, the PI-controller is designed with control parameters as given below:

$$T_{p_s,i} = \frac{\tan(\phi_{p_s,m})}{\omega_{p_s,c}} \quad (4.32)$$

$$k_{p_s,p} = \frac{\omega_{p_s,c}^2}{\gamma_{p_s} \sqrt{\frac{1}{T_{p_s,i}^2} + \omega_{p_s,c}^2}} , \quad (4.33)$$

where the cross-over frequency, $\omega_{p_s,c}$, the phase margin, $\phi_{p_s,m}$, and the integrator gain, γ_{p_s} , is to be determined.

The cross-over frequency is chosen through a worst case consideration of the time constant for the fuel actuator system. To ensure that the outer-loop is five times slower than the inner-loop, the cross-over frequency can be expressed as:

$$\omega_{p_s,c} = \frac{1}{5\tau_{fu}} , \quad (4.34)$$

where τ_{fu} expresses the time constant for the fuel actuator system. As the fuel actuators are PWM controlled and moves with constant speed, it is difficult to specify the time constant for the closed loop fuel actuator. Thus τ_{fu} is chosen as the timespan, where the fuel valve travels 10 %. This seems reasonable, as the change in valve stroke is small during normal operation. By using this definition of τ_{fu} to calculate the bandwidth of the steam pressure control loop, it is the fuel actuator that defines the overall boiler performance. Thus the design choice of 10 % might be reconsidered if the fuel actuator is either very fast or very slow.

To ensure stability even if the real system differs from the control model, a phase margin of 60 degrees is suitable. However, the fact that the fuel actuator is five times faster than the outer-loop system, gives rise to an influence on the outer-loop system, corresponding to a phase change of 11 degrees. Thereby, the phase margin of the outer-loop steam pressure controller, $\phi_{p_s,m}$, is chosen to 71 degrees.

Steam pressure closed loop analysis

At this point, a steam pressure controller is designed, to depend only on the measurable steam pressure integrator gain, γ_{p_s} , and the approximated time constant, τ_{f_u} . The integrator gain can be found using well known methods as parameter estimation or as a part of a self-tuning algorithm.

Using the estimated parameters for the fuel actuator, listed in Table 3.4 on Page 40, and the linear boiler model in Table D.1 on Page D1, the steam pressure controller parameters are calculated to:

$$T_{p_s,i} = 9.572 \quad \text{and} \quad k_{p_s,p} = 15 \cdot 10^{-3} . \quad (4.35)$$

This gives a closed loop system resulting in a step response and frequency response as shown in Figure 4.12a and 4.12b respectively.

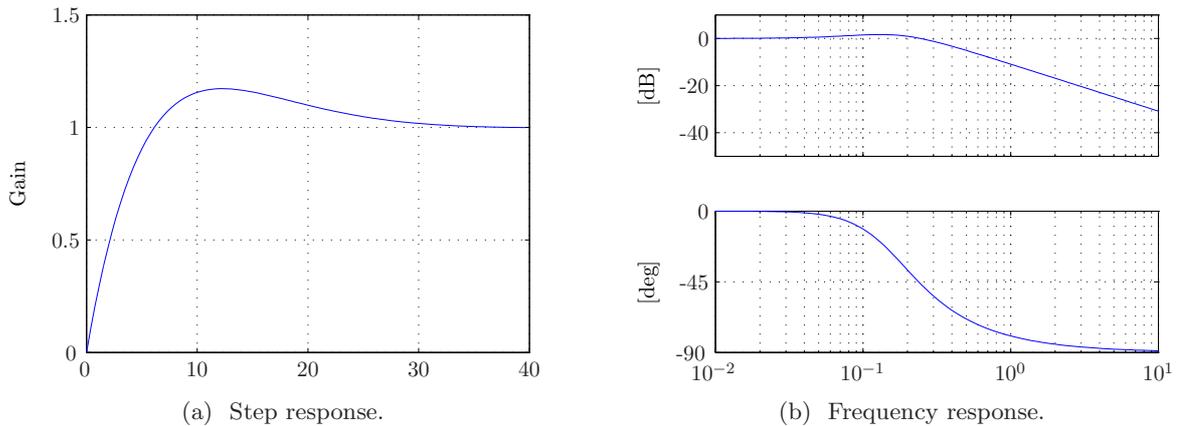


Figure 4.12: Step and frequency response for the closed loop steam pressure control loop.

These responses show that a fast response is obtained with a closed loop bandwidth of $\omega_{p_s,CL} = 0.385$ rad/s and an overshoot of approximately 17 %.

4.2.2 Water Level Control

This subsection contains first a derivation of a linear control model, followed by the water level controller design and finally the closed loop system response is shortly analysed.

Water level control model

The water level control model is derived from the mass balance equation for the water/steam part. From the output equation for the non-linear boiler model, Equation (3.62) derived in

Chapter 3, it is seen that the water level depends on the volume of water and steam bubbles, described by the equations given below:

$$\left(V_w \frac{\partial \rho_w}{\partial p_s} + V_{sb} \frac{\partial \rho_s}{\partial p_s} \right) \dot{p}_s + \rho_w \dot{V}_w + \rho_s \dot{V}_{sb} = w_{fw} - \frac{1}{t_{sb}} \rho_s V_{sb} \quad (4.36)$$

$$\left((V_{tot} - V_w - V_{sb}) \frac{\partial \rho_s}{\partial p_s} \right) \dot{p}_s - \rho_s \dot{V}_w - \rho_s \dot{V}_{sb} = \frac{1}{t_{sb}} \rho_s V_{sb} - w_s . \quad (4.37)$$

To simplify the non-linear equations, some of the terms are further analysed.

The partial fractions are calculated from values in a steam table [Schmidt, 1989, p. 32], resulting in the ratio $\frac{V_w \frac{\partial \rho_w}{\partial p_s} + V_{sb} \frac{\partial \rho_s}{\partial p_s}}{\rho_w} \approx 0.0117$ and thus the cross coupling from the steam pressure is ignored. Likewise the ratio $\frac{(V_{tot} - V_w - V_{sb}) \frac{\partial \rho_s}{\partial p_s}}{\rho_s} \approx 0.14$ and thus the first terms in the equations above can be neglected, and the equations reduces to:

$$\rho_w \dot{V}_w + \rho_s \dot{V}_{sb} = w_{fw} - \frac{1}{t_{sb}} \rho_s V_{sb} \quad (4.38)$$

$$-\rho_s \dot{V}_w - \rho_s \dot{V}_{sb} = \frac{1}{t_{sb}} \rho_s V_{sb} - w_s . \quad (4.39)$$

Generally the neglect of the cross coupling means that shrink and swell is not incorporated in the linearised boiler model. An expression for the change in water volume can be derived, by combining Equation (4.38) and Equation (4.39), which gives:

$$\dot{V}_w = \frac{1}{\underbrace{\rho_w - \rho_s}_{\beta_{L_w}}} (w_{fw} - w_s) . \quad (4.40)$$

Remaining is to derive an expression for the steam bubbles volume contribution, to the water level, L_w . To do this, the approximation of the mass flow from steam bubbles to steam, Equation (3.56) on page 27, is examined:

$$w_{sb \rightarrow s} = \frac{1}{t_{sb}} \rho_s V_{sb} . \quad (4.41)$$

In steady state, the mass flow from steam bubbles to steam equals the steam flow, $w_{sb \rightarrow s} = w_s$. Inserting this and rearranging gives the expression for the steam bubbles level, L_{sb} :

$$L_{sb} = \frac{t_{sb}}{\underbrace{\rho_s A_w}_{\alpha_{L_w}}} w_s . \quad (4.42)$$

The total water level control model reduces to a state space model as follows:

$$\dot{V}_w = \begin{bmatrix} \beta_{L_w} & -\beta_{L_w} \end{bmatrix} \begin{bmatrix} w_{fw} \\ w_s \end{bmatrix} \quad (4.43)$$

$$L_w = \frac{1}{A_w} V_w + \begin{bmatrix} 0 & \alpha_{L_w} \end{bmatrix} \begin{bmatrix} w_{fw} \\ w_s \end{bmatrix} . \quad (4.44)$$

Equation (4.43) and (4.44) is shown as a block diagram in Figure 4.13.

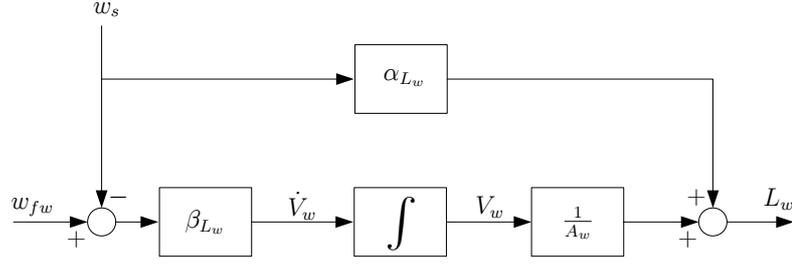


Figure 4.13: Water level control model.

Furthermore, the derived linear water level control model, given by Equation (4.43) and (4.44), is verified prior to using the control model for controller design. This is likewise documented in Appendix D, which contains a description of parameter estimation and a verification, that approves the water level control model for controller design.

The water level control loop is based on the water level control model, Equation (4.43) and (4.44). However, the model includes the steam flow, w_s , which is seen as an unmeasured disturbance. Thus, the placement of the disturbance is of no importance, and α_{L_w} is neglected. Thereby, the water level control loop is represented by a block diagram as in Figure 4.14.

As the inner control loop is neglected, the water level controller is designed adequately slow, to ensure that the feed water valve position reference, determined by the water level controller, can be obtained by the inner-loop.

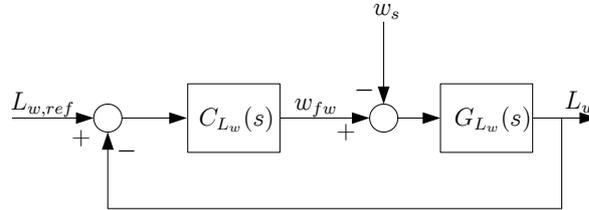


Figure 4.14: Block diagram of the water level control loop.

The system transfer function for the linear water level control model, $G_{L_w}(s)$, equals:

$$G_{L_w}(s) = \frac{\beta_{L_w,1}}{s}, \quad (4.45)$$

where

$$\beta_{L_w,1} = \frac{\beta_{L_w}}{A_w} = \frac{1}{\rho_w A_w}, \quad (4.46)$$

and thus only dependent on the boiler dimensions.

Water level controller design

The water level controller is designed to have a structure similar to the steam pressure controller, a PI-controller on the form:

$$C_{L_w}(s) = k_{L_w,p} \frac{s + \frac{1}{T_{L_w,i}}}{s}. \quad (4.47)$$

Based on the same stability criterion, as for the steam pressure controller given by Equation (4.30) and (4.31), the PI-controller is designed with control parameters as given below.

$$T_{L_w,i} = \frac{\tan(\phi_{L_w,m})}{\omega_{L_w,c}} \quad (4.48)$$

$$k_{L_w,p} = \frac{\omega_{L_w,c}^2}{\beta_{L_w,1} \sqrt{\frac{1}{T_{L_w,i}^2} + \omega_{L_w,c}^2}}, \quad (4.49)$$

where the cross-over frequency, $\omega_{L_w,c}$, and the phase margin, $\phi_{L_w,m}$, is to be determined.

The cross-over frequency is chosen from a worst case consideration of the time constant for feed water actuator system. To ensure that the outer-loop is five times slower than the inner-loop, the cross-over frequency can be expressed as:

$$\omega_{L_w,c} = \frac{1}{5\tau_{fw}}, \quad (4.50)$$

where τ_{fw} is the time constant for the feed water actuator system and the cross-over frequency, $\omega_{L_w,c}$, is in [rad/s].

To ensure stability even if the real system differs from the control model, a phase margin of 60 degrees is suitable. However, the fact that the feed water actuator is five times faster than the outer-loop system, gives rise to an influence on the outer-loop system corresponding to a phase change of 11 degrees. Thereby, the phase margin of the outer-loop water level controller, $\phi_{p_s,m}$, is chosen to 71 degrees.

Water level closed loop analysis

At this point, a water level controller has been designed to depend on the dimensions of the boiler and the time constant of the feed water actuator. No consideration has been taking, regarding the noisy measurements of the unavoidable fluctuating water level. The noisy measurements might influence the bandwidth of the closed loop, resulting in an even slower water level controller. Previously projects by [Andersen and Jørgensen, 2007] and [Solberg and Hvistendahl, 2004] treated a similar or the same Mission OSTM boiler system and documented, that due to the fluctuating water level the maximum obtainable system response is with a time constant of approximately 150 s.

Thereby the demand for the closed loop bandwidth of the water level control loop is redefined as:

$$\omega_{L_w,c} = \frac{1}{150}. \quad (4.51)$$

Using the parameters for the linear boiler model, given in Table D.1 on Page D1, and the surface area for the water in Table B.4 on Page B2, the water level controller parameters are calculated to:

$$T_{L_w,i} = 435.6 \quad \text{and} \quad k_{L_w,p} = 5.68. \quad (4.52)$$

This gives a closed loop system resulting in a step response and frequency response as shown in Figure 4.15a and 4.15b respectively.

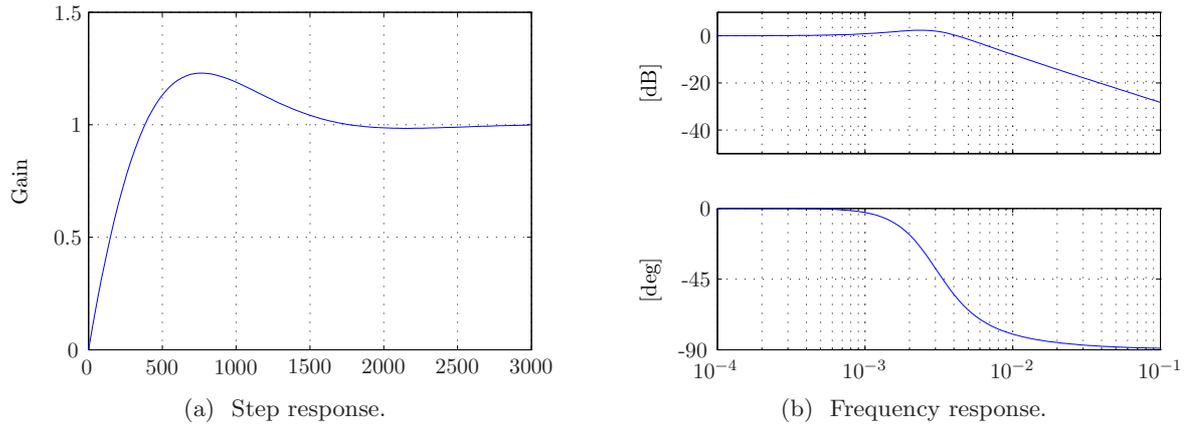


Figure 4.15: Step and frequency response for the closed loop water level control loop.

These figures show that the obtained response is slow compared to the steam pressure closed loop, which is expected according to the design. The closed loop bandwidth is $\omega_{L_w,CL} = 0.0059$ rad/s and the overshoot is approximately 23 %.

4.2.3 Oxygen Level Control

The purpose of the oxygen level controller is to maintain a steady oxygen level in the combustion, using a complementary feedback controller to a feed-forward curveband, as illustrated in Figure 4.16. The fuel actuator loops have already been designed, so it is assumed that the fuel and air flows are well defined relative to the oil valve and air damper positions.

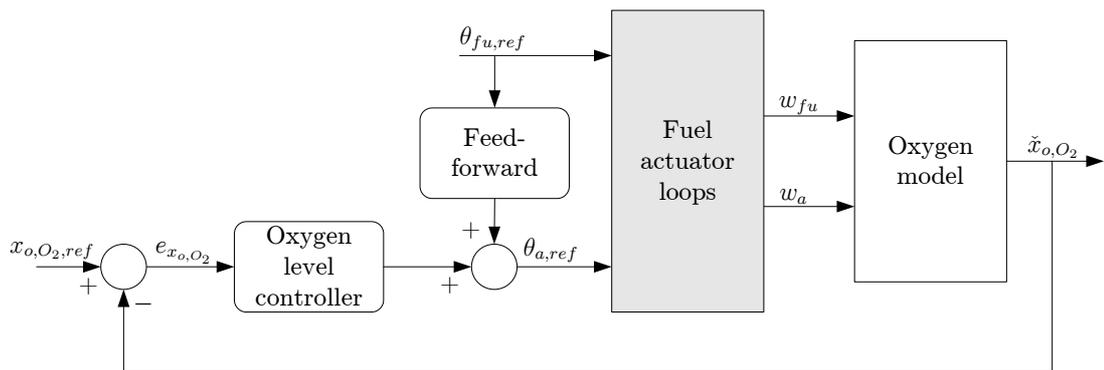


Figure 4.16: Block diagram of oxygen control using a curveband as feed-forward from oil valve position to air damper position and measured oxygen level as feedback.

The next two subsections contain a description of how the feed-forward and feedback controllers are designed.

Feed-forward Controller

The existing oxygen controller structure at AI consists only of a feed forward controller, or ratio controller, expressing the ratio between oil valve and air damper positions at a specified oxygen level. The ratio controller is referred to as a curveband. For each boiler system the curveband is manually adjusted by AI as the curveband changes by small alterations in the boiler setup, e.g. if a new burner unit is installed, valves are replaced or oil pressure is changed etc. Due to these issues the same practical approach, for implementing the feed-forward controller, is taken in this project.

The concept of determining the curveband is shown in Figure 4.17. When the boiler is operative, the oil valve and air damper positions can be manually adjusted. This way several data sets of oil valve and air damper positions is obtained and a polynomial can be fitted to a polynomial, or curveband, at a desired oxygen level. The fitted polynomial is used as curveband to control the oxygen level of the combustion.

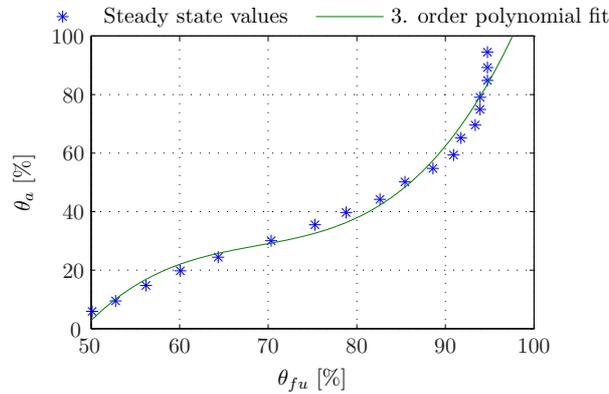


Figure 4.17: Curveband fitted to oil valve and air damper positions at a specified oxygen level.

The polynomial fit of the curveband in Figure 4.17 is a 3'th order non-linear polynomial, given by the general form:

$$\theta_a = f(\theta_{fu}) = a_{O_2,3} \theta_{fu}^3 + a_{O_2,2} \theta_{fu}^2 + a_{O_2,1} \theta_{fu} + a_{O_2,0} . \quad (4.53)$$

The feed-forward curveband should ideally secure a steady oxygen level, but different impacts could lead to a deviation of the oxygen level. Thus a feedback controller for the oxygen level is desired.

Oxygen Level Feedback Controller

If for some reason the curveband is inadequate or another oxygen level, different from the fitted curveband, is preferred, it is desirable to add a feedback controller as complementary to the feed-forward controller. The inadequacy of the curveband could e.g. be caused by use of another fuel type with a different viscosity resulting in another fuel flow or if the polynomial fit is not sufficient in the entire operating range.

Different from the steam pressure and water level controllers, the oxygen level controller is chosen to be an I-controller. Due to the fast response through the curveband it is not necessary to apply a proportional gain, but the integrator gain is required to eliminate any steady state errors.

Oxygen level control model

In order to design the I-controller, the non-linear oxygen model is linearised using a first order Taylor series expansion, which is shown in Appendix E, and the transfer function expressing the linear oxygen model is given by Equation (E.5).

In addition to the linear oxygen model, the gain of the air damper actuator loop has to be included, as illustrated in Figure 4.16. In the parameter estimation of the air damper, on Page 42, the gain is estimated to 0.009, which together with Equation (E.5) gives the final linear transfer function:

$$\frac{\Delta x_{o,O_2}}{\Delta w_a}(s) = \frac{0.009 \left(\frac{k_1 k_4 \bar{w}_{fu} + k_2 k_3 \bar{w}_{fu}}{k_3^2 \bar{w}_a^2 + k_4^2 \bar{w}_{fu}^2 + 2k_3 k_4 \bar{w}_a \bar{w}_{fu}} \right)}{\tau_{x_o,O_2} s + 1}, \quad (4.54)$$

where the constants, $k_1 \dots k_4$, are given in Appendix E.

Root Locus design method

As argued the feedback controller is chosen as a simple I-controller on the form:

$$C_{O_2}(s) = \frac{K_{O_2,i}}{s}. \quad (4.55)$$

To determine the integrator gain, $K_{O_2,i}$, for the I-controller, the Root Locus design method is used to obtain a desired closed loop response, on the basis of the open loop response given by:

$$G_{O_2OL} = C_{O_2}(s) \frac{\Delta x_{o,O_2}}{\Delta w_a}(s). \quad (4.56)$$

By adjusting the integrator gain, $K_{O_2,i}$, the closed loop poles are moved and the desired dynamic response can be obtained. Prior to determining the integrator gain, the desired closed loop response needs to be defined. The bandwidth of the oxygen level control loop has to be lower than the bandwidth of the oxygen model, defined by the time constant τ_{x_o,O_2} . Hence, by choosing the desired time constant of the closed loop oxygen level control to $5\tau_{x_o,O_2}$, a relatively slow but satisfactory closed loop response is obtained. From the designed margin with a bandwidth five times lower than the oxygen model dynamics, a possible larger time constant for the oxygen model is of less importance.

Oxygen level closed loop analysis

At this point, an oxygen level controller has been designed to depend on the relation between fuel and air flow, in form of a curveband defined by the polynomial coefficients:

$$\psi_{O_2} = [a_{O_2,3} \ a_{O_2,2} \ a_{O_2,1} \ a_{O_2,0}], \quad (4.57)$$

and the time constant, τ_{x_o,O_2} , for the oxygen model, used to calculate the gain for the I-controller. As mentioned, the curveband must be identified from a practical approach in form of a self-tuning algorithm. The time constant for the oxygen model is however not possible to identify from self-tuning due to the missing flow sensor, hence the theoretical value for τ_{x_o,O_2} from Equation (3.30) is used as reference.

Using Root Locus in MATLABTM and adjusting the integrator gain until the desired system response is obtained, the integrator gain becomes:

$$K_{O_2,i} = 20.27 . \quad (4.58)$$

Using the calculated integrator gain the closed loop system, when ignoring the feed-forward curveband, gives the step response and frequency response as shown in Figure 4.18a and 4.18b respectively.

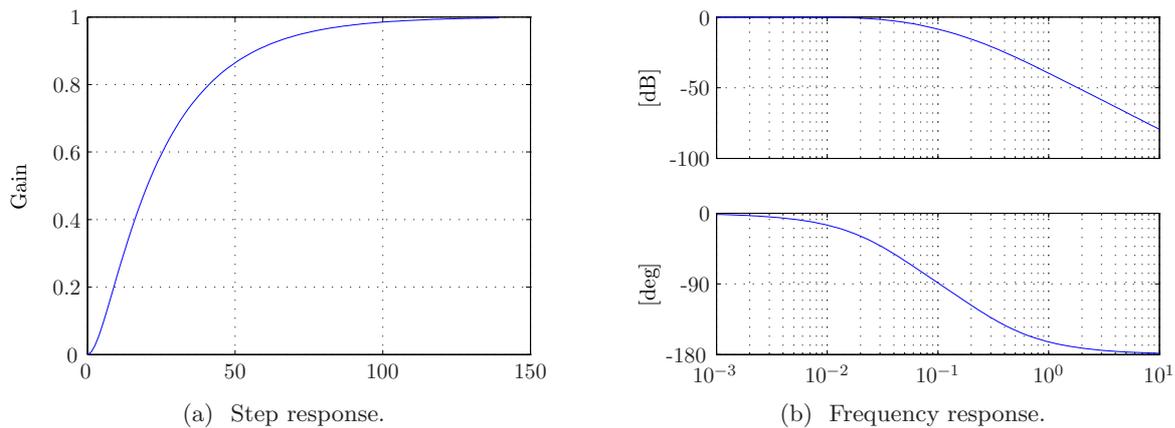


Figure 4.18: Step and frequency response for the closed loop oxygen level control loop.

The figures show that the obtained closed loop response is satisfactory according to the design. The closed loop bandwidth is $\omega_{O_2,CL} = 0.043$ rad/s and there is no overshoot because the obtained response is relatively slow.

Having designed all the outer-loop controllers their performance must be simulated to verify the designs.

4.2.4 Simulation of Outer-loop Controllers

To evaluate the performance of the designed controllers, a simulation, using the non-linear model to obtain the most reliable responses, is carried out.

The outer-loop controllers are simulated by a large step and a small step in steam flow, to simulate two different scenarios from a sudden change of steam consumption. The simulated closed loop system responses and the simulated inner-loop actuator responses are shown and commented in the following.

In the simulation of the steam pressure and water level controller performance, weighted white noise is added to the sensor measurements, to imitate the real system best possible. As previously discussed it is the water level measurements that limits the controller performance of the water level. The added weighted white noise is an approximate replica from the actual sensor measurements in the test center and specified as follows.

Sensor noise specification

For the steam pressure measurements the noise is defined as a zero mean Gaussian distributed random signal, $\mathcal{N}(0, 6.26 \cdot 10^6)$. The white noise is passed through a low pass filter given by:

$$p_{s,lowpass}(s) = \frac{1}{2\pi \cdot s + 1} . \quad (4.59)$$

The white noise together with the low pass filter constitutes the sensor noise for the steam pressure measurements.

For the water level measurements the noise is likewise defined as a zero mean Gaussian distributed random signal but with different characteristics, $\mathcal{N}(0, 0.0025)$. The white noise is passed through a low pass filter given by:

$$L_{w,lowpass}(s) = \frac{1}{\pi \cdot s + 1} . \quad (4.60)$$

The white noise together with the low pass filter constitutes the sensor noise for the water level measurements.

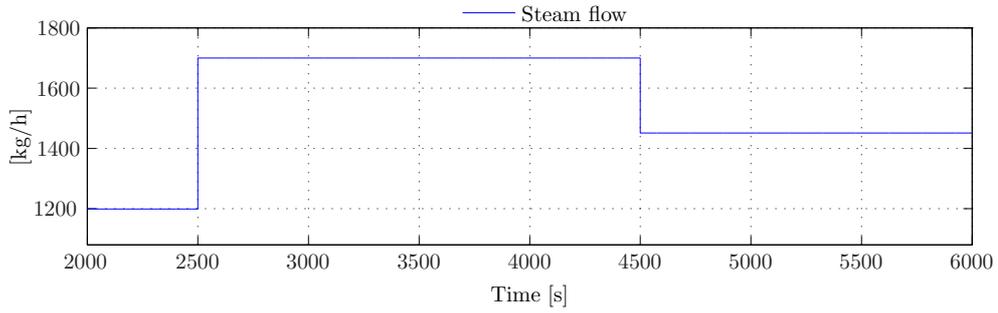
Simulation of Steam Pressure Controller

The simulated steam pressure controller performance is depicted in Figure 4.19 and 4.20. From the applied steps in steam flow, the simulated closed loop response of the steam pressure is shown in Figure 4.19b. As seen, the steam pressure controller is capable of maintaining the steam pressure within a very small deviation from the operating point, when a large step in steam flow is applied. This result is achievable because the fuel actuator system is very fast, in the non-linear simulation model, and thereby the steam pressure controller gives a corresponding high performance.

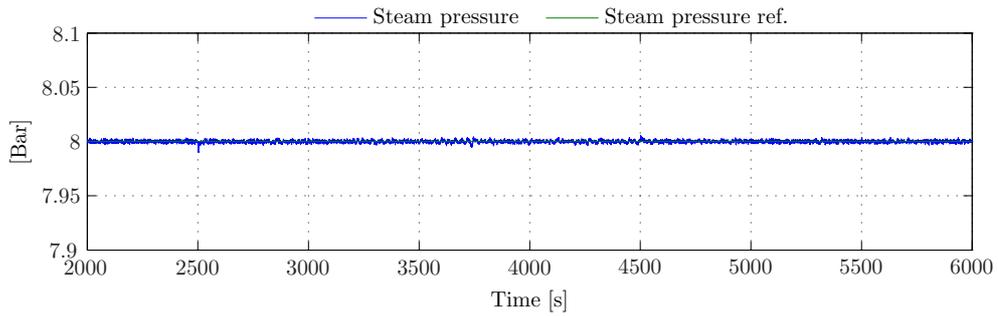
Looking at the inner-loop fuel actuator duty cycle in Figure 4.20b, clearly the steam pressure controller is not violating the oil valve and air damper controllers, as the duty cycle of the oil valve and air damper never saturates. However, when the large step in steam flow is applied, the duty cycle approaches the saturation limit. The resulting fuel actuator responses is shown in Figure 4.20a and the result looks satisfactory.

Simulation of Water Level Controller

In Figure 4.21b the simulated closed loop response of the water level is shown. It is evident that the water level controller is slower than the steam pressure controller, which is expected due to the controller design, affected by the fluctuating water level measurements. The water level controller is however able to keep the water level within a range of ± 2 cm from the operating point, when a small step in steam flow is applied. From the large step the water level deviates approximately ± 3 cm from the operating point, which is considered satisfactory. In Figure 4.21b it is furthermore seen that the water level controller is unable to overcome the shrink and swell phenomenon. When the steam flow increases the swell phenomenon can be seen as the water level shortly rises before decreasing.

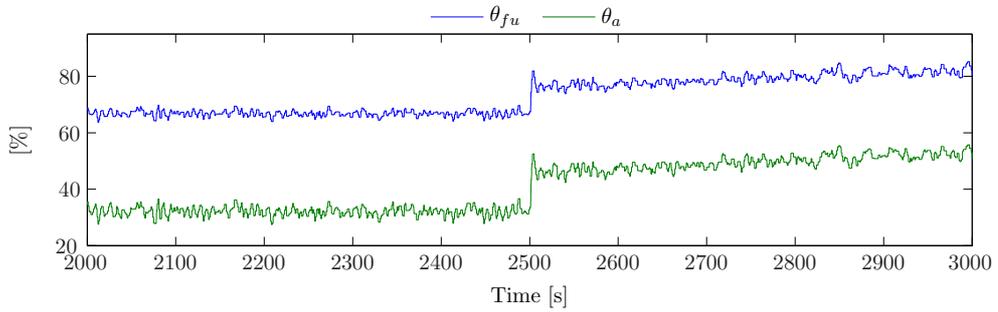


(a) Disturbance in steam flow.

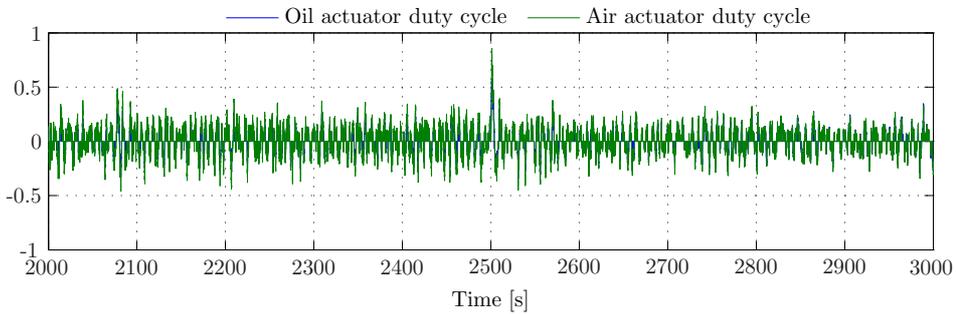


(b) Steam pressure response from disturbance.

Figure 4.19: Simulation of p_s controller for a large and small disturbance variation.



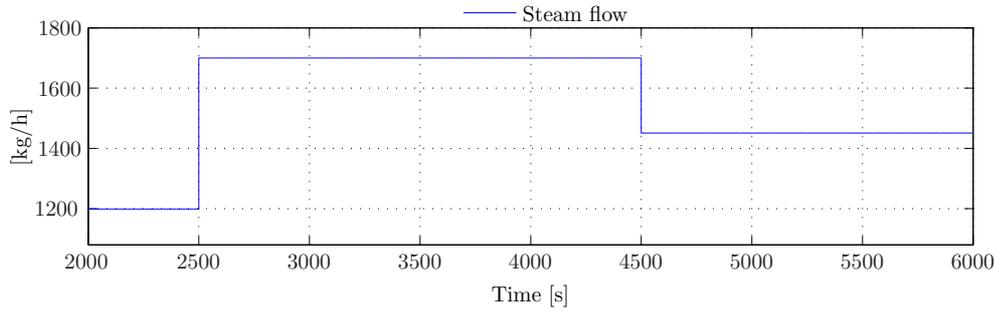
(a) Fuel actuator response from disturbance.



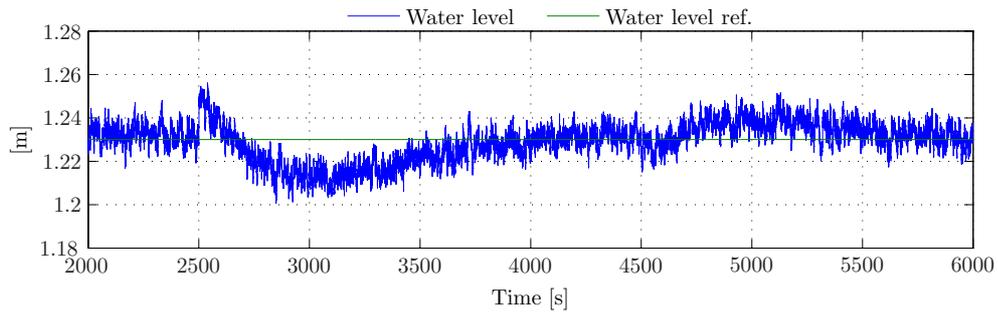
(b) Oil and air actuator duty cycle.

Figure 4.20: Simulation of actuator controllers for a large and small disturbance variation.

As for the steam pressure controller, the inner-loop feed water actuator response is also examined and shown in Figure 4.22a.

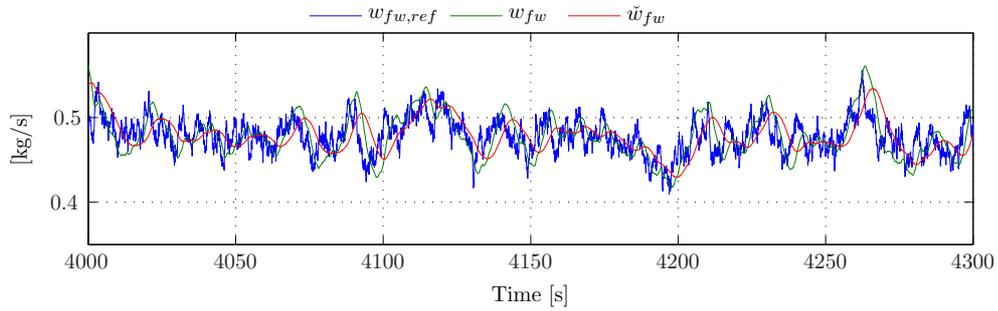


(a) Disturbance in steam flow.

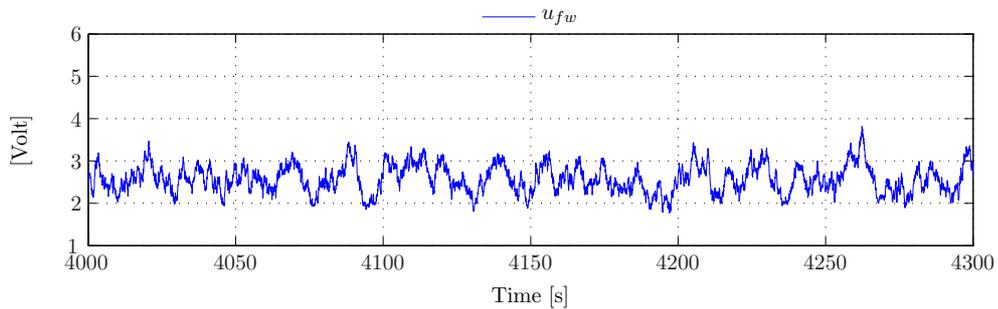


(b) Water level response from disturbance.

Figure 4.21: Simulation of water level performance for a large and small disturbance variation.



(a) Feed water actuator response.



(b) Feed water actuator control signal.

Figure 4.22: Simulation of feed water actuator performance.

Clearly the noisy measurements affects the reference, $w_{fw,ref}$, but compared to the feed water flow, w_{fw} , there is a satisfactory correlation. Thereby the water level controller is not violating

the inner-loop controller, which is consistent to the design of the water level controller. In Figure 4.22a it is furthermore seen that the measured feed water flow, \check{w}_{fw} , is delayed compared to the actual feed water flow, which is caused by the dynamics in the flow sensor. Finally in Figure 4.22b the reference voltage for the feed water actuator is shown. Also here it is seen that the water level controller is not violating the performance of the inner-loop, as the reference voltage resembles the noisy measurements.

Simulation of Oxygen Level Controller

The oxygen controller, consisting of the curveband and I-controller, is also simulated to verify the controller design. In Figure 4.23 the result from a simulation with only the curveband as oxygen controller is shown. The valve positions of the fuel actuator together with the references are shown in Figure 4.23a. The reference for the air damper is defined by a curveband which is fitted for an oxygen level of 5 %. By looking at the resulting oxygen level in Figure 4.23b, it is seen that the controlled oxygen level matches the reference very well.

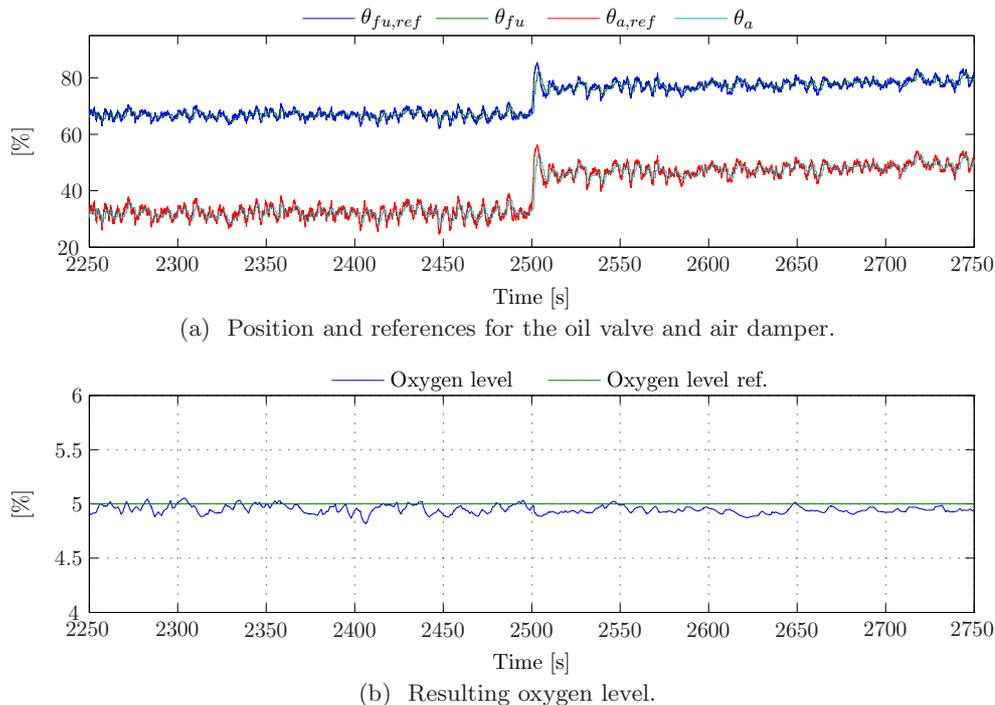
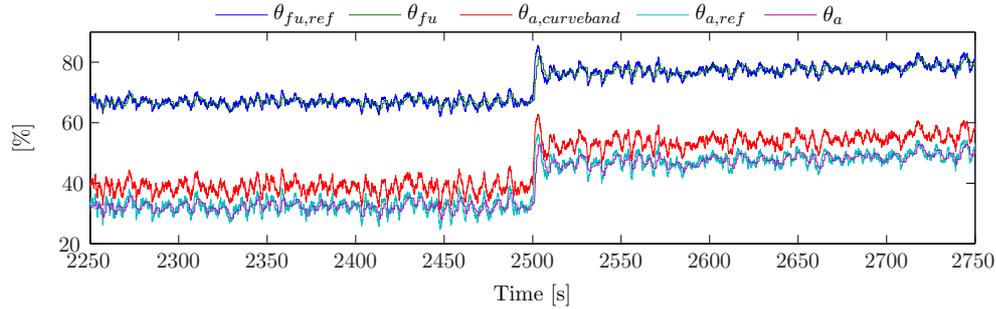


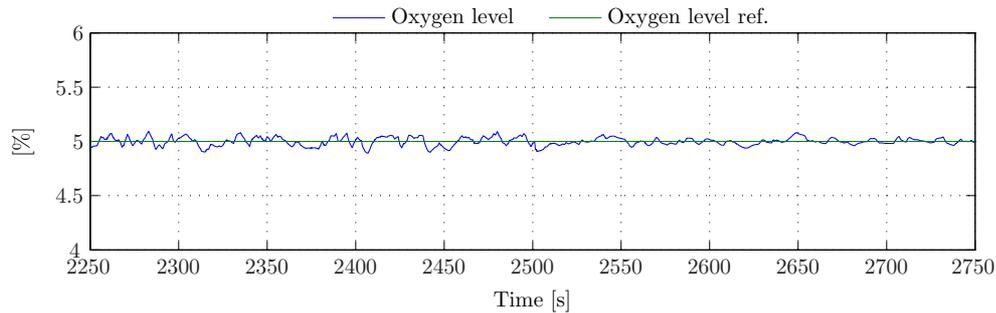
Figure 4.23: Simulation of curveband as oxygen controller.

To illustrate the effect of the feedback controller together with the curveband, Figure 4.24 shows another simulation.

Figure 4.24b shows how the controlled oxygen level still corresponds to the reference value. To illustrate the effect of the I-controller, the different valve references and positions are shown in Figure 4.24a. The curveband reference for the air damper is changed by an offset from the feedback controller, resulting in a total reference for the air damper that is consistent with the desired oxygen level. Thereby the purpose of the complementary feedback controller has the



(a) Position and references for the oil valve and air damper.



(b) Resulting oxygen level.

Figure 4.24: Simulation of curveband and oxygen feedback controller.

intended effect of correcting deviations from the curveband controller.

All the controllers have been designed and verified from non-linear simulation. In the following section a short resume of this chapter is given.

4.3 Controller Resume

A control scheme of three inner-loop and three outer-loop controllers has been presented. The inner-loop controllers are designed to control the feed water flow, the oil valve position and air damper position, all with the purpose of optimising the conditions for the outer-loop controllers, which are the steam pressure, water level and oxygen level controllers.

The feed water flow controller consists of a feed forward controller, to eliminate the non-linearities in the feed water valve, and a PI-controller to obtain fast and accurate response to a given reference. Furthermore a tracking anti wind-up scheme is designed to avoid integrator wind-up. From simulation the feed water controllers show satisfactory performance, with a small overshoot and a fast response.

The oil valve and air damper position controllers are designed in the same way due to similar operation. The position controllers are designed as P-controllers due to an existing integrator effect in the actuators. The actuators are PWM controlled, resulting in a saturation problem if the oil valve and air damper have unequal traveling times. This problem is solved by slowing the fast valve and thereby obtaining equal traveling times for the oil valve and air damper. Furthermore a dead band is designed for each PWM controller to avoid limit cycles. The

designed controllers show satisfactory performance but due to the PWM saturation solution, another problem arises if the curveband is non-linear. This is treated in Chapter 6.

Prior to designing the steam pressure controller, the non-linear boiler model is linearised from an analytical point of view. This results in a control model consisting of only an integrator. Parameters for the simple control model are estimated and the control model is verified. Based on the derived control model, the steam pressure controller is designed as a PI-controller based on a stability criteria in the frequency domain. Apart from the stability criteria the steam pressure controller is designed according to the inner-loop response.

The design of the water level controller is similar to the steam pressure controller design, and includes an analytical linearisation of the non-linear boiler model, resulting in a integrator control model. As for the steam pressure control model, the parameters for the water level control model are estimated and the model is verified. A PI-controller is designed from the same stability criteria but due to the fluctuating water level measurements, the controller design is reevaluated, resulting in the final and slower water level controller.

The final designed controller is the oxygen level controller, consisting of a feed-forward controller in form of a curveband and a slow feedback I-controller. The curveband is used to obtain a fast response from the burner load and is found using a practical approach. The I-controller is added to enable reference control and is designed using the Root Locus design method to obtain a desired closed loop response.

A simulation of the steam pressure, water level and oxygen level controller performances showed promising results. However, a final verification of the performance of the designed controllers is yet to be conducted, due to an emerged defect in the Morpheus control platform, which is still under development at AI. The next chapter describes a self-tuning scheme, designed to automatically tune the controllers for a random Mission OSTM boiler system.

Self-tuning

This chapter describes self-tuning of the controllers for the Mission OSTM boiler system. The self-tuning mainly covers tuning of the controllers for the actuator systems, as the physical dimensions of the boiler is well known. However, as not all of the input flows into the boiler are measurable, the self-tuning will also cover tuning of the steam pressure controller. The self-tuning is designed to be executed only during the start-up phase of the boiler system, and is thus not used as an online adaptive tuning method.

The self-tuning described in this chapter is developed, based on considerations from a previous project done by [Andersen and Jørgensen, 2007]. First, the parameters to identify are specified together with a description of their implementation in the existing start-up sequence. Next the specific algorithms to identify the specified parameters are described together with a verification of each algorithm. Finally the developed self-tuning scheme is evaluated by non-linear simulation, and limitations of the self-tuning scheme are presented.

During the project, the boiler setup at AI's test center has been undergoing several modifications, among others, replacement of burner units and servo motors. Hence the developed self-tuning scheme is tested and verified on a system of same structure as described in Chapter 2 and modelled in Chapter 3.1, but with different model parameters.

The general idea of the developed self-tuning scheme is to apply an input signal to a system, making it possible to identify parameters for a control model, based on the system response. The identified control model parameters are then used to generate parameters to a controller, capable of controlling the system. The general self-tuning scheme used to self-tune the Mission OSTM boiler system is illustrated in Figure 5.1.

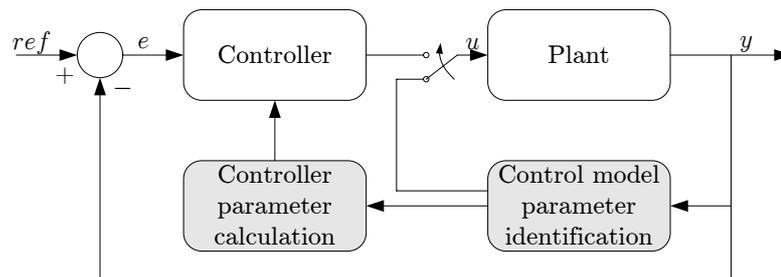


Figure 5.1: Illustration of the general self-tuning scheme.

When performing self-tuning of the Mission OSTM system, the above self-tuning scheme is applied to both the inner and outer control loops.

In Chapter 4, the controllers have been designed based on physical measurable parameters, thus enabling the control model parameters to be automatically identified.

The control model parameters to be identified are listed below.

Inner-loop control model parameters

1. Oil valve gain, K_{fu}
2. Air valve gain, K_a
3. Feed water actuator time constant, τ_{fw}
4. Feed water non-linearity, second order polynomial coefficients, ψ_{fw}

Parameter 1 and 2 are related to valve characteristics of valves in the fuel actuator system and are identified using Algorithm 5.1 in Section 5.1. Similar, parameter 3 and 4 are related to valve characteristics of the feed water valve and are identified using Algorithm 5.2 in Section 5.2.

Outer-loop control model parameters

5. Curveband fitted for oil/air relationship, given by the polynomial coefficients, $\psi_{fu \rightarrow a}$
6. Maximum and minimum load expressed in terms of maximum and minimum oil servo position, $\theta_{fu,max}$ and $\theta_{fu,min}$
7. Steam pressure integrator gain from oil valve position to steam pressure, γ_{ps}

Parameter 5 and 6 are identified using Algorithm 5.3 in Section 5.3, and finally, parameter 7 is identified using Algorithm 5.4 in Section 5.4.

All identification algorithms are implemented as S-functions written in C in MATLABTM Simulink.

How and when the parameters are identified during the start-up sequence of the Mission OSTM boiler system is in the following referred to as triggering. Triggering of the identification algorithms is implemented in StateFlow together with the existing start-up burner sequence implementation, as depicted in Figure 5.2, and a detailed view of the controller release procedure in Figure 5.3. The placement of each self-tuning trigger block is described together with the algorithms.

For each identification step, the status label is set to a number equal to or higher than 20, to distinguish between the status for normal operation and the status during self-tuning, i.e. during the first identification step, the status label is set to 20, during the next 22 and so forth.

In the following sections, each identification step is described and test results are presented.

In the following, the algorithm for the identification, in a one-valve case, is outlined.

Algorithm 5.1 Algorithm for valve characteristic identification in a one-valve case.

1. Apply input signal to open the valve. Stop when a valve position of 98 % is reached.
During the valve travel, the valve position and time is logged when the valve position passes 20 % and 80 %.
 2. Calculate valve gain, $K_{(\cdot)}$, using Equation (5.1)
 3. Using the valve position controller and the newly found parameters, close the valve.
 4. Set output flag to indicate that the self-tuning algorithm is performed successfully.
-

The full algorithm for determining the characteristics for the oil valve and the air damper is written as an S-function `FindValveGains.c`. Furthermore the designed controllers are implemented in an S-function `PDeadBand.c` reading the identified parameters, utilising self-tuned control of the oil valve and the air damper.

Algorithm Verification

In Figure 5.4, the oil valve and air damper positions, during the identification of the characteristics of the valves mounted in the setup in AI's test center, are shown as function of time.

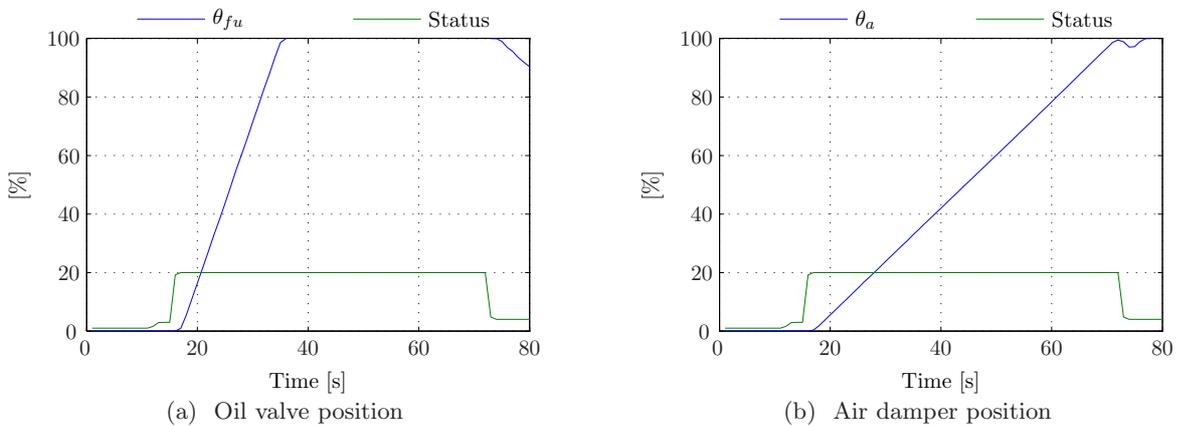


Figure 5.4: Oil valve and air damper positions as function of time during valve characteristic identification.

The oil valve and air damper controllers are designed in Subsection 4.1.2 on Page 47 and Subsection 4.1.3 on Page 49 respectively, as proportional controllers. The proportional gains are calculated using Equation (4.14) and (4.17) together with the dead band for the respective controllers. The model parameters obtained by self-tuning Algorithm 5.1 and the resulting controller parameters are listed in Table 5.1.

Ident. parameters		Ctrl. parameters	
K_{fu}	0.0552	$k_{fu,p}$	18.12
		$e_{fu,db}$	0.0035
K_a	0.0182	$k_{a,p}$	54.95
		$e_{a,db}$	0.0011

Table 5.1: Identified oil valve and air damper gains, listed together with the calculated proportional control parameters.

5.2 Feed Water Actuator Identification

As described in Section 4.1.1, the feed water actuator system is subject to non-linearities together with a time constant. These are to be determined automatically prior to operating the boiler. The identification of both the non-linearity and the time constant can be done using just one input sequence; a step sequence applied to the feed water actuator starting at 0 % valve position with stepwise increase of 10 % until the maximum valve position, 100 %, is reached. By logging the steady state values of both input to and output from the feed water system, the non-linearity can be determined.

The duration of each step depends on when a steady state feed water flow is reached, and thereby the time constant of the feed water actuator. By assuming that the valve behaves as a first order system, the steady state is reached after a time period of approximately 5τ . Obviously, the time constant must be identified prior to applying a new step.

For the identification of the feed water actuator characteristics, the steam pressure in the Mission OSTM boiler must be close to the operating point, i.e. 8 bar. However also the steam pressure integrator gain must be identified close to the operating point. Thus it is chosen to identify the feed water characteristics with a steam pressure of 6 bar. Thus the triggering of the identification algorithm is implemented as depicted in Figure 5.3.

Identification of Feed Water Actuator Time Constant

To identify the time constant, τ_{fw} , for the feed water actuator, parameter estimation is used. A method to obtain model parameters is by performing least squares estimation, where the model output is fitted to a measured system response. This is done by minimising a quadratic performance function, $V(\theta)$, [Knudsen, 1993, p. 32]:

$$V(\theta_{fw}) = \frac{1}{2n} \sum_{k=1}^n \epsilon(k, \theta_{fw})^2, \quad (5.2)$$

where θ_{fw} is a parameter vector. The estimation error $\epsilon(k, \theta_{fw})$ is given by:

$$\epsilon(k, \theta_{fw}) = \check{w}_{fw}(k) - \hat{w}_{fw}(k, \theta_{fw}), \quad (5.3)$$

where $\check{w}_{fw}(k)$ is the measured output and $\hat{w}_{fw}(k, \theta_{fw})$ is the optimal prediction, estimated on basis of the mathematical model. The optimal model prediction can be expressed on the form

[Knudsen, 1993, p. 38]:

$$\hat{w}_{fw}(k, \boldsymbol{\theta}_{fw}) = \begin{bmatrix} -\check{w}_{fw}(k-1) & u_{fw}(k-1) \end{bmatrix} \begin{bmatrix} a_{fw} \\ b_{fw} \end{bmatrix} \quad (5.4)$$

$$= \boldsymbol{\varphi}_{fw}^T \boldsymbol{\theta}_{fw}, \quad (5.5)$$

where $\boldsymbol{\varphi}_{fw}$ is denoted a signal vector and $\boldsymbol{\theta}_{fw}$ is the parameter vector.

To identify the elements, a_{fw} and b_{fw} , of the parameter vector, the linear part of the feed water actuator model:

$$G_{fw} = \frac{K_{fw}}{1 + \tau_{fw}s}, \quad (5.6)$$

where $K_{fw} = \frac{w_{fw}}{u_{fw}}$, must be rearranged to the form given by Equation (5.4), by discretising using backward Euler.

This gives the discretised model on the desired form:

$$\check{w}_{fw}(k) - \underbrace{e^{-\frac{T_{sys}}{\tau_{fw}}}}_{a_{fw}} \check{w}_{fw}(k-1) = \underbrace{\left(K_{fw} - K_{fw} e^{-\frac{T_{sys}}{\tau_{fw}}} \right)}_{b_{fw}} u_{fw}(k-1) + w_{fw}(k), \quad (5.7)$$

where $w_{fw}(k) \in \mathcal{N}(0, \sigma^2)$ is added stochastic noise.

In order to use Equation (5.2), a series of n measurements must be available.

For the specific use, the time constant must be identified prior to making a new step in valve position and thus the required measurements are not obtainable. Thus, the identification of the time constant for the feed water valve must be done recursively, which also reduces the computational requirements for the PPC.

The recursive least square estimation (RLSE) algorithm can be derived from Equation (5.2), and is given by [Knudsen, 1993, p. 65]:

$$\mathbf{P}(k) = \frac{1}{\lambda_{fw}} \left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1) \boldsymbol{\varphi}(k) \boldsymbol{\varphi}^T(k) \mathbf{P}(k-1)}{\lambda_{fw} + \boldsymbol{\varphi}^T(k) \mathbf{P}(k-1) \boldsymbol{\varphi}(k)} \right] \quad (5.8)$$

$$\hat{\boldsymbol{\theta}}_{fw}(k) = \hat{\boldsymbol{\theta}}_{fw}(k-1) + \mathbf{P}(k) \boldsymbol{\varphi}(k) \left[\check{w}_{fw}(k) - \boldsymbol{\varphi}^T(k) \hat{\boldsymbol{\theta}}_{fw}(k-1) \right], \quad (5.9)$$

where \mathbf{P} is a weighted covariance matrix and the expression in the square brackets in Equation (5.9) is the estimation error. The parameter $\lambda_{fw} \in [0; 1]$ is used as an exponential memory factor, that weights the old predictions lower than the new predictions. For the value $\lambda_{fw} = 1$, all predictions are weighted equal.

The initial values for the RLSE can be calculated using the following equations:

$$\mathbf{P}(k_0) = \left[\sum_{s=1}^{k_0} \lambda_{fw}^{k_0-s} \boldsymbol{\varphi}(s) \boldsymbol{\varphi}^T(s) \right]^{-1} \quad (5.10)$$

$$\hat{\boldsymbol{\theta}}_{fw}(k_0) = \mathbf{P}(k_0) \sum_{s=1}^{k_0} \lambda_{fw}^{k_0-s} \boldsymbol{\varphi}(s) \check{w}_{fw}(s). \quad (5.11)$$

However, to calculate the initial values, a series of measurement data is needed and the amount of measurement data must be least the number of parameters in the parameter vector θ_{fw} .

For the specific use, the measurement data for calculating initial values is not obtainable. Thus, the initial values are determined based on a rule of thumb by [Knudsen, 1993, p. 66]:

The parameter vector, θ_{fw} , should be initialised as zero:

$$\theta_{fw} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (5.12)$$

whereas P should be initialised with a diagonal matrix:

$$P = \begin{bmatrix} p & & \\ & \ddots & \\ & & p \end{bmatrix}, \quad (5.13)$$

with $p \in [1, 100]$.

The RLSE algorithm is written as a C-function `rlse.c` and used by Algorithm 5.2.

Identification of Feed Water Actuator Non-linearity

To identify the non-linearity, the steady state values of the feed water flow, w_{fw} , is logged at each step, making it possible to fit a polynomial using the least-squares method. Thereby, the polynomial describes the non-linearity of the feed water actuator; the feed water flow as a function of the feed water valve position.

A second order polynomial is chosen based on considerations made by [Andersen and Jørgensen, 2007, p.31-33] concerning the non-linearity in the feed water actuator system.

The algorithm containing both identification of the time constant and the non-linearity is outlined in Algorithm 5.2.

The above algorithm is implemented as an S-function `FindFWNonLin.c`. The inner-loop feed water flow controller is implemented in Simulink, with controller parameters calculated in an embedded MATLABTM function, reading the identified non-linearity and the time constant for the feed water actuator.

Algorithm Verification

To verify the developed algorithm, the input sequence together with the resulting feed water flow are depicted in Figure 5.5.

The measurement data shown in the figure, is obtained with feed water valve position increments of 5 %, due to a saturation in the output signal from the flow sensor, when the feed water valve

Algorithm 5.2 Algorithm for identification of feed water actuator characteristics.

1. Initialize feed water valve to 0 %.
2. Increase the feed water valve position by e.g. 10 %.
3. Estimate the time constant, τ_{fw} , using a recursive least square estimator implemented in the C-function `rlse.c`.
When the time corresponding to five times the estimated time constant has passed, the feed water valve position is logged together with the steady state value of the feed water flow. A new step in feed water valve position is initiated. Step 2 and 3 of the algorithm is repeated until 100 % valve position is reached.
4. Fit a polynomial to the logged feed water valve position and feed water flow, using the C-function `polyfit.c`.
5. Save the polynomial coefficients together with the time constant.
6. Set output flag to indicate that the self-tuning algorithm is performed successfully.

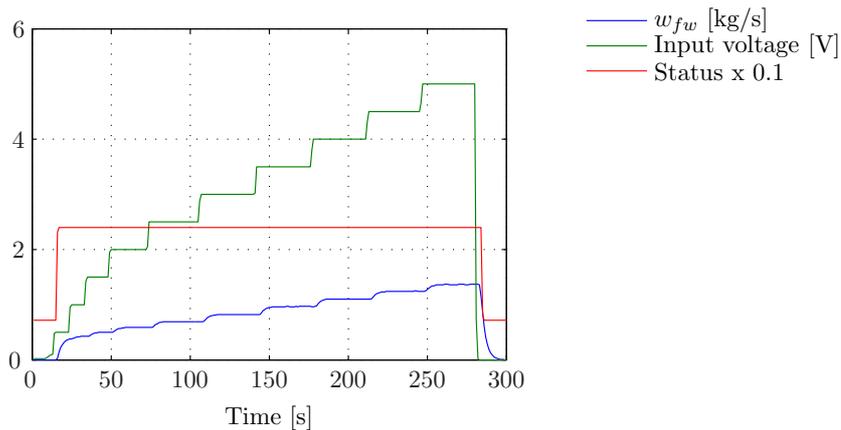


Figure 5.5: Illustration of input sequence for feed water valve characteristics identification and resulting feed water flow.

exceeds 50 % of full open. Such limitations or defects in the instrumentation is comprehensive to include into a self-tuning scheme, and is not taken into consideration in this self-tuning algorithm. Further limitations for the self-tuning scheme is presented in Section 5.6 on page 88.

For the feed water actuator system in AI's test center, the logged data points are shown in Figure 5.6a together with the determined second order polynomial fit.

The feed water flow controller is designed in Subsection 4.1.1 on page 44 as a PI-controller with feed-forward.

The feed-forward features the inverse non-linear valve characteristics and is fitted as a second order polynomial using Algorithm 5.2. The resulting feed-forward is determined by Equation (4.3) on page 45 with the coefficients listed in Table 5.2 and implemented in the controller in

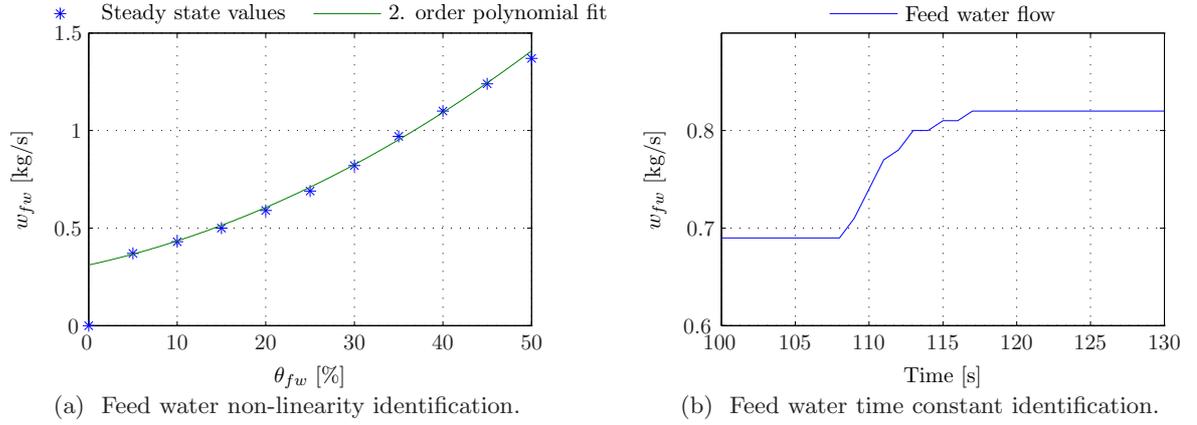


Figure 5.6: Result of feed water actuator characteristics identification.

Simulink as an embedded MATLABTM function.

The feed water time constant and gain is calculated from the RSLE of the a_{fw} and b_{fw} estimates, using the following equations:

$$\tau_{fw} = \frac{-T_{sys}}{\text{real}(\log(a_{fw}))} \quad (5.14)$$

$$K_{fw} = \frac{b_{fw}}{1 - e^{\frac{-T_{sys}}{\tau_{fw}}}}, \quad (5.15)$$

derived from Equation (5.7).

The PI-controller for the feed water actuator is designed with a proportional gain equal to the inverse estimated gain, K_{fw} , and the integrator time constant is equal to the estimated time constant, τ_{fw} , for the feed water actuator, also found using Algorithm 5.2. The calculation of controller parameters is implemented in Simulink using an embedded MATLABTM function.

	Ident. parameters	Ctrl. parameters
τ_{fw}	6.63	$T_{fw,i}$ 6.63
K_{fw}	0.28	$K_{fw,p}$ 3.6
ψ_{fw}	[0.00024 0.009952 0.310675]	

Table 5.2: Results of feed water actuator system identification.

5.3 Oil/Air Curveband Identification

The oil/air curveband is used as feed-forward for the oxygen level control to remove the non-linearities, caused by saturation in airflow and valve/damper characteristics.

When identifying the oil/air curveband, it is important that the oxygen level is held as constant as possible around 5 % to ensure a clean combustion during the identification. To avoid impure

combustion and thereby soot, it is preferred, that the oxygen level does not move below 5 % . A way to ensure that the oxygen level always stays above the level, is to adjust air prior to oil when increasing the oil flow and vice versa when decreasing the oil flow.

Identification of the oil/air curveband can take place after identification of oil and air valve characteristics and just after the ignition state, as depicted in Figure 5.2.

The basic idea of the identification is to step-wise increase the air valve position, θ_a , and adjust the oil valve position, θ_{fu} , to gain an oxygen level of 5 % . However, to obtain a curveband for an interval as large as possible, it is necessary to step-wise decrease the oil servo position, θ_{fu} , and adjust the air servo position, θ_a , until a lower bound of e.g. 10 % for either θ_a or θ_{fu} is reached. When the lower bound is reached, the air valve position is step-wise increased, until an upper bound is reached for either oil valve or air damper, to find the curveband.

In order to adjust the oil valve and air damper during the identification of the curveband, two conservative controllers must be designed. These controllers are only used during the identification of the curveband.

Design of Conservative Oxygen Level Controllers

As described in Chapter 4, the oxygen control model is simplified to a first order linear system. Assuming linear valve characteristics for the oil valve, a transfer function from oil valve position to oxygen level is given as:

$$G_{fu,O_2}(s) = \frac{x_{i,O_2}}{\theta_{fu}} = \frac{K_{O_2}}{\tau_{x_{o,O_2}}s + 1} . \quad (5.16)$$

Similar, the transfer function from air valve position to oxygen level is given by the same, but negative signed, transfer function, by assuming that the influence from oil and air servo is the same:

$$G_{a,O_2}(s) = -\frac{x_{i,O_2}}{\theta_a} = -\frac{K_{O_2}}{\tau_{x_{o,O_2}}s + 1} . \quad (5.17)$$

Introducing an oxygen level reference and feedback control, a close loop system is obtained as shown in Figure 5.7.

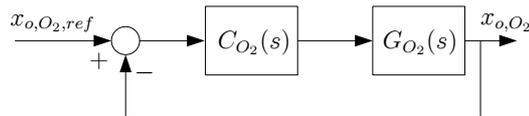


Figure 5.7: Oxygen feedback controller.

A simple controller to remove steady state error is an integrator, obtained by introducing a pole in origo and adjust the integrator gain, K_{i,O_2} .

Thus the controller is given by:

$$C_{O_2}(s) = K_{i,O_2} \frac{1}{s} . \quad (5.18)$$

The closed loop transfer function is then given by:

$$CL_{O_2}(s) = \frac{K_{O_2} K_{i,O_2}}{\tau_{O_2} s^2 + s + K_{O_2} K_{i,O_2}} = \frac{1}{\frac{\tau_{O_2}}{K_{O_2} K_{i,O_2}} s^2 + \frac{1}{K_{O_2} K_{i,O_2}} s + 1} . \quad (5.19)$$

Prior to identifying the curveband, it must be assumed, that only the time constant is known. Thus the controller design must be very conservative, consequently a controller design that guarantees no overshoot is made. To avoid overshoot the poles in the closed loop system must be real. This can be ensured by examining the denominator:

$$\tau_{O_2} s^2 + s + K_{O_2} K_{i,O_2} = 0 \quad (5.20)$$

$$s_p = \frac{-1 \pm \sqrt{1 - 4\tau_{O_2} K_{O_2} K_{i,O_2}}}{2\tau_{O_2}} . \quad (5.21)$$

For the solution to become real, the curveband oxygen control gain, K_{i,O_2} , is expressed in terms of the time constant:

$$K_{i,O_2} \leq \frac{1}{4\tau_{O_2} K_{O_2}} . \quad (5.22)$$

Remembering the similarity for oil and air valve, the gain, K_{i,O_2} , can be used both when adjusting oil and air valve position:

$$K_{i,O_2, fu} = \frac{1}{4\tau_{O_2} K_{O_2}} \quad (5.23)$$

$$K_{i,O_2, a} = -\frac{1}{4\tau_{O_2} K_{O_2}} . \quad (5.24)$$

Having derived the above controller, the algorithm of finding the curveband is outlined in Algorithm 5.3.

The logged oil and air servo positions are fitted to a polynomial from oil servo position to air servo position and used as a curveband to maintain the desired oxygen level.

The algorithm is implemented in Simulink using an S-function `OilAirOxygen.c`, which together with the S-function `curveband.c` constitutes the self-tuned curveband.

Algorithm 5.3 Algorithm for identification of oil/air curveband.

1. Step oil servo position down, step size e.g. 5 %.
2. Adjust air servo position using the designed controller.
Repeat 2-3 until a specified minimum oil servo position is reached. If minimum air servo position is reached before minimum oil servo position, the oil servo position is adjusted, so none of the lower bounds are violated.
3. Step air servo position up, step size e.g. 5 %.
4. Adjust oil servo position using the designed controller.
Repeat 4-5 until a specified maximum air servo position is reached and save oil servo position, air servo position and oxygen level at each step. If maximum oil servo position is reached at any time, the air servo position is adjusted.
5. Fit polynomial to the oil and air servo position data points using the C-function `polyfit.c`.
6. Set output flag to indicate that the self-tuning algorithm is performed successfully.

Algorithm Verification

The algorithm is tested in AI's test center, showing a system response as seen in Figure 5.8.

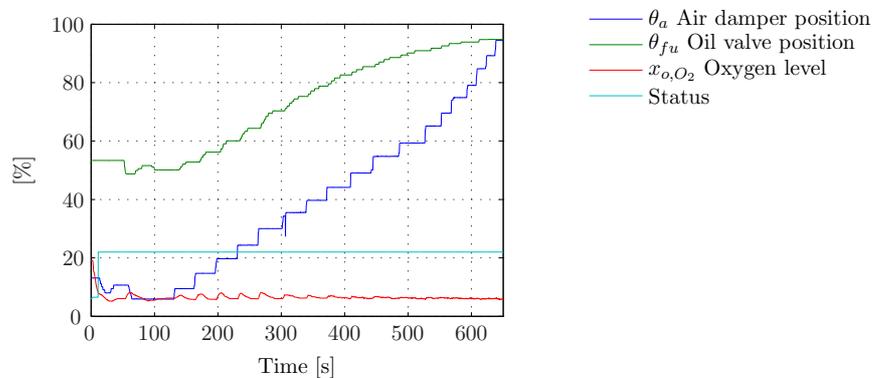


Figure 5.8: System response during oil/air curveband identification showing oil and air servo positions and oxygen level. Step size for both oil and air servo is 5 %.

The obtained oil and air servo positions are fitted as depicted in Figure 5.9.

Finally, the polynomial coefficients obtained from identification of the curveband are listed in Table 5.3 together with the determined load limits.

The derived polynomial is only valid within the operation of the valves during the identification, i.e. $\theta_{fu} \in [50.13; 94.75]$ and $\theta_a \in [5.88; 94.47]$. The interval for the oil servo position is used as saturation limits for the steam pressure controller output saturation.

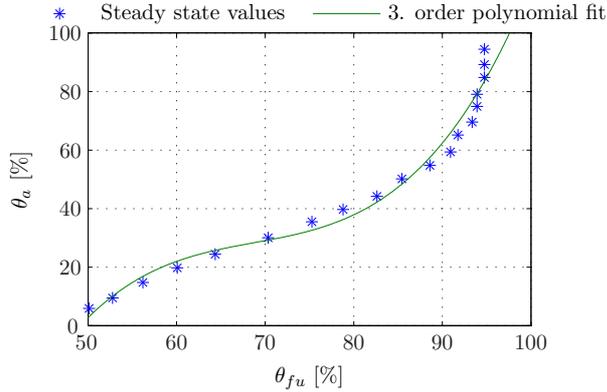


Figure 5.9: Curvebands fitted to obtained oil and air servo positions.

Identified parameters	
Curveband polynomial, $\psi_{fu \rightarrow a}$	[-0.0005 0.1008 -4.9010 65.8333]
Minimum burner load, $\theta_{fu,min}$	50.13
Maximum burner load, $\theta_{fu,max}$	94.75

Table 5.3: Results of curveband identification.

5.4 Steam Pressure Integrator Gain Identification

For the outer-loop steam pressure control, the control model was reduced to an integrator from oil valve position to steam pressure. Due to dependency on the steam pressure, p_s , the steam pressure integrator gain, γ_{p_s} , has to be found as close to the operation point of 8 bar as possible.

The integrator gain from oil valve position, θ_{fu} , to steam pressure, p_s , is found by measuring the time, Δt , to raise the steam pressure, Δp_s , at a given constant oil valve position, $\theta_{fu,\Delta t}$:

$$\gamma_{p_s} = \frac{\Delta p_s}{\Delta t \cdot \theta_{fu,\Delta t}}, \quad (5.25)$$

using SI units.

The steam pressure integrator gain is chosen to be identified in the time interval, when the steam pressure raises from 7 to 8 bar, i.e. one bar from the operating point. When the integrator gain is identified, the controller is released, as depicted in Figure 5.3 on page 73.

The algorithm for the identification is outlined in Algorithm 5.4.

The identification algorithm is implemented as an the S-function `FindGamma.c`, which together with the designed outer-loop steam pressure controller, implemented in Simulink as an embedded MATLABTM function, constitutes a self-tuned steam pressure controller.

Algorithm 5.4 Algorithm for identification of steam pressure integrator gain.

1. Fix oil and air valve positions.
2. Log the oil valve position together with the time and steam pressure.
3. Let the steam pressure increase to the operating point (8 bar).
4. Log the oil valve position together with the time and steam pressure.
5. Calculate the integrator gain, γ_{p_s} , using Equation (5.25).
6. Set output flag to indicate that the self-tuning algorithm is performed successfully.

Algorithm Verification

In Figure 5.10, the steam pressure is shown as function of time at a given constant oil and air valve position. It is seen, that the slope of the steam pressure curve is approximately constant and that the control model is adequate when the integrator gain is identified close to the operating point.

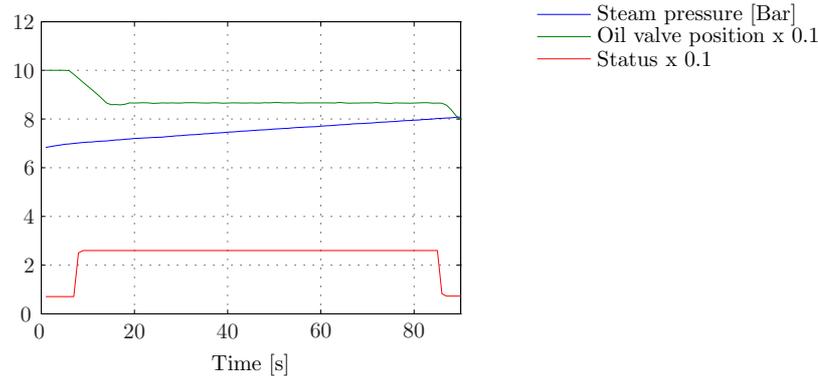


Figure 5.10: Measured steam pressure as function of time at a given constant oil and air valve position.

The steam pressure controller is in Subsection 4.2.1 on Page 54 designed as a PI-controller. Both the proportional gain and integrator gain, are calculated from the time constant for the fuel actuator and the steam pressure integrator gain, as in Equations (4.32)–(4.34) on Page 56.

The fuel actuator time constant is estimated from the valve gains, obtained by self-tuning Algorithm 5.1 and the steam pressure integrator gain is obtained by self-tuning Algorithm 5.4. The identified integrator gain and the resulting controller parameters are listed in Table 5.4.

Ident. parameters		Ctrl. parameters	
γ_{p_s}	13	$k_{p_s,p}$	$5.3 \cdot 10^{-3}$
		$T_{p_s,i}$	39.89

Table 5.4: Calculated steam pressure controller parameters, obtained from self-tuning.

5.5 Self-tuning Evaluation

In this section, the developed self-tuning scheme is evaluated based on both the previous verifications of each identification algorithm and general reflections regarding the performance of the developed self-tuning scheme.

To evaluate the developed self-tuning scheme a non-linear simulation of the controller performance is carried out. The parameters used for the non-linear simulation are the ones found during the various identification steps. Hence, the non-linear simulation will be a simulation of the final closed loop self-tuned system performance.

The simulation is carried out as the simulation of the designed controllers in Section 4.2.4 on Page 64. Hence, the outer-loop controller performance is simulated by a large step and a small step in steam flow, to simulate two different scenarios from a sudden change of steam consumption. Furthermore noise is added in accordance with the specifications in Section 4.2.4.

Simulation of Steam Pressure Controller

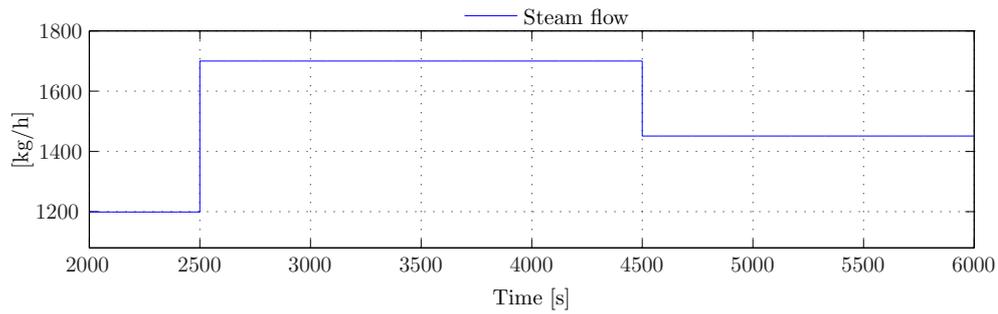
From the applied steps in steam flow, the simulated closed loop response of the steam pressure is shown in Figure 5.11b. As seen, the steam pressure controller is capable of maintaining the steam pressure within a deviation of approximately 0.02 bar from the operating point, when a large step in steam flow is applied. The minor step in steam flow results in a even smaller deviation and the steam pressure controller is considered acceptable.

Through the verification of the identification algorithms, the oil valve and air damper integrator gains was found to be small compared to the present actuator system, during verification of the system model in Section 3.3. This implies larger traveling times for both servos. The fact that the outer-loop steam pressure controller is designed with a bandwidth determined by the inner-loop oil and air actuator controllers, emerges as a slow outer-loop steam pressure control compared to the simulation in Section 4.2.4. However, the controller performs satisfactory taking the slow fuel actuator system into account. Furthermore, by looking at Figure 5.11c, it is seen, that the inner-loop actuator controllers are saturated for a short period of time, when the large step in steam flow is applied. Hence a transient deviation emerges in the oxygen level, depicted in Figure 5.11d. The transient is caused directly by the saturation of the fuel actuator.

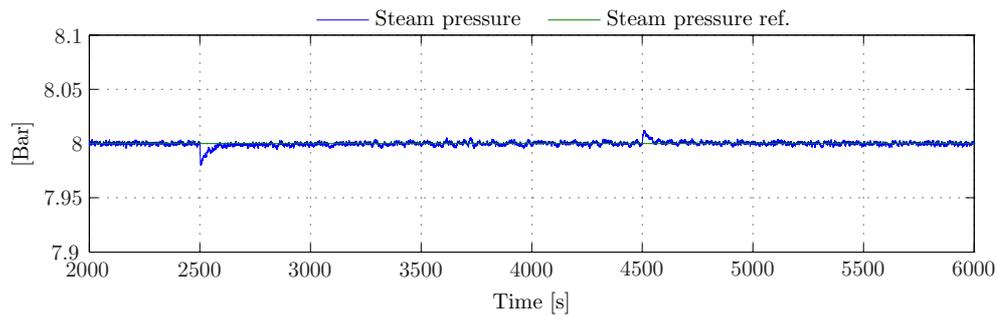
In attempt to cope with fuel actuator saturation, a model predictive servo control scheme is presented in Chapter 6.

Simulation of Water Level Controller

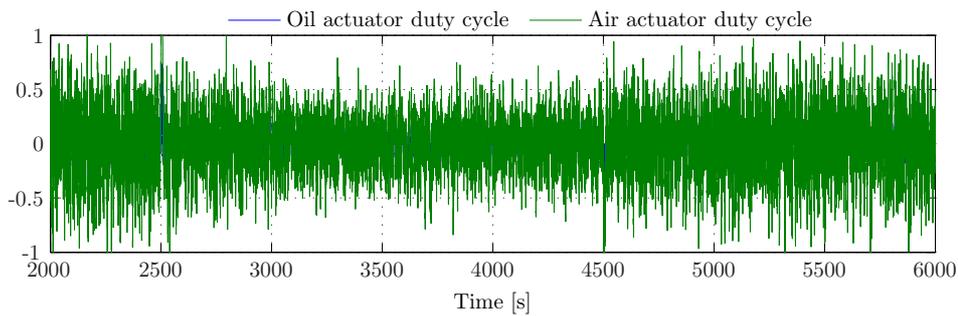
In Figure 5.12b the simulated closed loop response of the water level is shown. It is evident that the water level controller is slower than the steam pressure controller, which is expected due to the controller design, affected by the fluctuating water level measurements. The results of the simulation of the water level controller performance is very similar to the simulation presented in Section 4.2.4 on Page 64 due to a very small deviation between the estimated model parameters and the parameters identified by the self-tuning scheme.



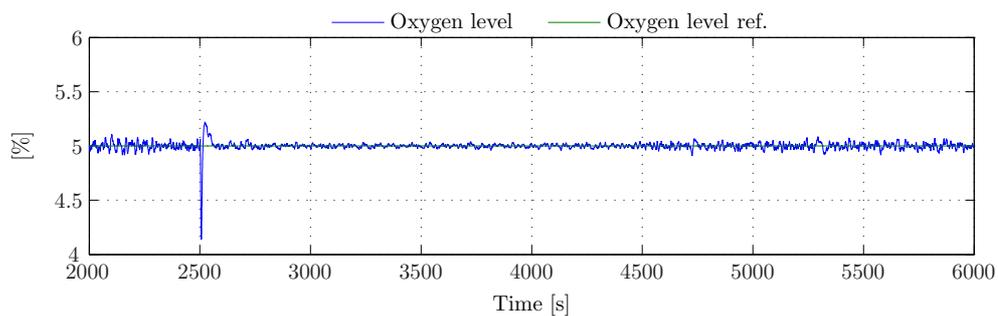
(a) Disturbance in steam flow.



(b) Steam pressure response from disturbance.



(c) Oil and air actuator duty cycle.

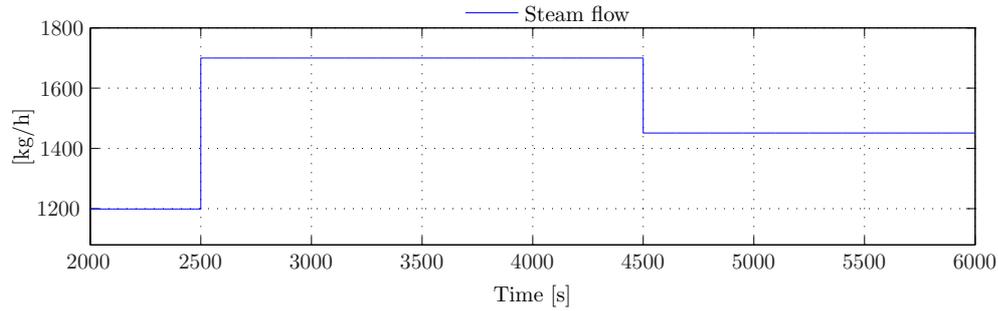


(d) Oxygen level.

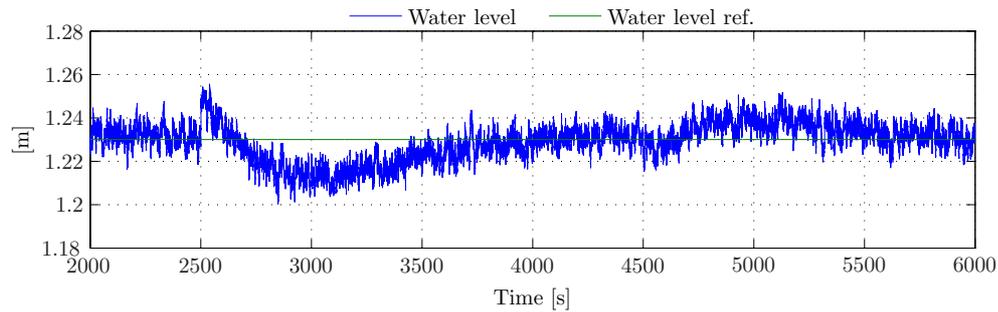
Figure 5.11: Simulation of p_s controller and actuator controllers from a large and small disturbance variation.

The water level controller is however able to keep the water level within a range of approximately ± 3 cm from the operating point, which is considered satisfactory.

With the presented simulation of the self-tuned controllers, the benefits from the developed self-



(a) Disturbance in steam flow.



(b) Water level response from disturbance.

Figure 5.12: Simulation of outer-loop water level performance from a large and small disturbance variation.

tuning scheme shows to be auspicious. However, a final test and verification of the outer-loop controllers are yet to be conducted in order to verify the complete self-tuning scheme.

As seen from the verification of the self-tuning identification algorithms, the self-tuned controller for the inner-loop fuel actuator system has showed good performance regarding tracking of the reference servo positions. However, actuator saturation and assumptions regarding linear curveband affects the oxygen level controller in an unfavorable manner. As previously mentioned, a solution to cope with the constraints introduced by the curveband, and actuator saturation limits, is presented in Chapter 6.

In the following section, a short review of potential limitations for the developed self-tuning scheme is presented.

5.6 Limitations of the Developed Self-tuning Scheme

The designed self-tuning algorithms are designed as a proof of concept. Thus some limitations in the developed self-tuning scheme exists. In this section, some of the most conspicuous limitations are presented, without leaving out the existence of others.

1. Instrumentation

The self-tuning algorithms generally do not take any malfunctioning of the instrumentation into consideration. E.g. the flow sensor for the feed water in the test center is saturated above a valve position of 70 %. As a result, the feed water valve characteristics are faulty identified and the controller is incorrectly tuned.

2. Error checking

The designed self-tuning algorithms are designed as a proof of concept. Thus, the degree of error checking is almost non-existing. However an a simple and efficient method for error checking can be implemented in state flow together with the triggering of each algorithm. This way, dead locks can be escaped i.e. using a timer. Faulty error handling or missing error handling can however be fatal by e.g. causing water overflow or too high steam pressure during identification.

3. Initial burner position The initial burner position defines the position of both oil valve and air damper position. The initial burner position must be known prior to performing the selftuning on the Mission OSTM boiler system in order to ignite the fuel supplied at ignition. However, as the identification of the curveband takes place as the first step shortly afterwards the ignition, the servo positions are rapidly adjusted to ensure a clean combustion.

4. PWM saturation

The pulse width resolution defined by $\frac{T_{PWM}}{T_{sys}}$ limits the inner-loop oil and air servo controllers ability to ensure equal servo traveling times. Thereby, the best performance of is obtained with servos manufactured to have same travel speed. The performance is lowered concurrently with the deviating traveling time for the two servos.

A self-tuning scheme has been developed and successfully tested using non-linear simulation. The developed self-tuning scheme is capable of identifying simple model parameters and using the parameters to calculate controller parameters. However, during the development of the self-tuning scheme, the importance of following the curveband was clarified. In order to follow a non-linear curveband, a generic solution is desired.

In attempt to develop a generic method for controlling the oil and air servos, to ensure that a non-linear curveband is obeyed, the problem is described as a constrained model predictive control problem in the following chapter.

Constrained Control of Fuel Valve and Air Damper

This chapter contains a description of a method of constrained control, developed to cope with the task of following a non-linear curveband regardless of fuel actuator saturation. First, the problem of oil/air ratio mismatch is outlined by a case study and a model predictive control problem formulation, to describe the constrained minimisation problem, is outlined. Finally, a simple method of reference correction, to solve the discrete minimisation problem, is developed and the method is verified through a case study.

When performing control of a fuel consuming process, the oxygen level in the exhaust gas must be constant. The task of ensuring constant oxygen level can be a challenge, in particular if the burner load is changed momentarily from one load to another. An easy methods to achieve this objective is to ensure uniform flow and servo characteristics for the air damper and the fuel valve. However this method relies on the mechanical design and the method is often impractical. Traditionally, to compensate for the difference in flow characteristics, a curveband is used to define the ratio between oil valve and air damper position [[Aalborg Industries, 2008](#)]. Thus only the servo characteristics influence the dynamic behaviour of the consumption. When introducing the curveband, the phenomenon of fuel/air ratio mismatch can occur. This phenomenon is described and analysed in the following section, followed by a model predictive problem formulation and an algorithmic scheme to cope with the phenomenon.

6.1 Fuel/Air Ratio Mismatch

The phenomenon of fuel/air ratio mismatch occurs when the burner load is momentary changed. In Subsection 4.1.4 on page 49, a solution for PWM saturation was presented. However, the presented method relies on linear and one-to-one fuel/air servo ratio and is applicable when equal traveling times is desired for the two servos. The presented method is however not capable of solving the problem of non-linear servo ratio constraints. To ease the understanding of the phenomenon, a case study is carried out.

Case Study

As an example, the oil valve and the air damper has integrator gains, $K_{fu} = 0.05$ and $K_a = 0.015$ respectively and the burner load is defined to be an interval $[0; 1]$ mapping θ_{fu} linear to the interval $[0.05; 0.98]$. Furthermore, a curveband defining the air damper position as a function of the oil valve position is given by the 2. order polynomial, $\theta_a = -0.21\theta_{fu}^2 + 0.44\theta_{fu} + 0.16$. The

curveband is depicted in Figure 6.1.

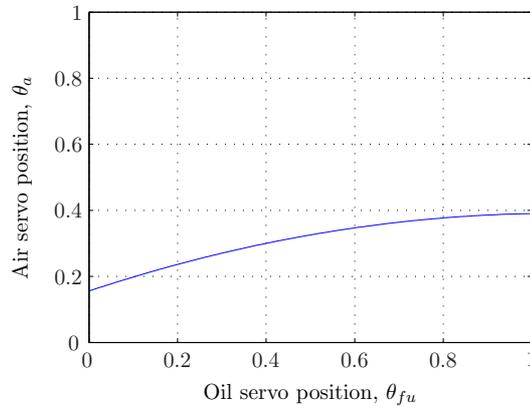


Figure 6.1: Curveband defining the air damper position as a function of the oil valve position.

A curveband, as depicted in Figure 6.1, shows that the air intake is oversized compared to the fuel intake, based on the intervals for the servos. Furthermore, the curveband is non-linear caused by e.g. low fuel pressure or simply a non-linear fuel valve.

In order to maintain constant oxygen level, the curveband must be followed at any time. However rate constraints for the position of the air damper and the oil valve causes the two actuators not to follow the specified curveband at any time. An example is given in Figure 6.2, where a step change in the reference is applied, causing the two servos to vary their respective positions freely. Thereby the phenomenon of fuel/air ratio mismatch, during the change of position toward the new set point, occurs.

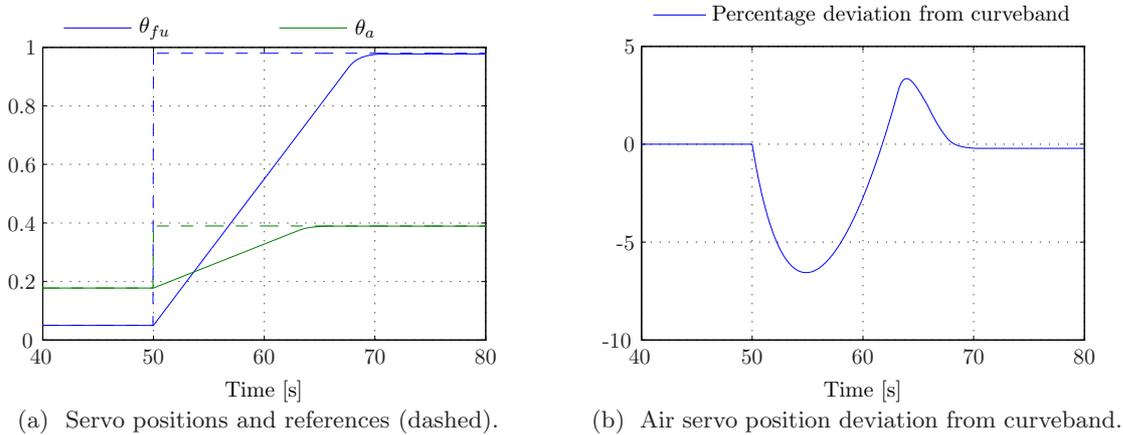


Figure 6.2: Air damper and oil valve position as a function of time with unconstrained control.

As seen in Figure 6.2a, the servo positions, θ_{fu} and θ_a , are changing by constant slope. Hence their respective references are reached at different time and clearly the servo positions do not follow the curveband in the timespan, from changing their reference to the reference is reached, for both servos. Figure 6.2b shows a worst case deviation from the curveband of approximately 5 %. Clearly the phenomenon is caused by constraints in the system, leading to describe the problem as a model predictive control problem.

6.2 Model Predictive Control Problem Formulation

For the use of model predictive control to formulate the problem, a linear discrete-time state space system model is derived for the fuel/air actuator. The derivation is based on simple integrator models, as described in Section 3.3 on page 38 during modeling of the fuel actuator system:

$$\begin{bmatrix} \theta_{fu}(k+1) \\ \theta_a(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \theta_{fu}(k) \\ \theta_a(k) \end{bmatrix} + \begin{bmatrix} K_{fu}T_s & 0 \\ 0 & K_aT_s \end{bmatrix} \cdot \begin{bmatrix} d_{fu}(k) \\ d_a(k) \end{bmatrix} . \quad (6.1)$$

The input vector, $u = \begin{bmatrix} d_{fu} & d_a \end{bmatrix}^T$ represents the duty cycle for the PWM part of the actuator system and is defined in the interval, $u \in [-1; 1]$. Positive duty cycle represents CW servo rotation and negative duty cycle represents CCW servo rotation.

The air damper and oil valve introduces range linear constraints on the controlled variables, i.e. servo positions, in the form:

$$0 \leq \theta_{fu} \leq 1 \quad (6.2)$$

$$0 \leq \theta_a \leq 1 , \quad (6.3)$$

defining the outer position limits of the servos. Furthermore, the servos introduces constraints on slew rates of the controlled variables as follows:

$$-K_{fu} \leq \Delta\theta_{fu} \leq K_{fu} \quad (6.4)$$

$$-K_a \leq \Delta\theta_a \leq K_a . \quad (6.5)$$

These slew rate constraints can however easily be transformed to linear input constraints by realising, that the maximum slew rate of the controlled variables is obtained with a PWM duty cycle of 1. Thus the constraints becomes:

$$-1 \leq d_{fu} \leq 1 \quad (6.6)$$

$$-1 \leq d_a \leq 1 . \quad (6.7)$$

The curveband introduces an equality constraint:

$$\theta_a = f(\theta_{fu}) , \quad (6.8)$$

which might be non-linear.

Finally, by introducing a feedback oxygen controller, correcting the air damper position by $\Delta\theta_a$, the equality constraint (6.8) simply becomes:

$$\theta_a = f(\theta_{fu}) + \Delta\theta_a , \quad (6.9)$$

and thereby changing by time.

To recapitulate, the constrained problem can be written as a constrained linear minimisation problem as follows:

$$\begin{aligned}
 \min_{d_a, d_{fu}} \quad & |\theta_{fu} - \theta_{fu,ref}| \\
 \text{s.t.} \quad & 0 \leq \theta_{fu} \leq 1 \\
 & 0 \leq \theta_a \leq 1 \\
 & -1 \leq d_{fu} \leq 1 \\
 & -1 \leq d_a \leq 1 \\
 & \theta_a = f(\theta_{fu}) + \Delta\theta_a .
 \end{aligned} \tag{6.10}$$

By rewriting the performance function $\theta_{fu} - \theta_{fu,ref}$ using the system model from Equation (6.1) the discrete performance function is given by:

$$\theta_{fu} - \theta_{fu,ref} = \theta_{fu}(k) + K_{fu}T_s d_{fu}(k+1) - \theta_{fu,ref}(k+1) . \tag{6.11}$$

Similar, the constraints given by Equation (6.2) and (6.3) are rewritten to:

$$\frac{-\theta_{fu}(k)}{K_{fu}T_s} \leq d_{fu}(k+1) \leq \frac{1 - \theta_{fu}(k)}{K_{fu}T_s} \tag{6.12}$$

$$\frac{-\theta_a(k)}{K_aT_s} \leq d_a(k+1) \leq \frac{1 - \theta_a(k)}{K_aT_s} . \tag{6.13}$$

Finally, the last constraint given by (6.9) is rewritten to:

$$\theta_a(k) + K_aT_s d_a(k+1) = f(\theta_{fu}(k) + K_{fu}T_s d_{fu}(k+1)) + \Delta\theta_a(k+1) . \tag{6.14}$$

Thereby, the minimisation problem is expressed in terms of the controlled variables, d_{fu} and d_a , as given below:

$$\begin{aligned}
 \min_{d_a(k+1), d_{fu}(k+1)} \quad & |\theta_{fu}(k) + K_{fu}T_s d_{fu}(k+1) - \theta_{fu,ref}(k+1)| \\
 \text{s.t.} \quad & \frac{-\theta_{fu}(k)}{K_{fu}T_s} \leq d_{fu}(k+1) \leq \frac{1 - \theta_{fu}(k)}{K_{fu}T_s} \\
 & \frac{-\theta_a(k)}{K_aT_s} \leq d_a(k+1) \leq \frac{1 - \theta_a(k)}{K_aT_s} \\
 & -1 \leq d_{fu}(k+1) \leq 1 \\
 & -1 \leq d_a(k+1) \leq 1 \\
 & d_a(k+1) = \frac{f(\theta_{fu}(k) + K_{fu}T_s d_{fu}(k+1)) - \theta_a(k) + \Delta\theta_a(k+1)}{K_aT_s} .
 \end{aligned} \tag{6.15}$$

Hence, some of the constraint depends on the current oil valve and air damper positions and are thus varying over time.

To ease the understanding of the minimisation problem, an example of the minimisation constraints is illustrated Figure 6.3, with $\Delta\theta_a = 0$ for simplicity. The figure illustrates the constrained value space for the duty cycles, d_{fu} and d_a . Furthermore, the time variation is illustrated

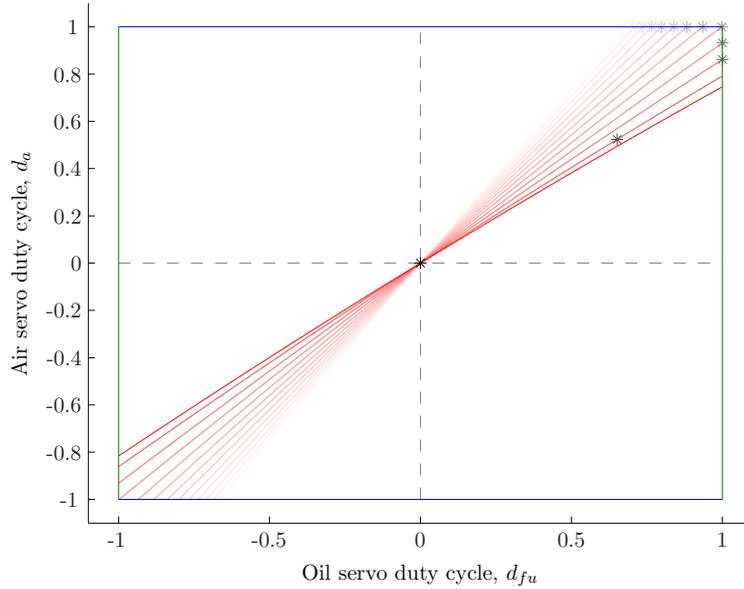


Figure 6.3: Value space for d_{fu} and d_a including the constraints for $d_{fu}(k+1)$ (green) and $d_a(k+1)$ (blue). Furthermore, the varying constraint given by the curveband is included (red). The desired solution to the minimisation problem is marked by *.

by toning the color of the time varying quantities, i.e. low toning represents the samples of the past and normal toning represents the current sample.

As depicted in Figure 6.3, the constraint related to the curveband varies over time. As seen, both duty cycles converges toward the origin, $(0,0)$, corresponding to servo positions converging toward the given references. Furthermore, it is seen that non of the linear inequality constraints are violated.

To lower the computation load, a reference correction control scheme is proposed to cope with the constrained problem, based on the given minimisation problem. This control scheme is described in the following section.

6.3 Model Predictive Control Scheme for Ratio-Constrained PWM Servo Positioning

The objective for the control scheme is to cope with the both linear and non-linear constraints of the fuel actuator system, by correcting a common reference to ensure that no constraints are violated.

In Figure 6.4, the principle of a common control scheme is outlined.

The ratio-constrained dual-servo positioner is to be designed to facilitate fast PWM servo control and comply with the servo ratio specified by the non-linear constraint, i.e. the curveband.

By treating the curveband purely as the ratio, the curveband is used to define the air servo position reference, $\theta_{a,ref}$, as a function of the oil servo position, $\theta_{fu,ref}$. Thus only one input

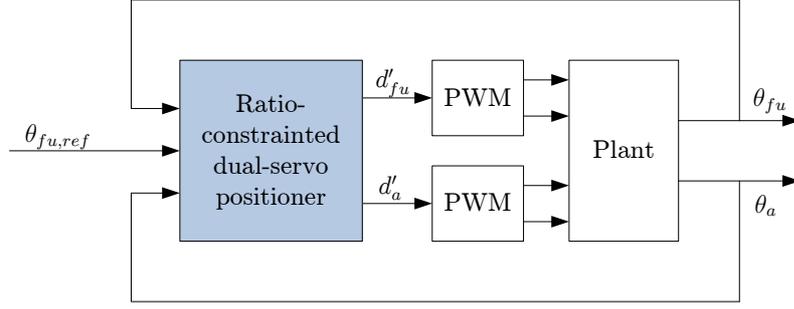


Figure 6.4: Block diagram illustrating the context of the servo positioning control scheme.

reference for the burner system is present, namely the reference to the oil servo position, $\theta_{fu,ref}$.

As described in Chapter 4, a suitable valve position controller is a P-controller with control gain inverse proportional to the specific valve gain. For PWM with a frequency of 1 Hz, this gives the following two control gains:

$$k_{fu,p} = \frac{1}{K_{fu}} \quad (6.16)$$

$$k_{a,p} = \frac{1}{K_a} . \quad (6.17)$$

Thus the corresponding PWM duty cycles are defined by:

$$d_{fu} = (\theta_{fu,ref} - \check{\theta}_{fu}) k_{fu,p} \quad (6.18)$$

$$d_a = (\theta_{a,ref} - \check{\theta}_a) k_{a,p} . \quad (6.19)$$

However, the constraints (6.6) and (6.7) causes the controller to enter a state of output saturation, where the saturated valve travels at full speed when the duty cycle is saturated. When the output of the controller is saturated, the non-linear constraint is violated. To avoid output saturation, the valve position references can be corrected based on system knowledge as depicted in Figure 6.5.

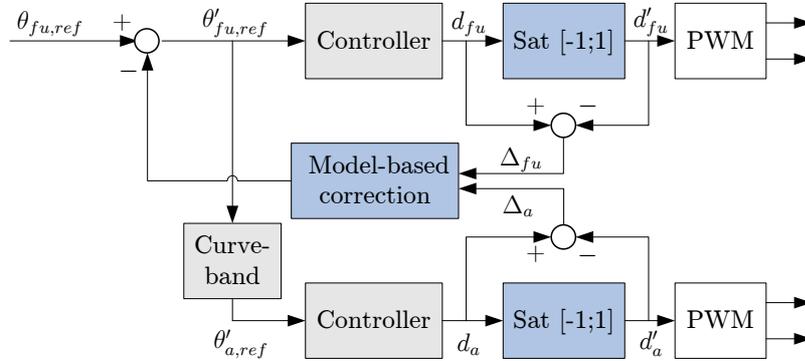


Figure 6.5: Block diagram illustrating the principle of the servo positioning control scheme with reference correction.

By inspection of the block diagram in Figure 6.5, the reference correction can be considered as an anti-saturation scheme, correcting the reference when the output is saturated, to ensure that the two involved servos follows the specified curveband.

A fundamental quantity of the model-based correction is the specific amount of saturation, here defined as $\Delta d_{fu} = d_{fu} - d'_{fu}$ and $\Delta d_a = d_a - d'_a$ for the two servos respectively. These quantities combined with knowledge to the system and the controllers are used in order to determine the correction of the common reference, $\theta_{fu,ref}$. The corrected reference is denoted $\theta'_{fu,ref}$.

Regarding the amount of saturation, Δd_{fu} and Δd_a respectively, it is important to notice the properties listed in Table 6.1.

	Saturation difference, Δd	Duty cycle, d	Common reference correction
Fuel	$\Delta d_{fu} \geq 0$	$d_{fu} \geq 1$	$\theta'_{fu,ref} \leq \theta_{fu,ref}$
	$\Delta d_{fu} \leq 0$	$d_{fu} \leq -1$	$\theta'_{fu,ref} \geq \theta_{fu,ref}$
	$\Delta d_{fu} = 0$	$-1 \leq d_{fu} \leq 1$	$\theta'_{fu,ref} = \theta_{fu,ref}$
Air	$\Delta d_a \geq 0$	$d_a \geq 1$	$\theta'_{a,ref} \leq \theta_{a,ref}$
	$\Delta d_a \leq 0$	$d_a \leq -1$	$\theta'_{a,ref} \geq \theta_{a,ref}$
	$\Delta d_a = 0$	$-1 \leq d_a \leq 1$	$\theta'_{a,ref} = \theta_{a,ref}$

Table 6.1: List of Δd properties.

By inspection of the properties listed in Table 6.1, the correction of the common reference can only contribute by decreasing $|d_{fu}|$ and $|d_a|$. Hence, the order of correction is of no importance and the model-based correction is chosen to first avoid saturating the air servo duty cycle, defined by constraint (6.6).

For the air servo constraint, the correction of the common reference is defined by:

$$\Delta \theta_{fu,ref|\Delta d_a} = f^{-1}(\Delta \theta_{a,ref|\Delta d_a}) \quad , \quad (6.20)$$

where $\Delta \theta_{a,ref|\Delta d_a}$ defines the correction of $\theta_{a,ref}$ in order to avoid violating the constraint. The function f^{-1} defines the inverse curveband function, i.e. the function that maps air servo position, θ_a , to oil servo position, θ_{fu} . By mapping the air servo correction, $\Delta \theta_{a,ref|\Delta d_a}$, using the inverse curveband, the correction, $\Delta \theta_{fu,ref|\Delta d_a}$, specifies the correction of the common reference, $\theta_{fu,ref}$, needed in order to avoid violating the constraint in duty cycle for the air servo, d_a .

For the remaining constraint (6.7), the correction, $\Delta \theta_{fu,ref|\Delta d_{fu}}$, is simply defined as a further correction of the newly corrected reference $\theta_{fu,ref} - \Delta \theta_{fu,ref|\Delta d_a}$, without need for any mapping function, as the correction is given directly as the needed correction of the common reference.

By recalling that the servo position controller is designed as a proportional controller, the two reference corrections can be expressed as linear dependent on the controller gain and thereby the model parameters, as given below:

$$\Delta \theta_{a,ref|\Delta d_a} = \frac{1}{k_{a,p}} \Delta d_a = K_a \Delta d_a \quad (6.21)$$

$$\Delta \theta_{fu,ref|\Delta d_{fu}} = \frac{1}{k_{fu,p}} \Delta d_{fu} = K_{fu} \Delta d_{fu} \quad . \quad (6.22)$$

With the above definitions clarified, a two step algorithm for correcting the reference is presented in Algorithm 6.1.

Algorithm 6.1 Algorithm capable of correcting a common reference in order to avoid PWM saturation.

1. Correct common reference with respect to first constraint (6.6)
 - (a) Calculate duty cycle, d_a , to reach reference $\theta_{a,ref} = f(\theta_{fu,ref})$:

$$d_a = k_{a,p} (f(\theta_{fu,ref}) - \check{\theta}_a)$$
 - (b) Calculate duty cycle saturation, Δd_a :

$$\Delta d_a = d_a - d'_a$$
 - (c) Calculate correction, $\Delta\theta_{a,ref|\Delta d_a}$, of $\theta_{a,ref}$ needed in order to avoid saturation:

$$\Delta\theta_{a,ref|\Delta d_a} = K_a \Delta d_a$$
 - (d) Using the inverse curveband, map the correction to the common reference:

$$\Delta\theta_{fu,ref|\Delta d_a} = f^{-1}(\Delta\theta_{a,ref|\Delta d_a})$$
 2. Further correct common reference with respect to second constraint (6.7)
 - (a) Calculate duty cycle, d_{fu} , to reach corrected reference, $\theta_{fu,ref} - \Delta\theta_{fu,ref|\Delta d_a}$:

$$d_{fu} = k_{fu,p} (\theta_{fu,ref} - \Delta\theta_{fu,ref|\Delta d_a} - \check{\theta}_{fu})$$
 - (b) Calculate duty cycle saturation, Δd_{fu} :

$$\Delta d_{fu} = d_{fu} - d'_{fu}$$
 - (c) Calculate correction, $\Delta\theta_{fu,ref|\Delta d_{fu}}$, of $\theta_{fu,ref}$ needed in order to avoid saturation:

$$\Delta\theta_{fu,ref|\Delta d_{fu}} = K_{fu} \Delta d_{fu}$$
-

By applying Algorithm 6.1, the corrected common reference is given by:

$$\theta'_{fu,ref} = \theta_{fu,ref} - f^{-1}(\Delta\theta_{a,ref|\Delta d_a}) - \Delta\theta_{fu,ref|\Delta d_{fu}} . \quad (6.23)$$

Reverting to the case study in Section 6.1, the result of a simulation, applying the developed reference correction scheme, is presented in Figure 6.6.

As seen in Figure 6.6a, the references for the two servos are corrected to avoid violating the duty cycle constraints. Hence, the reference correction scheme complies with the non-linear ratio constraint specified by the curveband. As seen in Figure 6.6b, the worst case percentage deviation from the specified curveband is, in the case study, by use of the reference correction scheme, reduced to the restrictions introduced by the dead band designed in Chapter 4.

To verify, that the developed method is capable of keeping the duty cycle within the limits specified by the constraints, the duty cycle for both the oil and the air servo is depicted in Figure 6.7.

From Figure 6.7 it is easy to verify, that the method corrects the references to ensure that none of the two constraints are violated.

Referring to the non-linear simulation of the performance of the self-tuned controllers in Section

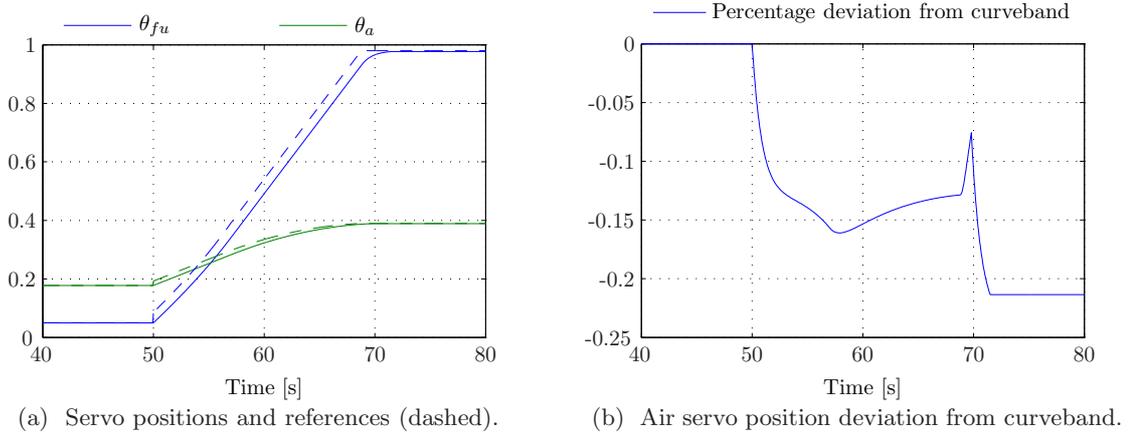


Figure 6.6: Air damper and oil valve position as a function of time with constrained control scheme.

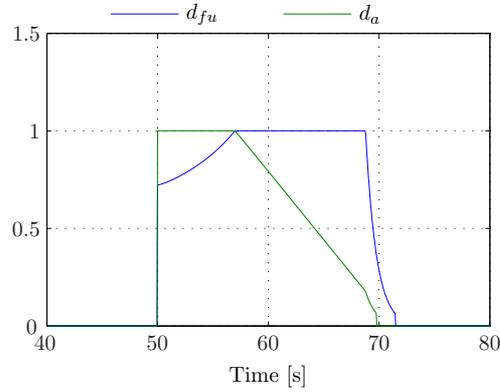
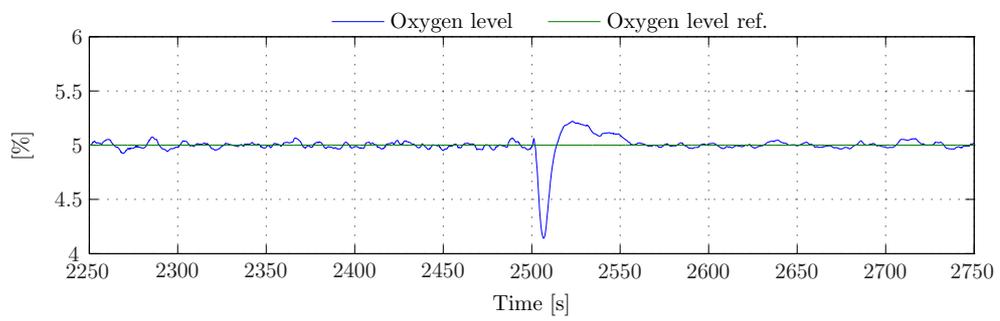


Figure 6.7: Duty cycle for oil and air servo, during reference correction for a step from minimum to maximum burner load.

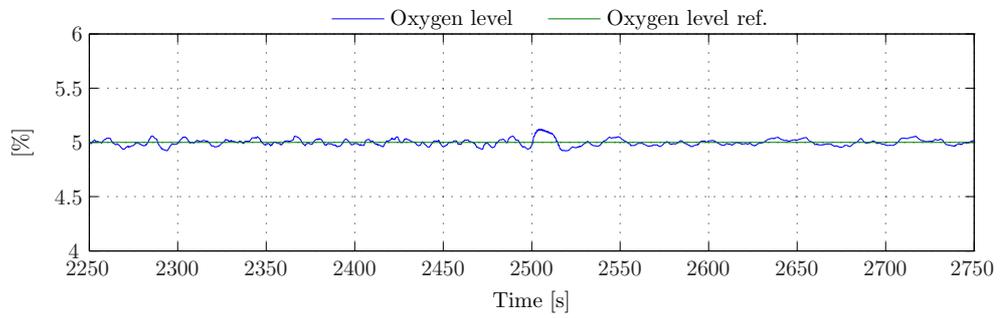
5.5 on page 85, the developed reference correction scheme aims to improve the oxygen level control. To examine the benefits by applying the developed reference correction scheme, a non-linear simulation is carried out. The resulting oxygen level, both with and without reference correction scheme, is depicted in Figure 6.8.

As seen in Figure 6.8, the reference correction scheme entails a more steady oxygen level. Hence by using the developed reference correction scheme, the minimisation problem described by Equation (6.15) is solved using a simple correction approach.

Referring to the objectives for the project, listed in the problem description in Chapter 2, the following chapter contains an analysis of the potential benefits of applying classic estimation techniques, in attempt to improve the control of the water level.



(a) Oxygen level without servo reference correction scheme.



(b) Oxygen level with servo reference correction scheme.

Figure 6.8: Simulation comparing oxygen level with and without servo reference correction scheme.

Kalman Estimation

This chapter contains a description of the potential benefits by the employment of classic estimation techniques in order to estimate i.a. the unmeasurable steam flow. Furthermore a linear Kalman estimator is designed, capable of both providing smoothed estimates of the mean water level and estimates of the unmeasurable steam flow. The performance of the designed Kalman estimator is simulated and the chapter is ended, by an assessment of the usage of the designed estimator, in order to improve the control of the water level in the Mission OSTM boiler.

The use of Kalman estimation originates from the objective, of obtaining an estimate of the mean water level and thereby achieve better control of the water level. The water level controller design in Chapter 4, was initially designed to have a bandwidth in relation to the inner-loop controller. However, the bandwidth was further reduced due to the degree of noise in the water level measurements, caused by a fluctuating water level.

By use of a linear Kalman estimation, existing sensor information can be filtered according to the system model and new information can be estimated by use of simple state augmentation. The objective for designing a linear Kalman estimator is to filter the water level measurements, to obtain a mean water level and to estimate the unmeasurable load disturbance, i.e. the steam flow.

Prior to designing the Kalman estimators a state space model of the Mission OSTM boiler is required. Appendix F contains a derivation, parameter estimation and verification of a linear state space model on the form:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) , \end{aligned} \tag{7.1}$$

suitable for Kalman estimation. The use of a linear state space model implies the use of linear Kalman estimation. In the case of a non-linear state space model, the use of Extended Kalman estimation or Unscented Kalman estimation would be beneficial.

The remaining of this chapter contains a description, design and verification of the linear Kalman estimator.

7.1 Linear Kalman Estimation

Kalman estimation, or Kalman filtering, is a mathematical tool usefull in many control related problems. The overall principle of the Kalman estimator is to combine knowledge from the system model and noisy sensor measurements, to give a statistical estimate of the system states, and possible non-measurable states, which thereby facilitate better control of the system. Hence

the Kalman estimator works as a recursive filter. The Kalman estimator is a state observer including statistical noise information to optimise the estimates, by propagating the probability distribution of the states through the filter. In case of a linear system with e.g. Gaussian probability distribution, the propagated probability distribution will also be Gaussian.

Figure 7.1 illustrates the general concept of the Kalman estimator. From a given input, \mathbf{u} , the

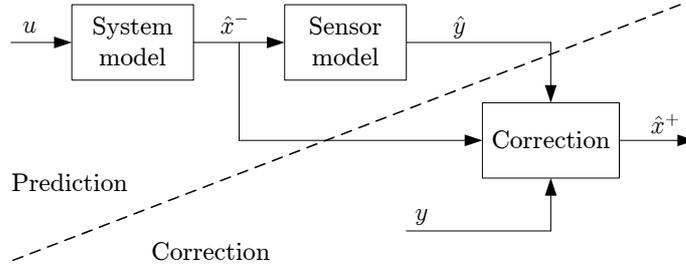


Figure 7.1: Block diagram of the Kalman estimator principle.

system states are predicted from the system model, and the predicted states, $\hat{\mathbf{x}}^- = \hat{\mathbf{x}}(k|k-1)$, are further used to predict the measured output from the sensor model. Finally the estimated states and estimated output, $\hat{\mathbf{y}}$, are corrected, using the sensor measurements, \mathbf{y} , resulting in an estimate of the current state of the system, $\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}(k|k)$. In brief, the Kalman estimator consists of a prediction step and a correction step, where the underlying calculations are based on probability theory.

The discrete state space model, derived in Appendix F, is derived from the control models in Chapter 4 and the resulting state space model is given by Equation (7.2) and (7.3).

$$\begin{bmatrix} \dot{p}_s \\ \dot{V}_w \\ \dot{w}_s \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\beta_{p_s} \\ 0 & 0 & -\beta_{L_w} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \\ w_s \end{bmatrix} + \begin{bmatrix} \gamma_{p_s} & 0 \\ 0 & \beta_{L_w} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_{fu} \\ w_{fw} \end{bmatrix} \quad (7.2)$$

$$\begin{bmatrix} p_s \\ L_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{A_w} & \alpha_{L_w} \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \\ w_s \end{bmatrix}, \quad (7.3)$$

which is on the form of Equation (7.1). In the state space model, the state vector is augmented to also include the steam flow, w_s , which is a necessity in order to estimate the unmeasurable steam flow using Kalman estimation. The state augmentation is described in Appendix F.

Looking at the parameters in Equation (7.2) and (7.3), most of them are already known or identified during the self-tuning of the Mission OSTM boiler system. Only the parameter β_{p_s} is unknown. However, β_{p_s} is by definition dependent on the mass of water, steam bubbles, steam and the metal of the boiler. Thus a qualified estimate of β_{p_s} is obtainable by evaluation of β_{p_s} in the operating point of the boiler. Hence, a Kalman estimator, based on the linear model, has potential to be tuned as a part of the self-tuning scheme for the Mission OSTM boiler system.

7.1.1 Linear Kalman Estimator Algorithm

Referring to the notation of Equation (7.1), the Kalman estimator is given by Algorithm 7.1 [Grewal and Andrews, 2001, p. 121] and shortly described in the following.

Algorithm 7.1 Linear Kalman estimator.

1. Prediction step:

$$\hat{\mathbf{x}}^-(k) = \Phi \hat{\mathbf{x}}^+(k-1) + \Gamma \mathbf{u}(k-1) \quad (7.4)$$

$$\hat{\mathbf{y}}(k) = \mathbf{H} \hat{\mathbf{x}}^-(k) \quad (7.5)$$

$$\mathbf{P}^-(k) = \Phi \mathbf{P}^+(k-1) \Phi^T + \mathbf{Q} \quad (7.6)$$

2. Correction step:

$$\mathbf{K}(k) = \mathbf{P}^-(k) \mathbf{H}^T (\mathbf{H} \mathbf{P}^-(k) \mathbf{H}^T + \mathbf{R})^{-1} \quad (7.7)$$

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k) (\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \quad (7.8)$$

$$\mathbf{P}^+(k) = (\mathbf{I} - \mathbf{K}(k) \mathbf{H}) \mathbf{P}^-(k) . \quad (7.9)$$

The prediction step consists of three steps. The first two steps is simply the discrete state space model, used to estimate the a priori states, $\hat{\mathbf{x}}^-(k)$, and outputs, $\hat{\mathbf{y}}(k)$. The last step in the prediction step is the calculation of the covariance matrix, $\mathbf{P}^-(k)$, expressing the probability distribution of the states, corresponding to the uncertainty of the estimated states.

The correction step also consists of three steps. First the Kalman gain is calculated from the covariance and sensor model. Secondly the Kalman gain and prediction error is used to correct the estimate of the system states. Finally the covariance matrix is corrected, also by using the Kalman gain. Compared to an ordinary state observer, the Kalman gain corresponds to the observer gain, but the Kalman observer gain is calculated using the stochastic properties of the disturbance and measurement noise.

In the recursive Algorithm 7.1, the steps of calculating the Kalman gain and covariance matrix can be considered unnecessary, as the values of the Kalman gain and covariance matrix converges after relatively few computations. Alternatively the Kalman gain can be found from testing the Kalman estimator in advance, and use the obtained gain as a constant factor to ease the computations of the algorithm. However, if the Kalman estimator is used in a self-tuning scheme, the update of the Kalman gain and covariance matrix must be included.

The two matrices, \mathbf{Q} and \mathbf{R} , in Algorithm 7.1 are covariance matrices for the model or process noise, \mathbf{w} , and sensor noise, \mathbf{v} , respectively. These are the only design parameters in the Kalman estimator and are used to weight whether the model output or the noisy measurements are to be trusted the most. The design of the matrices \mathbf{Q} and \mathbf{R} are in this project based on the following considerations.

There are three states, $\mathbf{x} = [p_s \ V_w \ w_s]$, in the state space model. From previously tests at AI's test center the steam pressure measurements are known to be sufficiently smooth and thus the sensor measurements are trusted a bit more than the model. For the water volume the

model is trusted over the measurements, as the water level measurements are known to be quite fluctuating. The last state is the unmeasurable steam flow, and because no measurements are available, the model has the dominating weight. This gives the following matrices:

$$\mathbf{Q} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 1 \cdot 10^{-12} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}, \quad (7.10)$$

which are used in the linear Kalman estimator.

7.1.2 Simulation of Linear Kalman Estimator

Simulating the designed Kalman estimator in MATLABTM, using the m-file `linearKalmanaug.m`, gives the results shown in Figure 7.2.

As expected, the steam pressure measurements and estimated steam pressure in Figure 7.2a are very alike, due to the choice of weights in \mathbf{Q} and \mathbf{R} . The figures of most interest are Figure 7.2c and 7.2e, showing the results of the last two states, the water level and steam flow.

Looking at Figure 7.2c the estimated water level is clearly smooth, compared to the measured water level, which is the purpose of the designed Kalman estimator. Thereby the Kalman estimator facilitates possible improved control of the water level.

The second purpose of the linear Kalman estimator is to estimate the steam flow, as shown in Figure 7.2e. The estimation shows a very satisfactory result, where the Kalman estimator is tracking the measured steam flow. On AI's commercial boilers no flow sensor is present to measure the steam flow. However, the measurements are obtained in AI's test center, where a steam flow sensor is mounted, as described in the system description in Chapter 2.

The remaining subfigures in Figure 7.2 show the estimation error for the respective states. Common for the estimation errors is a larger variation whenever a step is executed, but generally the estimation error is low. To further examine the effectiveness of the Kalman estimator, the error of the estimated steam flow is of particularly interest whenever a step is executed. Besides from a desired low estimation error, the tracking time is relevant for the effectiveness. Figure 7.2g shows that approximately 7 seconds after an applied step, corresponding to 14 samples in this simulation, the Kalman estimator has tracked the steam flow again.

7.1.3 Evaluation of the Linear Kalman Estimator

Conclusively, the linear Kalman estimator shows satisfactory results for both smoothening the water level measurements and estimating the unmeasurable steam flow. In order to improve the 7 seconds tracking time of the unmeasurable steam flow, requires a re-evaluation of either the uncertainty model, describing the steam flow, or another weight factor in the \mathbf{Q} matrix. If the model is trusted less, compared to the above example, the tracking time is reduced at the expense of a generally larger estimation error. The generally larger estimation error is caused by the noisy water level measurements, which are propagated back to the steam flow in the augmented state space model, given by Equation (7.2) and (7.3).

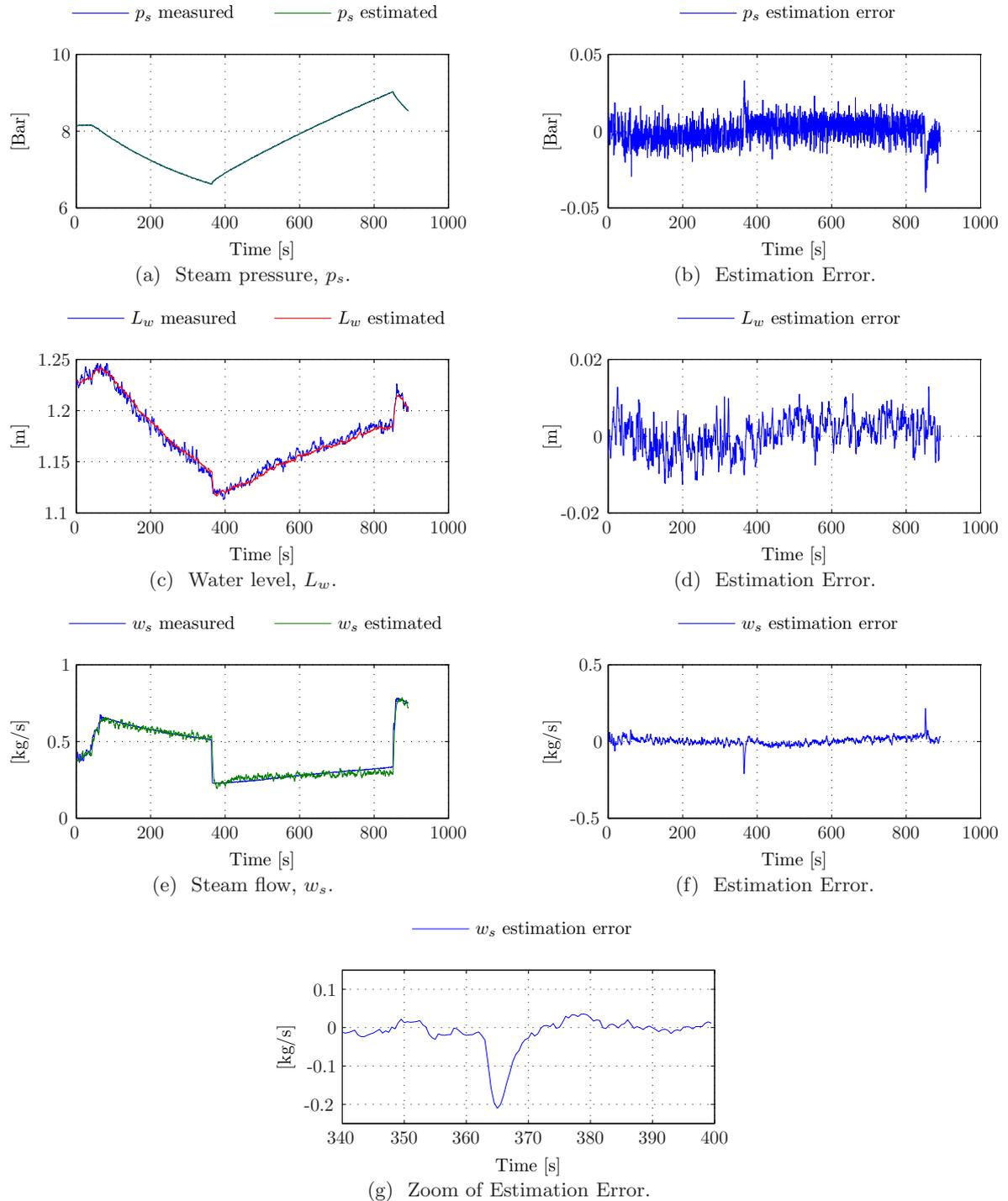


Figure 7.2: Simulation results from Kalman estimator.

In brief, the performance of the Kalman estimator depends on the choice of uncertainty model and design of the weight matrices, chosen on basis of the purpose of the Kalman estimator.

In the following, an assessment of the benefits of applying the designed Kalman estimator is carried out.

By smoothening the water level measurements, the fluctuation, caused by e.g. waves and air bubbles breaking the water surface, is smoothened in order to obtain the mean water level. The use of the estimated mean water level, as input to the water level controller, enables the design of a water level controller with higher bandwidth and thereby facilitates better level control. However, when designing the water level controller, it is important to realise that the water level deviates from the estimates.

In steady state, the feed water flow, w_{fw} , must equal the steam flow, w_s , in order to maintain a constant water level. As a result, the unmeasurable steam flow is of special interest when controlling the water level. On the basis of the evaluation of the designed Kalman estimator, the steam flow can however be estimated. Estimating the unmeasurable steam flow facilitates the possibility to use the information e.g. as feed forward to the water level controller. However, the feed forward should not be used without a feedback controller, as the estimated steam flow might deviate from the actual consumption. Hence, combining the estimate of the mean water level and the estimated steam flow, constitutes a solid basis for obtaining a better control of the water level.

Conclusion

The main objective for this project was to develop a proof of concept self-tuning system, based on a simplified system model, capable of automatically tuning the controllers, prior to operating the Mission OSTM boiler system. Other objectives were to ensure a clean combustion using a curveband, specifying the relation between oil valve and air damper positions, and to design the actuator controllers to obey the curveband at any time. The final objective for the project was to assess the benefits of using classic estimation techniques, in attempt to improve the control of the water level in the Mission OSTM boiler. To comply with the objectives for this project, it was necessary first to derive a model of the entire Mission OSTM boiler system.

The Mission OSTM boiler system was modelled as three separate non-linear models; a model of the Mission OSTM boiler, the feed water actuator system and the fuel actuator system. The purpose of the non-linear Mission OSTM boiler model was to express the steam pressure, water level and oxygen level, based on the input flows from the actuator systems, i.e. fuel, air and feed water.

To derive the non-linear boiler model, a number of control volumes were identified and the model was divided in two parts. One part describing the dynamics of the furnace and the other describing the dynamics of the water/steam. For the furnace an oxygen model was derived, based on the chemical reaction during combustion, together with an efficiency factor, describing the energy transferred from the furnace to the water/steam part. By use of energy and mass balance equations, the dynamics of the water/steam part was described. The resulting boiler model was presented on descriptor form and implemented in Simulink for verification, which showed satisfactory similarities compared to obtained measurements data.

The feed water actuator system was modelled as a non-linear function, describing the feed water flow as a function of steam pressure and input voltage to a valve positioner. Thereby neglecting the dynamics of the actuator and considering the flow sensor dynamics as dominating. The fuel actuator model consists of an oil actuator model and an air actuator model. The two actuators were similar of structure and thus also similar modelled, as integrators. For the oil actuator, a flow sensor was modelled as a second order nonminimum phase system, due to delay between forward and return oil flow. The flow characteristics of the oil actuator was modelled as a static third order polynomial. Parameters for both actuator models were estimated and the models were verified against measurement data. The models had similar dynamic behaviour as the actuator system responses. Thereby the entire non-linear Mission OSTM boiler system was successfully modelled, verified and applicable for controller design.

A control scheme of three inner-loop and three outer-loop SISO controllers were presented. The inner-loop controllers were designed to control the feed water flow, the oil valve position and air damper position, all with the purpose of optimising the conditions for the outer-loop controllers for steam pressure, water level and oxygen level.

The feed water flow controller was designed using feed-forward, to eliminate the non-linearities

of the feed water valve, and a PI-controller to obtain a fast response and eliminate steady state errors.

The servos for controlling the oil valve and the air damper were Pulse Width Modulation (PWM) controlled, and two position controllers were designed as P-controllers. The two position controllers were designed to facilitate equal response time, even in case of unequal native servo travelling times.

Prior to designing the steam pressure and water level controllers, the non-linear boiler model was reduced to a simple integrator model, which was verified using obtained measurement data. Both the steam pressure and water level controllers were designed as PI-controllers, based on a stability criteria in the frequency domain and the bandwidth of the actuator systems. Furthermore, the water level controller was designed taking the fluctuating water level into account.

To obtain a steady oxygen level, an oxygen level controller was designed, utilising both a curveband and an slow feedback controller to eliminate steady state errors.

A common goal for the controller designs was to facilitate self-tuning of the controllers, by simple model parameter identification. Finally, the performance of the designed controllers was simulated, using the derived non-linear simulation model, and showed satisfactory results.

An objective was to design a self-tuning system capable of determining system parameters and automatically tune the designed controllers. To accomplish this, four self-tuning algorithms were implemented during the start-up sequence of the Mission OSTM boiler system.

The first algorithm identified the traveling times for the oil valve and air damper, and from these the proportional gains were calculated for the fuel actuator controllers. Furthermore, the traveling times were used to determine a dead band, for the respective actuators, to avoid limit cycles. Finally, the bandwidth of the steam pressure control loop was determined using the identified parameters.

The second algorithm was used to identify the curveband for the oxygen controller, by automatically adjusting the oil valve and air damper positions, to obtain a desired oxygen level, in the entire range of the servos.

The third algorithm analysed the feed water actuator and estimated the non-linear ratio between valve position and feed water flow, used as feed-forward. Furthermore the time constant of the feed water actuator was estimated and used to calculate the PI-controller parameters for the feed water actuator controller.

The last self-tuning algorithm was designed to identify the steam pressure integrator gain, by maintaining a fixed oil valve position, used to calculate the steam pressure controller parameters. From tests at Aalborg Industries A/S, the self-tuning algorithms showed promising results, regarding identification of the control model parameters, and the self-tuned controllers were verified by non-linear simulation. The simulation showed an overall satisfactory performance. However, the chosen oil valve and air damper position controller design, causes PWM saturation in case of a non-linear curveband or sudden changes in burner load. To cope with PWM saturation, and thereby avoiding transient variation in the oxygen level, a method utilising constrained control was developed.

In attempt to solve the PWM saturation issue, a case study was carried out to proper describe the problem. Through the case study, the issue was analysed and found to be solved by a minimisation problem, constrained by the saturation limits of the fuel actuator system and the curveband. Hence, a model predictive control scheme was developed to resolve the problem using reference correction. The developed reference correction scheme was verified to solve the minimisation problem, and thereby defeat the problem of curveband violation during PWM

saturation. Using the developed method, the performance of the steam pressure controller might be reduced by the compliance of the curveband. However, the method ensures a steady oxygen level, and thereby a clean combustion.

Finally, a study of the benefits of using classic estimation techniques was carried out. A linear Kalman filter was designed to estimate the mean water level and the unmeasurable steam flow. A verification of the designed Kalman estimator showed satisfactory performance and thus forms a solid basis for improvement of the water level control.

Altogether, a model of the Mission OSTM boiler system, including actuator systems, has been derived and simplified to form a basis for a developed self-tuning scheme, capable of identifying model parameters during the start-up phase of the boiler system. Furthermore, a model predictive control scheme has been developed to comply with the curveband, implying a clean combustion. Finally, the designed Kalman estimator showed ability to estimate a mean water level and the unmeasurable steam flow, facilitating improved water level control.

Discussion

This chapter contains a discussion of future work, on the basis of the work and developed schemes during the project.

Further Development of the Designed Self-tuning Scheme

The self-tuned controller performance is verified by non-linear simulation, and showed satisfactory performance. However to finally verify the self-tuning scheme, a real-life test using the Mission OSTM boiler system must be carried.

During the development of the self-tuning scheme, several limitations were made to simplify the design. Two of the limitations is of interest, regarding future work, namely error checking during the identification and the assumption of functioning instrumentation. When relying on a self-tuning scheme, the necessity of fully operating hardware is important, and faults can be fatal for the performance of the system. Hence, it is evident to design a fault detection scheme, to deal with possible hardware errors.

Another possibility for further development of the self-tuning scheme, is to inherit the ideas of the developed self-tuning scheme to an online adaptive controller structure. Thereby, it is possible to adjust the controller parameters to the actual state of the boiler system.

Finally, it is of interest to examine the opportunities of self-tuning a Kalman estimator, used in order to improve the water level control.

State Estimation Using Kalman

A Kalman estimator was designed in this project, to estimate the mean water level in the boiler and the unmeasurable steam consumption. The purpose was to assess the potential benefits of applying the estimated quantities, to improve the water level control.

In continuation of the assessment, a future task is to design a control structure, using the estimated states, in order to achieve better water level control. It is evident to use the estimated steam flow as feed forward to the feed water, and use the estimated mean water level as feedback, to eliminate steady state errors, caused by e.g. a small steam flow estimation error. Furthermore, the use of Kalman estimation in order to filter various sensor inputs, is assessed to be beneficial.

Further use of Kalman estimation can also encompass inverse Kalman filtering, in order to estimate the covariance of sensor inputs. This information can e.g. be included in a more advanced self-tuning scheme, in order to avoid actuator stress.

Advanced Controller Types

The existing controllers on the Mission OSTM boiler system mainly consists of classic PI-controllers, which is an advantage for the developed self-tuning scheme. However, the use of more advanced controller types has through previous projects shown to be beneficial. [Solberg, 2008] developed MIMO controllers for a boiler system, and thereby included the cross coupling between water level and steam pressure. Hence, self-tuning of MIMO controllers might further improve the overall performance of a self-tuned boiler system.

During the project, the non-linearities have been remedied by use of feed-forward controllers. However, non-linear control might be beneficial to use, as the main part of the boiler system reveals non-linear behavior.

Bibliography

- Aalborg Industries [2008]. Statements from our contact at Aalborg Industries A/S.
- Andersen, P. and Pedersen, T. S. [2007], ‘Modeldannelse’. Lecture Notes.
URL: <http://www.control.auc.dk/pa/kurser/PR6model/modelnote.pdf>
- Andersen, S. and Jørgensen, L. [2006], ‘Implementation of control platform and nonlinear modeling’. 9. semester pre-Master Thesis at Aalborg University, Department of Electronic Systems, Section of Automation and Control.
- Andersen, S. and Jørgensen, L. [2007], ‘Scheme for auto tuning control of marine boilers’. Master Thesis at Aalborg University, Department of Electronic Systems, Section of Automation and Control.
- Bohn, C. and Atherton, D. [1995], ‘An analysis package comparing pid anti-windup strategies’. Paper.
- Davies, R. M. and Taylor, G. [1949], ‘The mechanics of large bubbles rising through extended liquids and through liquids in tubes’. Excerpt from a book.
URL: <http://www.jstor.org/stable/98449>
- Franklin, G. F., Powel, J. D. and Emami-Naeini, A. [2006], ‘Feedback control of dynamic systems’. Fifth Edition.
- Grewal, M. S. and Andrews, A. P. [2001], ‘Kalman filtering - theory and practice using matlab’. Second edition.
- Knudsen, M. [2004], ‘Experimental modelling of dynamic systems’. Lecture note.
URL: <http://www.control.aau.dk/te02pdf.pdf>
- Knudsen, T. [1993], ‘Systemidentifikation’. Lecture note.
URL: <http://www.control.aau.dk/jsat/Teaching/SysId/si.pdf>
- Schmidt, E. [1989], ‘Properties of water and steam in si-units’. 4th edition.
- Skogestad, S. [2002], ‘Simple analytic rules for model reduction and pid controller tuning’. Paper.
- Solberg, B. [2008], ‘Optimisation of marine boilers using model-based multivariable control’. Industrial Ph.D. Thesis at Aalborg Industries A/S and Aalborg University, Department of Electronic Systems, Section of Automation and Control.
- Solberg, B. and Hvistendahl, P. U. [2004], ‘Modelling and multi variable control of a marine boiler’. Master Thesis at Aalborg University, Department of Electronic Systems, Section of Automation and Control.

Boiler Dimensions

Figure A.1 shows a drawing of the Mission OSTM boiler. Furthermore some of the dimensions of the boiler is illustrated, which are used for calculating different volumes of the Mission OSTM boiler. The notation N.W. in the figure indicates the normal water level.

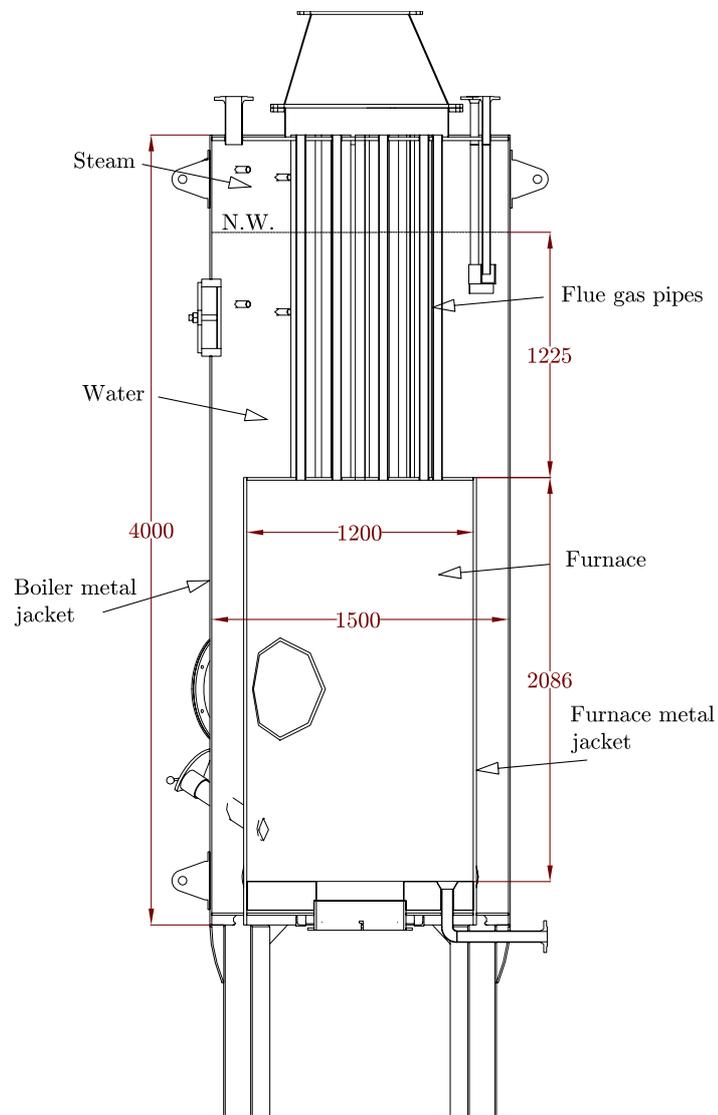


Figure A.1: Drawing of the Mission OSTM boiler and some of the dimensions. Inspired by [Aalborg Industries, 2008].

Constants

This appendix contains various tables with useful coefficient values, used for different purposes throughout the report.

Operating Point Values

Table B.1 shows the operating point values for different loads in the steam flow and the corresponding input mass flows. The input flows are calculated as the mean value over a time period of 1200 seconds in steady state, during a simulation with noise added to the steam pressure and water level measurements.

State/input	$w_s = 0.33 \frac{\text{kg}}{\text{s}}$	$w_s = 0.403 \frac{\text{kg}}{\text{s}}$	$w_s = 0.472 \frac{\text{kg}}{\text{s}}$	Unit
V_w	1.385	1.370	1.357	m^3
V_{sb}	0.063	0.077	0.091	m^3
w_{fw}	0.327	0.401	0.471	$\frac{\text{kg}}{\text{s}}$
w_{fu}	0.027	0.033	0.038	$\frac{\text{kg}}{\text{s}}$
w_a	0.453	0.553	0.648	$\frac{\text{kg}}{\text{s}}$

Table B.1: Operating point values for different steam loads at the operating point; $p_s = 8 \text{ bar}$, $L_w = 1.23 \text{ m}$, $x_{o,O_2} = 5 \%$.

Steam Table Coefficients

Table B.2 contains coefficients from a steam table, used as operating point values in the control model derivation and thereby controller design.

Constant	Value	Unit
h_w	720.94	$\frac{\text{J}}{\text{kg}}$
h_s	2767.5	$\frac{\text{J}}{\text{kg}}$
h_{fw}	105	$\frac{\text{J}}{\text{kg}}$
ρ_w	896.86	$\frac{\text{kg}}{\text{m}^3}$
ρ_s	4.16	$\frac{\text{kg}}{\text{m}^3}$

Table B.2: Steam table coefficients at 8 bar, 170.41°C , [Schmidt, 1989, p. 32], except h_{fw} .

Oxygen Model

Table B.3 contains coefficient values used for the oxygen model. The coefficients are mass fractions and molar masses for different elements of the periodic table.

Constant	Value	Unit
x_C	15/47	-
x_H	32/47	-
x_{O_2}	21/100	-
x_N	79/100	-
M_C	12.01	$\frac{\text{g}}{\text{mole}}$
M_H	1.007	$\frac{\text{g}}{\text{mole}}$
M_O	15.999	$\frac{\text{g}}{\text{mole}}$
M_N	14.01	$\frac{\text{g}}{\text{mole}}$

Table B.3: Coefficients used for oxygen model calculation.

Non-linear Boiler Modelling

In Table B.4 the coefficients used in the modelling of the non-linear boiler system and their values are listed.

Constant	Value	Unit
H_{fu}	$40e^3$	$\frac{\text{J}}{\text{g}}$
c_{fu}	$2+0.003*(T_{fu}-100)$	$\frac{\text{J}}{\text{g}\cdot\text{C}}$
c_a	1.005	$\frac{\text{J}}{\text{g}\cdot\text{C}}$
T_{fu}	10/283.15	$^{\circ}\text{C}/^{\circ}\text{K}$
T_o	20/293.15	$^{\circ}\text{C}/^{\circ}\text{K}$
T_{fl}	350/623.15	$^{\circ}\text{C}/^{\circ}\text{K}$
κ_{fu}	16.9	.
L_b	4	m
r_b	0.75	m
A_w	1.6389	m^2

Table B.4: Coefficients used for non-linear boiler model calculation.

Parameter Estimation using Senstools

This appendix describes how model parameter values, for the Mission OSTM boiler model derived in Chapter 3, are estimated from measurements obtained in AI's test center. The MATLABTM toolbox Senstools, developed by Professor Morten Knudsen, is used for the parameter estimation in this project. Thus a short introduction to Senstools is given in the following.

Senstools

Generally seen, parameter estimation is a part of experimental modelling of dynamical systems, which can be divided into the following five steps [Knudsen, 2004, p. 2]:

1. **Model structure determination**

The model structure is determined by use of physical laws and empirical considerations.

2. **Experiment design**

An appropriate input signal is chosen for the specific system. The frequency of the input signal is chosen in the range, where the accuracy of the model is most important, that is the crossover frequency. If the model is considered linear, the amplitude of the input signal is not important.

3. **Experiment**

The system is actuated by the chosen input signal, and the input and output sequences are saved for use during the parameter estimation.

4. **Parameter estimation**

The parameters for estimation are adjusted until a weight of the error, between the model output and the output of the system, is at a minimum. This is where Senstools is used. In short terms, Senstools minimizes a performance function with the use of Gauss-Newton method, when adjusting the parameters.

5. **Model validation**

The derived model and the estimated parameters are evaluated. This is most often done by comparing system response and model output, using new measurement data.

In this appendix, only parameter estimation is of interest. The procedure for parameter estimation is illustrated by the blockdiagram in Figure C.1 and shortly described in the following.

An input signal, $u(t)$, is applied to the system, resulting in a system response, $y(t)$. With a given frequency, the input signal is sampled and used as input for the simulation model. The model

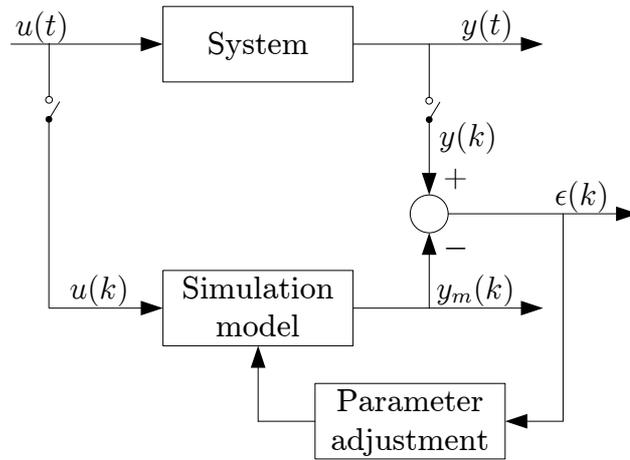


Figure C.1: Blockdiagram of the parameter estimation procedure.

output is compared to the sampled system response, $y(k)$, and the model error, $\epsilon(k)$, leads to an adjustment of the model parameters.

The parameter estimation procedure is performed offline, meaning the input signal and system response is obtained beforehand.

Senstools estimates the model parameters by minimizing the performance function given by:

$$P(\theta) = \frac{1}{2N} \sum_{k=0}^N \epsilon^2(k, \theta) . \quad (\text{C.1})$$

As mentioned previously, Senstools uses the Gauss-Newton method when minimizing the performance function. The Gauss-Newton method is iterated until the relative parameter update of each parameter is less than 0.1 %. When finished, `mainest.m` returns the estimated parameters, the normed root mean square output error and a plot of the measured output versus the model output. The root mean square error is defined as:

$$\text{ERRN} = \sqrt{\frac{\sum_{k=0}^N (y(k) - y(k, \theta))^2}{\sum_{k=0}^N y(k)^2}} . \quad (\text{C.2})$$

Control Model Validation

This appendix contains a validation of the linear boiler model consisting of the steam pressure and water level control models. These models have been analytically linearised in Section 4.2 in order to use linear controller design methods. Hence, the parameters of the linear model must be estimated and the resulting linear boiler model is simulated, to compare the linear model response with measurement data.

Parameter Estimation

From the linear steam pressure and water level control models, only three parameter values needs to be estimated, and β_{L_w} is calculated from values in a steam table. The linear control models are implemented in Simulink and using Senstools the parameter values are estimated. The parameters and the obtained parameter values are listed in Table D.1.

Parameter	Estimated value	Unit
α_{p_s}	0.0069	$\frac{\text{kg}}{\text{s}\cdot\%}$
β_{p_s}	2767.6	$\frac{\text{Pa}}{\text{kg}}$
α_{L_w}	0.15	$\frac{\text{m}\cdot\text{s}}{\text{kg}}$
$\beta_{L_w}^*$	$1.11 \cdot 10^{-3}$	$\frac{\text{m}^3}{\text{kg}}$

Table D.1: The estimated (calculated) parameter values for the linear boiler model.*

The parameter values are calculated as a mean value from several measurement series with steps in the different flows. Using the estimated parameter values, a simulated boiler response is compared to measurement data from the test center at AI. The result is shown and commented in the following.

Step in steam flow

Figure D.1 shows a comparison of the linear model response and the measured response from a step in steam flow, w_s .

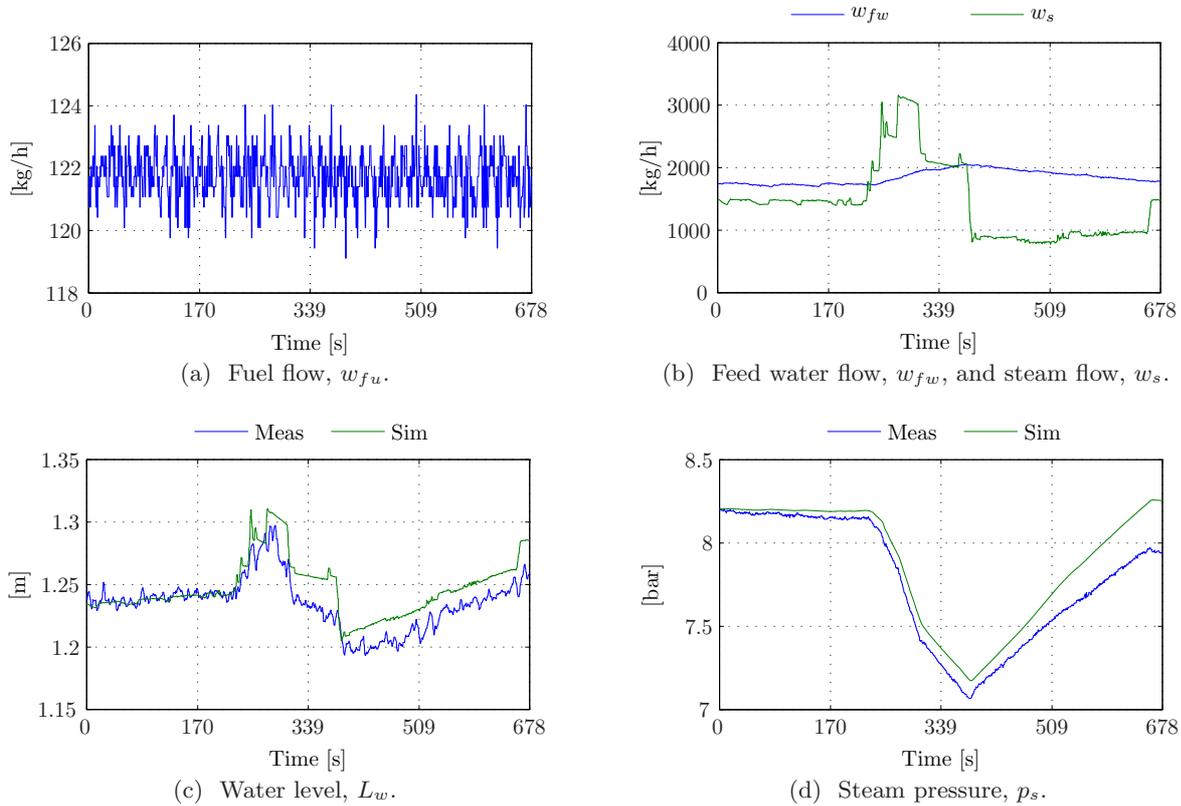


Figure D.1: Response of linear boiler model, from a step in steam flow.

Figure D.1a and D.1b shows the input flows and in Figure D.1c and D.1d the resulting simulated and measured responses are shown. Both the simulated water level and simulated steam pressure shows satisfactory similarity to the measurements. This result is expected as the steam flow affects both the water level and steam pressure control models.

Step in feed water flow

Figure D.2 shows a comparison of the linear model response and the measured response from a step in feed water flow, w_{fw} .

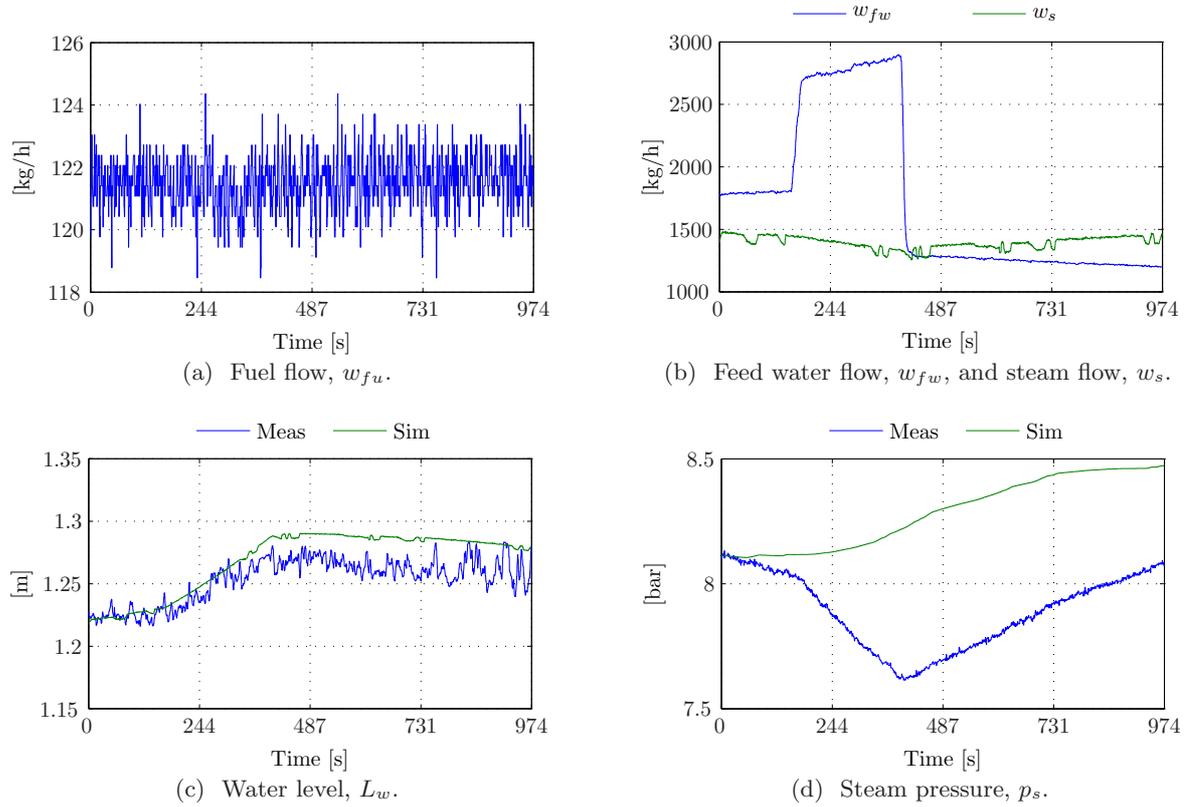


Figure D.2: Response of linear boiler model, from a step in feed water flow.

Figure D.2a and D.2b shows the input flows and in Figure D.2c and D.2d the resulting simulated and measured responses are shown. The simulated water level response is similar to the measurements, but the simulated steam pressure shows greater variation. However, as the linear steam pressure control model does not encompass the cross coupling from the water/steam part, the simulated response is expected and considered acceptable.

Step in fuel flow

Figure D.3 shows a comparison of the linear model response and the measured response from a step in fuel flow, w_{fu} .

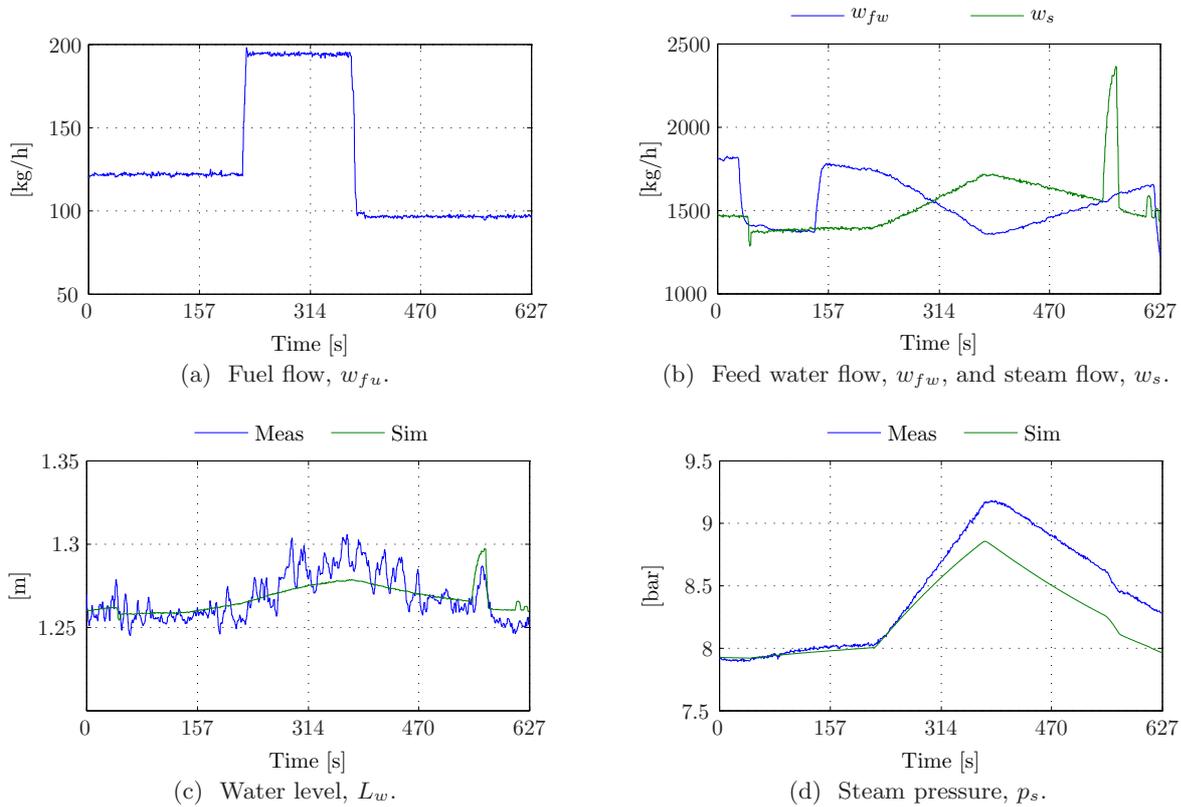


Figure D.3: Response of linear boiler model, from a step in fuel flow.

Figure D.3a and D.3b shows the input flows and in Figure D.3c and D.3d the resulting simulated and measured responses are shown. Both the simulated water level and simulated steam pressure shows satisfactory similarity to the measurements. In spite of the water level control model encompass no cross coupling from the steam part, the water level response is dependent on the steam flow, which gives the resulting characteristics.

Conclusion

From the comparison of the linear boiler model response and the measurement data, it is obvious that the linear control models are well suited for controller design. Conclusively the non-linear boiler system can easily be modelled by two simple integrators without losing the essential dynamic behaviour.

Linearisation using Taylor Series Expansion

This appendix describes how first order Taylor series expansion is applied to linearise the non-linear oxygen model and thereby facilitating the use of linear controller design.

Non-linear Oxygen Model

The non-linear oxygen model, derived in Subsection 3.1.3 on page 23, is repeated here and the constant terms are replaced by $k_{(\cdot)}$:

$$\begin{aligned}\dot{x}_{o,O_2} &= \frac{1}{\tau_{x_{o,O_2}}} \left(\frac{\frac{x_{O_2}}{M_a} w_a - \frac{x_C + \frac{1}{4} x_H}{M_{fu}} w_{fu}}{\frac{x_{O_2} + x_N}{M_a} w_a + \frac{\frac{1}{4} x_H}{M_{fu}} w_{fu}} \right) - \frac{1}{\tau_{x_{o,O_2}}} x_{o,O_2} \\ &= \frac{1}{\tau_{x_{o,O_2}}} \left(\frac{k_1 w_a - k_2 w_{fu}}{k_3 w_a + k_4 w_{fu}} \right) - \frac{1}{\tau_{x_{o,O_2}}} x_{o,O_2} .\end{aligned}\quad (\text{E.1})$$

Equation (E.1) can also be represented on a more general form, as a function of the input variables:

$$\dot{x}_{o,O_2} = f(w_a, w_{fu}, x_{o,O_2}) .\quad (\text{E.2})$$

First Order Taylor Series Expansion

The non-linear model can be linearised using a first order Taylor series expansion, where the small signal model is given by:

$$\begin{aligned}\overset{\Delta}{\dot{x}}_{o,O_2} &= \left. \frac{\partial f(w_a, w_{fu}, x_{o,O_2})}{\partial w_a} \right|_{w_a = \bar{w}_a} \cdot \overset{\Delta}{w}_a \\ &+ \left. \frac{\partial f(w_a, w_{fu}, x_{o,O_2})}{\partial w_{fu}} \right|_{w_{fu} = \bar{w}_{fu}} \cdot \overset{\Delta}{w}_{fu} \\ &+ \left. \frac{\partial f(w_a, w_{fu}, x_{o,O_2})}{\partial x_{o,O_2}} \right|_{x_{o,O_2} = \bar{x}_{o,O_2}} \cdot \overset{\Delta}{x}_{o,O_2} ,\end{aligned}\quad (\text{E.3})$$

where \bar{w}_{fu} and \bar{w}_{fu} denotes the operating point value and small signal value respectively.

By calculating each partial derivative term in Equation (E.3) and reducing the expression, the linearised small signal model for the oxygen model becomes:

$$\dot{\hat{x}}_{o,O_2} = \frac{(k_1 k_4 \bar{w}_{fu} + k_2 k_3 \bar{w}_{fu}) \hat{w}_a - (k_2 k_3 \bar{w}_a + k_1 k_4 \bar{w}_a) \hat{w}_{fu}}{\tau_{x_o,O_2} (k_3^2 \bar{w}_a^2 + k_4^2 \bar{w}_{fu}^2 + 2k_3 k_4 \bar{w}_a \bar{w}_{fu})} - \frac{1}{\tau_{x_o,O_2}} \hat{x}_{o,O_2} \quad . \quad (\text{E.4})$$

Laplace Transformation

By Laplace transforming the linearised small signal model and assuming constant oil flow, $\hat{w}_{fu} = 0$, a transfer function from air flow to oxygen level is given by:

$$\frac{\hat{x}_{o,O_2}(s)}{\hat{w}_a} = \frac{\left(\frac{k_1 k_4 \bar{w}_{fu} + k_2 k_3 \bar{w}_{fu}}{k_3^2 \bar{w}_a^2 + k_4^2 \bar{w}_{fu}^2 + 2k_3 k_4 \bar{w}_a \bar{w}_{fu}} \right)}{\tau_{x_o,O_2} s + 1} \quad . \quad (\text{E.5})$$

Linear Oxygen Model Verification

To verify the derivation of the linear oxygen model (E.5), the response is compared to the non-linear model response, as showed in Figure E.1.

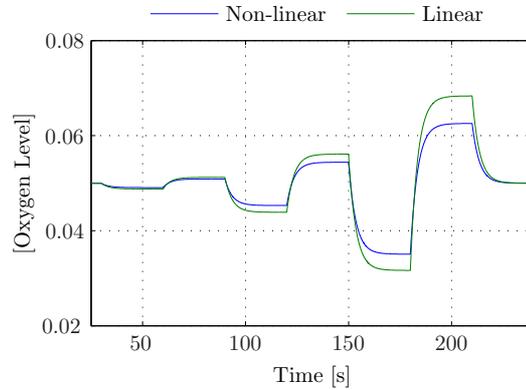


Figure E.1: A comparison of the non-linear and linear oxygen model response.

As expected the responses are very alike close to the operating point of 5% oxygen level. However, the similarity of the responses is decreased when moving away from the operating point. Thus the linear oxygen model (E.5) is considered suitable for linear controller design.

Derivation of Linear State Space Model

This appendix describes the derivation and analysis of a linear state space model of the Mission OSTM boiler, used for Kalman estimation.

Linear State Space Model

The state space model is outlined from the control models, derived in Chapter 4, and are repeated here:

$$\dot{p}_s = \underbrace{\frac{h_s - h_w}{\left(m_w \frac{\partial h_w}{\partial p_s} + m_{sb} \frac{\partial h_s}{\partial p_s} + m_s \frac{\partial h_s}{\partial p_s} + m_m c_m \frac{\partial T_{sat}}{\partial p_s}\right)}}_{\beta_{ps}} \left(\underbrace{\frac{\eta_{fu} \beta_{fu} k_{fu}}{h_s - h_w}}_{\alpha_{ps}} \theta_{fu} - w_s \right) \quad (\text{F.1})$$

$$\dot{V}_w = \frac{1}{\underbrace{\rho_w - \rho_s}_{\beta_{Lw}}} (w_{fw} - w_s) \quad (\text{F.2})$$

$$L_w = \frac{1}{A_w} V_w + \frac{t_{sb}}{\underbrace{\rho_s A_w}_{\alpha_{Lw}}} w_s . \quad (\text{F.3})$$

These three equations are rearranged into a state space form as given below:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{d}(t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{F}\mathbf{d}(t) + \mathbf{v}(t) , \end{aligned}$$

where $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are uncorrelated zero mean white noise. The disturbance $\mathbf{d}(t)$ is here regarded as the steam flow, which is only measurable in AI's test center. A block diagram of the state space model is illustrated in Figure F.1.

Using Equations (F.1)–(F.3) gives the linear state space model:

$$\begin{bmatrix} \dot{p}_s \\ \dot{V}_w \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \end{bmatrix} + \begin{bmatrix} \gamma_{ps} & 0 \\ 0 & \beta_{Lw} \end{bmatrix} \cdot \begin{bmatrix} \theta_{fu} \\ w_{fw} \end{bmatrix} + \begin{bmatrix} -\beta_{ps} \\ -\beta_{Lw} \end{bmatrix} \cdot w_s \quad (\text{F.4})$$

$$\begin{bmatrix} p_s \\ L_w \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{A_w} \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_{fu} \\ w_{fw} \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_{Lw} \end{bmatrix} \cdot w_s . \quad (\text{F.5})$$

Here the added white noise is omitted and the steam flow, w_s , is considered as an unknown disturbance as input to the system model and output model.

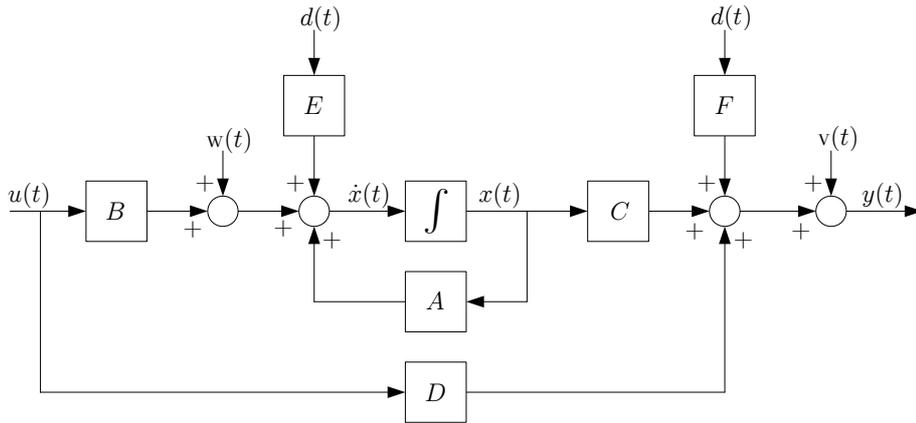


Figure F.1: Block diagram of the linear state space model of the Mission OS™ boiler.

Linear State Space Model Verification

To verify the linear state space model, the model is discretised using the MATLAB™ function `c2d()` and the discrete model parameters are fitted using Senstools. For the verification of the model, also the steam flow is measured. A plot of the measured output and the model output is shown in Figure F.2, and the result is satisfactory.

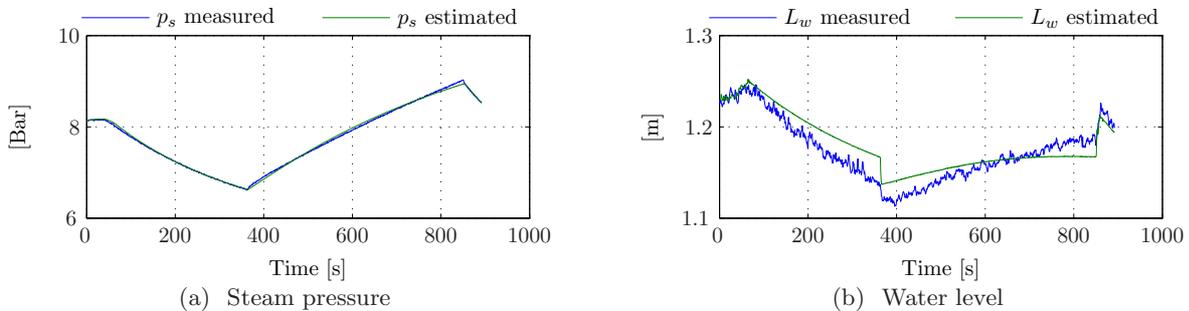


Figure F.2: Verification of the linear state space model.

Linear State Space Model Augmentation

In order to estimate the steam flow, w_s , the disturbance must be relocated to the state vector, equivalent to augmenting the state vector. This requires a model of the steam flow and by assuming the steam flow can be modelled as a constant, the following augmented state space model is obtained:

$$\begin{bmatrix} \dot{p}_s \\ \dot{V}_w \\ \dot{w}_s \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\beta_{p_s} \\ 0 & 0 & -\beta_{L_w} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \\ w_s \end{bmatrix} + \begin{bmatrix} \gamma_{p_s} & 0 \\ 0 & \beta_{L_w} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_{fu} \\ w_{fw} \end{bmatrix} \quad (\text{F.6})$$

$$\begin{bmatrix} p_s \\ L_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{A_w} & \alpha_{L_w} \end{bmatrix} \cdot \begin{bmatrix} p_s \\ V_w \\ w_s \end{bmatrix}, \quad (\text{F.7})$$

where the parameter values of the augmented state space model are given in Table [F.1](#).

Element	Value
β_{p_s}	3251
β_{L_w}	0.00157
γ_{p_s}	46016
α_{L_w}	0.1052
A_w	1.6389

Table F.1: Parameters for the augmented linear discrete state space model.

A more accurate disturbance model for the steam flow could probably be used, but this would require an extensive analysis of the steam consumption on different vessels. This is considered too comprehensive for this objective, as the designed Kalman estimator in Chapter [7](#) shows satisfactory results.

Observability of Augmented State Space Model

In order to use the discrete linear state space model for Kalman estimation, the system must be observable. From the augmented discrete state space form:

$$\begin{aligned} \mathbf{x}_a(k+1) &= \mathbf{\Phi}_a \mathbf{x}_a(k) + \mathbf{\Gamma}_a \mathbf{u}_a(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{H}_a \mathbf{x}_a(k) + \mathbf{v}(k), \end{aligned}$$

the observability can be examined from the canonical observability matrix, \mathcal{O} , given by [\[Franklin et al., 2006, p. 502\]](#):

$$\mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{\Phi} \\ \vdots \\ \mathbf{H}\mathbf{\Phi}^{n-1} \end{bmatrix}. \quad (\text{F.8})$$

In order for the system to be observable, the observability matrix must have full column rank. The column rank is verified using the MATLABTM function `rank()`, which is 3 and hence all the states are observable. Equations [\(F.6\)](#) and [\(F.7\)](#) are the augmented linear discrete state space model used for Kalman estimation in Chapter [7](#).