

Jørn Bøni Hofstad

Waste incineration

Optimal
control of process dominated
by delay and uncertainty

Master's thesis in MTTK
Supervisor: Fredrik Dessen
August 2020

Jørn Bøni Hofstad

Waste incineration

Optimal
control of process dominated
by delay and uncertainty

Master's thesis in MTTK
Supervisor: Fredrik Dessen
August 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics



Waste incineration: Optimal control of process dominated by delay and uncertainty

Combustion of municipal waste comes with the issues of great uncertainty when it comes to the properties of the waste. Noxious gases are released if the combustion is incomplete. As a result, there are strict regulations the maximum concentration of CO , acidic gases, like NO_x , HCl and HF and heavy metals, like Cd and Hg . Since the properties of the waste are in-homogeneous, it becomes difficult to maintain constant steam production

Abstract

The process of burning MSW¹ is a rather complex one, with multiple objectives, non-minimum phase responses long time-constants in combination with very short ones, large disturbances because of varying waste quality. During the last years, stricter regulations have been put in with the regards to the release of hazardous materials. As a result, SINTEF Energy Research has done further research on the subject of modelling and controlling combustion processes. This project attempts to use the additional information that is gained from having a sensor estimating the heating value of the waste, as well as some of the additional information gained from using a more complex model when controlling the plant.

¹Municipal Solid Waste

Sammendrag

Forbrenning av restavfall² er en ganske kompleks prosess, som er preget av å ha både lange og korte tidskonstanter, flere referanser som skal følges samtidig, i tillegg til ikke-minimum fase-responser og store forstyrrelser på grunn av avfallets varierende sammensetning. I løpet av de siste årene har det kommet strengere reguleringer for utslipp av skadelige stoffer ved avfallsforbrenning. På grunn av dette har SINTEF Energi gjort videre forskning på modellering og regulering av denne prosessen. Dette prosjektet forsøker å utnytte den ekstra energien av å ha en sensor som estimerer varme-verdien til avfallet som forbrennes, sammen med den ekstra informasjonen som kommer av å regulere et anlegg med en mer kompleks modell.

²Restavfall oversettes ofte til Municipal Solid Waste

Contents

Waste incineration: Optimal control of process dominated by delay and uncertainty	i
Abstract	ii
Sammendrag	iii
Preface	xi
1 Introduction	1
1.1 Motivation	1
1.1.1 Literature review	1
1.1.2 Assumptions	2
1.1.3 Contributions	3
1.1.4 Outline	3
2 Process overview	5
2.1 Process overview	5
3 Features of the combustion process	9
3.1 Input step-responses	9
3.2 Disturbance step-responses	12
3.3 Some observations	14

3.3.1	Justifying the use of an HHV-estimator	14
3.3.2	Usage of air when controlling steam production	14
4	Parameter estimation	19
4.1	Eigensystem Realisation Algorithm	20
4.1.1	Handling Hysteresis	26
5	Controllers	29
5.1	Model Predictive Control	29
5.1.1	Linear MPC	31
5.1.2	Stabilising an MPC	34
5.1.3	Reference-tracking MPC	34
5.2	PID-control	36
5.2.1	Skogstad's Internal Mode Control	36
5.2.2	Simplifying plant models	36
5.2.3	PID-rules	38
5.2.4	PID-tuning by using bode-plots	40
5.2.5	Gain and phase margins of stable systems	43
6	Automated tuning	49
6.1	Nelder-mead	49
6.1.1	Theory	50
6.1.2	Issues	52
7	Experimental results	53
7.1	Parameter estimation results	54
7.1.1	Choosing a model order	54
7.1.2	Expanding the model with disturbances	58
7.2	PID-results	61
7.2.1	AB-controller	61
7.2.2	Cascaded A/B controller	65
7.3	MPC results	69

7.3.1	Different LQE structures	70
7.3.2	Integral LQR	72
7.3.3	LQR with integral cost on $F_{aI} - F_{aII}$	75
7.3.4	Discrete MPC	77
7.4	Robustness analysis	79
7.4.1	Combined estimator and physical system	79
7.4.2	The root locus plots	81
8	Conclusions and future work	85
8.1	Conclusion	85
8.2	Future work	86
8.3	Possibly useful improvements	86
A	Linear Quadratic Regulator	89
B	Feed forward control	93
	References	101

List of Tables

List of Figures

2.1	Overview of the plant	6
3.1	Step-response form the Manipulable inputs	10
3.2	Change in outputs, given step-changes in the inputs. Adjusted for the operating point, and scaled for the step-amplitude	11
3.3	Step-response to the different kinds of disturbances	13
3.4	Change in outputs, given step-changes in the the disturbances. Adjusted for the operating point, and scaled for the step-amplitude . . .	15
3.5	Change in $H\hat{H}V$ and F_{st} as percentage of total change. Given as percentage to make them comparable.	16
3.6	Change in outputs, given step-changes in the the disturbances. Adjusted for the operating point, and scaled for the step-amplitude . . .	17
5.1	Poles ρ and zeros λ of a transfer-function. The existence of ρ_3 makes it unstable	42
5.2	A generic bode-plot of $\frac{s-0.3}{s^2+s+1}$	43
5.3	N for a poorly controlled process	45
5.4	M for a poorly controlled process	47
6.1	The 4 possible update-steps in Nelder-Mead's optimisation algorithm	51
7.1	$\frac{\sum_{i,i}}{\sum_{j=1}^N(\Sigma_{j,j})}$ from the SVD $U\Sigma V^* = H_1$	55

7.2	Approximation from un-scaled measurements. The lack of scaling causes an over-fixation on $H\hat{H}V$ and v_{grate}	56
7.3	Approximation from scaled measurements. The relative error is now much smaller	57
7.4	$\frac{\sum_{i,i} \dots}{\text{sum}_{j=1}^N(\Sigma_{j,j})}$ from the SVD $U\Sigma V^* = H_1$	59
7.5	Model that also shows the dynamics of disturbances	60
7.6	Norm of $y(t) - y_{\text{ref}}(t)$ at each improved time-step of the Nelder-Mead optimisation.	62
7.7	A/B controller structure. Setpoints are assumed	62
7.8	A/B controller disturbance step response	63
7.9	A/B controller disturbance step response	64
7.10	Cascaded A/B controller structure	66
7.11	Cascaded A/B controller disturbance step response	67
7.12	Cascaded A/B controller disturbance step response	68
7.13	All controllers that assume all disturbances to only come from Q_{grate} perform poorly. Only the last one is not based on doing that	71
7.14	Outputs: LQR versus PIDs	73
7.15	Inputs: LQR versus PIDs	74
7.16	Outputs: PID vs LQR with integral cost on $F_{aI} - F_{aII}$	75
7.17	Inputs: PID vs LQR with integral cost on $F_{aI} - F_{aII}$	76
7.18	Outputs: PID vs MPC with integral cost of $F_{aI} - F_{aII}$	77
7.19	Inputs: PID vs MPC with integral cost of $F_{aI} - F_{aII}$	78
7.20	Root locus plot for delays on all plant inputs	82
7.21	Root locus plot for linear scalings of B	83
7.22	Root locus plot for unmodeled time-delays	84
B.1	Comparison of the different controllers with stochastic disturbances	95
B.2	Idealised output of a feed forward controller with $\alpha = 1$	96
B.3	Amplitude of $F_{aI} - F_{aII}$ for the idealised steam feed forward controllers, given the same input-step as in figure B.2	97
B.4	Outputs: Practical feed forward. The results became worse.	98

B.5 Inputs: Practical feed forward. The inputs change a lot more than what is normal for ordinary operation 99

Preface

This master's thesis is a part of the requirements for the master degree at the Department for Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). The thesis was done under the supervision of Associate professor Fredrik Dessen and in collaboration with SINTEF Energy Research under the additional supervision of Dr Elisa Magnelli, and Dr Michael Becidan.

The thesis is not based on a previous specialisation project. However it is based on previous work at SINTEF Energy Research, as well as a report of a previous summer-student, Herman Øie Kolden. As with all of the reports from the summer-students at SINTEF, the report by Øie Kolden (2019) was never published.

SINTEF provided an in-house simulator developed during the Waste-to-Energy 2030 project, co-funded by industry and public partners and the Research Council of Norway under EnergiX program (WtW, 2030, 280949).

The model was expanded to include an implementation of an A/B-controller and a cascaded A/B controller through the work done in Øie Kolden (2019). This model was used in chapter 7. A Model Predictive Controller had also been developed using one of MATLAB's toolboxes, but due to the relatively long sampling-time, it had no way to keep up with the performance of the A/B-controller. The implementation of the Eigensystem Realisation Algorithm was based on the video-lectures on YouTube by Steve Brunton. All plots were made using MATLAB's built-in functions. The same goes for most complex mathematical operations that have a build-in MATLAB-function, like singular value decomposition, and conversion between continuous and discrete

Linear Time-Invariant systems.

I would especially like to thank one of my supervisors, Elisa, for taking time out of her day, helping me with the mysterious problems that resulted from the simulator. Also, I'd like to thank Fredrik, as well, for answering most of my control-related questions and for advising me to take more time, so that I could deliver a more functioning product. Finally, I'd like to thank my family for giving me the time, space and support needed for completing this project.

*Jørn Bøni Hofstad
Trondheim, August 2020*

Chapter 1

Introduction

1.1 Motivation

Combustion of Municipal Solid Waste(MSW) is a process that has gained increased attention in the last years. This is both as a result of a steady increase in produced waste, as well as stricter regulations, which means that the plants have to follow stricter standards with regards to both gas emission and management of the solid ash left behind after the combustion. One way to ensure all conditions are met is to find some operating point that is know for causing the system to behave in the desired manner, and then using a controller to make sure the system stays near that state, regardless of any disturbances that may affect the process. A solution using cascaded PID-controllers has already been made by SINTEF. But that model will not be able to take full advantage of how to combine different inputs in such a way disturbances are suppressed, while not affecting other outputs too much.

1.1.1 Literature review

There has already been quite a bit on work on the subject of waste incineration, like in Leskens (2013). Furthermore, there has been some interest in using additional sensors

to the ones that are normally used when controlling such a plant. In Øie Kolden (2019), an estimate of the heating value was used to give a controller that could handle changes in waste-quality more effectively. Leskens (2013) tested several different implementations for how to control waste-combustion plants. Waste-incineration suffers from the typical problem that when one tries to increase one of the measured variables by increasing one of the inputs, there will usually be a rather large overshoot, as well as some states decreasing before they start to increase again. This is what is known as a non-minimum phase system and they are usually more difficult to control. Furthermore, changing one input usually affects all outputs. This makes it so that there are other controllers than PID-controllers that may be able to give better results. This thesis will attempt to improve on the solution in Øie Kolden (2019) by estimating a system with the method given in Juang and Suzuki (1988), and then combining the acquired model with standard linear dynamical systems theory, like what is found in Boyd (2008b) to make a better controller. Finally, the estimated model may also be used for gaining a feeling for how sensitive the rest of the system is to disturbances, measurement and modelling errors. The method will be repeated on a selection of different model-based linear controllers with slightly different architecture to find the one that has the best performance while also having some indicator of its robustness.

1.1.2 Assumptions

Since the main implementation is based on a linear model, the biggest assumption about the plant is that it behaves sufficiently similar to a linear time-invariant system. Furthermore, it is assumed that any time-delay in the control-loop remains within certain borders. Additionally, it is assumed that all modelling errors are on the form of pure time-delays or in the form of incorrect scaling for the input-matrix B , and that these errors vary so slowly that they can be seen as piece-wise constant when stabilising the plant and that these periods are so long that the controller gets the time to stabilise the plant. The implementation that was used also assumes that the process disturbances have a much larger effect on measured signals than the measurement-noise as long as a low-pass filter with a cutoff-frequency of $1 \left[\frac{rad}{s} \right]$ is used. However,

this assumption can be changed later by using a new filter and re-tuning the weights in the controller and the state estimator.

1.1.3 Contributions

The contributions from this project mainly come in the form of implementations of the different controllers, as well as an implementation of the Eigensystem Realisation Algorithm. All steps needed for the algorithm to work can be found in Juang and Suzuki (1988). Two different estimated systems were implemented. One that only considers how the controlled disturbances affect the outputs, and one that also uses the measured disturbances from the simulation to give an optimistic estimate of the increase in performance if they could be measured. It can also be used for getting a better estimate of how the process-disturbances affect the output, which can be used when designing the estimator.

Making a Linear Quadratic Estimator that is decoupled from the rest of the controller also allows for properly utilising the sensor measurements for state-estimation, since a Model Predictive Controller needs more computation time than a Kalman filter.

Finally, some different controller architectures were explored, which had the purpose of making the controller follow the desired steady-state, since some combinations of inputs can only be allowed to handle transient inputs, and should not be used to make the controller reach its steady-state. Finally, some robustness analysis was also done on the linear, estimated system to give some kind of guess to how robust the controller might be.

1.1.4 Outline

The thesis is divided into N chapters.

- 2: Process overview
- 3: Features of the process
- 4: Parameter estimation

- 5: Controllers
- 7: Experimental result
- 8: Conclusion and future work

Chapter 2 will explain what happens in the process, without going in-depth about what mathematical model is required to simulate the given system. This is mostly due to the complexity of a combustion process, which results in a proper explanation of how the process works being outside the scope of this thesis.

Chapter 3 will cover the qualitative properties of the system's step responses when the system is excited around the operating point. This may give some information as to what issues will be the most pressing when trying to control the plant.

The different controller architectures will be discussed in chapter 5, namely two PID-controller architectures, as well a few methods for optimal control. Namely, the Model Predictive Controller and the Linear Quadratic Regulator. Different architectures can be tested, depending on if there is supposed to be some kind of cost on the system for excessively increasing the primary air while reducing the secondary air.

Chapter 7 describes the process of tuning the controllers, as well as how the resulting closed-loop system respond to disturbances and noise. Because of the system being somewhat inaccurate when it comes to representing how the system will most likely act in reality.

Finally, Chapter 8 covers suggestion for future work that could expand and improve the results of the current project.

Chapter 2

Process overview

A detailed description of the dynamic model used in the present work can be found in *Dynamic modeling of municipal solid waste incineration* (2020). As a result, there will only a quick summary of the process will be given in this thesis.

2.1 Process overview

As seen in Figure 2.1, the waste is normally delivered to the waste-bunker (1). There, a set of cranes are used to mix it, to make it more homogeneous, decreasing how much the waste-composition varies, and therefore the disturbances. Some of it is then fed into the charging hopper (3) by the same cranes. A ram (4) pushes the waste onto the waste-grates (5). While on the grate, the primary air-flow (17) dries the waste and provides oxygen for the primary combustion. The conversion process taking place on the grates can be seen as divided into three sections. The waste is dried in the first section. On the second section of the grate, the primary combustion happens, where most of the heat is produced. The third part of the grate, any post-combustion happens to ensure that the amount of combustible material in the ash is below a

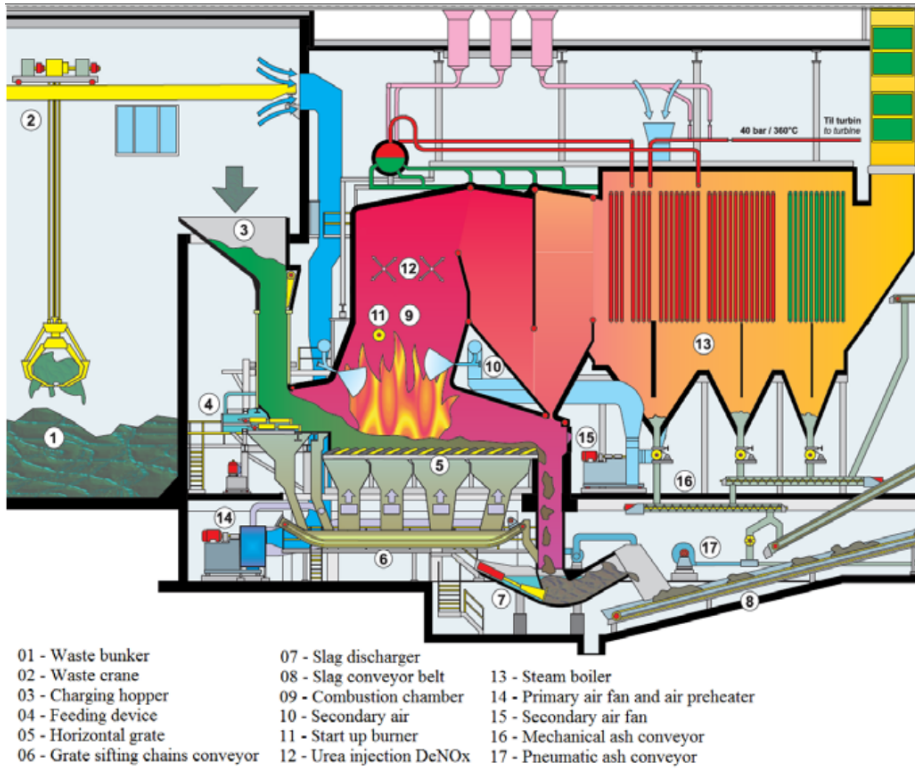


Figure 2.1: Overview of the plant

certain threshold¹. The non-combustible ash is then removed(17) from the system to be cleaned and disposed of. The combustion also results in the production of various gasses, which together with the air that did not react make up the flue-gas. Somewhere above the second part of the grate, a secondary airflow is also applied to the volatile gases resulting from the primary combustion. This is where secondary combustion occurs, producing more heat through the further conversion of the gases produced by conversion on the grate (e.g. CO , H_2 and CH_4). The primary and secondary airflow(10) provide oxygen to the fire so that the combustion is complete. The two air-flows are added at two different points to allow for better mixing between flue gases and oxygen. Proper mixing is a requirement for having complete combustion and for reducing the formation of NO_x and other harmful gases. The heat-exchange between the burning waste and the flue gas happens through contact, the mass-transfer of gases produced during combustion and by radiation. The flue gas in the combustion chamber(9) has a temperature of roughly $850^{\circ}C$. By law, the flue gas need to stay above $850^{\circ}C$ for at least 2 seconds. This is to ensure that no dangerous gases are produced. The flue gas travels through multiple bends, called passes. The passes make up the first section of the boiler (i.e. evaporator). The walls in the passes are made up of pipes where saturated water flows. When the gas flows past the pipes, it gives off heat to the walls, turning the saturated water into saturated steam. The second part of the boiler(13) (i.e. superheater), is where the gas is used to heat a flow of saturated steam that is extracted from the boiler drum. The final heat exchange section is called the economizer. Here, the cold water that is feed into the boiler drum is pre-heated to better take advantage of all the heat in the flue gas. Finally, the flue gas exits out of the system that is to be controlled in this thesis and into the part of the plant where it is cleaned before being released into the atmosphere. The composition of the flue gas that is released will depend on the amount of oxygen, how well oxygen and flue gas was mixed during combustion, the temperature in the combustion chamber, as well as the composition of the waste that is being burned.

¹In addition to the environmental damage that may come from exceeding these limits for too long, the limits are also enforced by law

Chapter 3

Features of the combustion process

3.1 Input step-responses

Before doing some actual system identification, it might be useful to do some experiments to get an overview of how the system behaves, as it might give some intuition as to how to control the plant. Figure 3.2 shows the result of an experiment where the first 19000 seconds were used to allow the plant to reach a stationary point, while a constant input-value was held, without any disturbances. At 20000 seconds, one of the input-variables is increased. The point at which the step occurs is shown as a red cross on the graph. Both the inputs and the outputs operate at completely different operating points, and with at completely different orders of magnitude. To better illustrate what is going on, figure 3.1 shows a scaled version of the step-response, where the operating point has been subtracted and the output have been scaled to correspond to what "would have been" if the step-amplitudes had been unity instead.

As can be seen from figure 3.1, the speed at which the plant reacts with is very different, depending on the input and the output. The steam-production normally

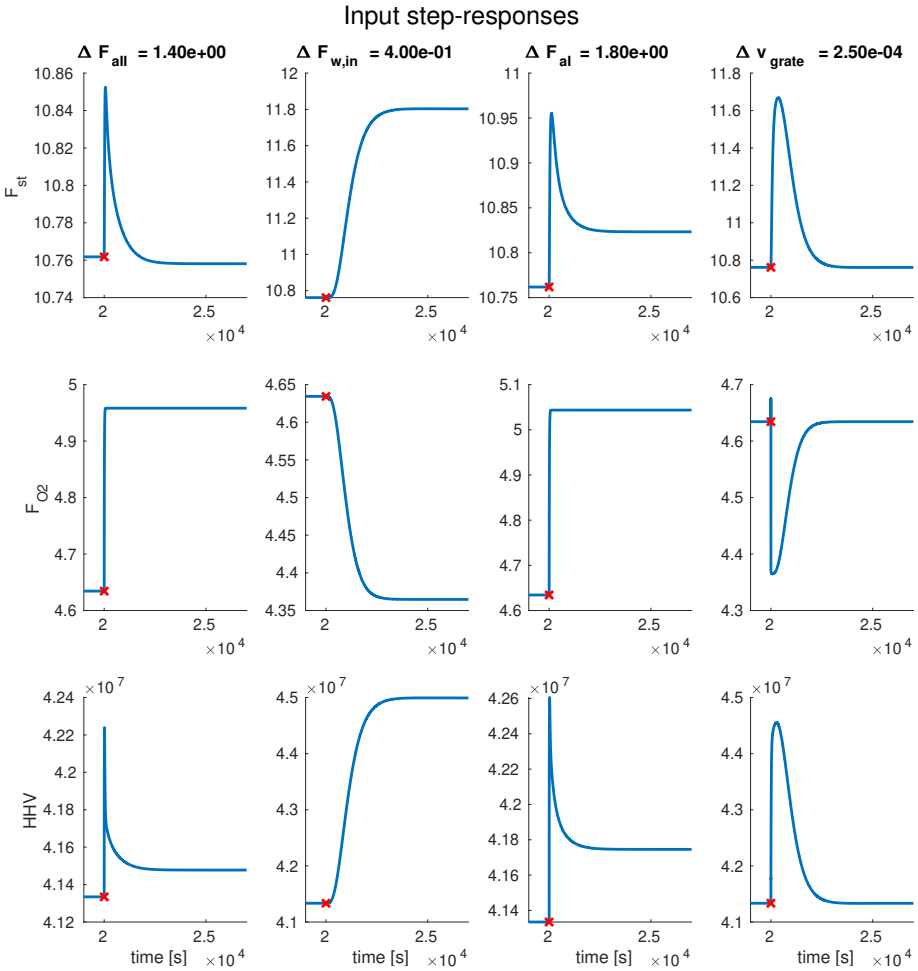


Figure 3.1: Step-response form the Manipulable inputs

reacts rather slowly to any changes in the amount of waste fed into the plant. This is usually in the realm of thousands of seconds. On the other hand, changing the input of primary or secondary air works almost instantaneously on the mass-flow of air, while

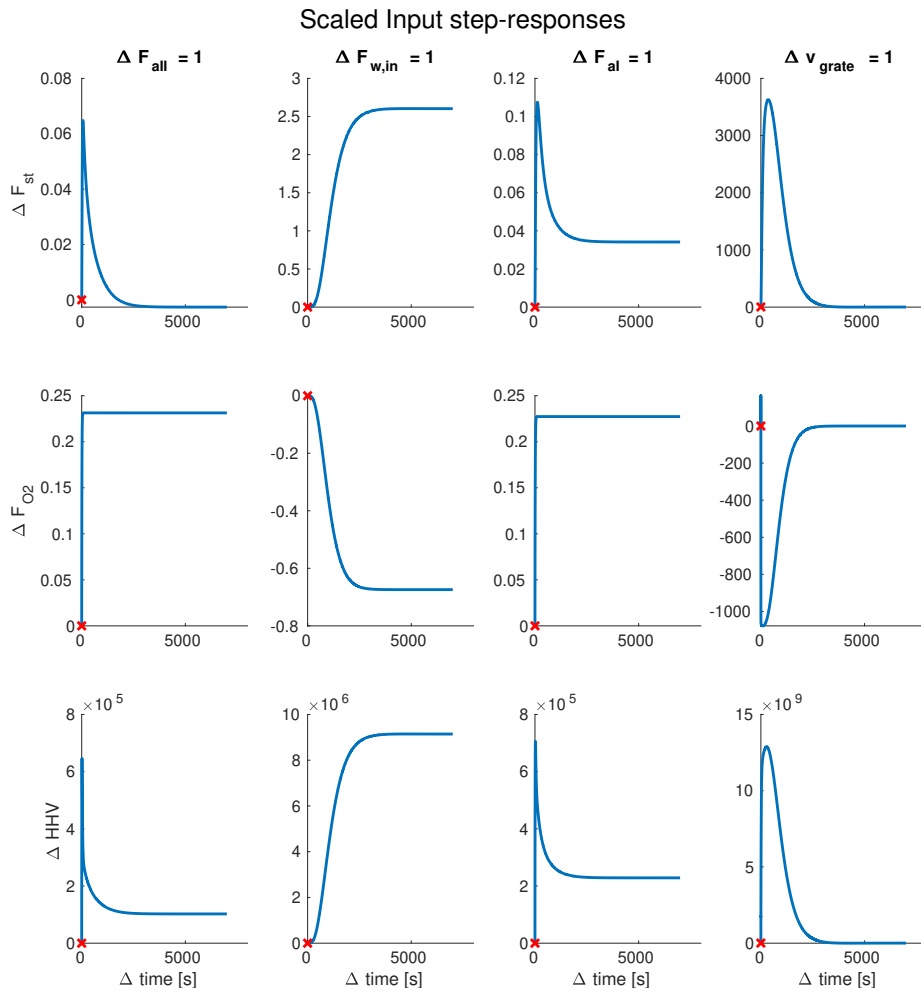


Figure 3.2: Change in outputs, given step-changes in the inputs. Adjusted for the operating point, and scaled for the step-amplitude

also having a fast effect on the steam-production.

The primary air is already pre-heated. Additionally, an increased flow of oxygen

flowing through the waste also helps to make more oxygen available for the primary combustion process, which speeds up the process. *Dynamic modeling of municipal solid waste incineration* (2020) explains that increasing oxygen concentration increases the reaction rate. The effect rapidly decreases again, as the available waste/gases are "used up".

Since more cold air is added to the new flue gas, the result will be that it gets more diluted, and the temperature will drop somewhat as a result. The result will be that the delivered power will go back to roughly the old amount.

It is worth noting that the plant is very sensitive to changes in v_{grate} . This is because it is normally only made to operate at a couple millimetres per second. As a result, it is desirable to only perform small changes to the grate speed. Changing the grate speed still has some advantages. The biggest one is the fact that both the steam-production and the oxygen concentration react a lot more quickly to changes in the grate speed than in the amount of waste fed into the plant. Because of this, one aspect where a model-based controller might be able to improve a bit is in its use of the grate-speed. The air-flows are good for controlling steam-production on a short time-horizon. Changing the waste-flow is good when controlling over a long time-horizon. The grate speed may therefore be used to supplement both of these, by helping to control the steam production over a medium time-horizon.

3.2 Disturbance step-responses

The first two columns in figure 3.3 show the effects of changes in the temperature of the primary and secondary air flow. The three different Q_{grate} variables represent heat-exchange in the different sections of the grate. In practice, there should also be other variables as well, to represent sudden changes in the concentration of combustible materials or moisture in the waste. But the disturbances that are there show a few very interesting points. The first observation is the small effect that air-temperature has on the steam-production. Figure 3.4 show that a change of one degree in the temperature of the primary or secondary air will only change the steam-production by $4 \left[\frac{\text{kg}}{\text{s}} \right]$. On the other had, the three values for Q_{grate} fluctuate more in the range of 10^4 , so all

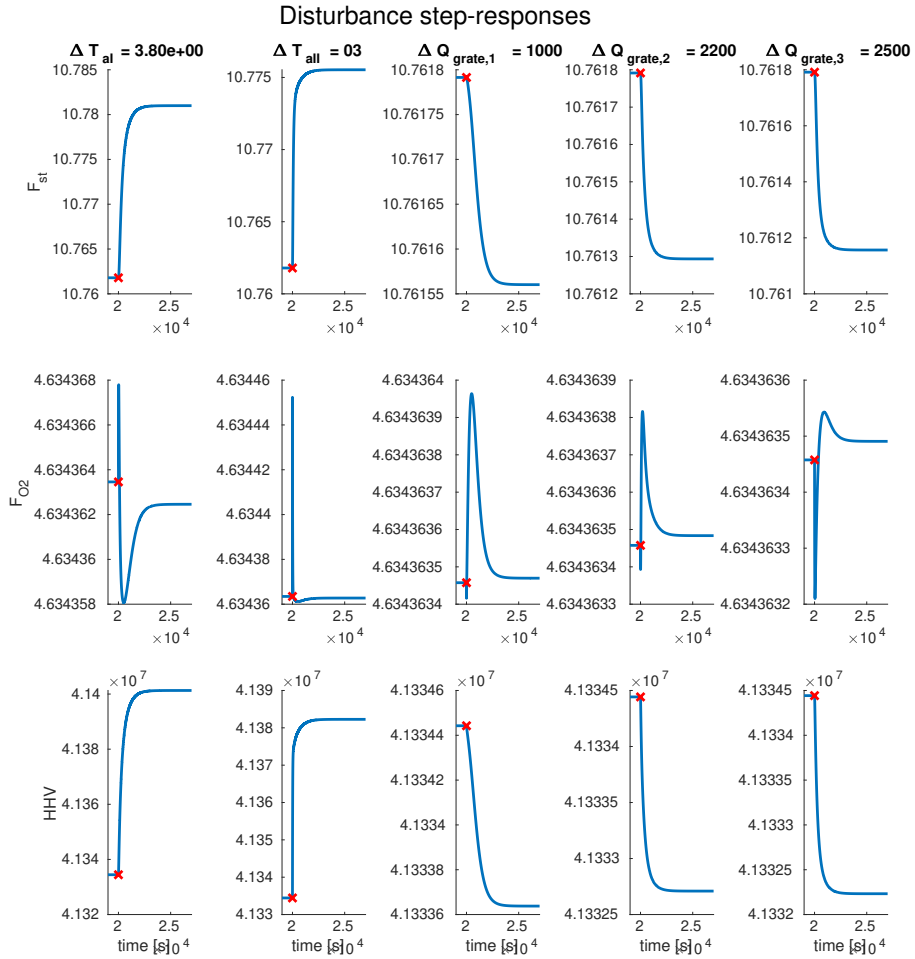


Figure 3.3: Step-response to the different kinds of disturbances

changes in air temperature become completely dominated by the disturbances from Q_{grate} . Because of this, the disturbances resulting from changes in air temperature will not be taken into consideration when designing controllers.

Finally, the biggest takeaway from plotting the disturbances can be seen when looking at the effects both of them have on F_{O_2} . As a result, the disturbances almost entirely affect the production of steam, while most changes to the mass-flow of air in the flue gas are a result of the attempts to correct the disturbances. This means that a feed-forward controller that tries to cancel the effects of Q_{grate} can only focus on the steam-production, and then another feed forward controller can be used to negate the changes in F_{O_2} that result from trying to correct for the changes in F_{st}

3.3 Some observations

3.3.1 Justifying the use of an HHV-estimator

When looking at figures 3.2 and 3.3, it can be seen that F_{st} and $H\hat{H}V$ are remarkably similar, except for their amplitude. Since F_{st} is one of the controlled outputs, the question becomes why $H\hat{H}V$ is necessary.

Figure 3.5 shows the step-responses of F_{st} and $H\hat{H}V$, without the mean, and represented as percentage of the infinite-horizon change. The two plots are very similar, but F_{st} lags behind $H\hat{H}V$ by about a minute. In the case of PID-controller, this opens up for using a cascaded PID-controller, as was done in Øie Kolden (2019), which allows for higher bandwidth, while also keeping the robustness of the previous controller. A model-based controller will be able to use the extra measurement of $H\hat{H}V$ to detect any changes in plant-disturbances earlier, but also to as a means to mitigate the effects of measurement-noise, as long as the two measurements are uncorrelated.

3.3.2 Usage of air when controlling steam production

The results that will be discussed in section 7.3 show that some controllers will attempt to primarily use air to control the steam-production around the operating point. This is normally bad, but the way it is done is somewhat interesting. Instead of using F_{aI} , like it has been done in the previous controllers, it instead uses $F_{aI} - F_{aII}$ to increase the production of steam. There are two main reasons for doing this. Primarily, because changing $F_{aI} - F_{aII}$ does not affect the total amount of oxygen added to the

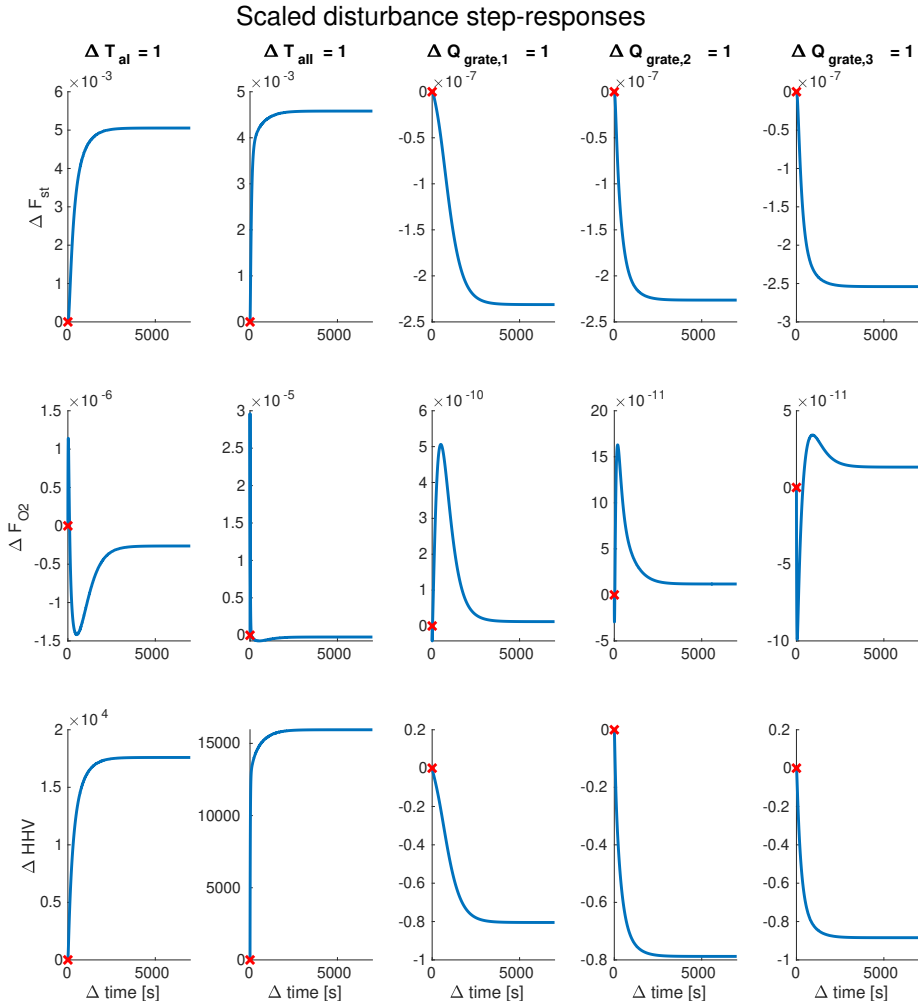


Figure 3.4: Change in outputs, given step-changes in the the disturbances. Adjusted for the operating point, and scaled for the step-amplitude

flue gas, leading to fewer changes in F_{O_2} . The second reason can be seen in figure

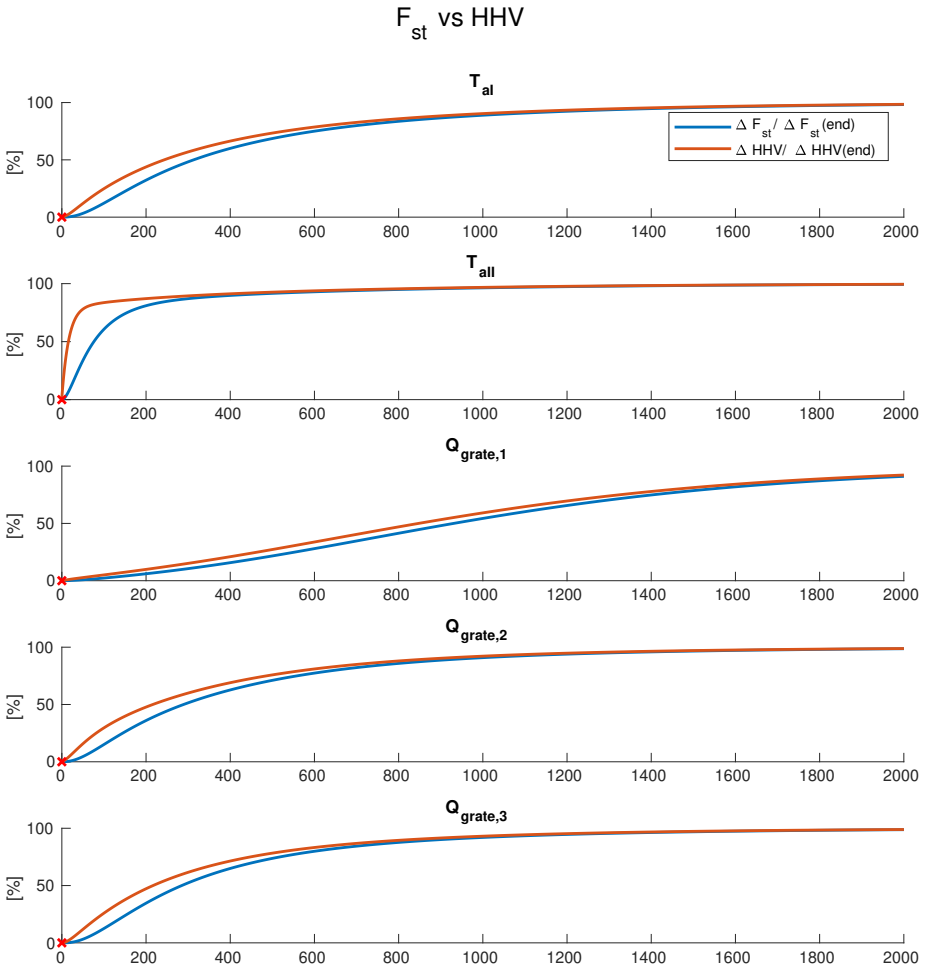


Figure 3.5: Change in $\hat{H}V$ and F_{st} as percentage of total change. Given as percentage to make them comparable.

3.6. By keeping the total amount of oxygen injected into the system constant, the step-response becomes more well-behaved, potentially leading to a controller that can

be allowed to be more aggressive.

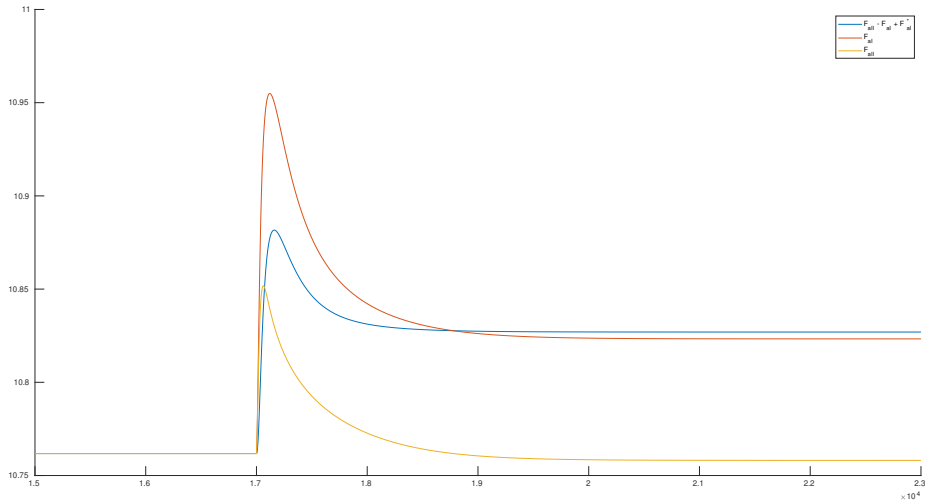


Figure 3.6: Change in outputs, given step-changes in the the disturbances. Adjusted for the operating point, and scaled for the step-amplitude

A PID-controller with this premise was not implemented in this project, but it could potentially be an avenue for further research.

Chapter 4

Parameter estimation

Even if most physical constants are known, there are always some that are specific for each plant. They have to be measured or estimated, since they can be dependant on properties unique to each plant, like size, mass, etc. There are be methods for estimating the dynamics of nonlinear plants, but linear approximations are often much easier to perform, and has well-known methods that can be used without full state measurement. As long as the plant is sufficiently well-behaved and the state remains close enough to the operating point linear parameter estimation is preferred.

There are several robust methods for approximating the parameters of MISO systems or MIMO-systems where the outputs are loosely coupled, as shown in Ioannou and Sun (2012). But, in the case of the MSWC-plant, the fact that the amount of oxygen consumed in the and the amount of heat produced makes it steam somewhat unreasonable that these two output variables should be independent. As a result, a MIMO-scheme for parameter estimation is needed instead. There are several different methods of performing linear parameter estimation, like the state-space approach and a transfer-function based approach, as presented in Stoica and Jansson (2000). However, in this project, the Eigensystem Realisation Algorithm (ERA) was used instead. This is mostly because it can estimate systems without full-state measurement. Additionally, the fact that it is somewhat easier to understand and implement made it the preferred

candidate for parameter estimation.

4.1 Eigensystem Realisation Algorithm

The ERA, as described in Juang and Suzuki (1988), is a method for estimating linear, discrete systems when the impulse response is known for some measurements of a system. It also allows for the creation of reduced-order models, that describe the dominant dynamics of the impulse-response, even if the original system was of a higher order than the estimated one.

The ERA assumes that the system is linear on the form:

$$\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k \quad (4.1)$$

$$\vec{y}_k = C\vec{x}_k + D\vec{x}_u \quad (4.2)$$

Where

$$x \in \mathfrak{R}^n$$

$$u \in \mathfrak{R}^m$$

$$y \in \mathfrak{R}^p$$

y and u must be measured, but the states x can be unknown. Unless full-state measurement is available, the states x in the estimated system will correspond to some abstract values that do not need to correspond to any of the physical states of the real plant. The ERA is based on the impulse-responses of the discrete system. It uses the impulse-response from each separate input. If $\vec{u}_{0,i}$ is a Kronecker delta at element i and 0 everywhere else, then the impulse-response y_i for that input can be written as:

$$y_{0,i} = D\vec{u}_{0,i} \quad (4.3)$$

$$y_{k,i} = CA^{k-1}B\vec{u}_{0,i} \forall k = 1, 2, \dots \quad (4.4)$$

All possible impulse-responses at one time-step can be combined into one matrix Y_k .

$$Y_k = \begin{bmatrix} y_{k,1} & y_{k,2} & \dots & y_{k,m} \end{bmatrix} \quad (4.5)$$

The concatenation of all input-vectors $u_{k,i}$ is the same as

$$I = \begin{bmatrix} u_{0,1} & u_{0,2} & \dots & u_{0,m} \end{bmatrix} \quad (4.6)$$

By combining equation 4.4 and 4.6, Y_k can be written quite conveniently as a matrix-product.

$$Y_k = CA^{k-1}B \quad (4.7)$$

All the impulse responses can be written into a $r \times s$ block matrix, on the form of a Hankel -matrix.

$$H_k = \begin{bmatrix} \vec{Y}_{k+1} & \vec{Y}_{k+2} & \dots & \vec{Y}_{k+s} \\ \vec{Y}_{k+2} & \vec{Y}_{k+3} & \dots & \vec{Y}_{k+s+1} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{Y}_{k+r} & \vec{Y}_{k+r+1} & \dots & \vec{Y}_{k+r+s} \end{bmatrix} \quad (4.8)$$

This also means that the Hankel-matrix can ideally also be expressed by using $[A, B, C]$. For instance The first matrix, H_1 can be written as:

$$H_1 = \begin{bmatrix} CB & CAB & \dots & CA^{m-1}B \\ CAB & CA^2B & \dots & CA^mB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m-1}B & CA^mB & \dots & CA^{2m-2}B \end{bmatrix} \quad (4.9)$$

H_1 can equivalently be written as $H_1 = \mathbb{O}\mathbb{C}$, which is the product of the adjoint impulse response \mathbb{O} and the direct impulse response matrix \mathbb{C} . These are given by:

$$\mathbb{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{m-1} \end{bmatrix} \vec{u}_0 \quad (4.10)$$

$$\mathbb{C} = \begin{bmatrix} B & AB & \dots & A^{m-1}B \end{bmatrix} \quad (4.11)$$

The matrices \mathbb{O} and \mathbb{C} are also the same as the observability matrix and the controllability matrix. Any following matrix H_k can be described as $H_k = \mathbb{O}A^k\mathbb{C}$. Since A , B and C are unknown, the goal is to find some matrices \hat{A} , \hat{B} and \hat{C} which is able to produce the same (Or a similar) Hankel-matrix.

Y_{k+1} can be extracted quite easily from H_k . If the matrix $E_p^T = [I_p, 0_p, \dots, 0_p]$ and $E_m^T = [I_m, 0_m, \dots, 0_m]$, then

$$Y_{k+1} = E_p H_k E_m = E_p \mathbb{O} A^k \mathbb{C} E_m \quad (4.12)$$

This structure will be used to find good good matrices \hat{A} , \hat{B} and \hat{C} .

In Juang and Suzuki (1988) a matrix $H^\#$ is found, which has the properties

$$\mathbb{O}H^\#\mathbb{C} = I_n \quad (4.13)$$

The matrix $H^\#$ satisfies being the pseudoinverse of H_1 , but we will not prove that in this thesis.

Another important tool will be the Singular Value Decomposition (SVD) of H_1

$$H_1 = U\Sigma V^* \quad (4.14)$$

The SVD has the property that all the column vectors in the matrices U and V are orthonormal to the other vectors from the same matrix. This is because the Hankel-matrix is purely real. This means that $VV^* = I$ and $U^T U = I$. Σ is a diagonal matrix of positive elements. Finally, we introduce the matrices U_d and $U_d^\#$

$$U_d = U\Sigma \quad (4.15)$$

$$U_d^\# = \Sigma^{-1}U^T \quad (4.16)$$

Juang and Suzuki (1988) also shows that $H^\# = VU_d^\#$. By inserting equation 4.13 into equation 4.12, a new expression can be found.

$$Y_{k+1} = E_p \mathbb{O}CH^\# \mathbb{O}A^k CH^\# \mathbb{O}CE_m \quad (4.17)$$

Next, $\mathbb{O}\mathbb{C}$ can be contracted, and $H^\#$ can be replaced by $VU_d^\#$.

$$Y_{k+1} = E_p H_1 V U_d^\# \mathbb{O}A^k \mathbb{C} V U_d^\# H_1 E_m \quad (4.18)$$

Next, a trick is needed to get rid of the expression containing A^k . To use this trick, we will have to show that $[U_d^\# H_2 V]^k = [U_d^\# \mathbb{O}ACV]^k$ is the same as $U_d^\# \mathbb{O}A^k \mathbb{C}V$. A polynomial $[U_d^\# \mathbb{O}ACV]^k$ can be written as $(U_d^\# \mathbb{O}ACV) \dots (U_d^\# \mathbb{O}ACV)$. By opening the

parenthesis and replacing $VU_d^\#$ with $H^\#$, and then replacing any $\mathbb{C}H^\#\mathbb{O}$ with I_p , the result is $= U_d^\#\mathbb{O}(AI_p)^k\mathbb{C}V$. As a result the two expressions are the same and equation 4.17 can be rewritten as:

$$Y_{k+1} = E_p H_1 V \left[U_d^\# H_2 V \right]^k U_d^\# H_1 E_m \quad (4.19)$$

Finally, since $U_d^\# = \Sigma^{-1}U^T$, the square root of Σ can be taken, and extracted from the exponential term, giving

$$Y_{k+1} = E_p U \Sigma^{\frac{1}{2}} \left[\Sigma^{-\frac{1}{2}} U^T H_2 V \Sigma^{-\frac{1}{2}} \right]^k \Sigma^{\frac{1}{2}} V^T E_m \quad (4.20)$$

This final equation gives a formulation that makes it possible to extract estimates of A , B and C that will have the same impulse-response. We will denote these estimates with a symbol hat.

$$\hat{A} = \Sigma^{-\frac{1}{2}} U^T H_2 V \Sigma^{-\frac{1}{2}} \quad (4.21)$$

$$\hat{B} = \Sigma^{\frac{1}{2}} V^T E_m \quad (4.22)$$

$$\hat{C} = E_p U \Sigma^{\frac{1}{2}} \quad (4.23)$$

$$(4.24)$$

Under idealised circumstances, the rank of the Hankel-matrix can be found when performing the SVD, as all elements after a given index will be 0. As a result, it will not matter if Σ , U and V are all truncated after that given index. The matrices can be divided into two parts. The part that is used to estimate the system, and the part that can be truncated.

$$\Sigma = \begin{bmatrix} \tilde{\Sigma} & 0 \\ 0 & \Sigma_{tr} \end{bmatrix} \quad (4.25)$$

$$U = \begin{bmatrix} \tilde{U} & U_{tr} \end{bmatrix} \quad (4.26)$$

$$V = \begin{bmatrix} \tilde{V} & V_{tr} \end{bmatrix} \quad (4.27)$$

$$(4.28)$$

If all sub-matrices with subscript tr truly are equal to 0, then

$$U\Sigma V^* = \tilde{U}\tilde{\Sigma}\tilde{V}^* \quad (4.29)$$

Consequently, the truncated matrices can be used to find a minimal representation of $[\hat{A}, \hat{B}, \hat{C}]$. In reality, noise, non-linearities, and numerical errors are inevitable, so the rank of the Hankel-matrix can not be given certainly. Furthermore, in the case of this project, it will be highly desirable to have a model of reduced order, since it will speed up an MPC-controller significantly.

So, in practice, some index is simply chosen for where to cut off $\tilde{\Sigma}$. A common way to find a decent index is to choose some value ϵ , such that

$$\frac{\text{sum}(\text{diag}(\tilde{\Sigma}))}{\text{sum}(\text{diag}(\Sigma))} \geq (1 - \epsilon) \quad (4.30)$$

And then making $\tilde{\Sigma}$ to be as small as possible, while satisfying the condition. By plotting the values of $\tilde{\Sigma}$, it also becomes a lot easier to see what is to be gained by adding more states to the estimated model.

Finally, it is important to remark that unless all states are observed, the states that result from the SVD will not necessarily represent physical values, but rather some linear combination of underlying properties of the system. This can easily be proven.

For any non-singular matrix T, the triplet $[TAT^{-1}, TB, CT^{-1}]$ will result in the

same Hankel-matrix as $[A, B, C]$

$$CT^{-1} (TAT^{-1})^k TB = CA^k B \quad (4.31)$$

This means that any true system representing a physical process can be transformed into infinitely many other systems that will have the same input-output response.

4.1.1 Handling Hysteresis

If the system suffers from a small amount of hysteresis, or nonlinearities at the at first when responding to an impulse, then the impulse-response of the physical system may not reflect the dynamics of the linearized system very well. The solution to this is to use some kind of more permanent excitation, and then using that to find some other kind of excitation and then using that to get an estimate of what the impulse-response would have been without the transient nonlinearities. The normal method is to use Observer Kalman Filter Identification (OKID) to find out what the impulse-response "would have been" for a linear system, given some pseudorandom input. The asymptotic memory-usage of OKID is $O(n^2)$, so if the number of samples is too large, the required space needed for system identification will be several gigabytes. In those cases, a simple trick with the step-response can be used instead. If a linear model should be valid, then the principle of superposition should also be somewhat true. This means that the response y from a sum of inputs and initial sates is the same as the sum of the responses that would have resulted from each separate signal.

$$y(u_1 + u_2) = y(u_1) + y(u_2) \quad (4.32)$$

$$\alpha y(u) = y(\alpha u) \quad (4.33)$$

An impulse is just two step-inputs subtracted by each other if they have the same amplitude, but different delay.

$$\delta(T_i) = u(T_i) - u(T_{i-1}) \quad (4.34)$$

Therefore, the superposition-principle allows us to find the impulse-response by subtracting a step-response by a delayed version of itself.

If the plant is asymptotically BIBO (Bounded Input, Bounded output)-stable, then this allows us to avoid some of the issues related to hysteresis, or non-linearities at the beginning of a response.

Chapter 5

Controllers

There are several methods for controlling multivariable systems. While basic PID-controllers or linear controllers may be employed, it may in some cases be more useful to use a controller that allows the user to prioritise certain control variables, as well as enforcing strict or semi-strict limits on both inputs and states. Since there are strict limits that dictate a minimum oxygen concentration in the flue gas, a Model Predictive Controller (MPC) will be tested in this project.

5.1 Model Predictive Control

The main idea of an MPC is that it contains some kind of model of the process that it is supposed to control. If we let \vec{x} be a vector denoting the state of a system, \vec{u} its input and \vec{d} the disturbances, a model can be written as:

$$\vec{x}_{k+1} = f_K(\vec{x}_k, \vec{u}_k, \vec{d}_{disturbance,k}) \forall k \in [T_0, \dots, T_n] \quad (5.1)$$

Where k represents all time-steps from the current time-step T_0 to some prediction horizon T_n that may be defined by the one who makes the MPC. Even though the

system is discretized, it is not required that an MPC has a uniform length between each time-step. As a result, each equation f_k can be different, if there is a need for higher resolution at a shorter time-horizon. In this thesis, all time-steps will have the same length, so $f_i = f_j \forall i, j \in [T_0, \dots, T_n]$

On a similar form to f , the expected output from the system will be given by:

$$\vec{y}_k = g_K \left(\vec{x}_k, \vec{u}_k, \vec{d}_{noise,k} \right) \quad (5.2)$$

The MPC always works on minimising some kind of objective function, that may take in a combination of inputs, outputs and states. as a result, it is necessary to make vectors containing all states in a prediction.

$$\vec{X} = \left[\vec{x}_{T_0}^T \quad \vec{x}_{T_1}^T \quad \dots \quad \vec{x}_{T_n}^T \right]^T \quad (5.3)$$

$$\vec{U} = \left[\vec{u}_{T_0}^T \quad \vec{u}_{T_1}^T \quad \dots \quad \vec{u}_{T_n}^T \right]^T \quad (5.4)$$

$$\vec{Y} = \left[\vec{y}_{T_0}^T \quad \vec{y}_{T_1}^T \quad \dots \quad \vec{y}_{T_n}^T \right]^T \quad (5.5)$$

Finally, an MPC may also be under a set of equality constraints h and inequality constraints g

$$g_i(\vec{X}, \vec{U}) \leq 0 \quad (5.6)$$

$$h_i(\vec{X}, \vec{U}) = 0 \quad (5.7)$$

The constraints are normally only constraining the states or inputs at one time-step, but they can also be used to limit the rate of change of \vec{u}_k or to lump together inputs if the MPC has a lower resolution closer to the prediction horizon.

If the function V is used to describe the cost the MPC tries to minimise. The

resulting problem becomes:

$$\begin{aligned}
 & \min_{\vec{U}} V(\vec{X}, \vec{U}, \vec{Y}) \\
 & \text{s.t} \\
 & \vec{x}_{i+1} = f\left(\vec{x}_k, \vec{u}_k, \vec{d}_{disturbance,k}\right) \\
 & g_i(\vec{X}, \vec{U}) \leq 0 \forall i \in [1, \dots, N_{ieq}] \\
 & h_i(\vec{X}, \vec{U}) = 0 \forall i \in [1, \dots, N_{eq}]
 \end{aligned} \tag{5.8}$$

Equation 5.8 is quite generic and can represent a wide range of potential MPC-problems. Depending on the type of restriction that is set for the problem, more efficient algorithms may be used, which may speed up the speed of the MPC considerably. Because of the estimated model being linear, only linear MPCs will be considered in this thesis.

5.1.1 Linear MPC

The Eigen-system Realisation Algorithm that was described in section 4 makes it possible to find a system representation from measured data. If such a model is combined with a Kalman-filter, it becomes possible to make a relatively fast model predictive controller.

The standard form for the problem that a linear MPC solves is on the form of a quadratic problem. Let m be the number of inputs and N is the number of prediction steps. H is a positive definite matrix $H \in \mathfrak{R}^{mN \times mN}$ and F is some matrix $F \in \mathfrak{R}^{mN}$, while U are all inputs over all time-steps within some prediction horizon, such that $\vec{U} \in \mathfrak{R}^{mN}$, then the problem can be described as:

$$\begin{aligned}
& \min_{\vec{U}} \vec{U}^T H \vec{U} + F(x_0, \text{ref})^T \vec{U} \\
& \text{s.t. :} \\
& A_{ge} \vec{U} \geq b_{ge} \\
& A_{eq} \vec{U} = b_{eq}
\end{aligned} \tag{5.9}$$

Unlike a lot of other model predictive controllers, the sequence of states X can be implicitly expressed by a linear function of x_0 and the sequence of inputs \vec{U} . If the model is discrete, then the exact solution can be found by matrix-multiplications with a set of pre-prepared matrices.

$$\vec{x}_{T_{i+1}} = A \vec{x}_{T_i} + B u_{T_i} \tag{5.10}$$

Simply by adding all the inputs, and multiplying with A an appropriate amount of times, a general expression can be found.

$$\vec{x}_{T_i} = A^i \vec{x}_{T_0} + \sum_{j=1}^{i-1} A^{i-j} B u_{T_j} \tag{5.11}$$

So if one matrix \mathfrak{M} is constructed for the response due to the initial state x_0 , and another one, \mathfrak{C} for the convolution between the input signal and A , then all states in \vec{X} can be represented by a few simple matrix operations

$$\vec{X} = \mathfrak{M} x_0 + \mathfrak{C} \vec{U} \tag{5.12}$$

If N is the number of time-steps that the MPC uses in its prediction, then the two

matrices \mathfrak{M} and \mathfrak{C} are defined as:

$$\mathfrak{M} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \quad (5.13)$$

$$\mathfrak{C} = \begin{bmatrix} B & 0 & \dots & 0 & 0 \\ AB & B & \dots & 0 & 0 \\ A^2B & AB & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{N-2}B & A^{N-3}B & \dots & B & 0 \\ A^{N-1}B & A^{N-2}B & \dots & AB & B \end{bmatrix} \quad (5.14)$$

Because of this, any cost that would normally be written as a function the inputs and states can instead be expressed entirely by the inputs and the initial state.

$$\vec{X}^T \hat{Q} \vec{X} = (\mathfrak{C}U + \vec{\mathfrak{M}}x_0)^T \hat{Q} (\mathfrak{C}U + \vec{\mathfrak{M}}x_0) \quad (5.15)$$

The same goes for constraints as well

$$A_{eq} \vec{X} = A_{eq} (\mathfrak{C}U + \vec{\mathfrak{M}}x_0) \geq b_{eq} \quad (5.16)$$

$$\Rightarrow A_{eq} \mathfrak{C}U \geq b_{eq} - A_{eq} \vec{\mathfrak{M}}x_0 \quad (5.17)$$

Linear MPCs have the advantage of being able to use more efficient algorithms for solving the QP, than what would be possible for a generic non-linear problem. The result is that a faster sampling-frequency or longer prediction-horizons can be used.

5.1.2 Stabilising an MPC

A finite-horizon MPC is usually not guaranteed to stabilise a plant. Using a long prediction-horizon may solve this, but the number of prediction steps is dependant on the plant. Additionally, if the terminal state does not have a cost that reflects how the MPC will behave later, some strange behaviour might occur.

Proof of instability: Let $x[T_{i+1}] = 2 \cdot x[T_i] + u$ a scalar difference equation. If the cost-function is $J = x[T_{i+1}]^2 + 3u^2$, simple differentiation gives that the best u is $u = -\frac{1}{2}x[T_i]$, which is not enough to stabilise the plant.

As explained in appendix A, an LQR solves the same kind of problem an MPC does, but over an infinite horizon, and without constraints. If all constraints remain inactive after the end of the prediction-horizon, and the terminal state-cost of the MPC is the same as the infinite-horizon state cost for the LQR, then the two controllers will have the exact same sequence of inputs. Since the LQR is guaranteed to stabilise the plant, the MPC will be guaranteed to do the same.

If only the state-feedback matrix K_{lqr} is available, the infinite-horizon cost can be found by solving a Lyapunov-equation instead.

$$Q_\infty = (A - BK_{lqr})^T Q_\infty (A - BK_{lqr}) + Q_\infty + K_{lqr}^T R K_{lqr} \quad (5.18)$$

Luckily, MATLAB implemented methods for solving both equation Lyapunov-equations and Riccati equations

$$K_{lqr} = \text{lqr}(A, B, Q, R) \quad (5.19)$$

$$Q_\infty = \text{dlyap}((A - B * K_{lqr})^T, Q + K_{lqr}' * R * K_{lqr}); \quad (5.20)$$

5.1.3 Reference-tracking MPC

MPC systems do not inherently track references. The way this is normally done is by giving a cost for changing the inputs, instead of using a cost for allowing the inputs to deviate from the reference point. This can the MPC to act differently from how the

LQR that is used to make the infinite-horizon cost would have acted. One solution to stationary feed-forward input v_{ff} that would cause the closed-loop system with and LQR.

If an LQR is to track a reference, it usually means that the linearisation point has to be changed. One way to solve this easily is to find a matrix that would cause the closed-loop LQR to converge asymptotically to the wanted output. If the closed loop is asymptotically stable, then the stationary output measurements can be found by:

$$Y_{\infty} = C (I + BK - A)^{-1} B \quad (5.21)$$

It is possible to simply use the pseudo-inverse of $(C (I + BK - A)^{-1} B)$ to find a matrix that can transform the set of desired outputs into the corresponding set of required inputs. A better solution can be found by solving a constrained QP¹ that causes the reference to be tracked as cheaply as possible according to the original cost function.

$$\begin{aligned} \min_{v_{ff}} & ((I + BK - A)^{-1} v_{ff})^T (Q + K^T R K) (I + BK - A)^{-1} v_{ff} + v_{ff}^T R v_{ff} \\ \text{st :} & \\ & C (I + BK - A)^{-1} v_{ff} = y_{ref} \end{aligned} \quad (5.22)$$

Additional constraints may be added if the problem is constrained. No proof was found in this thesis that would have allowed for tracking y_{ref} as cheaply as possible without recalculating the QP each time the reference changes.

If the cost of the input-cost of an MPC is set to be given by $u - v_{ff}$, instead of u and the state-cost is set to be given by $x - (I + BK - A)^{-1} B v_{ff}$, then the cost of the MPC will be zero when the system tracks the reference.

¹Quadratic Problem

5.2 PID-control

As has been seen in previous sections, it can be difficult to model the system. If it is possible to find a method that may control the plant, without having to rely on overly complex models or unreliable estimates, that would be desirable. Fortunately, normal cascaded PID-controllers can very often do an excellent job without an exact model. PID-controllers also often have the advantage of being robust against modelling errors. Unfortunately, tuning the three parameters in a PID-controller is known for being a difficult task, depending on the type of process. As a result, several different methods have been developed for tuning PID-controllers. This section will explain how to tune a PI or PID-controller by using Skogstad's Internal Mode Control (SIMC).

5.2.1 Skogstad's Internal Mode Control

SIMC is a set of rules for tuning PID-controllers that were originally used for teaching-purposes, but who have proven to work very well in practice. The method is based, quite simply on exciting the system from a stationary state with a step-response. A step-response is often more well-behaved on most systems, and usually gives a response that is more in line with that of an actual linear system. The SIMC are the tuning rules for how to tune the constants of a PID-controller if a simple approximation of the model is already known. As a result, there are two parts of SIMC-tuning. The system approximation and the actual tuning. Because parts of the system are also affected by a rather large non-minimum phase response, it will also be necessary to expand upon the normal system approximation to handle the large transients.

5.2.2 Simplifying plant models

Skogstad's method normally relies on some transfer function model of the system to tune the system. If τ represents the lag in the system and $e^{\theta_0 s}$ represents the delay and k the plant gain, then Skogstad's method is based on a SISO (Single Input Single Output) system that can be described as:

$$y = k \cdot \frac{\prod_{i=1}^n (-T_i^{inv} s + 1)}{\prod_{j=1}^n (\tau_{j,0} s + 1)} e^{-\theta_0 s} \quad (5.23)$$

Where $\tau_i \geq \tau_{i+1} \geq 0$ and $T_i^{inv} \geq T_{i+1}^{inv} \geq 0$ (Positive roots in the numerators are explained in subsection 5.2.2.1). Since Skogstad's method only revolves around a PID-controller, only the most dominant terms are taken into consideration. Any factor after the second one will instead be approximated by a the Taylor-approximation of a time-delay and added to the already existing time-delay.

$$e^{\tau_j s} = 1 + \tau_j s + \dots + \tau_j^i s^i + \dots \quad \approx 1 + \tau_j s \quad (5.24)$$

$$e^{-\tau_j s} = \frac{1}{e^{\tau_j s}} \quad \approx \frac{1}{\tau_j s + 1} \quad (5.25)$$

Depending on the properties of the plant, it is possible to perform different simplifications. If the lag τ_1 dominates the process (usually $\tau_1 > 8\theta$), then the estimation $\frac{k}{\tau_1 s + 1} \approx \frac{k'}{s}$ can be used. This can also allow you to end the experiment early instead of waiting for convergence. A discretely sampled with a sampling time h , can be simplified into a continuous system with an additional time-delay $\frac{h}{2}$ if the time-delay is not too dominant. Additionally, the largest neglected pole can be evenly distributed between the smallest lag that is being used and the time-delay.

All these rules can be summarised into the much simpler rules, described in equations 5.26 and 5.27. The tuning rules are lifted directly from equations (10) and (11) in Skogstad (2004).

Second order approximation :

$$\tau_1 = \tau_{1,0}; \tau_2 = \tau_{2,0} + \frac{\tau_{3,0}}{2}; \theta = \theta_0 + \frac{\tau_{3,0}}{2} + \sum_{i \geq 4} \tau_{i,0} + \sum_j T_{j0}^{inv} \frac{h}{2} \quad (5.26)$$

First order approximation:

$$\tau_1 = \tau_{1,0} + \frac{\tau_{2,0}}{2}; \theta = \theta_0 + \frac{\tau_{2,0}}{2} \sum_{i \geq 3} \tau_{i,0} + \sum_j T_{j0}^{inv} \frac{h}{2} \quad (5.27)$$

5.2.2.1 Positive time-constants in the numerator

In Skogstad (2004) positive roots in the numerator ($T_0s + 1$) are handled by allowing the term in the numerator to cancel one in the denominator (The closest, larger one). The delay θ is the final delay after all cancellations have been made. Because of this, the cancellations have to be made iteratively with the new delays.

$$\frac{T_0s + 1}{\tau_0s + 1} \approx \begin{cases} \frac{T_0}{\tau_0} & \text{if } T_0 \geq \tau_0 \geq \theta \\ \frac{T_0}{\theta} & \text{if } T_0 \geq \theta \geq \tau_0 \\ 1 & \text{if } \theta \geq T_0 \geq \tau_0 \\ \frac{T_0}{\tau_0} & \text{if } \tau_0 \geq T_0 \geq 5\theta \\ \frac{(\tilde{\tau}_0/\tau_0)}{(\tilde{\tau}_0 - T_0)s + 1} & \text{if } \tilde{\tau}_0 \triangleq \min(\tau_0, 5\theta) \geq T_0 \end{cases} \quad (5.28)$$

If there is no larger time-constant in the denominator, the closest smaller one is used instead.

5.2.3 PID-rules

SIMC uses the IMC tuning rules for generating the PID-parameters after a first or second-order model has been found. A more complete derivation of the rules is found in Skogstad (2004), but the short version is:

Usually, a desired response with time-constant τ_c is selected. τ_c is a tuning constant

of the SIMC, that is the desired time-constant of the closed-loop system.

$$\left(\frac{y}{y_s}\right)_{desired} = \frac{1}{\tau_c s + 1} e^{\theta s} \quad (5.29)$$

Let $g(s)$ be the transfer-function that describes the plant. An ideal controller $c^*(s)$ is chosen such that if

$$g(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{\theta s} \quad (5.30)$$

Then, the idealised controller can be written as:

$$\frac{g(s)c^*(s)}{1 + g(s)c^*(s)} = \left(\frac{y}{y_s}\right)_{desired} \quad (5.31)$$

$$c^*(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{\tau_c s + 1 - e^{-\theta s}} \quad (5.32)$$

The Taylor-approximation is used to get rid of the $e^{-\theta s} \approx 1 - \theta s$, such that the approximated controller $c(s)$ becomes

$$c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k} \frac{1}{(\tau_c + \theta) s} \quad (5.33)$$

becomes $((\tau_c + \theta) s)$. However, if the lag is the most dominant part of the process, this kind of tuning will result in a very slow settling-time. This is solved by approximating the system as a pure integrating process with no $\frac{k}{\tau_1 s + 1} e^{-\theta s} \approx \frac{k}{\tau_1 s} = \frac{k'}{s}$. Afterwards, $c(s)$ can be set as a PI-controller with gains that cause the closed loop to have a damping factor ξ of 1

$$\frac{g(s)c(s)}{1 + g(s)c(s)} = \frac{\frac{k'}{s} K_c \left(1 + \frac{1}{\tau_I s}\right)}{1 + \frac{k'}{s} K_c \left(1 + \frac{1}{\tau_I s}\right)} \quad (5.34)$$

Sorting out the terms gives

$$\frac{\tau_I s + 1}{\frac{\tau_I}{k'K_c} s^2 + \tau_I s + 1} \quad (5.35)$$

A system on standard second order form is written as $\tau_0^2 s^2 + 2\tau_0 \xi s + 1$. If $\xi = 1$, any system with denominator on that form is critically damped. So, in the case of equation 5.34

$$\tau_0 = \sqrt{\frac{\tau_I}{k'K_c}} \quad (5.36)$$

$$\xi = \frac{1}{2} \sqrt{k'K_c \tau_I} \quad (5.37)$$

So if K_c is chosen as in equation 5.33 to be $\frac{1}{k} \frac{\tau_1}{\tau_c + \theta}$, then $\tau_I = 4(\tau_c + \theta)$. Both these tuning rules can be combined into one very simple one.

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} \quad (5.38)$$

$$\tau_I = \min \{ \tau_1, 4(\tau_c + \theta) \} \quad (5.39)$$

$$\tau_D = \tau_2 \quad (5.40)$$

If a PI-controller is needed, a simpler model is used instead, such that $\tau_2 = 0$. Skogstad (2004) mentions that $\tau_c \in [\theta, \infty)$ can be chosen freely, but that it is a trade-off between speed and robustness (as well as smaller variations in inputs). The paper highlights that $\tau_c = \theta$ is a good trade-off between speed and robustness.

5.2.4 PID-tuning by using bode-plots

Although Skogstad's method can be very useful, it might sometimes struggle with systems where the poles or zeros are complex, since those cases are not defined in the original method. There are proposed methods for handling such cases, but it is also possible to simply tune a stable system by observing the phase-response of a stable system and choosing K_c and T_I such that a sufficient phase and gain margin can be

achieved. If the system is stable and the controller is a simple PI-controller, then it is also possible to find a set of rules that can automate the tuning-process, so that it is robust against time-delays and modelling errors.

5.2.4.1 A quick reintroduction to bode-plots (Nyquist's criteria of stability)

Let the transfer-function of the plant be $h_p(s)$, and the transfer-function of the controller be $h_c(s)$. Let $h_0(s) = h_p(s)h_c(s)$. An instability in the system is caused if the transfer-function of the closed loop

$$M(s) = \frac{h_0(s)}{1 + h_0(s)} \quad (5.41)$$

has a pole in the left half-plane. $1 + h_0(s)$ can be divided into a numerator and denominator, and the resulting closed-loop will have a set of poles and zeros

$$1 + h_0(s) = 1 + \frac{n_0(s)}{d_0(s)} = \frac{d_0(s) + n_0(s)}{d_0(s)} = \frac{(s - \lambda_1) \dots (s - \lambda_n)}{(s - \rho_1) \dots (s - \rho_n)} \quad (5.42)$$

If one performs integration of s along the imaginary axis, and then along a circle with an infinitely large radius in the right half-plane, only poles and zeros in the right half-plane will give a net contribution to the phase, as described in Jens G. Balchen (2016). This can be seen in figure 5.1. All zeros λ will give a net negative contribution to the angle, while all poles ρ will give a positive contribution.

$\Delta\angle(1 + h_0(s))$ is the total angular change when integrating along the half-circle. N_p and N_n are the total number of poles in the right half-plane in the open and closed system respectively. The resulting relation is

$$\Delta\angle(1 + h_0(s)) = -2\pi (N_n - N_p) \quad (5.43)$$

So, Nyquist's stability criteria uses the total phase-contribution of $1 + h_0(s)$, and the number of unstable poles in the open-loop system to determine the number of unstable

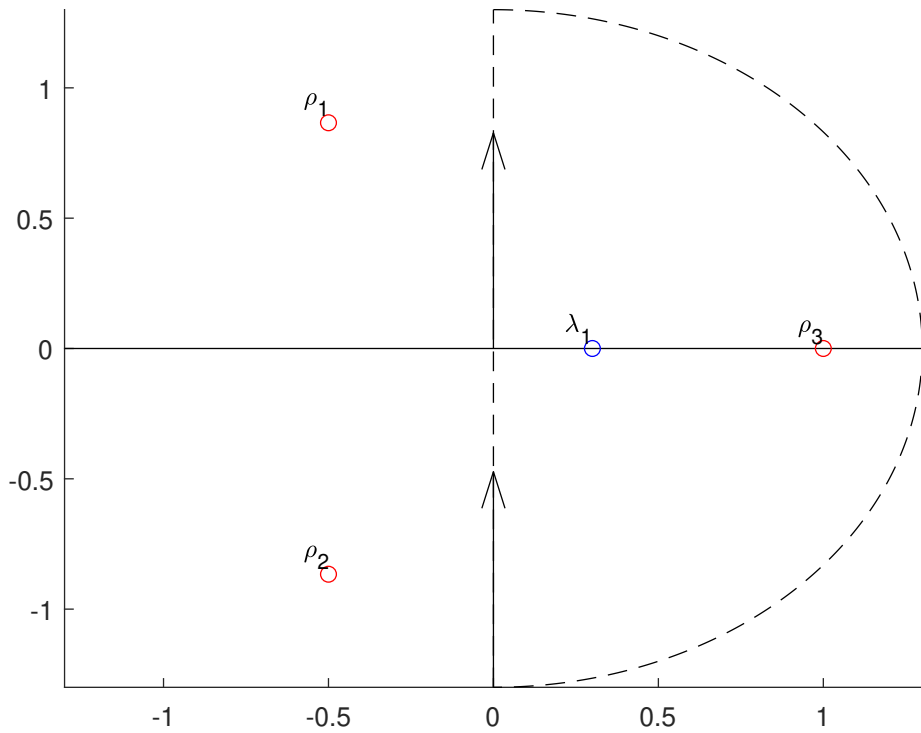


Figure 5.1: Poles ρ and zeros λ of a transfer-function. The existence of ρ_3 makes it unstable

poles in the closed-loop

$$N_n = N_p - \frac{\Delta \angle(1 + h_0(s))}{2\pi} \quad (5.44)$$

To analyse if an open loop system will be stable when the loop is closed, a bode-plot, like in figure 5.2 is normally used. A bode-plot consists of one plot of the amplitude of the open-loop, and the phase of the open-loop system. Both of them are plotted against an angular velocity, ω .

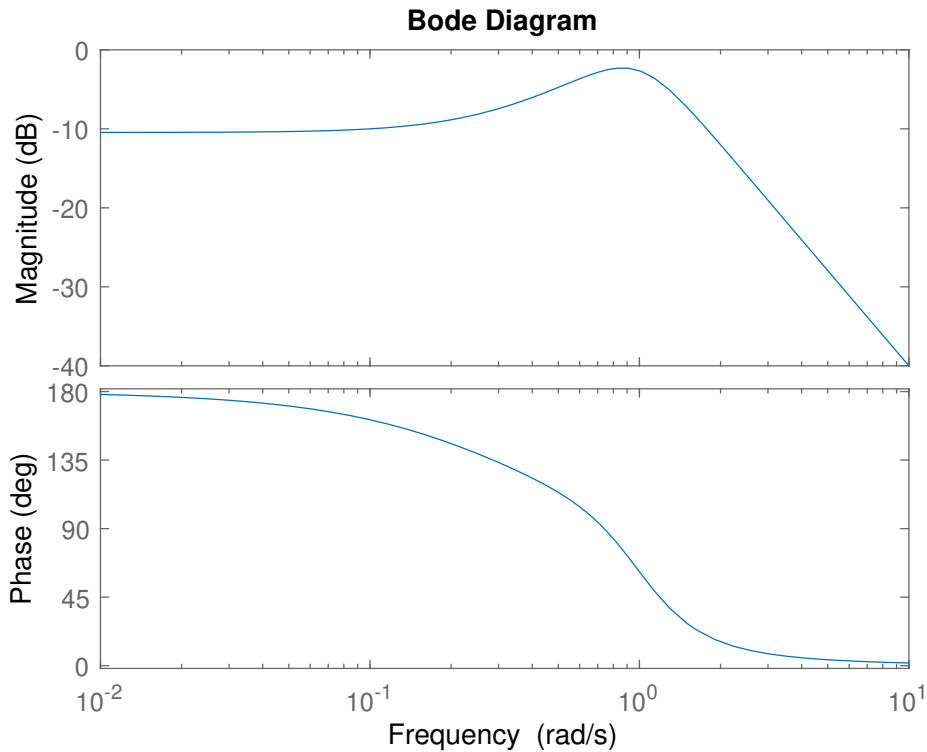


Figure 5.2: A generic bode-plot of $\frac{s-0.3}{s^2+s+1}$

A bode-plot makes it very clear a stable system will be stable in a closed-loop as well. If the amplitude of $h_0(s)$ is greater than 1 (0 dB) when the phase is modulo 360 is -180 degrees, then the system is unstable.

5.2.5 Gain and phase margins of stable systems

The plant that is being controlled is known to be stable, so the only condition that has to be satisfied is that the amplitude of the system is less than 0dB whenever the phase

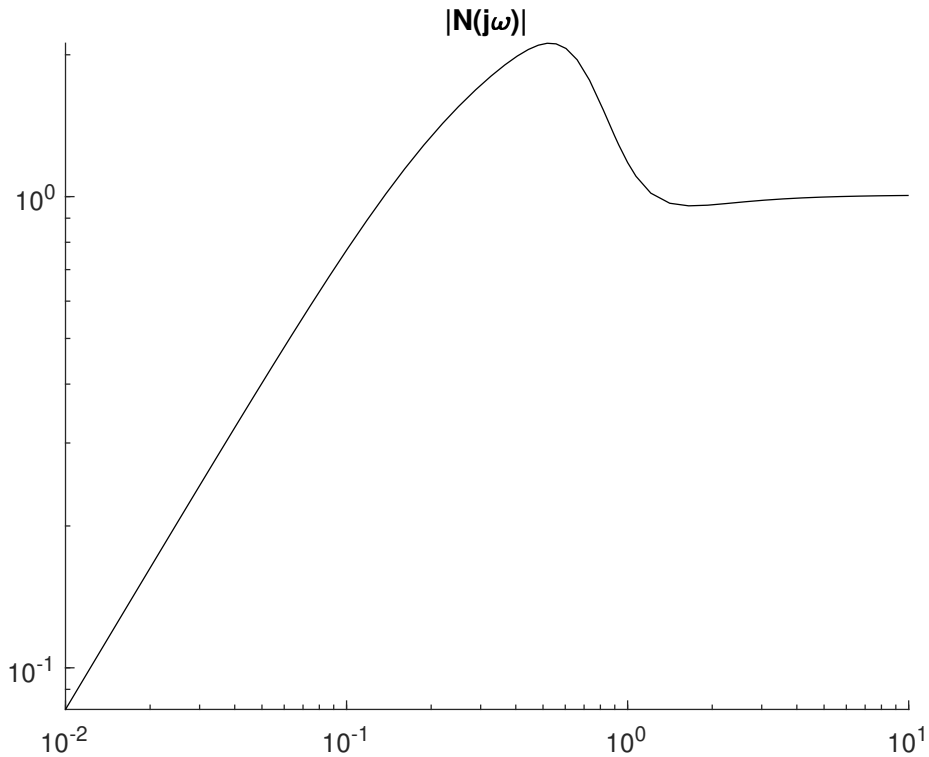
of the system crosses -180° . It is never possible to know exactly how one might model the system incorrectly, but bode plots allow saying something about the robustness against time-delays and against unknown gains affecting the system. If a system has robustness in both of these regards, then it should also have some robustness to in regards to the modelling errors with regards to the system dynamics as well.

Since time-delays are written as $e^{\theta j\omega}$, they only contribute with an addition to the phase in the bode-plot. This means that a good phase-margin will give robustness against time-delays. Usually, it is said that a phase margin of 60° and a gain margin of 20dB is a good balance between performance and robustness.

5.2.5.1 Choosing a good controller

One way to say something about a controller is by using the control ratio $N(s) = \frac{1}{1+h_0(s)}$ and the following ratio $M(s) = \frac{h_0(s)}{1+h_0(s)}$. M tells how the system will track changing references of a given frequency, as a result, it will also say how much the system reacts to measurement noise. Meanwhile, N says how well noise or disturbance will be suppressed. When working with a system that is subject to both noise and process disturbances, both $N(s)$ and $M(s)$ are necessary. Ideally, $N(s)$ has such a shape that all disturbances become 0 while ignoring any noise. This is impossible, but it gives a hint as to how the controller and filters would be chosen. If it is known that some noise has a very specific frequency, then band-stop filters or notch-filters can be added to the controller as well.

In figure 5.3, a plot of $N(j\omega)$ is given for some arbitrarily chosen controller. As can be seen in the plot, it does suppress disturbances at low frequencies. Additionally, it can be seen that for high frequencies, the disturbances are simply let through. However, a rather large peak at around $0.3 \frac{rad}{s}$ can be seen as well. This peak around where the crossover-frequency, which means that $h_0(j\omega)$ almost entirely negative. As long as the phase of $h_0(j\omega)$ crosses -180° , there is always a point where the amplitude of $N(j\omega) > 1$. The way to control the size of this peak is by making sure the amplitude of $h_0(s)$ is sufficiently small around -180° , since if it were to be close to 1, then the system would amplify any noise or disturbance of that frequency by an almost absurd amount.

Figure 5.3: N for a poorly controlled process

There are two possible measures that can be taken to suppress the resonance around the unwanted frequencies. The first is to simply increase the gain margin. If the amplitude of $h_o(j\omega)$ is small at $\angle h_o(j\omega) \approx -180$, then the fraction $\frac{1}{1+h_o(j\omega)}$ does not grow as large. A phase margin of $20dB$ will ensure that the resonant amplitude will be less than $\frac{1}{1-\frac{1}{10}} \approx 1.11$.

The second option is to use the D-part of the controller to "lift" the phase of the combined controller and plant. Normally, most plants attenuate high-frequency signals more than low-frequency ones, so by keeping the phase above -180° for

longer, $|h_0(s)|$ can be made to be smaller at the new crossover frequency. Simply adding a differentiating term can sometimes lead to other problems, since it makes the system react more to high-frequency noise. The solution is to add a low-pass filter to the controller as well. An additional advantage of using a low-pass filter is that a differentiation term will usually require some numerical differentiation, that might be ill-behaved. A differentiation term in series with a low-pass filter gives a variation of a high-pass filter, which is a proper transfer function², that can be written as a causal system and discretized in a more well-defined manner.

The need for low-pass filters can be shown in figure 5.4. Even if the plot for $N(s)$ looked good, $M(s)$ shows that high-frequency noise are not sufficiently attenuated, making it susceptible to noise, which is very often present in the form of electrical noise.

²Proper transfer function: The order of the denominator (Highest degree of s) is greater or equal to the order of the numerator

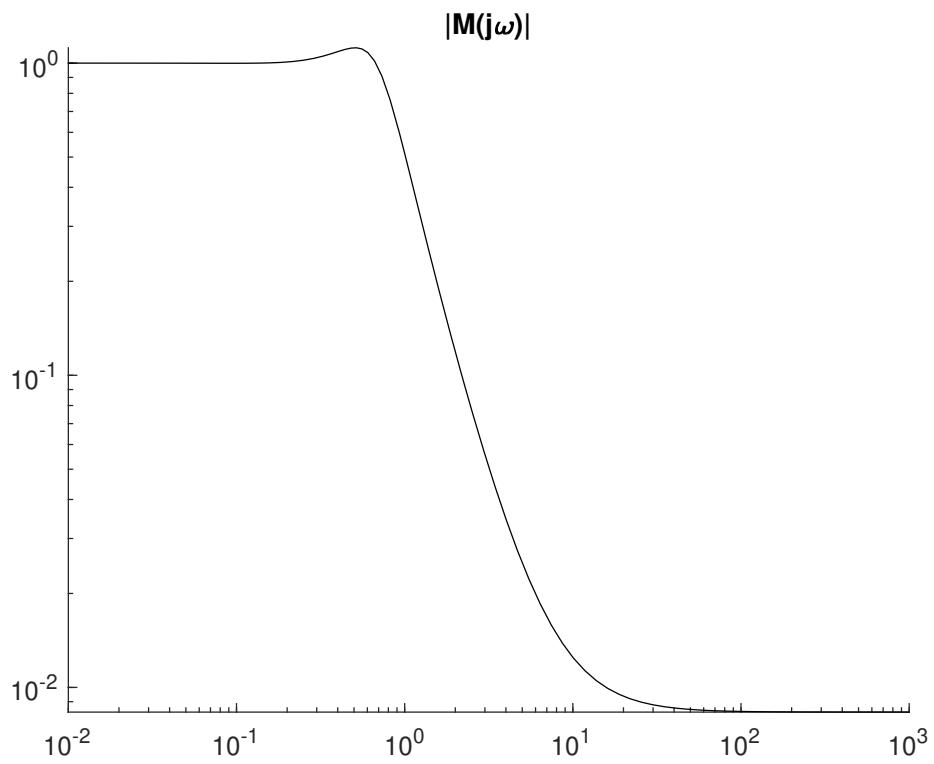


Figure 5.4: M for a poorly controlled process

Chapter 6

Automated tuning

The methods from Chapter 5 do not necessarily give controllers that work perfectly right away. Because of this, some manual tuning may be needed as well. Some of the required changes may be intuitive, while others may be dependant combinations of parameters, as well as some trial and error. As a result, the tuning process can be quite time-consuming. One solution to this is to make up some kind of measure of the quality of the system response, and then use an optimisation algorithm, like Nelder-Mead to minimise said cost.

6.1 Nelder-mead

The Nelder-Mead optimisation algorithm is a very popular method for optimising convex nonlinear problems. The main selling-point of Nelder-Mead is that the search of the optima is done entirely by a set of measured points and the corresponding values of the function, without the use of an approximated or an analytical gradient. This means that the algorithm tends to be a bit slower than most other optimisation algorithms, like SQP¹. Instead, it has the advantage of being usable on a wide set of convex functions without much prior knowledge about the function. It should be

¹SQP: Sequential Quadratic Programming

noted that according to McKinnon (1996), even though the algorithm works well in practice, it is possible for it to converge towards non-stationary points.

6.1.1 Theory

Lagarias et al. (1998) summarises the Nelder Mead algorithm quite nicely, so it will only be covered very briefly in this chapter.

Nelder-mead optimises an N -dimensional function by using a set $\mathbf{X} = \{x_1, x_2, \dots, x_{n+1}\}$ of $N + 1$ points. The points are at all times sorted in such a way that

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1}) \quad (6.1)$$

At iteration k , the set \mathbf{X}^k is updated to \mathbf{X}^{k+1} by replacing the worst the worst point x_{n+1}^k in the set. x_{n+1}^k is reflected over or drawn to the "centre of mass" $\bar{x} = \sum_{i=1}^N \frac{x_i}{N}$ off the other points, depending on if it improves the point or not. The new point x_{new}^k will be given by

$$x_{\text{new}}^k = \bar{x}^k + \rho(\bar{x}^k - x_{n+1}^k) \quad (6.2)$$

Several possible candidates along the line are tested. If the function $g^k(\cdot)$ is defined as

$$g^k(\rho) = f(\bar{x}^k + \rho(\bar{x}^k - x_{n+1}^k)) \quad (6.3)$$

the update-rule for a single iteration is described in algorithm 6.1.1.

Afterwards, the new set \mathbf{X}^{k+1} is given by

$$\mathbf{X}^{k+1} = \{x_1^k, x_2^k, \dots, x_n^k\} \cup \{x_{\text{new}}^k\} \quad (6.4)$$

Lagarias et al. (1998) also expands the algorithm by adding tie-breaking rules for if two function values in \mathbf{X} are the same.

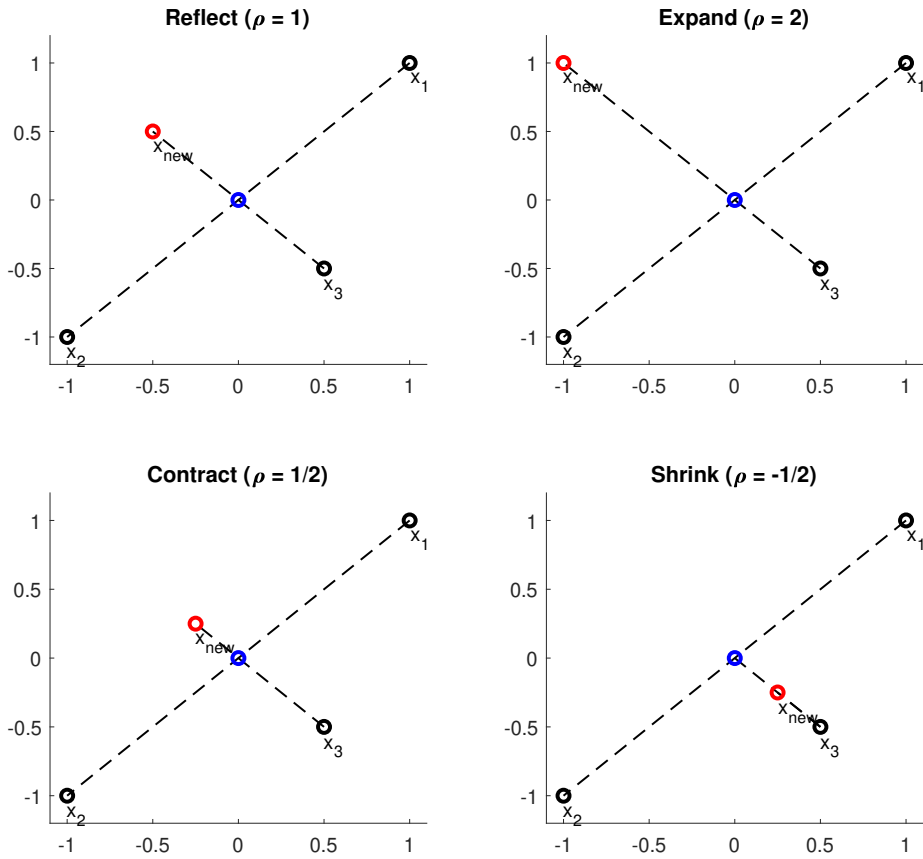


Figure 6.1: The 4 possible update-steps in Nelder-Mead's optimisation algorithm

Algorithm 1 Find $x_{\text{new}}^k = \bar{x} + \rho(\bar{x}^k - x_{n+1}^k)$

Require: $x_{n+1}^k, \bar{x}^k, g^k(\cdot), f(x_{n+1}^k)$

Ensure: $y = x^n$

- 1: **if** $g(1) < f(x_{n+1}^k)$ **then**
 - 2: **if** $g(2) < g(1)$ **then**
 - 3: $\rho = 2$
 - 4: **else**
 - 5: $\rho = 1$
 - 6: **end if**
 - 7: **else if** $g(1/2) < f(x_{n+1}^k)$ **then**
 - 8: $\rho = 1/2$
 - 9: **else**
 - 10: $\rho = -1/2$
 - 11: **end if**
-

6.1.2 Issues

In the two-dimensional case McKinnon (1996) proved that it is possible for the simplex shrink repeatedly into a line, and that this line can be orthogonal to the steepest decent direction. As a result, some precaution should be taken with the optimised function, since the underlying function of a simulation is unknown. If these issues become far too big, other methods, like simulated annealing or genetic algorithms can also be used, but they normally need a lot more samples when going from a poor solution to a decently .

Chapter 7

Experimental results

7.1 Parameter estimation results

Since everything in the simulator is inside Simulink, it is not too hard to get noise-free measurements, which means no experiments had to be repeated to suppress noise or disturbances. A step-response seemed to work better with the simulator than an impulse-response, so that was used instead. The diagonal of Σ after a singular value decomposition can be used to say something about how much adding additional states in a model will help in lowering the estimation error.

From the plot in figure 7.1, it can be seen that the diagonal elements of Σ decrease quite rapidly. Using too many parameters will often cause a mode to be overfitted, which can lead to problems. As a result, models of order 5, 7, 10, 12 and 15 were tested. To validate the estimated models, they were plotted against the original step-response. As can be seen in figure 7.2, even the 10th order models struggles with estimating certain outputs. This is because the inputs and outputs have not been scaled properly, so the total error will go down significantly more if the error in the *HHV* is reduced a little, than if the error in Y_{O_2} is eliminated. The solution to this is to scale the inputs and outputs somewhat. One formulation for the goal of the scaling can be set is to make as many input-output relations have an amplitude of 1 for some norm (max, \mathcal{L}_2 , etc.). There is probably a method to finding the optimal solution to this problem, but for the sake of simplicity, it is also possible to alternate between scaling the inputs and the outputs, such that the largest norms, are equal to 1. The exact method is not that important, as long as the *HHV* or v_{grate} does not dominate the estimation.

After using ERA to estimate the system, it can be scaled back with the same vectors. The result, as seen in figure 7.3. The resulting approximated system has a far lower relative error. This scaling is necessary for creating an accurate low-order model that can be used for controlling all outputs of the plant, not just the estimated *HHV*-value.

7.1.1 Choosing a model order

Picking a model order can be somewhat difficult. From what can be seen in figure 7.3. All estimates struggle with matching some of the outputs. Most importantly is the relation between the input air and steam production. Using a model with 7 states

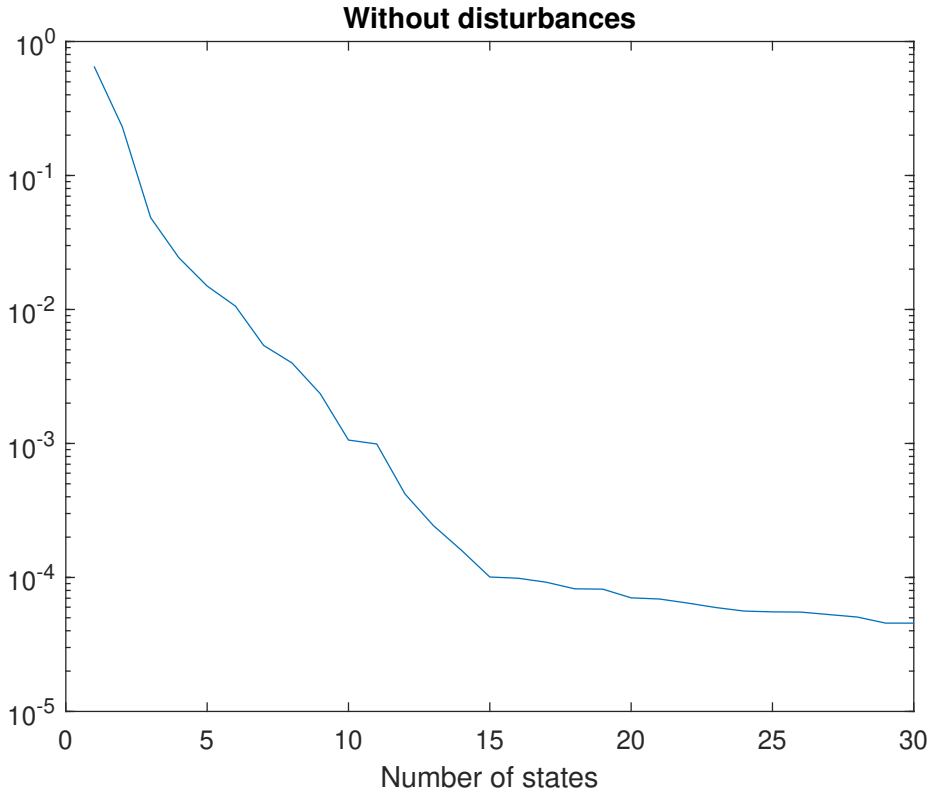


Figure 7.1: $\frac{\sum_{i,i} \sigma_{i,i}}{\sum_{j=1}^N (\sigma_{j,j})}$ from the SVD $U\Sigma V^* = H_1$

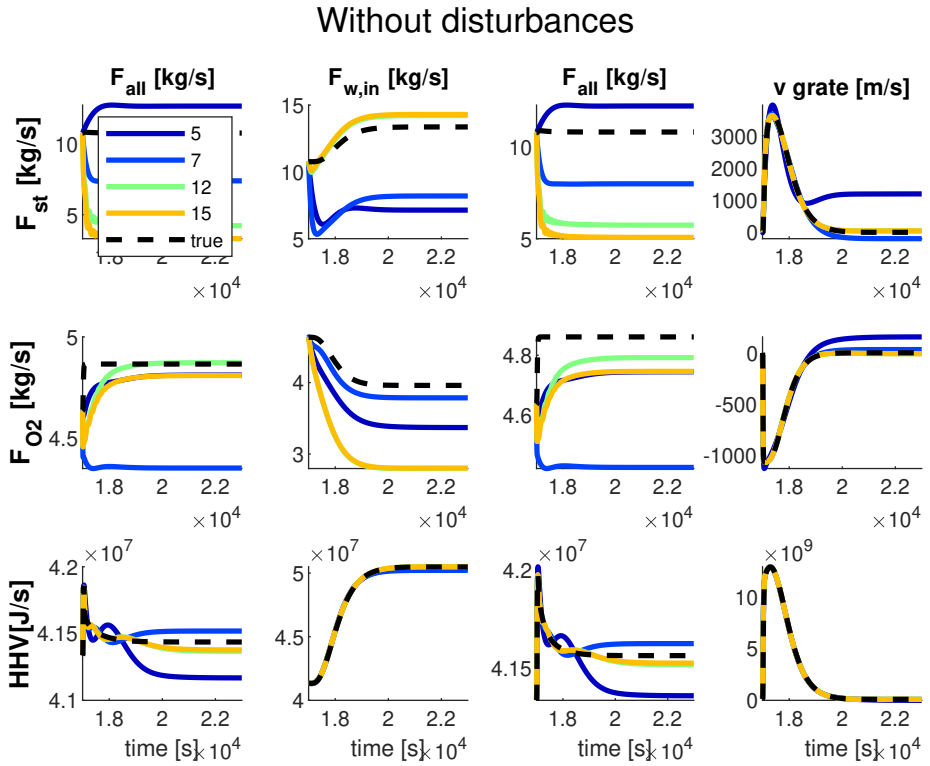


Figure 7.2: Approximation from un-scaled measurements. The lack of scaling causes an over-fixation on HHV and v_{grate}

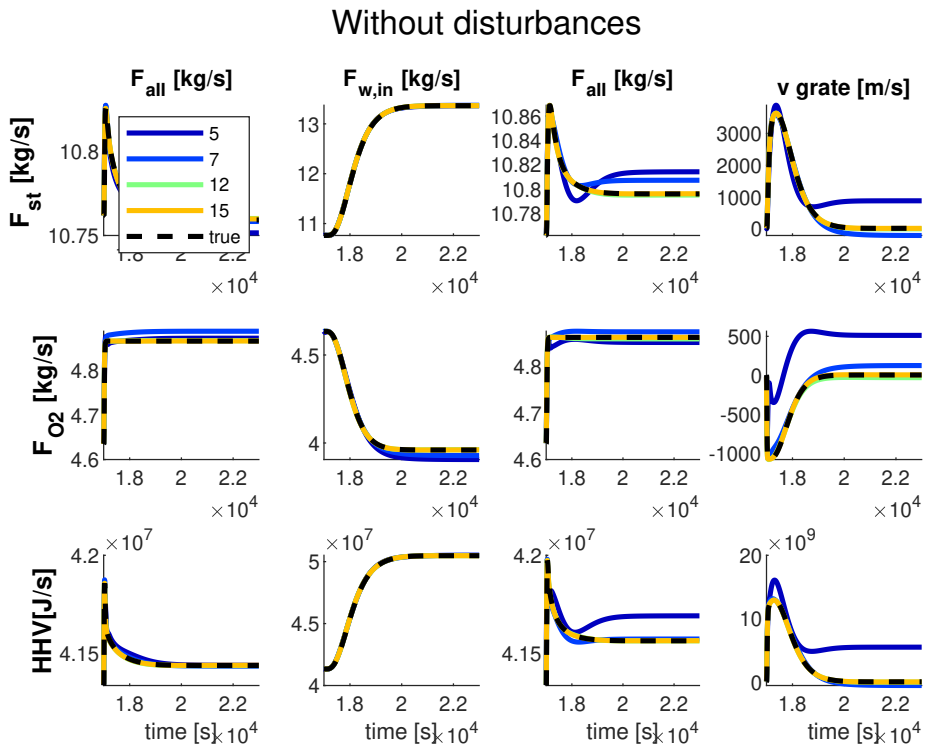


Figure 7.3: Approximation from scaled measurements. The relative error is now much smaller

might look sufficient at first, but later experiments show that a model order of 12 was found to greatly outperform it when attempting to control the plant. The reason for why the model has to be so precise is most likely because the controller can get an immediate increase in steam production by increasing F_{aI} while decreasing F_{aII} . If the controller uses this method to suppress immediate changes in steam-production, then this part of the model has to be as precise as possible.

7.1.2 Expanding the model with disturbances

Additional experiments were also performed in an attempt to estimate a model for both process disturbances and normal inputs. Figure 7.4 shows how much each state contributes to the new approximation. Figure 7.1 and 7.4 are remarkably similar, but the model without disturbances decreases faster in the beginning, which is the reason for why a simpler model can be chosen.

Just like in the previous model, the inputs and outputs have to be scaled to get a decent model. Additionally, the extra inputs representing the measurements require a more complex model if everything is to be modelled somewhat properly. As a result, a model of order 15 is used instead of a model of order 12.

The estimated response reinforces the notion that Q_{grate} does not affect F_{O_2} very much. This is also the reason as for why the model seems to struggle more with those outputs than most of the other input-output relationships.

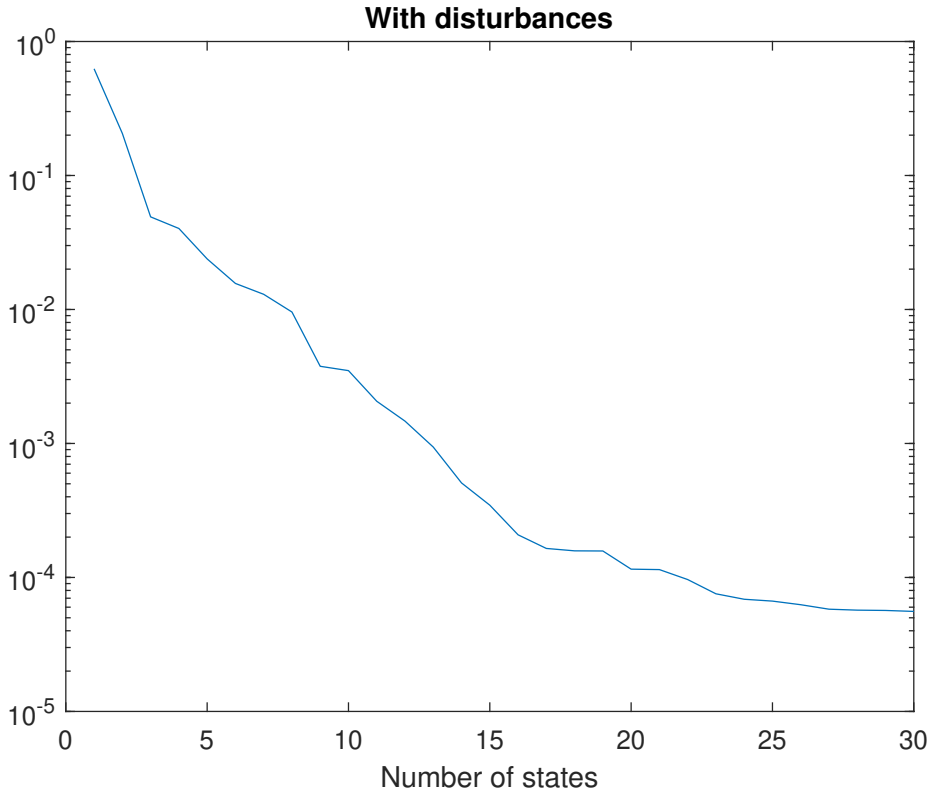


Figure 7.4: $\frac{\sum_{i,i}}{\sum_{j=1}^N(\Sigma_{j,j})}$ from the SVD $U\Sigma V^* = H_1$

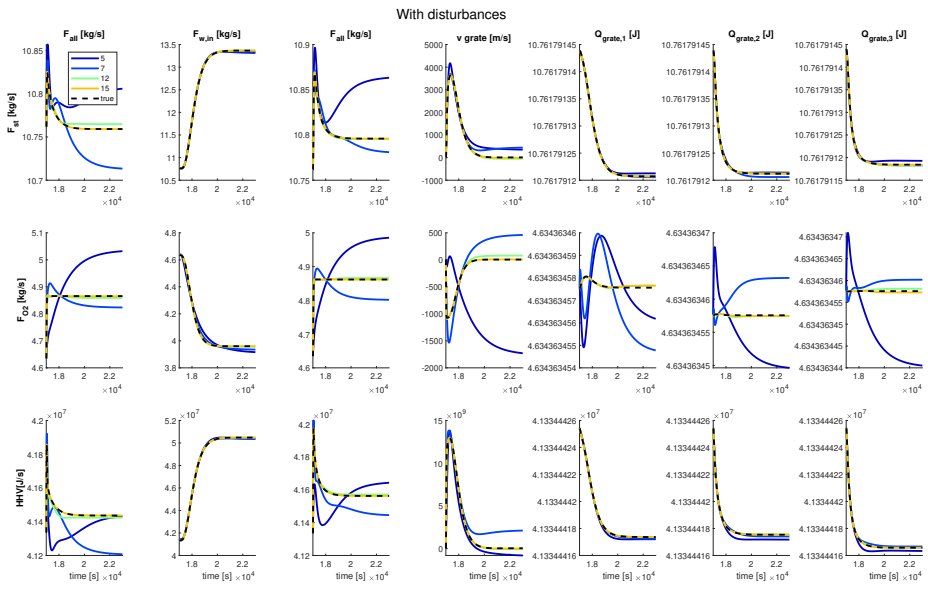


Figure 7.5: Model that also shows the dynamics of disturbances

7.2 PID-results

From an earlier project done by a summer-student at SINTEF, another controller has already been implemented. As a result, implementing SIMC or similar tuning methods is almost only interesting from the perspective of showing that it can be done. An easier approach is to make an automatic tuning algorithm that tries to improve on the parameters that have already been used.

If the simulator does not take too much time for each simulation, it is possible to automate parts of the tuning process and improve the controller by brute force if some kind of measure of the quality can be found. Normally, a good measure for the quality of a controller will involve looking at overshoot, settling time and potential stationary deviation. In this case, the cost of the controller is instead deviation e of the theoretical noise-free measurement $y_{\text{noise-free}}$ from the reference value.

$$\text{cost} = \left(\sum_{j=1}^m \sum_{i=1}^N \alpha_j (e_{T_i,j})^2 \right) + \sum_{j=1}^m \beta \max_{T_i} e_{T_i,j} + \sum \gamma_j \cdot \left| \sum_{i=1}^N e_{T_i,j} \right| \quad (7.1)$$

$$(7.2)$$

MATLAB has an implementation of the Nelder Mead algorithm in accordance to how it was described in Lagarias et al. (1998). This makes it possible to get good controllers by automatically tuning the parameters during the night.

The main drawback is that the algorithm still requires at least around 1000 iterations to find a really good solution. Additionally, the optimisation does not take robustness or noise-rejection if it is not made into a part of the cost.

An example of the progress of an automated tuning process can be seen in figure 7.6

7.2.1 AB-controller

Each output-variable is connected to one input-variable, with the secondary air being used to suppress any changes in the mass-flow of oxygen in the flue gas F_{O_2} and the

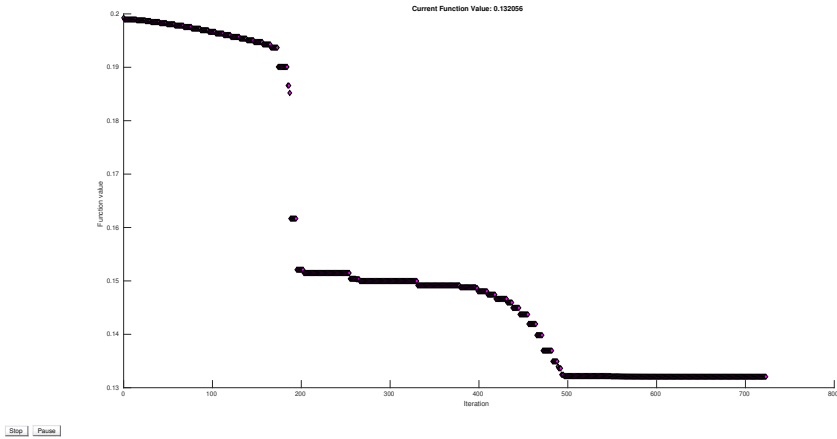


Figure 7.6: Norm of $y(t) - y_{ref}(t)$ at each improved time-step of the Nelder-Mead optimisation.

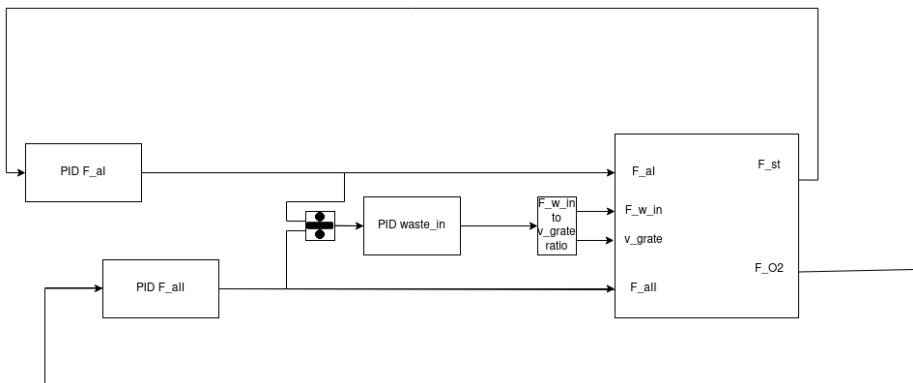


Figure 7.7: A/B controller structure. Setpoints are assumed

primary air being used to suppress changes in the production of steam F_{st} . Using the primary air to control the steam production over long time-horizons is usually a bad

idea since both a decent flow of primary and secondary air is needed for the gases to mix properly and for complete combustion to occur. As a result, the ratio between primary and secondary air $\frac{F_{al}}{F_{all}}$ is used as the final controlled variable in the plant. The controlled input-variable that is used to make this ratio follow the desired reference is a combination of grate-speed and the mass-flow of waste fed into the combustion chamber. A scaled difference could also have been used, but using a ratio may be desirable. Using too much secondary air is not too bad, beyond the loss of energy that comes from mixing the hot flue gas with ambient air. But using too little is potentially very bad since it might mean producing noxious gases.

Outputs

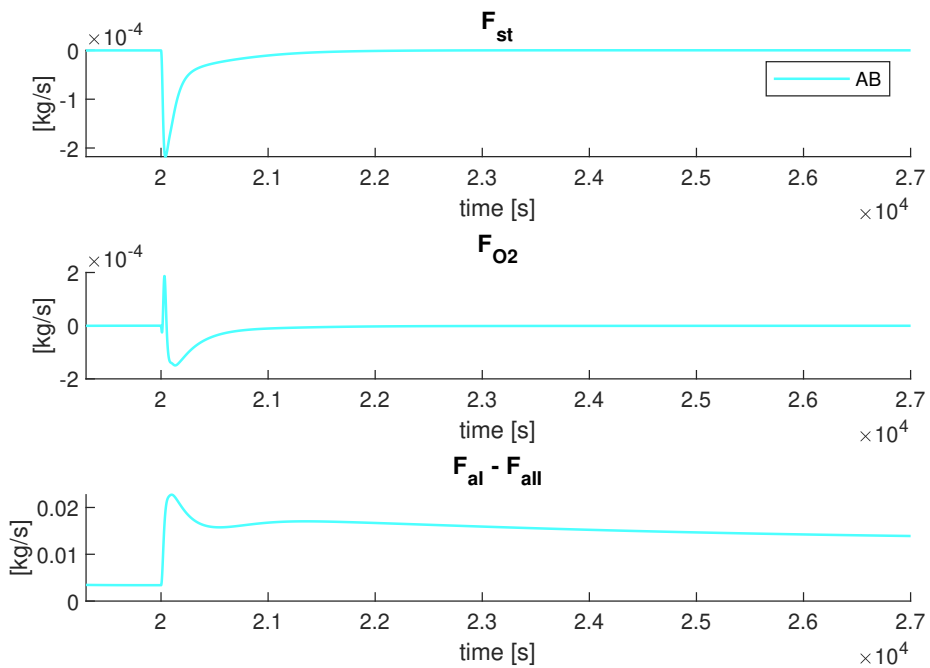


Figure 7.8: A/B controller disturbance step response

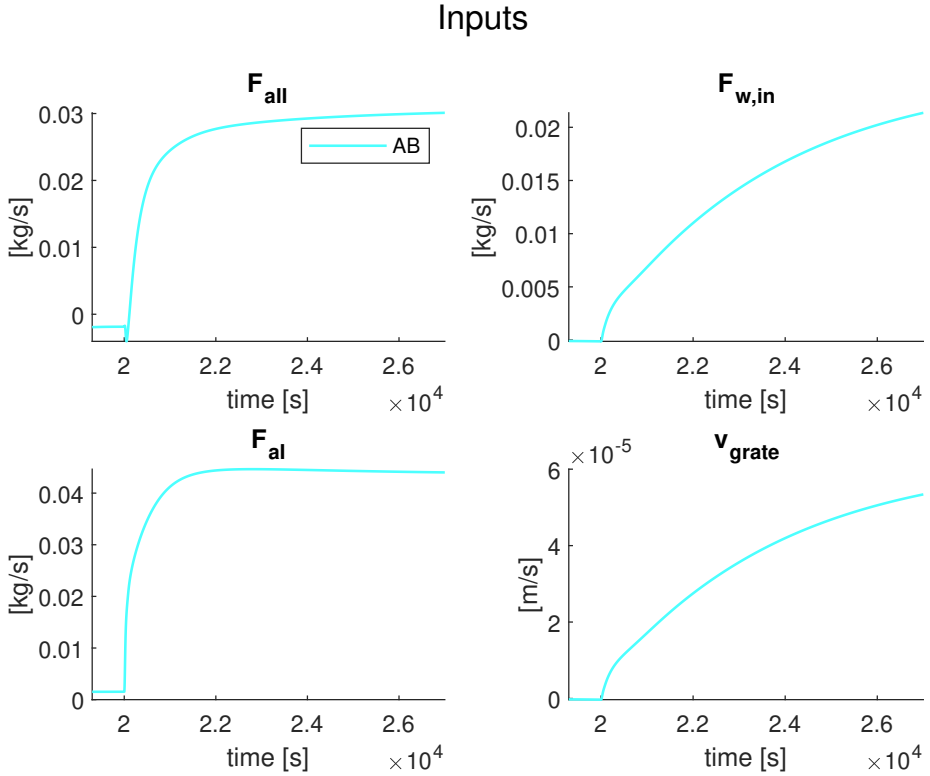


Figure 7.9: A/B controller disturbance step response

Given a sudden step in process-disturbances, the change can be seen in 7.8. The change involves all three values for Q_{grate} increasing by 50%. Regardless of the amount of measurement-noise affecting the controller, low-pass filtering the signals still desirable as a safety-precaution. Due to the nature of the simulator having some pure feed-through terms, it is also necessary for completing a simulation. A basic continuous, linear, time-invariant low-pass filter with a time-constant of 0.1 was chosen.

$$\tau_{\text{low-pass}} = 0.1 \quad (7.3)$$

The noise-spectrum may be very different, depending on the type of method that is used for measuring each variable. A common for of measurement-noise is the hum from the 50Hz AC-grid. The low-pass filter does serve the rejecting the noise from the AC-noise, but if the noise has a frequency-component around 1Hz that is too large, then the result will be far less pretty.

$$\left| \frac{1}{(50 \cdot 2\pi) 0.1j + 1} \right| \approx 0.0318 \quad (7.4)$$

$$\left| \frac{1}{(1 \cdot 2\pi) 0.1j + 1} \right| \approx 0.84673 \quad (7.5)$$

7.2.2 Cascaded A/B controller

Øie Kolden (2019) also went on to investigate the possibilities of using a theoretical sensor measuring the heating value of the waste HHV when controlling the plant. The measure proposed there was an estimate

$$H\hat{H}V = \sum_j C p_j F_{fg,j} (T_{fg} - T_{amb}) \quad (7.6)$$

which is the power delivered to the by a flow of flue-gas. The controller structure, as seen in figure 7.10 is the same as in 7.7, with the exception that the primary air is used to control $H\hat{H}V$, while the controller that tries to control the steam-production gives instead a reference to $H\hat{H}V$. The main advantage of using this controller should be that it can detect changes in the power delivered form combustion earlier, which allows for better control of the steam production.

As can be seen, from figure 7.10, both controllers have a rather similar response to changes in the waste quality when it comes to the oxygen concentration. The main

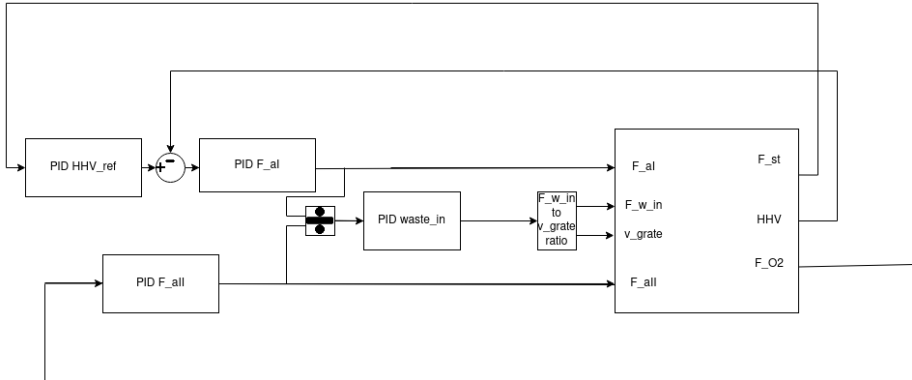


Figure 7.10: Cascaded A/B controller structure

advantage of using an estimate of the HHV-value when controlling the plant is the fact that the HHV-value changes before the steam-production, so the cascade controller can suppress the negative peak in the production of steam that happens for both the A/B-controller and the cascaded A/B-controller.

Figure 7.12 has a noticeable property in its response. It does not change the total amount of time needed to make the F_{st} converge back to its reference, but the maximum amplitude of the error has decreased significantly. Additionally when looking at the changes in inputs needed to achieve this, figure 7.11 shows that the inputs are almost identical, with the only difference being very slight differences in timing, with the cascaded controller being marginally more aggressive in its usage of primary air.

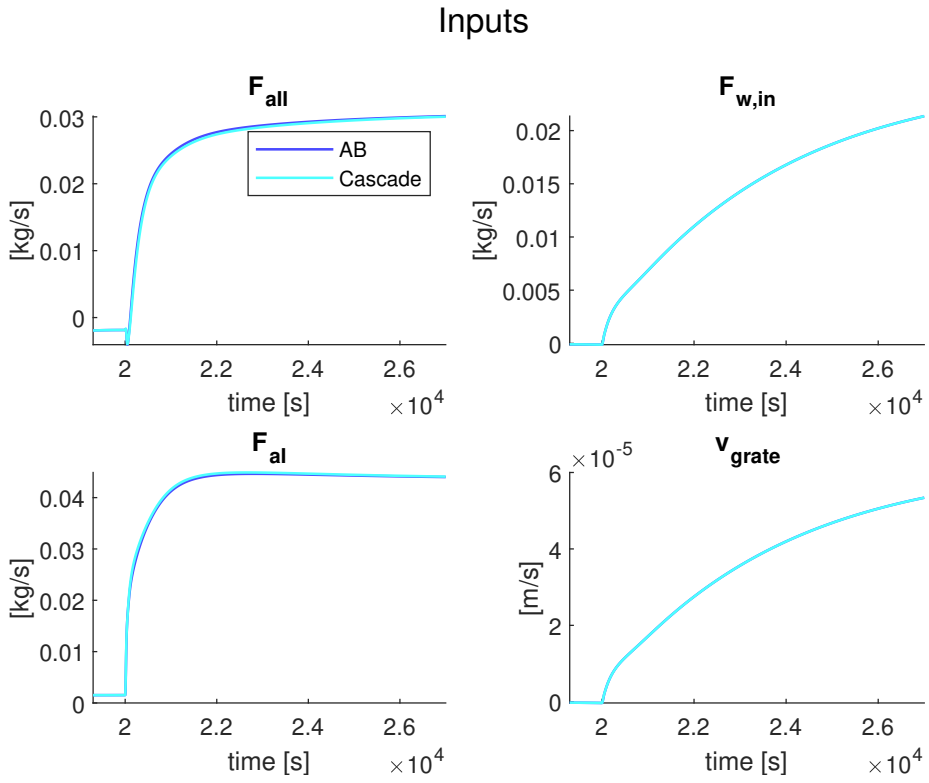


Figure 7.11: Cascaded A/B controller disturbance step response

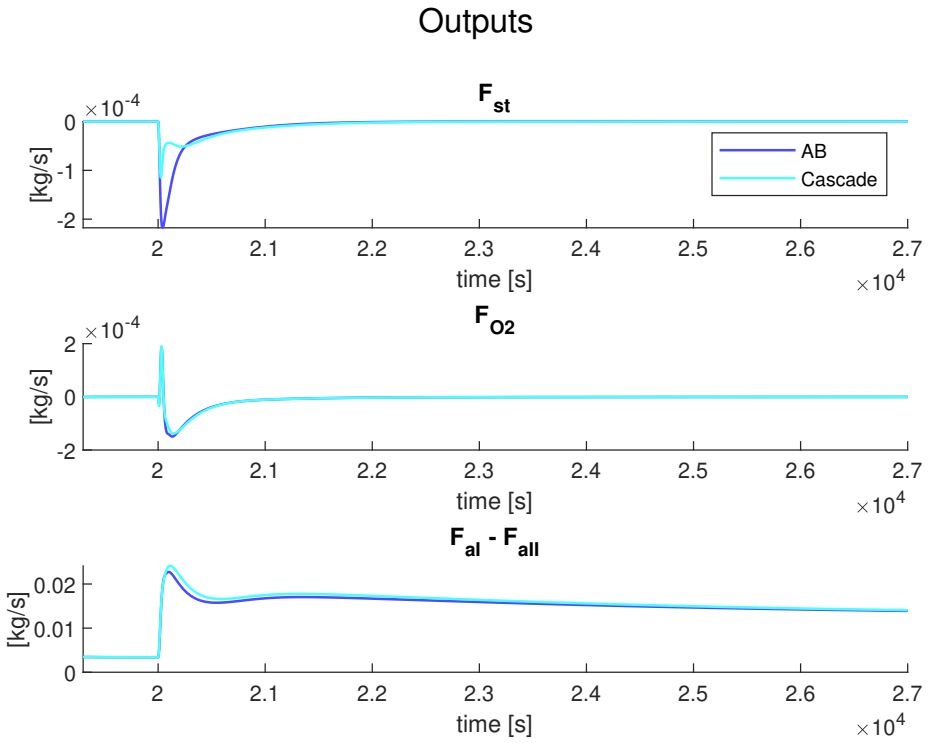


Figure 7.12: Cascaded A/B controller disturbance step response

7.3 MPC results

MPC offers a structure for designing a controller that should be able to stabilize the plant. Nevertheless, there is a multitude of different architectures which may provide different properties. In combination with this, some weights have to be tuned for the controller to prioritize the objectives we want to follow.

Multiple criteria should be upheld.

- F_{st} and F_{O_2} should be stabilized and should reject disturbances.
- Non-zero mean disturbances should be rejected completely.
- The controller should be verifiable through the use of a simulator.
- The ratio between F_{aI} and F_{aII} should only be able to deviate from the pre-determined reference ratio temporarily
- In the case where there are hard safety-constraints, the controller should try to uphold these.
- The controller should be well-behaved, even if the inputs have saturation limits or limits to the rate of change.

Normally, an MPC differs somewhat from an LQR in that it can express a cost for the change in variables. The LQR usually has to use high-pass filters to achieve a similar effect. Fortunately, a low-pass filter on the input is already required to control the plant, since both MPCs and simulators are bad at handling feed-through terms through the plant and controller. As a result, the low-pass filter can be used to make a high-pass filter:

$$1 - \frac{1}{\tau_{\text{low pass}}s + 1} = \frac{\tau_{\text{low pass}}s}{\tau_{\text{low pass}}s + 1} = h_{\text{high pass}}(s) \quad (7.7)$$

Matlab already has a function `lqry`, which returns the gain K_{lqr} . K_{lqr} optimizes the Ricatti equation if an output-cost Q_y and a state-cost R is provided. An optional

cost matrix, N , which punishes or rewards certain combinations of x and u is also used to implement the punishment for changing the inputs too much.

$$K_{lqr} = \text{lqry}(\text{lqry_plant}, Q_y, R, N) \quad (7.8)$$

`lqry` should be used instead of normal `lqr`, since performing the multiplication $Q = C^T Q_y C$ may break the positive semi-definiteness of Q , due to numerical errors. Any cost that may instead result from changes in input or output can be expressed by expanding the observed states

$$C_{lqr} = \begin{bmatrix} C \\ C_{\text{input_filter}} \\ C(A - I) \end{bmatrix} \quad (7.9)$$

Nelder-Mead can be used to optimise the costs in the controller and estimator. Because MPCs and LQR are usually tuned by multiplying the weights of factors 10, it is paramount that the optimised vectors represent exponents of 10, as the alternative would lead to very poor convergence. As for now, it is assumed that all forms of noise and disturbances are uncorrelated, so both R_{lqe} and Q_{lqe} are made to be diagonal matrices. The same is done for Q_y and R , which is used by the controller. Tuning the plant completely blindly can often be a bad idea since it can lead to several unneeded steps before reaching any decent results. There is no guarantee that the problem is truly convex, so it might be preferable to tune the controller manually until it does a decent job.

7.3.1 Different LQE structures

Section 7.1 showed two different kinds of models, one where the disturbances were measured, and one where they are almost ignored completely. An LQE usually needs some measure of variance with regards to measurement noise and process disturbances.

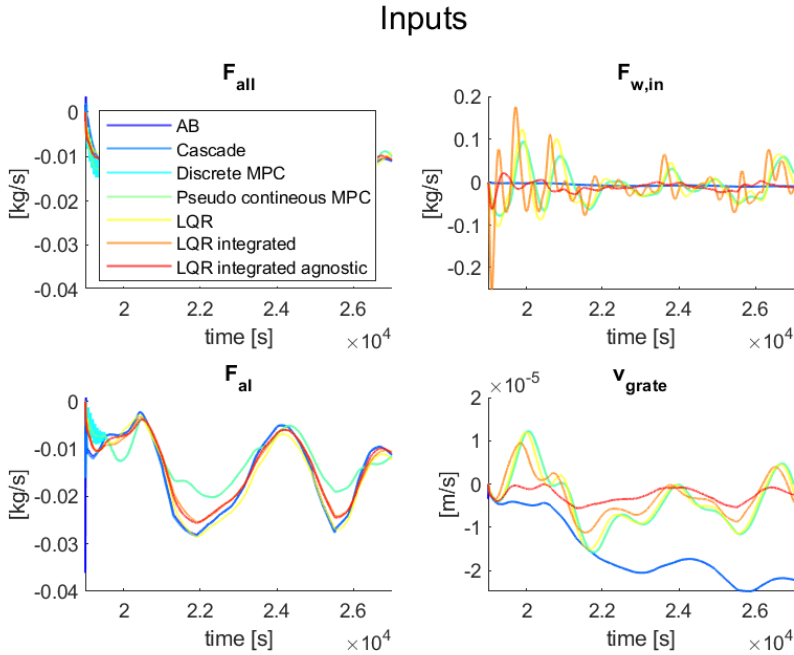


Figure 7.13: All controllers that assume all disturbances to only come from Q_{grate} perform poorly. Only the last one is not based on doing that

By using data that is somewhat representative of process-data, it should, in theory, be possible to create a good LQE only by using the estimated model and those measures. In practice, the resulting controllers struggled significantly and performed worse than the PID-controllers, as shown in figure 7.3.1.

The solution to this would normally be a rather unpleasant one, as it involves tuning the expected disturbance for each state, and the expected noise for each input. Luckily, it is not so hard to make an estimator that functions decently, and that can stabilise the plant in combination with a gentle controller. As a result, tuning the estimator becomes part of the optimisation tuning problem.

7.3.2 Integral LQR

Almost any model predictive controller that is not overly aggressive should be able to fulfil point 1. The second one is a bit more tricky, but the normal solution to this is to add integral states to all controlled outputs. This also has the added benefit of allowing reference tracking, even without explicitly creating a feed-forward matrix. The easiest way to get integral action for the outputs is to expand the state-space

$$A_I = \begin{bmatrix} A & 0 \\ I & 0 \end{bmatrix} \quad (7.10)$$

$$B_I = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (7.11)$$

$$C_I = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \quad (7.12)$$

$$D_I = \begin{bmatrix} D \\ 0 \end{bmatrix} \quad (7.13)$$

$$(7.14)$$

As can be seen in figure 7.3.2, the linear quadratic controller still loses to the PID-controllers on some points. Mostly in regards to steam production and O_2 concentration. The reason for this may come from the fact that the MPC uses one filter on the input with the cutoff-frequency of 7 seconds to mirror the filter that had to be used during parameter estimation to avoid aliasing, as well as the filter on the plant input, that is used to make the estimated plant proper. The PID only uses a filter with a time-constant of 0.1 seconds, so an aggressive response is required to compete, then the PID-controller has an advantage.

The main draw-back is that the LQR-controller has an unintended change in the amount of air released. This also happens to the PID, but that is because the selected

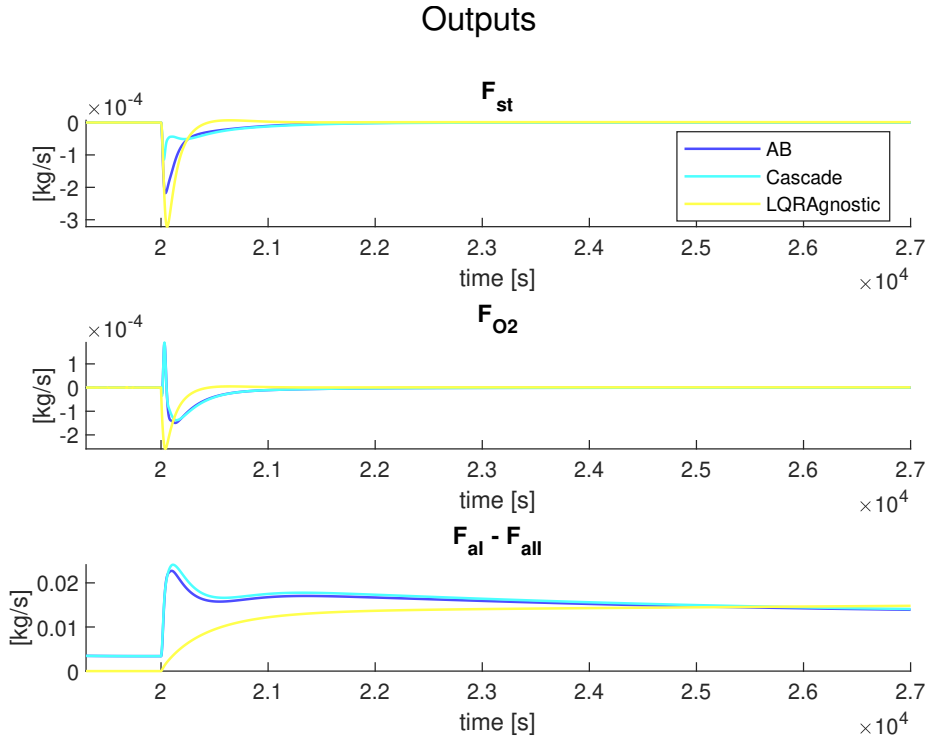


Figure 7.14: Outputs: LQR versus PIDs

ratio between primary and secondary air is chosen to be different from 1. Even if aggressive usage of primary and secondary air can be a useful tool for handling disturbances, changes in grate-speed and waste-flow should be the preferred method for handling persistent changes in waste-composition. It is possible to give an additional cost for whenever F_{al} and F_{all} . The resulting controller, as seen in figure 7.3.2 loses a lot of its ability to reject disturbances properly, while also being somewhat robust.

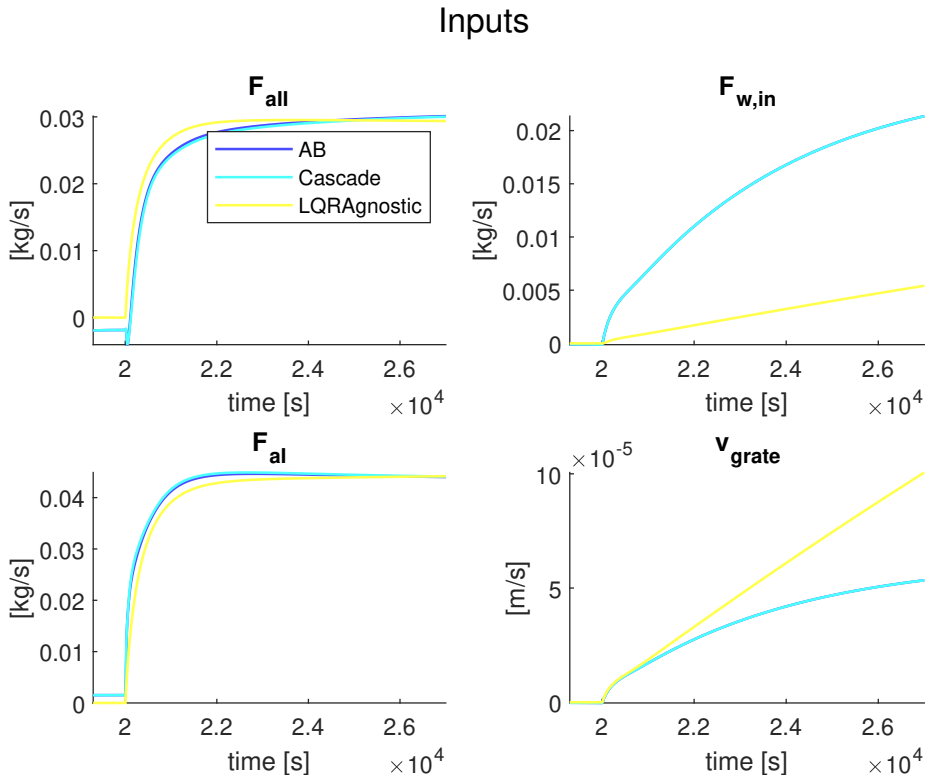


Figure 7.15: Inputs: LQR versus PIDs

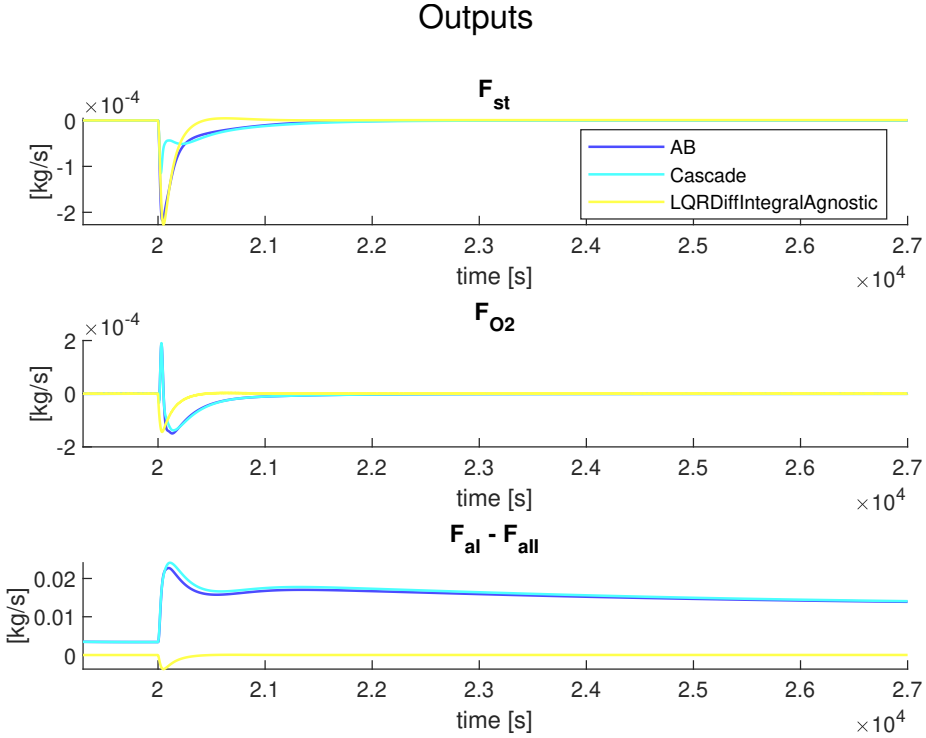


Figure 7.16: Outputs: PID vs LQR with integral cost on $F_{aI} - F_{aII}$

7.3.3 LQR with integral cost on $F_{aI} - F_{aII}$

Balancing the normal costs, trying to reject disturbances, while also enforcing a decent ratio between F_{aI} and F_{aII} may prove to be a pointless endeavour. It may also mean not using a very useful tool for rejecting disturbances since allowing the two air-flows to temporarily deviate has a faster response than changing the amount of waste ted into the furnace. The intuitive solution to this is to add an integral cost to $F_{aI} - F_{aII}$ instead, and then giving it a rather low priority compared to the other objectives. The resulting response is shown in figure 7.3.3

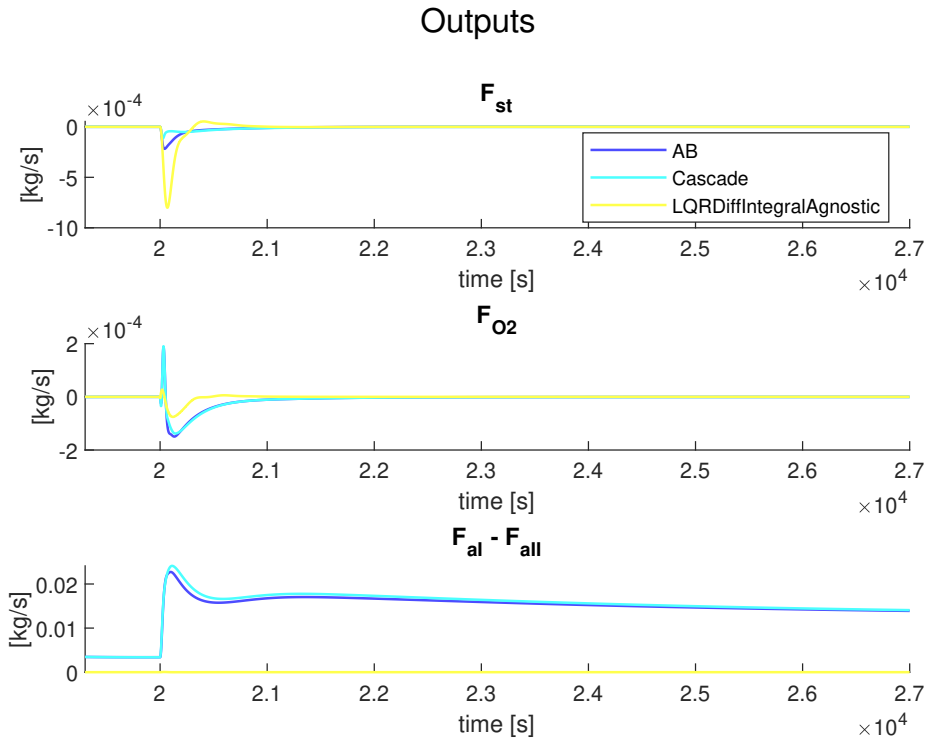


Figure 7.17: Inputs: PID vs LQR with integral cost on $F_{al} - F_{all}$

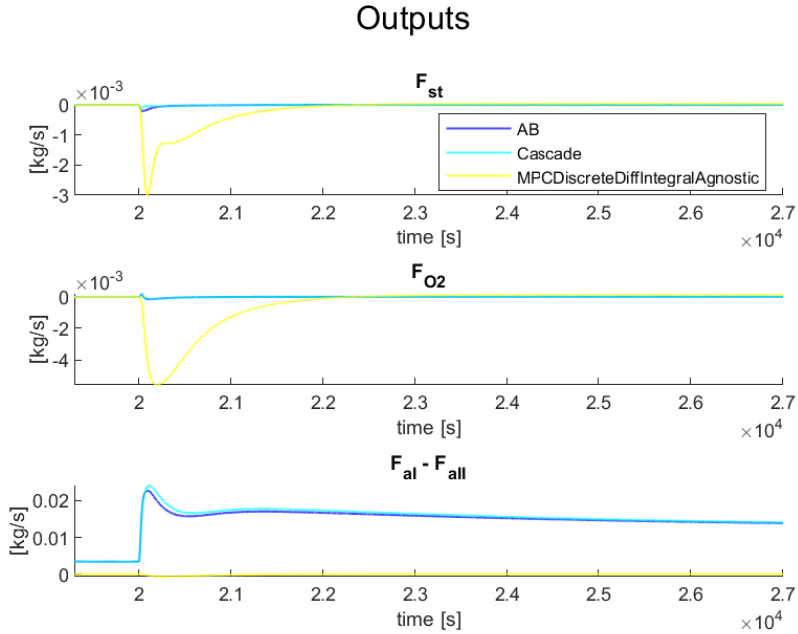


Figure 7.18: Outputs: PID vs MPC with integral cost of $F_{al} - F_{all}$

7.3.4 Discrete MPC

Even if a rather low sampling-time might work well for an MPC in practice, it does not work as well with the simulator, due to the stiffness of the problem. As a result, a sampling-time of 20 had to be chosen, even though that is noticeably slower than the preferable sampling-time of somewhere around one second. As a result, the MPC will have objectively worse performance than the LQR as long as all inequality-constraints are inactive. The delay means that the weight used for the LQR may not work as well for an MPC. The simulation will also run more slowly when an MPC is involved.

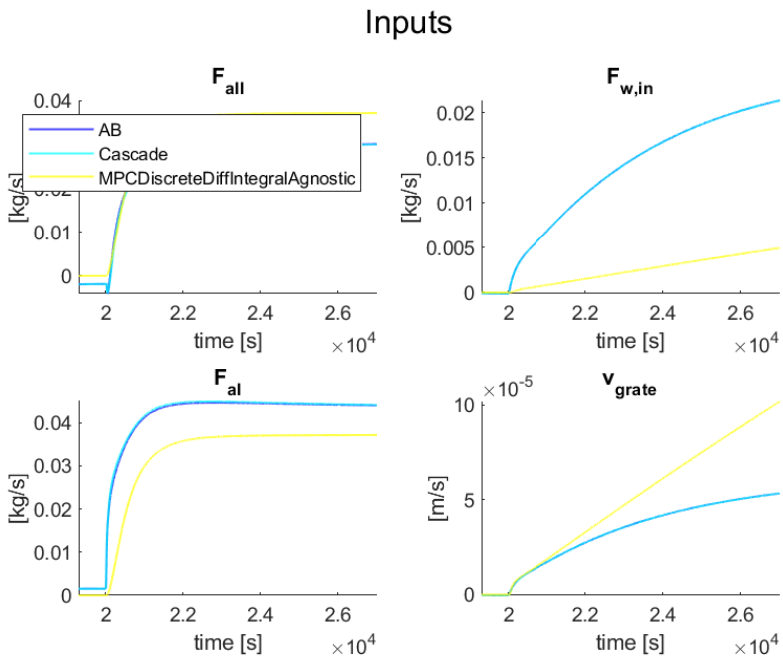


Figure 7.19: Inputs: PID vs MPC with integral cost of $F_{\text{al}} - F_{\text{all}}$

7.4 Robustness analysis

Even though the system is stable with the model that has been given, the assumptions that were made may be somewhat incorrect. As a result, the constructed controller should preferably also have at least some robustness against modelling errors. This analysis will only focus on errors in input gain and on time-delays. All models described in this section will be of continuous LTI systems.

7.4.1 Combined estimator and physical system

The physical plant, and the one used by the MPC are not entirely the same. As a result, some extra steps have to be taken when representing both the physical system and the estimated one in the same state-space. This analysis will only go over input-delays and not output-delays.

Let A, B_p, C and D represent the physical system and the matrix B input matrix of the estimated system. Finally, let K_{lqe} represent the feedback matrix used by the linear quadratic estimator. Then the open-loop system between the real state and the estimated state can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ K_{lqe}C & A - K_{lqe}C \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B_p \\ B \end{bmatrix} u \quad (7.15)$$

The controlled system is not necessarily the same as the estimated system. A low-pass filter has to be used by the controller or estimator to make remove all feed-through terms from LTI-system. This low-pass filter is represented by the matrices A_F , B_F and C_F . Additionally, the controlled states are not necessarily the same as the measured outputs, so a different pair of matrices C_c D_c are used to extract them from the estimate. Said measurements are used for making the integrator states of the plant. Finally, a padé-approximation is needed for the delay between the input-filter and the rest of the process. The delay will be estimated by A_D , B_D , C_D and D_D .

The full system is divided into several smaller sub-systems.

$$x_{full} \begin{cases} x_{\text{physical}} \\ x_{\text{estimated}} \\ x_{\text{input filter}} \\ x_{\text{integrated outputs}} \\ x_{\text{padé approximation}} \end{cases} \quad (7.16)$$

$$A_{full} = \begin{bmatrix} A & 0 & B_p D_D C_F & 0 & B_p C_D \\ K_{lqe} C & A - K_{lqe} C & B C_F & 0 & 0 \\ 0 & 0 & 0 & A_F & 0 \\ C_c & 0 & D D_D C_F & 0 & D C_D \\ 0 & 0 & B_D C_F & 0 & A_D \end{bmatrix} \quad (7.17)$$

$$B_{full} = \begin{bmatrix} 0 \\ 0 \\ B_F \\ 0 \\ 0 \end{bmatrix} \quad (7.18)$$

Instead of using normal states as outputs, it is also possible to translate them directly into the corresponding optimal set of inputs. The matrix K_{lqr} uses the estimated states, as well as the filtered states and the integrated states.

$$C_{full} = \begin{bmatrix} 0 & K_{lqr} & 0 \end{bmatrix} \quad (7.19)$$

Because of the low-pass filter, D_{full} is always 0. The stability of the feedback loop can be analysed by adding a padé approximation of a delay in the feedback between the optimal next input given out by C_{full} and the input used on B_{full} .

7.4.2 The root locus plots

With a fully functioning state-space system of both the physical system and the estimated one, it becomes possible to say something about how the poles evolve, depending on delay and static gain error. To test the static gain error, it is simply sufficient to multiply B_p by some constant. The robustness to delay can be tested by adding a padé- approximation in the feedback-loop, of some pre-determined order, and then setting the desired delay. In this analysis, order 9 was chosen, although there may be reasons for choosing other padé-approximations as well.

In this section, the LQR-controller with integral cost on $F_{aI} - F_{aII}$ will be covered, since it was one of the best-performing architectures.

The controller is quite sensitive to time-delays from the results in figure 7.4.2, becoming unstable if all inputs suffer an input-delay of only 0.3 seconds. If it can be guaranteed that there will be some inputs without delays, some stronger guarantees can be given instead. A delay of 5 seconds can be tolerated if there is no delay in the grate-speed. Sadly, this input that gives a notable improvement is if v_{grate} .

The fact that v_{grate} is one of the most problematic inputs was not something that could be solved in this project. It is especially bad, since it would seem more reasonable that the fans have a lot less variance in their delay compared to the ram and the grates, simply because they are a lot faster. In theory, decreasing the reliance on v_{grate} should make it better, but no solution was found during this project that also resulted in performance that could rival the previous controller. There may be some solution, using a different architecture or bu using a different set of weights.

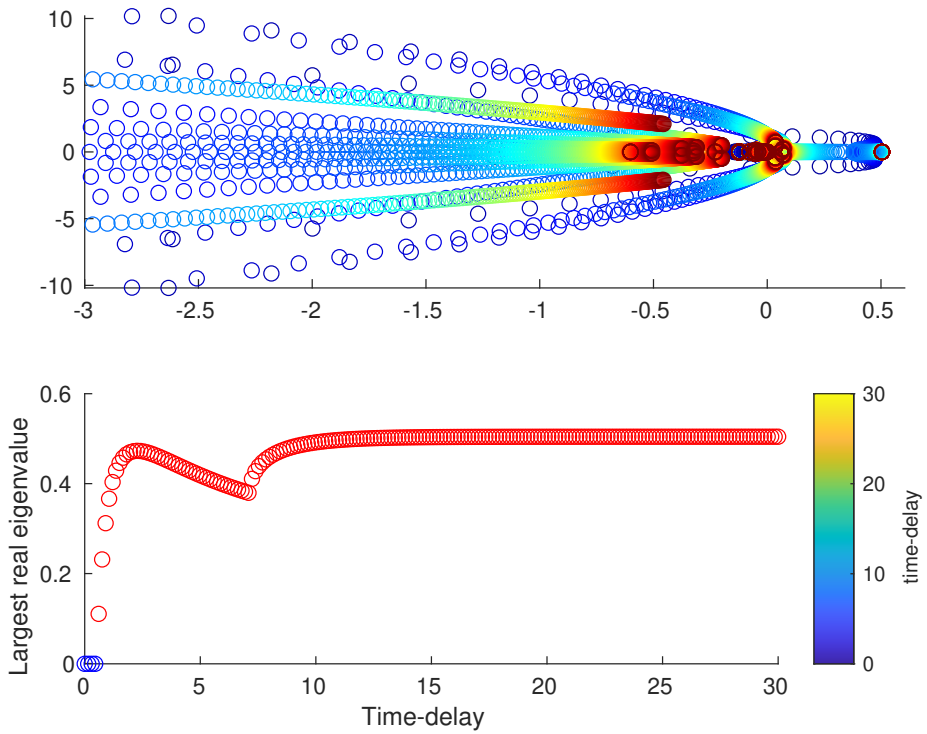


Figure 7.20: Root locus plot for delays on all plant inputs

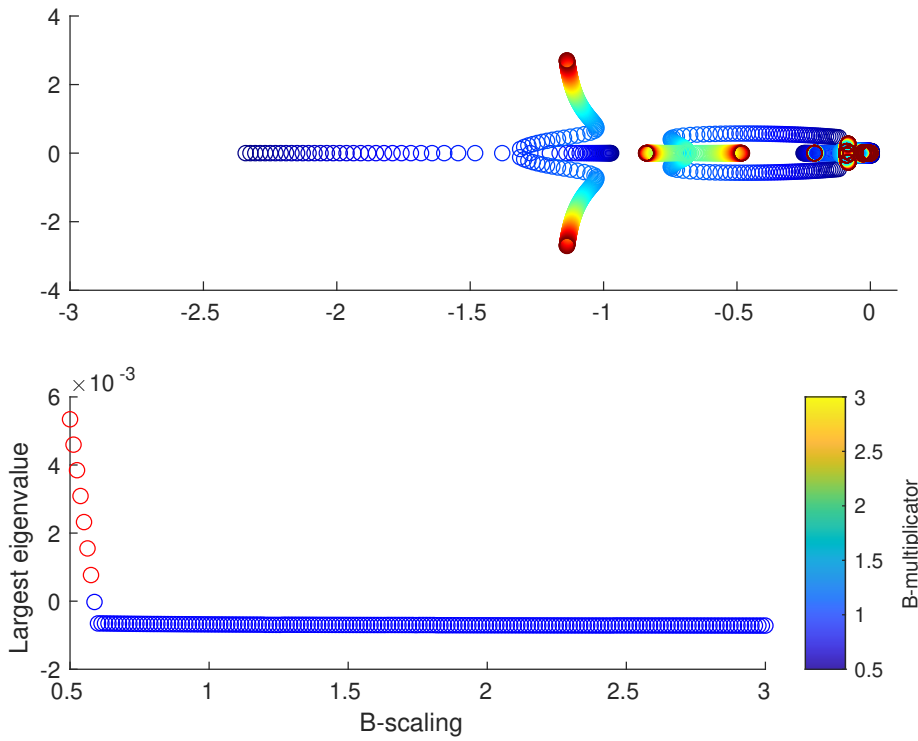


Figure 7.21: Root locus plot for linear scalings of B

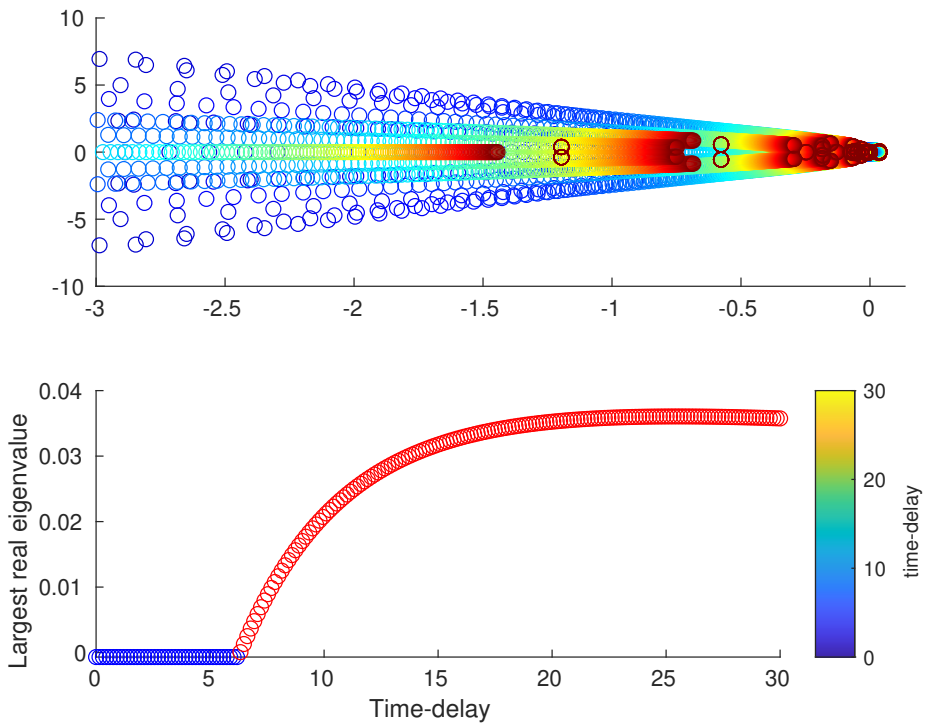


Figure 7.22: Root locus plot for unmodeled time-delays

Chapter 8

Conclusions and future work

8.1 Conclusion

The task of controlling an MSW combustion process proved to be a difficult one. This project provided the luxury of working with a simulator, which gave a rich source of potential data, as well as possibilities to verify the results. This also proved to be at the core of the issue, since certain kinds of controllers could not be implemented, due to the simulation time going towards infinity, or the simulator crashing with unknown errors, if the controller had discrete samples that were too frequent.

In the project, the ERA was implemented, allowing for estimating the dynamics of a linear system with unmeasured internal states was implemented. Furthermore, several different LQR and MPC controllers were implemented. Hard limits on the input-variables were also added to the MPC to better allow it to handle saturation, but this never became relevant with the disturbance test-data that was available. Hard limits on the measured outputs, with slack variables were also implemented, but the results were never discussed properly, due to them never being enforced. Due to what is assumed to be minor errors in implementation, the LQR-controller was not able to outperform the PID-controllers. There were, however several more positive outcomes. A root-locus method was implemented for estimating the stability of a closed-loop

estimated system. Additionally, a simple method for automatically tuning the PID-controller and the optimal controllers was also implemented.

8.2 Future work

The purpose of this project was to develop a controller that could outperform the current PID-controller. This was only achieved in part since the current PID-controller was improved by the process of automated tuning. There are a multitude of different things that could be done if more time is was available.

- The most important improvement is the fact that there was most likely something wrong in how the implementation of the MPC which causes it to under-perform. It may be because of more aggressive filtering, but the result is that
- Implement some estimate for the robustness of the current PID-controller. Most likely by using the ERA to make open-loop models that can be analysed.
- Feed-forward control on the plant was implemented, but did not work (see appendix). The nature of the problem may require a more advanced MIMO feed forward controller.
- Using integrators instead of low-pass filters could, in theory, implement a better way to limit the controller from changing the inputs too quickly.

8.3 Possibly useful improvements

Some of the potential future work are relatively large tasks, and should not be pursued unless there are explicit observations that indicate that they might be needed.

- If the true test-data make the non-linearities in the plant more prominent, a simplified nonlinear model could be implemented with nonlinear methods like simple linear regression, or the more advanced SINDy-algorithm. But these methods usually require full-state measurement.

- Proper analysis of process-disturbances and measurement-noise should be implemented since it might show the need for specialised band-stop filters.
- Exploration of how the performance degrades if the estimated HHV-value is not available.
- An exploration of what measurement-techniques are available, and if those measures could be made to behave well in combination with a linear controller.
- If only process data is available on future unknown plants, Observer Kalman Filter Identification may be used required for the ERA to work in the presence of process disturbances and measurement noise.

Appendix A

Linear Quadratic Regulator

The Linear Quadratic Regulator is a much-explored part of optimal control. It has the advantage of being able to minimize equation A.1 for any x , simply by pre-computing a state-feedback matrix K_{lqr} . This can be particularly useful, since a discrete MPC with an infinite time-horizon and an LQR will give the same sequence of inputs, given the same costs and same sequence of measurements.

The LQR minimizes a quadratic cost.

$$J = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k + x_k^T N u_k \quad (\text{A.1})$$

Most sources do not explain why the Riccati equation gives an optimal feedback matrix K_{lqr} , simply that it does. But Boyd (2008a) gives does give a proof. Since the source is a set of lecture notes, and not a piece of published work, the proof ill has to be reiterated here.

Given a controllable plant

$$x_{k+1} = Ax_k + Bu_k \quad (\text{A.2})$$

If the plant is controllable, for some $n \leq \infty$

$$\exists u_1, u_2, \dots, u_n : x_n = 0 \quad (\text{A.3})$$

Since all linear systems have their equilibrium at 0,

$$x_0 = 0 \quad (\text{A.4})$$

$$x_{k+1} = Ax_k + Bu \quad (\text{A.5})$$

$$\Rightarrow x_k = 0 \forall k \geq 0 \quad (\text{A.6})$$

This implies that if a plant is controllable, there exists an input sequence which ensures that $J \leq \infty$. The way this optimal J is found is by a recursive function. Imagine a single-step horizon problem, where the cost Q of the current state and of the symmetric cost of the next state P_1 are both known. Then the problem looks like

$$J_1 = x^T Qx + u^T Ru + (A + Bu)^T P_1 (A + Bu) \quad (\text{A.7})$$

$$Q = Q^T \geq 0, R = R^T \geq 0 \quad (\text{A.8})$$

The best single input u can be found by taking the gradient of J with respect to u and solving for 0. By using $\nabla_u(u^T Au + b^T u) = (A + A^T)u + b$ from Petersen and Pedersen (2012), and the symmetry of R and Q , $\nabla_u J$ becomes

$$\nabla_u J_1 = 2 \left(R + B^R P_1 B \right) u + 2 \left(B^T P_1 A x \right) \quad (\text{A.9})$$

Setting $\nabla_u J = 0$, and separating u to the left-hand side gives

$$\left(R + B^T P_1 B \right) u = - \left(B^T P_1 A \right) x \quad (\text{A.10})$$

R is positive definite, while P_1 is guaranteed to be positive semidefinite. Therefore,

no eigenvalue can be zeros and the matrix must be invertible. The optimal input u is proportional to the state x . The transformation from the current state to optimal input therefore becomes a trivial matrix-multiplication, with the matrix $K_{lqr,1}$

$$K_{lqr,1} = - \left(R + B^T P_1 B \right)^{-1} \left(B^T P_1 A \right) \quad (\text{A.11})$$

As the reader might have guessed, the single-horizon cost can be used to recursively find the cost of an arbitrary number of steps.

$$P_{k+1} = (A - BK_{lqr,k})^T P_k (A - BK_{lqr,k})^T + K_{lqr,k}^T R K_{lqr,k} + Q \quad (\text{A.12})$$

Because of equation A.3 and A.6, there exists a sequence of inputs that gives a finite infinite-horizon cost, so the optimal solution has to be just as good or better. Consequently P should converge to some positive definite matrix P_∞ as enough iterations have been done. After the cancelling the redundant terms, the equation for P_∞ becomes the Riccati equation

$$P = A^T P B \left(R + B^T P A \right)^{-1} B^T P A \blacksquare \quad (\text{A.13})$$

The matrix equation will always have at least more than one valid solution, but only one of them is positive definite, and therefore the correct one. The state feedback gain can easily be found from the solution.

$$K_{lqr} = - \left(R + B^T P B \right)^{-1} \left(B^T P A \right) \quad (\text{A.14})$$

There is also a similar proof in Boyd (2008b), which is based on the idea of approximating the dynamics over a time-step $h \rightarrow 0$, which gives an optimal input

$$K_{lqr,cont} = -R^{-1} B^T P_t \quad (\text{A.15})$$

And then also solving the Riccati differential equation A.16 backwards in time

$$-\dot{P}_t = A^T P_t + P_t A - P_t B R^{-1} B^T P_t + Q \quad (\text{A.16})$$

The solution can also be found by setting $\dot{P}_t = 0$, and simply solving the matrix equation.

Appendix B

Feed forward control

This was never implemented into the controller, which is why it is in the appendix.

An alternative to trying to suppress most disturbances to estimate them through output-measurements, is if they can be measured directly. If the plant is sufficiently well-behaved, then it is possible to make a feed-forward controller that can cancel the effects of the process disturbance. Most feed-forward control is done on SISO¹ systems, like what is seen in figure B. In this model, the process dynamics from the controllable input to the output is $P(s)$, the dynamics from the process disturbance to the output is $G(s)$ and the feed-forward controller is $F(s)$, while the feed-back controller is $C(s)$. In a SISO-system, if $P(s)$ has no zeros in the right half-plane the easiest way to make a controller that will cancel the disturbance is by making

$$F(s) = \frac{1}{P(s)}G(s) \tag{B.1}$$

But this is usually not possible if any of the zeros are in the right half-plane. The result to this is normally to split $P(s)$ into two $P(s) = P_{mp}(s)P_{nmp}(s)$, where P_{mp} is all the state that make up a phase system², while the P_{nmp} only the poles in the right half-plane. It

¹Single Input, Single Output

²Minimum phase will mean that all zeroes are in the left half-plane

is not possible to get rid of the bad poles, so instead, a set of stable zeros can be used to counteract $P_{nmp} = (-T_1s + 1) \cdots (-T_ns + 1)$. Depending on what $G(s)$ is, some factor α can be chosen as to decide how aggressive the feed-forward controller should be.

$$F_{\text{not proper}}(s) = G(s) \frac{1}{P_{mp}(s)} \frac{1}{(\alpha T_1s + 1) \cdots (\alpha T_ns + 1)} \quad (\text{B.2})$$

If $P(s)$ has a higher relative degree than $G(s)$, then a set of low-pass filters is also needed to make the feed-forward controller realisable.

$$F(s) = F_{\text{not proper}}(s) \frac{1}{(\tau_{lp,1}s + 1) \cdots (\tau_{lp,N}s + 1)} \quad (\text{B.3})$$

Where M is the difference in relative degrees between the controllers. α should be chosen such that the feed-forward controller is not too careful as to not have any effect at all, but also so that the input does not cause an over-correction in the other direction. The initial value-theorem can give an indicator as to how the plant might react to the sudden change. By replacing all s with $\omega \rightarrow \infty$ and allowing, it is possible to see what the initial value to an impulse-response will be. In the case of this project, $G(s)$ is rather slow, and would be able to correct for a lot the reaction in the opposite direction. As a result, $\alpha = 1$ would possibly be a decent choice.

In the idealised case, the step-response from an input would look like what it does in figure B.2, which is the result of a feed-forward controller on the linearized plant in MATLAB. The undamped response is from the closed-loop system, from a step-response on the different changes in Q_{grate} . The damped one is if there is a feed forward controller there as well. Three controllers try to correct for the changes in steam production caused by one of the disturbances. These controllers change $F_{aI} - F_{aII}$ to achieve this damping effect. A fourth feed forward controller measures $F_{aI} - F_{aII}$ and tries to nullify its effect on the oxygen concentration by changing $F_{aI} + F_{aII}$

Problems can be seen when this feed forward controller is implemented into the closed-loop system in figure B.4. In theory, F_{st} and F_{O_2} should both improve from the

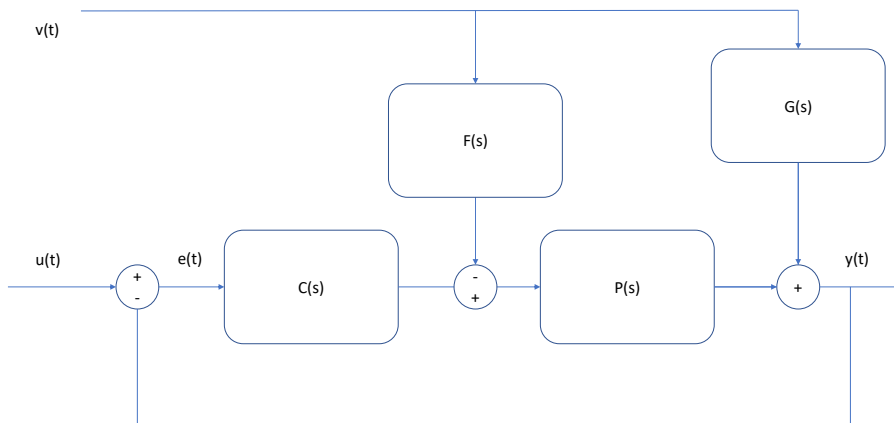
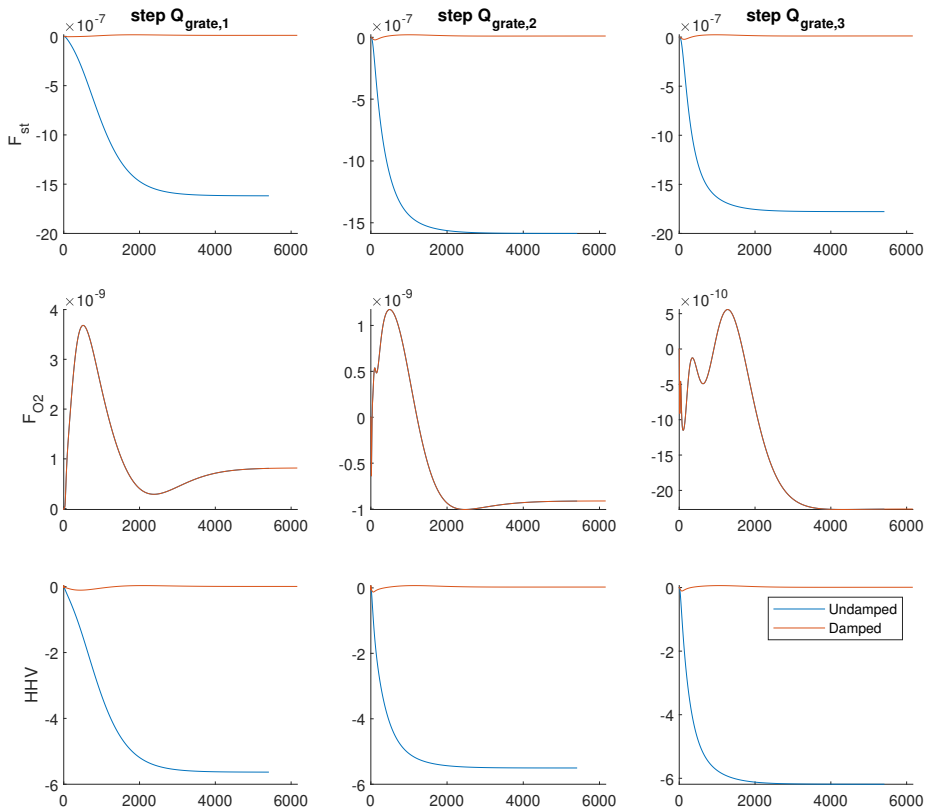


Figure B.1: Comparison of the different controllers with stochastic disturbances

feed forward controller, but in practice, it only became worse. This may be due to modelling errors or due to interactions with that LQE³, which did not measure any disturbances. Additionally, the inputs used were also too large for normal operation, as seen in figure B.2. An actual implementation would most likely require the usage of the other two inputs as well. Possibly a MIMO⁴ method for feed forward control would be required.

³Linear Quadratic Estimator

⁴Multiple Input Multiple Output

Figure B.2: Idealised output of a feed forward controller with $\alpha = 1$

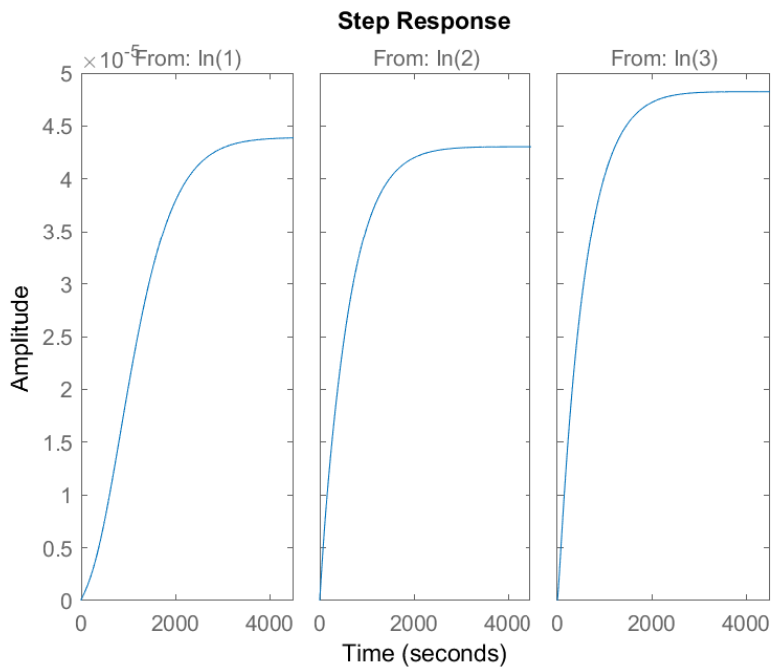


Figure B.3: Amplitude of $F_{aI} - F_{aII}$ for the idealised steam feed forward controllers, given the same input-step as in figure B.2

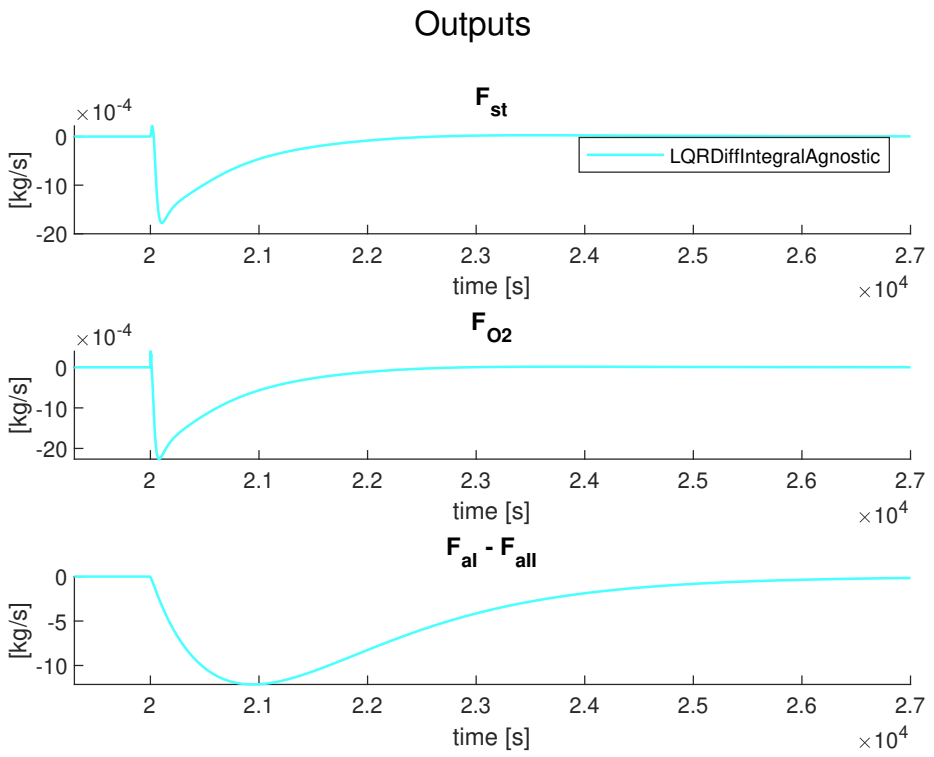


Figure B.4: Outputs: Practical feed forward. The results became worse.

Inputs

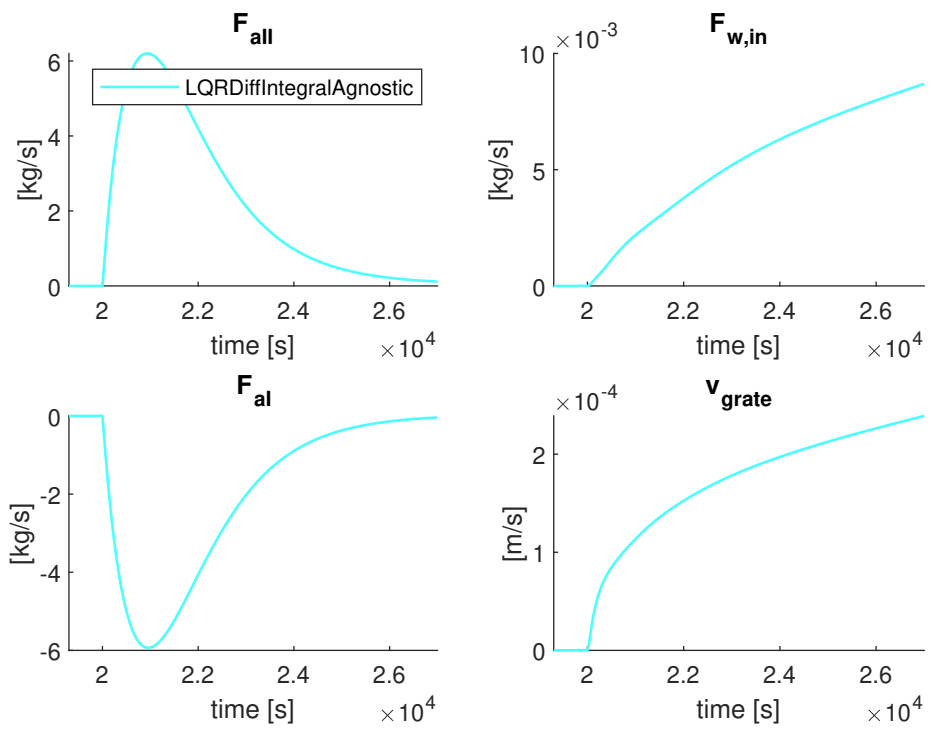


Figure B.5: Inputs: Practical feed forward. The inputs change a lot more than what is normal for ordinary operation

References

Boyd, S. (2008a). Lecture 3 infinite horizon linear quadratic regulator. Lecture slides published on the internet.

Boyd, S. (2008b). Lecture 4 continuous time linear quadratic regulator. Lecture slides published on the internet.

Dynamic modeling of municipal solid waste incineration (2020). *sciencedirect* "**In press**" ("In press").

Ioannou, P. A. and Sun, J. (2012). *Robust Adaptive Control*, Dover publications.

Jens G. Balchen, Trond Andrese, B. A. F. (2016). *Reguleringsteknikk*, Norges teknisk-naturvitenskapelige universitet (NTNU).

Juang, J.-N. and Suzuki, H. (1988). An eigensystem realization algorithm in frequency domain for modal parameter identification.

Lagarias, J., Reeds, J., Wright, M. and Wright, P. (1998). Convergence properties of the nelder-mead simplex method in low dimensions, *SIAM Journal on Optimization* **9**: 112–147.

Leskens, M. (2013). *Improved Economic Performance of Municipal Solid Waste Combustion Plants by Model Based Combustion Control*, PhD thesis.

McKinnon, K. (1996). Convergence of the nelder-mead simplex method to a non-stationary point.

- Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook. Version 20121115.
URL: <http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html>
- Skogstad, S. (2004). Simple analytic rules for model reduction and pidcontroller tuning, *Modeling, identification and control* **25**(2).
- Stoica, P. and Jansson, M. (2000). MIMO system identification: State-space and sub-space approximations versus transfer function and instrumental variables, *Signal Processing, IEEE Transactions on* **48**: 3087 – 3099.
- Øie Kolden, H. (2019). Disturbance rejection in mswc plants.

