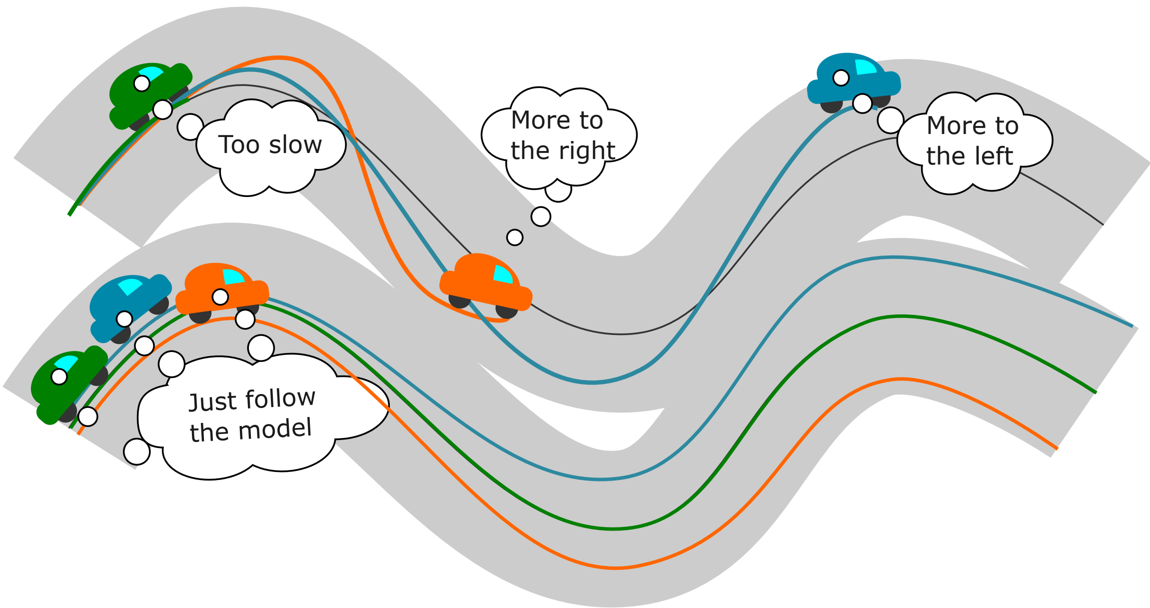
CHALMERS



A Comparison Between MPC and PID Controllers for Education and Steam Reformers

Ylva Lindberg

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden

EX049/2014

Abstract

This thesis examines the use of Model Predictive Control (MPC) compared to traditional PID control. The purpose of the work is twofold: to enhance the understanding of MPC in the industry and to investigate the possibility to improve the performance of a steam reformer process with MPC.

To give the industry a better knowledge of MPC an educational material for operators was developed. This complements an already existing lab tutorial for PID-controllers. It exemplifies the controllers with two tank systems — a single-tank system and a double-tank system. The educational material consists of a simulation of the tanks with Model Predictive Control and a manual for a lab tutorial. This part has also been used to study MPC in preparation for deriving a Model Predictive Controller for the steam reformer

A steam reformer is a process, reforming methane-rich natural gas into hydrogen gas and purge gas. The part of the steam reformer addressed here intended to handle is the uneven flow of purge gas into the combustion chamber. The models of the systems, both of the tanks and of the steam reformer, were theoretically developed, and a General Predictive Controller (GPC) was then implemented in MATLAB. The controller for the steam reformer system was also verified with a HYSYS simulation.

The most important lesson from this thesis is how important it is to make a correct model for a Model Predictive Controller. We can also see that a Model Predictive Controller is superior when it comes to handling changes in setpoint and keeping constraints. My conclusion is that it is useful to derive a Model Predictive Controller for the steam reformer, but not in the way presented here. Instead, I suggest a more holistic approach.

Sammanfattning

Det här examensarbetet jämför modellbaserad prediktiv reglering (Model Predictive Control, MPC) med traditionell PID-reglering. Syftet med uppsatsen är tudelat: att öka kunskapen om MPC i industrin och att undersöka möjligheten att förbättra en ångreformeringsprocess med MPC.

För att förbättra industrins kännedom om MPC har ett utbildningsmaterial tagits fram. Det kompletterar en redan existerande laborationshandledning för PID-reglering. Där exemplifieras regleringen med två tanksystem – ett med en enkeltank och ett med en dubbeltank. Utbildningsmaterialet består av en simulering av tanksystemen med modellbaserad prediktiv reglering samt en laborationshandledning med en introduktion till MPC. Den här delen av arbetet har också använts som förstudie inför utvecklandet av den modellbaserade prediktiva regulatorn för en ångreformeringsprocess.

En ångreformeringsprocess omvandlar metanrik naturgas till vätgas och restgaser. Den modellbaserde prediktiva regulatorn i det här arbetet syftar till att hantera det ojämna flödet av restgaser in i förbränningskammaren. Modellerna av systemen, både tanksystemen och ångreformeringsprocessen, togs fram ur teoretiska modeller. Sedan implementerades modellbaserade prediktiva regulatorer av typen Genral Predictive Controller (GPC) i Matlab. Regulatorn för ångreformeringsprocessen verifierades slutligen med hjälp av en Hysys-simulering.

Den viktigaste lärdommen från arbetet var hur viktig en god modell av systemet är för modellbaserade prediktiva regulatorer. Vi kan också se att modellbaserade prediktiva regulatorer är överlägsna för att hantera förändringar i börvärde och för att hålla sig inom givna gränser. Min slutsats är att en modellbaserad prediktiv regulator skulle vara mycket användbar för ångreformeringsprocessen, men inte på det sätt som den är genomförd här. Istället föreslår jag att man angriper problemet med ett större helhetsperspektiv.

Contents

Acknowledgments

I would like to express my greatest gratitude to the people how has helped and supported me in this project. First I would like to mention my supervisors Dr. Anders Hellgren at ÅF Group and Prof. Bengt Lennartson at Chalmers.

A special thank to Pär Sivertsson and Patrik Kerttu who gave me the opportunity to do this M.Sc. thesis at ÅF Group. I would also like to thank Antun Hroh, Martin Lundström, Johan Särnbrink, and Tobias Peterson at ÅF Group who with knowledge in their respective special areas have helped me to complete this project. I would also like to thank Torsten Wik for thesis LaTeX template. It has saved me a lot of time and sorrows.

I would like to thank my parents, Anna and Leif, for always believing in me, no matter what I have set my mind to do. Last, but not least, I want to thank my beloved husband, Johan, for his constant and invaluable support.

Göteborg, January 28, 2017

Ylva Lindberg

1 Introduction

When automatic control was first introduced there was a great resistance among the affected personnel. The new technique was called PID control, named after its three components: Proportional, Integral, and Derivative. Despite the first sceptics, the new control method was implemented, and for a long time PID control was the main method to control industrial systems. During the second half of the 20th century, Model Predictive Control (MPC) was introduced, and at many industries also this method was met with great resistance among the affected personnel. Nevertheless, MPC is flourishing and the Model Predictive Controllers are here to stay.

This thesis examines the use of MPC compared to traditional PID control. The first part of this thesis demonstrates how a Model Predictive Controller can be used in a simple system with one or two tanks compared to a traditional PID controller. This is intended to be included in the educational material for a shorter control theory course e.g. for operators in the industry or as a part of an advanced vocational training (KY). A suggestion for a Guide for Laboratory Practical is also included. This serves as an examination of the benefits and drawbacks of Model Predictive Controllers as a preparation for the second part.

The second part discusses MPC and the feedback of a hydrogen gas producing steam reformer. The specific unit in this project is located at Akzo Nobel Pulp and Performance Chemicals AB in Bohus, Sweden. ÅF Group has the responsibility to make the steam reformer more efficient in order to save both money and environment. This thesis examines if it is possible to do so using MPC. The intention is to make the results general enough to be applicable to other similar production facilities.

In a steam reformer, methane-rich natural gas is heated, which releases hydrogen gas. To extract the hydrogen gas from the other gas, here called *purge gas*, it is filtered through a catalyst. The residual purge gas is collected in a buffer tank and used as fuel to heat more natural gas. The flow of purge gas is not constant and it therefore has to be controlled and evened out with new natural gas, all to match the gas flow with a constant air flow. The air provides oxygen to the combustion. If the air flow is too low the combustion will not be complete. If it is too high the combustion costs extra energy without improved performance. The task in this work is to control the total gas flow to be constant using MPC and compare its performance with the PID controller currently in use.

In 1956 Coales and Noton published a first article in what today is seen as MPC [7]. After this several people have suggested similar solutions. The type that is of special interest in this project, the General Predictive Control (GPC), was developed by Clarke *et al.* in 1987 [6][13, ch. 1]. This new control technique soon became popular, especially in the chemical industry. At that point it was primarily with models based on step or impulse response [5, ch. 1]. Today MPC is considered to be the appropriate method when working with productions as large and expensive as the steam reformer at Akzo Nobel. GPC is a well studied type and it is considered stable [5, ch 2]. With the characteristics of the problem at hand it is thus natural to think of an MPC application of GPC type for this case.

The purpose with the educational material is to make MPC more well-known in industry and to give the operators confidence to work with this kind of controller. If this technique becomes accepted and thus more used, the industry would have access to more suitable controllers in the future.

ÅF Group has a general idea to make this steam reformer more beneficial and environmentally friendly, and also to sell energy conservation solutions in a broader sense. As a step towards this another M.Sc. thesis has been initiated to simulate the whole steam reformer at Akzo Nobel with Hysys and then improve the PID controller for the purge gas feedback presently implemented [11]. This thesis will instead study the benefits of implementing MPC for this part of the process. The Hysys model will be used for its verification.

Chapter 2 presents the general theory of MPC and its pertaining concepts are presented. Then Chapter 3 discusses the modelling of the tanks used in the educational material, the development and analysis of the Model Predictive Controllers. Then they are compared with PID controllers. Chapter 4 derives how the model of the steam-reformer process is derived and implemented in the Model Predictive Controllers. The results are compared to the PID controller presently implemented. Chapter 5 discusses the result of the Model Predictive Controllers of the tank systems and the steam-reformer process. Finally, Chapter 6 lists the conclusions.

Notation

Abbreviations

ASA	Active Set Algorithm
GPC	General Predictive Control
IPA	Interior Point Algorithm
MPC	Model Predictive Control
SISO	Single Input single Output

Nomenclature

$\mathbf{A}$	Output-signal matrix in state space model
A_{tank}	The bottom area of a tanks in the tank systems
$\mathbf{B}$	Control-signal matrix in state space model
c	Factor to convert flow in m^3/s to pressure difference in bar
g	The gravitational acceleration
H_{c}	Control horizon
H_{p}	Prediction horizon
h_i	Tank level in Tank i in the tank systems
K_d	The derivative factor in a PID controller
K_i	The integral factor in a PID controller
K_p	The proportional factor in a PID controller
K_v	Flow factor: a design parameter of a control valve
N	Pump speed in the tank systems
P_{c}	Pressure in combustion chamber in the steam-reformer system
P_{r}	Pressure in purge gas tank in the steam-reformer system
q_{c}	Flow into the combustion chamber in the steam-reformer system
q_{in}	Flow into a tank
q_{max}	Maximal flow through a valve
q_{out}	Flow out from a tank
T_{s}	The time step
$\mathbf{u}$	Control signal
v_i	Opening level of Valve i in the tank systems
v_{n}	Opening level of valve by the natural gas grid
v_{r}	Opening level of valve by the purge gas tank
$\mathbf{y}$	Output signal
λ	Penalizing parameter for the control signal
τ	Design parameter of a control valve of equal percentage type

2 Theory

The term Model Predictive Control (MPC) includes several control strategies with the same general idea. This chapter discusses the theory of control engineering in general and MPC in particular. First, these concepts will be presented briefly. This is followed by a review of the benefits and drawbacks of MPC as found in the literature. Thereafter MPC will be described more rigorously and with mathematical detail. The increment method and optimisation with an active-set algorithm are also explained, since these are important concepts for MPC.

MPC differs from traditional PID control in many ways, but they have one important thing in common: their purpose. All controllers exist to make a system follow a desired behaviour by adjusting some actuator, such as a valve, a pump, or a motor. How this adjustment is done is what makes the difference between different controllers. The choice of controller dramatically affects the system's behaviour.

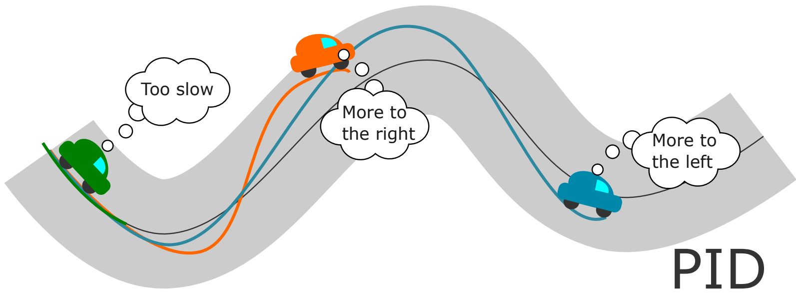
Whereas a PID controller uses the current and past states of the system to adjust the actuator, a Model Predictive Controller tries to predict the future using information from the current states and a model of how the system behaves. It is a bit like playing chess – you do not care about what has happened but only about how you should go on, by thinking about how different moves would affect the present game situation. In Figure 2.1 the difference is illustrated with a car which should follow a path under either MPC or PID control. Exactly how the adjustment of the car's path is done is decided by the control law. In a PID controller, this control law is

$$u(t) = K_p e + K_i \int e \mathrm{d}t + K_d \frac{\mathrm{d}e}{\mathrm{d}t} \tag{2.1}$$

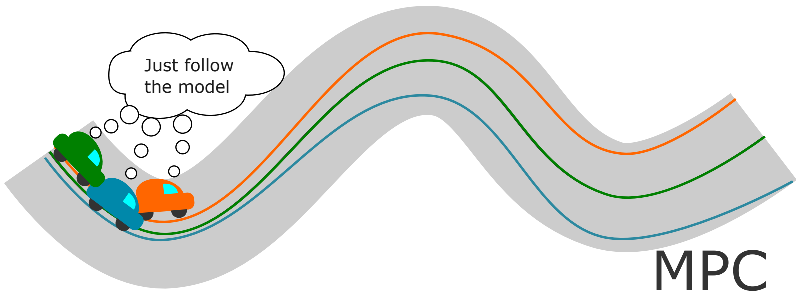
where u is the desired control signal and $e = e(t) = r(t) - y(t)$ is the measured control error between the reference r and the output y [12, ch. 8].

The control law of Model Predictive Controllers builds, as the name indicates, on a *prediction model* – a model that will predict the behaviour of the system. The method of developing this model makes out the main difference between different Model Predictive Controllers. It can be formulated from the impulse response, step response, transfer function or state space model to mention some possibilities. In this project the model of the controllers has been derived from theoretical physical laws. The model is very important as it sets the standards for the whole controller – *A bad model leads to a bad controller*, so one has to be sure to make as good a model as possible of the system and all known disturbances.[5]

An optimisation algorithm is used upon the model to find the best future track for the control signal. In this step it is also possible to include constraints. The algorithm will then take those into account when deriving the control signal. This procedure is then repeated for every sample: predict behaviour from the model, find the optimal control signal for the future. Only the control signal calculated for the



(a) The PID knows only what has happened and adjusts the controller based on the measured error.



(b) The MPC follows its model and the present measured value. If the model is good (green car, in the middle) it follows the setpoint well, but a small error in the model can lead the system astray (orange and blue car, on the sides).

Figure 2.1: The difference between PID control and MPC. [10]

very next time step is implemented, and the prediction always sees the same number of time steps into future. It is, in other word, a receding horizon. The number of time steps seen by the prediction is called *the prediction horizon*.

2.1 Benefits and Drawbacks of Predictive Controllers

The theory of MPC was first discussed during the middle of the 20th century and has developed since, particularly because of the development of computational hardware. Some general benefits and drawbacks of this very general theory are presented here.

MPC needs on-line calculations. When the technique was new, it was possible to use it only for simple Single Input Single Output (SISO) systems and problems with very low update rate. Larger and faster systems have to use a faster optimisation algorithm or more computational power to find the solution on time for the system to perform optimally. Now a days with faster computers this is no longer a limitation.[5, ch. 1]

The main feature of MPC type controllers type is their natural way of taking constraints into account. This could be physical limits such as maximum opening of valve or highest level allowed in a tank, as well as limits on the control signal. This way, the controllers can work close to the constraints. This is extra beneficial in all applications where the optimal work point lies close to a limit, e.g. a heating process which has to be warm enough but all extra heating is an unnecessary expense. In this project, the problem is a fuel flow which should perfectly match a steady air flow. If it is too low the extra air will be heated to no avail, but if it is too high the combustion will be incomplete. [13, ch. 1]

The implementation and tuning of Model Predictive Controllers is considered to be simple and can therefore be operated also by staff with a limited knowledge of control theory. On the other hand, a drawback is that the controller can be quite complex to derive compared to a PID controller. [5, ch. 1].

Another pitfall of MPC is finding a correct model and a good disturbance estimation. If this is not obtained, the estimation will be erroneous and the controller will not work properly. For systems with large unpredictable disturbances MPC is a bad choice. On the other hand, the propagation of measurement noise is lower with Model Predictive Controllers than with PID controllers. [5, ch. 1]

Furthermore systematic robustness and stability analysis known from traditional control theory are not possible. The control law is time-varying and can therefore not be written on standard closed-loop form [5, ch. 1]. For historical reasons, many commercial predictive controllers just assume the plant is stable [13, ch. 1]. The general conclusion is that for MPC to be a practical solution it has to be possible to make a good model of the system including disturbances. MPC is superior when it comes to set-point tracking and MPC is also beneficial for systems which have constraints, especially when the optimal working point lies close to these constraints.

2.2 Mathematical Description of MPC

As explained in the beginning of this chapter MPC relies on a model of the system. From this model a prediction of the system performance is made. Then optimisation is used to find the best control signal [5, ch. 1]. Here, I will elaborate more on the mathematical details of how this is done. An overview of the MPC framework is presented in Figure 2.2. A hat, ˆ, is used to indicate that a value is predicted rather than actual or measured.

The Model Predictive Controller consists of three different elements: the prediction model, the objective function and the control law [5, ch. 2], which will be explained more below. The objective function and derivation of control law is what previously was labelled as “optimisation”. The objective function is constructed from the prediction model as something which should be minimised. The control law is what is obtained when the algorithm has done the minimisation. An analogy of objective function in PID control is the control error which should be minimised.

The time frame for exactly how far into the future we want to see is called *prediction horizon,* H_p. Sometimes the control values are not calculated for the whole

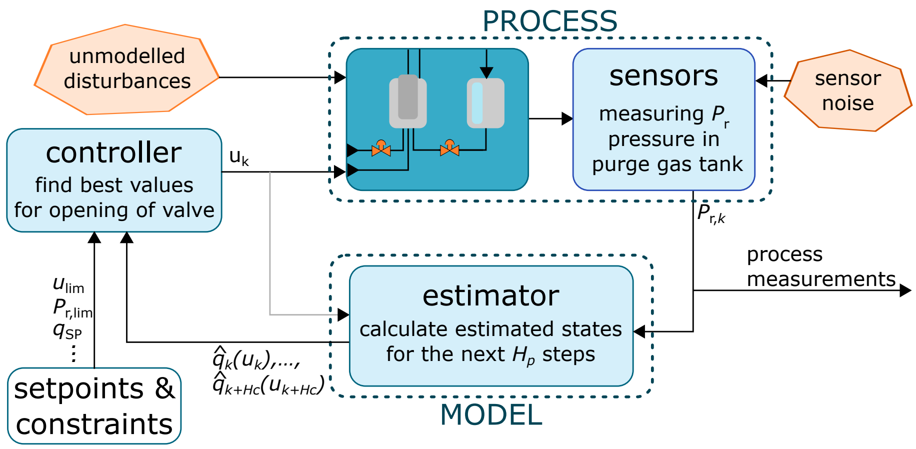


Figure 2.2: Scheme over the operation of a Model Predictive Controller [1] when applied to the steam reformer in Chapter 4. The model consists of the prediction model and the objective function. The controller follows the control law and calculates the control output.

prediction. Another time frame called *control horizon*, $H_c \leq H_p$ decides how many control values should be calculated. Using a shorter control horizon than prediction horizon will save computational power. Then, the system output is calculated as if all control signals after the control horizon are the same as the last calculated one. In both parts of this project General Predictive Control (GPC) based on a state space model was used. Here I described how these the different elements of the Model Predictive Controller can be found for a GPC. [13, ch. 1]

The Prediction Model has to fully capture the process dynamics to allow a good estimation of the future. To derive the prediction model a process model of the physical system is needed. This can be derived in many different ways and, every possible way has a given MPC formulation. [5, ch. 2]

Here, the derivation starts with the linear representation of a physical system, a process model, on state space form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \end{aligned} \tag{2.2}$$

where $\mathbf{x}$ is the state variable, $\mathbf{y}$ is the output and $\mathbf{u}$ is the control signal [12, p. 89]. Without loss of generality, it can be assumed that $\mathbf{D} = 0$ [13, ch. 2.1].

Discretisation gives

$$\begin{aligned} \mathbf{x}(k+1|k) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k|k) \\ &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\left(\mathbf{u}(k-1) + \Delta\mathbf{u}(k|k)\right) \end{aligned}$$

where the notation $|k$ indicates that this is an estimation done in the k-th time step.

Continuing this analogy we get:

$$\begin{aligned}
\mathbf{x}(k+2|k) &= \mathbf{A}\mathbf{x}(k+1|k) + \mathbf{B}(\mathbf{u}(k|k) + \Delta\mathbf{u}(k+1|k)) \\
&= \mathbf{A}^2\mathbf{x}(k) + (\mathbf{A}+\mathbb{I})\mathbf{B}\Delta\mathbf{u}(k|k) + \mathbf{B}\Delta\mathbf{u}(k+1|k) + (\mathbf{A}+\mathbb{I})\mathbf{B}\mathbf{u}(k-1) \\
&\vdots \\
\mathbf{x}(k+H_c|k) &= \mathbf{A}^{H_c}\mathbf{x}(k) + (\mathbf{A}^{H_(c-1)} + \ldots + \mathbf{A} + \mathbb{I})\mathbf{B}\Delta\mathbf{u}(k|k) + \ldots + \\
&\quad \mathbf{B}\Delta\mathbf{u}(k+H_c-1|k) + (\mathbf{A}^{H_c-1} + \ldots + \mathbf{A} + \mathbb{I})\mathbf{B}\mathbf{u}(k-1)
\end{aligned}$$

[13, p. 54-55].

For clarity this is best written as a matrix equation. Thus the prediction model is:

$$\begin{bmatrix} \mathbf{x}(k+1|k) \\ \vdots \\ \mathbf{x}(k+H_c|k) \\ \vdots \\ \mathbf{x}(k+H_p|k) \end{bmatrix} = \underbrace{\mathbf{E}\mathbf{x}(k) + \mathbf{F}\mathbf{u}(k-1)}_{\text{past}} + \underbrace{\mathbf{G}\begin{bmatrix} \Delta\mathbf{u}(k|k) \\ \vdots \\ \Delta\mathbf{u}(k+H_c-1|k) \end{bmatrix}}_{\text{future}}$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{A}^1 \\ \vdots \\ \mathbf{A}^{H_c} \\ \vdots \\ \mathbf{A}^{H_p} \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_c-1}\mathbf{A}^i\mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_p-1}\mathbf{A}^i\mathbf{B} \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{B} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_c-1}\mathbf{A}^i\mathbf{B} & \ldots & \mathbf{A}^0\mathbf{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1}\mathbf{A}^i\mathbf{B} & \ldots & \sum_{i=0}^{H_p-H_c}\mathbf{A}^i\mathbf{B} \end{bmatrix}$$

[13, p. 55].

The matrices sizes are dependent of the horizons such that **E** and **F** are vectors with length H_p and **G** is a matrix of size $H_\mathrm{p} \times H_\mathrm{c}$. The practice to control $\Delta\mathbf{u}$ and not **u** itself is called the Increment Method and eliminates a steady-state control error. This will be further explained below. The same way $\mathbf{y} = \mathbf{C}\mathbf{x}$ can be estimated with the **x** predicted above.

The Objective Function is a concept in mathematical optimisation. When a problem is been optimised, it is rephrased such that the best solution is found through minimising the function. This rephrased version of the problem is called the cost function. The optimisation problem may depend on several variables and can also have constraints [14, ch. 1]. To find the optimal control signal, the state space model is rephrased as a minimisation problem – and the cost function represents in this context *the objective function*.

The goal with the controller is to eliminate the error e between the output y and the set point r, hence to eliminate $e = r - y$, by making an appropriate choice of the input values, **u**. Preferably the elimination should occur when the system is in steady-state where the change in control signal, $\Delta\mathbf{u}$, also is eliminated. Therefore the objective function becomes

$$\min_{\Delta\mathbf{u}} J = \Delta\mathbf{u}'\mathbf{H}\Delta\mathbf{u} + \Delta\mathbf{u}'\mathbf{f} \tag{2.3}$$

[5, ch. 4.2] that means “Find the value of $\Delta\mathbf{u}$ that gives the smallest value if J.”, where

$$\begin{aligned}\mathbf{H} &= 2(\mathbb{I}_{H_c}\lambda^2 + \mathbf{G}'\mathbf{G})\\ \mathbf{f} &= 2\mathbf{G}'(\mathbf{E}\mathbf{y}(k-1) + \mathbf{F}\mathbf{u}(k-1) - r)\end{aligned}$$

and λ is a parameter to penalise the control signal. $\mathbf{H}$ has the purpose to minimise control signal and $\mathbf{f}$ to minimise the error. A objective function on this quadratic form is typical for GPC. Sometimes, the set point is a vector instead of a point and is then referred to as *the reference trajectory* [13, ch. 1]. To use a reference trajectory gives the possibility to announce a change in setpoint in advance, and the system can make the transition as smooth as possible. The variables, $\Delta\mathbf{u}$, which should be chosen are the change in control signals for the future.

Increment method Both in the model and in the objective function $\Delta\mathbf{u}$, rather than $\mathbf{u}$, has been used as variable. This correspond to introducing integral action in the PID case and has the purpose of eliminating the steady-state error. If not, Equation 2.3 may find a local minimum instead, since J depends both of the error (present value minus set point), and on the control signal.

To be able to introduce integral effect and eliminate the steady-state error, the control signal is expressed as the last control signal plus the difference

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta\mathbf{u}(k).$$

Then J can be minimised with respect to the difference in control signal and there will be no steady state error as long as the model is correct.

The Control Law is the explicit expression to calculate the next value of the control signal. The minimisation of J, Equation 2.3, is needed to find the control law, and thus also to set the optimal values of $\mathbf{u}(k+i|k)$. To do this analytically would be theoretically possible using the GPC approach, as long as there are no constraints [13, ch. 4][5, ch. 2]. However the problem at hand, has constraints and it is therefore solved with an iterative algorithm. Usually an Active Set Algorithm or an Interior Point Algorithm are used [4]. In this project, the minimisation of the objective function, Equation2.3, will be done with an Active Set Algorithm.

2.3 Quadratic programming: Active Set Algorithm

Active Set Algorithm (ASA) is an iterative optimisation method to find a solution given some equal and unequal constraints. This is one of the most common algorithms for MPC applications [9]. To solve the problem an “active set” is formed by setting some of the inequality constraints to be equal, those are the “active constraints”. Thus the optimisation is done for some constraints at a time. To optimise, the algorithm starts at a point in the feasible area and from there takes a step in an optimal direction. Step size depends on the allowance from all side conditions. Such steps are calculated and taken until a solution is found for that active set. Then the whole process is repeated iteratively until an optimal solution is found with respect to all constraints. [14, ch. 16.5]

A pseudo code of the Algorithm [14, ch. 16.5]:

Chose a feasible starting point, x_0;
Set $\mathcal{W}_0$ to be a subset of the active constraints at the starting point;
for $k = 0, 1, 2 \ldots$ **do**
  Solve the objective function (2.3) to find Δu.
  if $\Delta u_k = 0$ *or close enough* **then**
    Compute Lagrange multipliers, $\hat{\lambda}_i$, for the active set.
    if $\hat{\lambda}_i \geq 0 \forall i \in \mathcal{W}_k$ **then**
      The found point is optimal. **return**
    else
      Remove constraints with negative Lagrange multipliers for $\mathcal{W}_k$ to form $\mathcal{W}_{k+1}$ and $x_{k+1} \leftarrow x_k$
    end
  else
    Compute an optimal step α_k and search direction p, with respect to the side-conditions.
    $\Delta u_{k+1} \leftarrow \Delta u_k + p\alpha$
    if *there are no blocking constriants* **then**
      Obtain $\mathcal{W}_{k+1}$ by adding one constraint to $\mathcal{W}_k$
    else
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k$
    end
  end
end

You can read more about Lagrange multipliers in [14]. Very shortly this can be concluded in four steps:
- Solve the problem defined by $\mathcal{W}$, the active set.
- Compute the Lagrange multipliers $\hat{\lambda}_i$.
- Remove constraints with negative Lagrange multipliers from the active set.
- Search for infeasible constraints and adjust the active set.
Repeat until the solution is optimal enough.

In the MPC case when a function J (Equation 2.3) is minimised, the new output lies as close to the reference value as possible given the side conditions. When the plant has reached the set point, the difference between the output and the reference is zero. Also the change in control signal, $\mathbf{\Delta u}$, is zero. This means that there is no error left, and the control signal has stabilised in a steady state. In this project a MATLAB built-in version of this algorithm was used.

2.4 Summary

We have here seen that Model Predictive Control is built on three parts: the predictive model that is a description of the system used predict the future; the objective function that describes what we want to obtain reformulated to a mathematical optimization problem; and finally the control law that described how the problem should be optimized and the optimal control value found, here obtained with an active set algorithm.

We have also looked into the differences between Model Predictive Control and traditional PID control, found in the literature. In the next chapter it can be see how this theory can be used do develop a Model Predictive Controller for two different tank systems, and what performance it gives the system compared to a PID controller.

3 The Single and Double Tank Systems

ÅF Group has developed an educational material for a brief control engineering course, primarily intended for operators and students in advanced vocational training (KY). This material contains simulations of simple one- and two-tank problems, controlled with P, PI, and PID controllers, to show the properties of the different types. To enhance the knowledge about MPC in industry in general and among the operators specifically, a part of this project is to extends the material with an MPC simulation.

The work consists of constructing a theoretical model of the systems with one and two tanks and implementing Model Predictive Controllers to control the tank levels. The already existing parts of the educational materials was implemented in WideQuick, a JavaScript based simulation tool developed by Kentima. The idea is that also Model Predictive Controllers will be implemented here eventually, but with the drawbacks of JavaScript, primarily with respect to matrix handling, this was decided to be left outside the scope of this project.

Instead both the Model Predictive Controller and a PID controller for these systems have been implemented in MATLAB and compared. This system might differ slightly from the system implemented in WideQuick, due to uncertainties in the specification. Nevertheless, their concepts are identical and the systems within the MATLAB models correspond and they can thus be compared. The comparison and evaluations are similar to those that the student can do during the laboratory practical. The guide for this can be found in Appendix A, but this is only available in Swedish.

First in this chapter the method used for constructing the models of both systems and all controllers is described. Then their behaviour is analysed. The analysis contains disturbance rejection with respect to measurement noise and input disturbance. Then, I review how well the controllers work close to a limit, i.e. an almost full tank. As the tank systems do not exist as physical systems in this context but only as a model these Model Predictive Controller is a built on a model of a model. This gives an insight into how well a controller of this type can perform when it is built on a faultless model.

3.1 Modelling

First the model of the single-tank system was made. The double-tank system was divided into two parts of which each of them separately has a lot in common with the single tank. Thus the model of the double-tank system was derived from the single-tank system

All the tanks, valves, and pumps in both systems are identical. The tanks are 5 m tall and have bottom areas of $1\,\mathrm{m}^2$. The valves have maximum opening area of $0.0275\,\mathrm{m}^2$ and they take 15 s from being fully closed to fully opened and vice versa. The pumps can give a maximum flow of $98.1\,\ell/\mathrm{s}$ and they use 30 s to go from full speed to zero speed and vice versa. Both the pumps and the valves are assumed to be linear, which means that the percentage of the opening area is equal to the percentage of maximal flow with that opening. This theoretical construction is suitable as the purpose here is education about control strategies, not about control valves. If there is such an interest it should not be too difficult to implement other kinds of valves in this application.

3.1.1 Single tank

The single-tank system consists of one tank, one pump and two valves, see Figure 3.1. Normally only one of the valves, Valve 1 in the figure, is used. The flow through Valve 2 is considered to be a disturbance. This valve can be used in case of emergency if the tank needs to be emptied fast. The pump (orange in the figure) is regulated by the controller to keep the tank level at the setpoint. The openings of the valves are set manually.

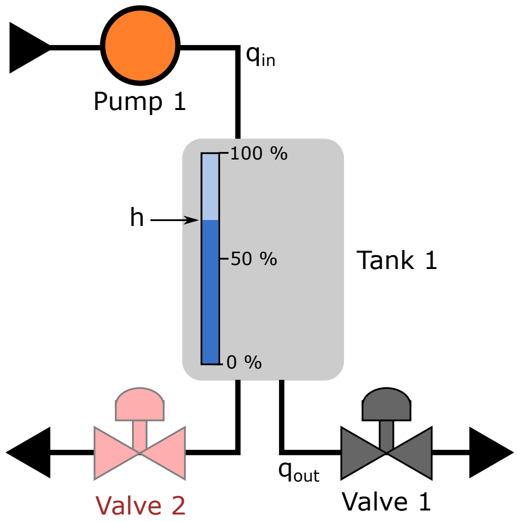


Figure 3.1: The single-tank system. The tank level is controlled by the orange pump, Pump 1. The opening of the valves are set manually.

When Valve 2 is assumed to be closed, the system is described as

$$A_{\text{tank}}\dot{h} = q_{\text{in}} - q_{\text{out}}.$$

where A_{tank} is the bottom area of the tank, h is the level of the tank, q_{in} is the inflow from Pump 1, and $q_{\text{out}} = v_1\sqrt{2gh}$ is the outflow from Valve 1. v_1 is the opening level of Valve 1 and g is the gravitational acceleration. Since the pump is linear $q_{\text{in}} \sim N$, where N is the pump speed. Since the valves are identical, opening of Valve 2 would give $q_{\text{out}} = (v_1 + v_2)\sqrt{2gh}$.

When linearised and discretised the system is described as

$$h_n = \left(1 + \frac{v_1 T_s}{A_{\text{tank}}}\sqrt{\frac{g}{2h_{n-1}}}\right) h_{n-1} + \frac{T_s}{A_{\text{tank}}} q^{\text{in}}_{n-1}. \tag{3.1}$$

where T_s is the time step. The linearisation is done with respect to the tank level at the last time step, h_{n-1}, since the controller should be able to work over a large range of different levels. Ergo, the linearised model is recalculated at each time step, but in the MPC context this is a minor part of the computational work.

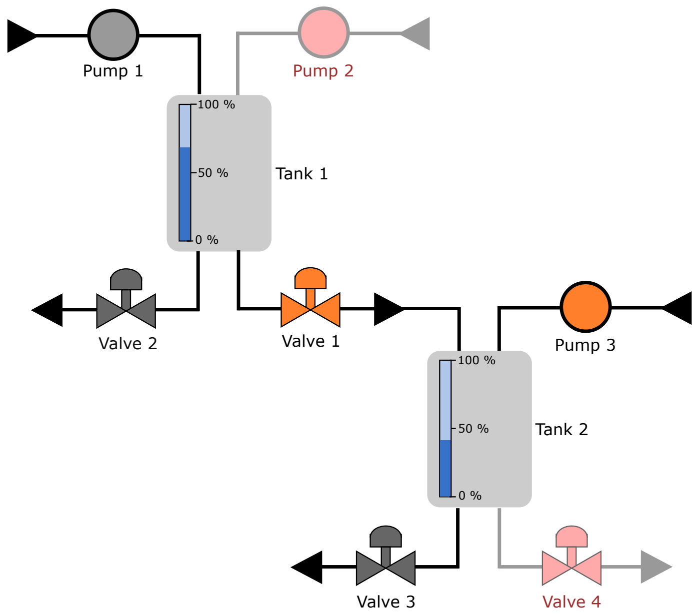


Figure 3.2: The double tank system. Valve 1 and Pump 3 (orange) are controlled by the controller, Pump 2 and Valve 4 (light red) are only for emergency situations. The rest (dark grey) are set manually.

3.1.2 Double tank

The double-tank system consists of two tanks, three pumps and four valves, see Figure 3.2. Valve 1 and Pump 3 (orange in the figure) are controlled by one controller each. Pump 1, Valve 2, and Valve 3 (dark grey in the figure) are set manually. Pump 2 and Valve 4 (light red in the figure) are considered as a disturbances, and can be used in case of emergency – Pump 2 as a back-up and Valve 4 if the tank needs to be emptied fast.

The system can be separated into two parts: one upper part with Tank 1, Pump 1 and 2, and Valve 1 and 2; and one lower part with Tank 2, the inflow from Valve 1, Pump 3, and Valve 3 and 4. Each of these parts are similar to the single tank system, with some important differences.

The model of the upper part of the system differs from that of the single tank as the pump (Pump 1) is set manually, while the output valve (Valve 1) is controlled by the controller. This gives, with linearisation around the last value of the tank level h_1:

$$h_{1,n} = \left(1 + \frac{q_{1,\text{in}} T_s}{A_{\text{tank}} h_{n-1}}\right) h_{1,n-1} - \frac{h_{1,n-1} T_s}{A_{\text{tank}}} \sqrt{\frac{g}{2h_{1,n-1}}} v_{1,n}. \tag{3.2}$$

The lower part of the system differs from that of the single-tank system with an extra inflow, which depend on opening of Valve 1 and the level of Tank 1. This

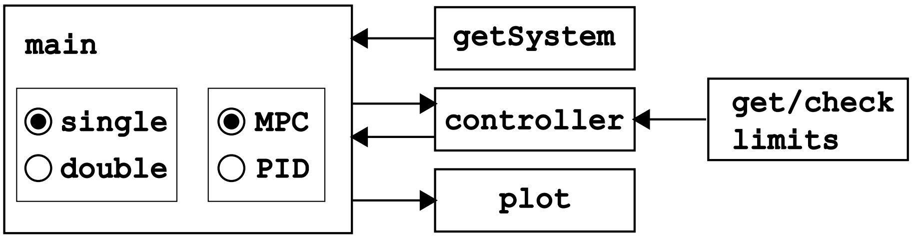


Figure 3.3: Overview of the MATLAB software for the tank systems and their controllers.

gives, with linearisation around the last value of the tank level h_2:

$$h_{2,n} = \left(1 - \frac{v_3 T_s}{A_{\text{tank}}}\sqrt{\frac{g}{2h_{2,n-1}}} + \frac{v_{1,n}}{h_{2,n-1}}\sqrt{2gh_{1,n}}\right) h_{2,n-1} + \frac{T_{\text{s}}}{A_{\text{tank}}} q_{3,\text{in}}. \tag{3.3}$$

3.2 Control

The Model Predictive Controllers for the tank systems were developed following the theory of Chapter 2. This project is focused on MPC and therefore the most of the work has been on developing proper a Model Predictive Controller for the problems at hand. The development of the PID controller will only be mentioned briefly.

The general function of the Model Predictive Controller program is outlined in Figure 3.3. There are four different kinds of main functions which uses all the combinations of the two different tank systems and the two different control types. In the main functions the controllers are specified with design parameters, such as horizons and time step. The controllers iteratively acquire state space matrices, $\mathbf{A}$ and $\mathbf{B}$, of the selected system, single or double tank. Those are sent to the selected controller, PID or Model Predictive. The Model Predictive Controller gets the constraints from an external function to get the control values. The PID controller checks the boundary values with an external function and, if necessary, adjusts the control signal after it has been calculated so that the control signal fulfils the physical constraints.

3.2.1 PID control

For both tank systems, classical PID controllers were implemented for comparison to the MPC. In the pre-existing educational material, both Ziegler-Nichols method and the lambda method for tuning are presented. Since these processes have such simple dynamics the K_i (see Equation 2.1) will become too high with Ziegler-Nichols method [12, ch. 8]. Therefore, the lambda method was used instead, with $\lambda = 2$ to get a stable but not too slow controller. After some manual fine-tuning the control parameters were set as in Table 3.1.

The constraints on the pump speed are fulfilled by a simple if-statement. If the

	Pump	**Valve**
K_p	0.063	0.02
K_i	0.0015	0.00054
K_d	0.001	0.001

Table 3.1: Tuning parameters of the PID controllers. The controllers are named after their actuators. The single tank and the lower tank in the double tank have controllers of “Pump” type, while the upper tank in the double tank has a “Valve” controller.

control signal is higher or lower than physically possible it is just set to the highest or lowest possible value. This may cause a windup, and if this PID was a real implementation it would be wise to make adjustments for this, e.g. with an anti-windup function [12, ch. 12].

3.2.2 MPC of the Single Tank System

The prediction model is constructed from Equation (3.1), since this is both the system and the perfect model of the system, where

$$\mathbf{A} = 1 + \frac{v_1 T_\mathrm{s}}{A_\mathrm{tank}}\sqrt{\frac{g}{2h_n}} \quad \mathbf{B} = \frac{T_\mathrm{s}}{A_\mathrm{tank}}$$

From this, the objective function is formed and the control law obtained. The size of the objective function is decided by the prediction and control horizons. The exact appearance of the objective function changes as it is reconstructed on-line by the controller in every time step.

3.2.3 MPC of the Double Tank System

Since the original PID controller of the double tank was divided in two parts, the MPC will also be handled that way. The upper part of the double-tank system is derived from Equation 3.2, and is hence described by

$$\mathbf{A}_\mathrm{upper} = 1 + \frac{q_\mathrm{in} T_\mathrm{s}}{A_\mathrm{tank} h_{1,n-1}} \quad \mathbf{B}_\mathrm{upper} = -\frac{h_{1,n-1} T_\mathrm{s}}{A_\mathrm{tank}}\sqrt{\frac{g}{2h_{1,n-1}}}.$$

The lower part of the double-tank system is constructed from Equation 3.3, and is hence described by

$$\mathbf{A}_\mathrm{lower} = 1 - \frac{v_3 T_\mathrm{s}}{A_\mathrm{tank}}\sqrt{\frac{g}{2h_{2,n-1}}} + \frac{v_{1,n}}{h_{2,n-1}}\sqrt{2gh_{1,n}} \quad \mathbf{B}_\mathrm{lower} = \frac{T_\mathrm{s}}{A_\mathrm{tank}}.$$

The prediction model is derived from these state space models. The objective function is then formed and the control law is obtained, as in the single-tank case.

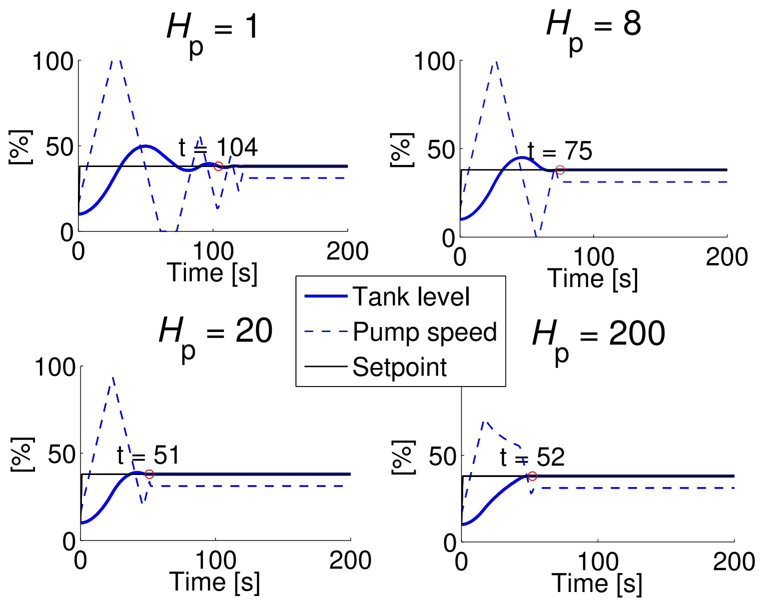


Figure 3.4: Step response with different prediction horizons, H_p. The step in tank level set point is from 10% to 50% of full tank. $H_\mathrm{c} = 1$ for the first, $H_\mathrm{c} = 8$ for the others.

3.2.4 Tuning

Tuning of a Model Predictive Controller is to choose appropriate values for H_c, H_p, and λ. A common method used for industry purposes is to look at a step of the unregulated system and see how long time it takes before it reaches steady state again, that is to measure the time constant, T. The horizon is then chosen to be same size as T. In this way, the controller predicts the behaviour of all the way to the steady state. [5, ch 5] [13, ch 7]

The single-tank system has $T \approx 200\,\mathrm{s}$, when a step in the pump speed occurs. The exact value depends on the opening degree of the valve at the bottom of the tank. This is also valid for the lower part of the double tank system. The upper part of the double-tank system has also $T \approx 200\,\mathrm{s}$. The exact value depends on the speed of the pump at the top of the tank.

In Figure 3.4, the step responses of the single tank system with MPC with different H_p values have been plotted. The step is from 10% to 50% in tank level set point. $H_\mathrm{c} = 1$ where $H_\mathrm{p} = 1$, and $H_\mathrm{c} = 8$ in the other cases.

With a low H_p the controller behaves as a poorly-tuned PID controller. When $H_\mathrm{p} = 200$, the controller is not as aggressive as in the other cases. This behaviour is eliminated if a more appropriate control horizon is chosen such, as $H_\mathrm{c} = 20$. If it is desired to have a lighter control it is better to do this with a higher penalising λ. Thus, $H_\mathrm{c} = 20$ is used in the analysis of the controller.

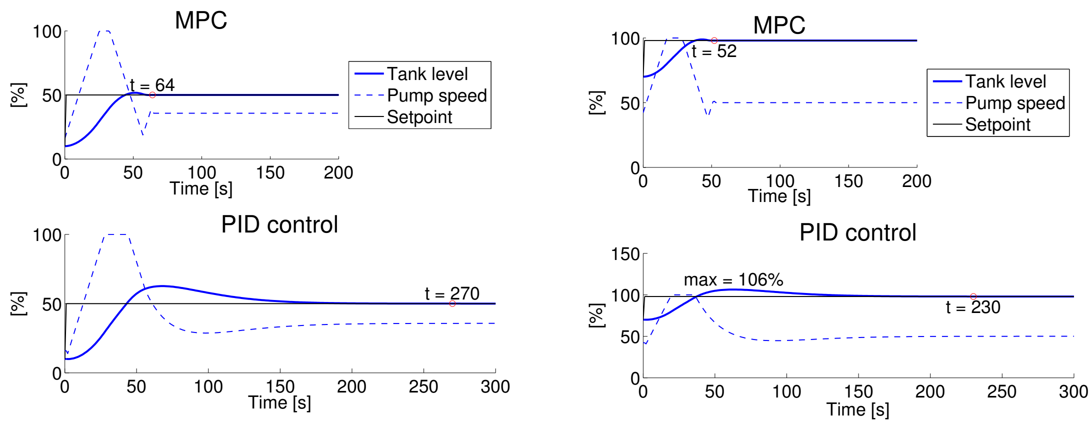
The penalising factor λ is chosen depending on how aggressive the controller should be. Most times it is chosen as a constant, and that is the case also here, but sometimes it can be exponentially increasing [5, ch 2]. The sampling interval is chosen with respect to computational power and aliasing effects. Here it is set to 1 s. It is short enough and the computational power is not an issue for making calculations at this rate.

3.3 Results

The performance of the controllers has been analysed with respect to a step, with different prediction horizons and also with different types of disturbances. If nothing else is stated, the design parameters are $\lambda = 0.1$, $H_\mathrm{p} = 200$ and $H_\mathrm{c} = 20$, and the time step is 1 s for all controllers.

3.3.1 Step response

The single-tank system has been tested with two different steps in setpoint from steady state, see Figure 3.5. In this analysis the openings of Valve 1 is set to 36% and Pump 1 is working at 30% of maximum speed.



(a) Response of step from 10% to 50% in tank level setpoint.

(b) Response of step from 70% to 98% in tank level setpoint.

Figure 3.5: Step response of single tank system with PID and Model Predictive Controller. The red circle indicates when the error gets below 10^{-4} m.

With a step in tank level setpoint at $t = 0$ from steady 10 % to 50 %, it takes more than four times longer time to reach an error below 10^{-4} m with the PID controller than with the Model Predictive Controller (see Figure 3.5a). In Figure 3.5b the controllers are keeping the tank at 70 % and at $t = 0$ the setpoint is changed with at step to 98 %. Not only is the Model Predictive Controller finished in a much shorter time, but it also keeps the tank level under the maximum. In an actual case the PID controller would cause the tank to flood, while the Model Predictive Controller keeps in within the given constraints. This is a good example of how the predictive controller inherently handles constraints. Not only constraints on the control signal but also constraints on the states and outputs.

The double tank system has been tested with step from 10 % to 40 % in Tank 1 and from 20 % to 50 % in Tank 2 (see Figure 3.6). During the examination of the double tank system, Pump 1 is working at 30 % of maximum speed while Valve 3 at the bottom of the second tank is completely open. When the $t = 0$, the controllers had reached a steady state.

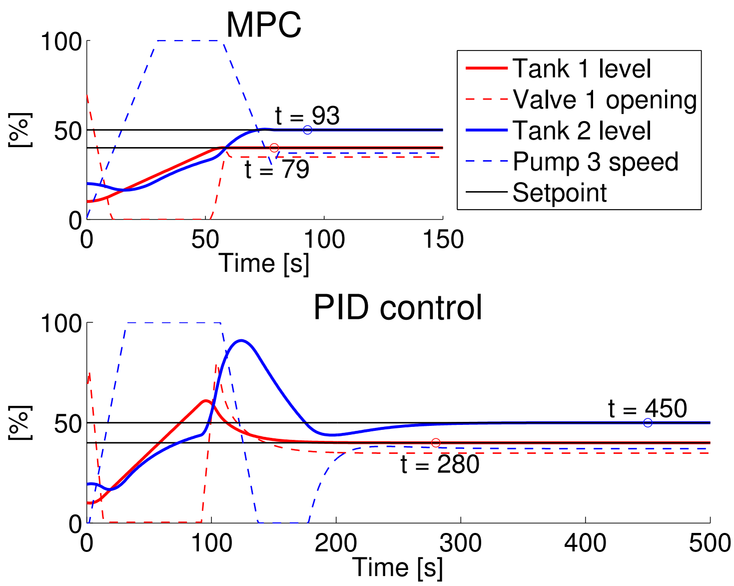


Figure 3.6: Step response of double tank system with PID and Model Predictive Controller. Step from 10 % to 40 %in Tank 1 and from 20 % to 50 % in Tank 2. The small circles indicate when the error is gets below 10^{-4} m.

In this double tank system it is even more clear that the Model Predictive Controller sees ahead and therefore is neater then the PID controller – the time to reach steady state at the new setpoint is several times shorter.

3.3.2 Disturbance rejection

From the literature it is known that Model Predictive Controllers do not handle un-modelled disturbance very well. The rejection of measurement noise is expected to be somewhat better, but constant disturbances will give a residual error. [5, ch. 1]

In this section the different controllers' capability of handling unmodelled disturbances, both constant and varying, has been analysed. The figures in this section show the single-tank system, but the same principles apply also to the double-tank system, only resulting in messier plots.

When adding a constant disturbance, for instance if the pump gives more flow a or valve is opened more than expected, the PID controller corrects this after some time, while the Model Predictive Controller leaves a small error. Figure 3.7 illustrates how the respective controllers react to an increase in the flow by 10 percentage points. In that case, the error left by the Model Predictive controller is 0.017 m above the setpoint.

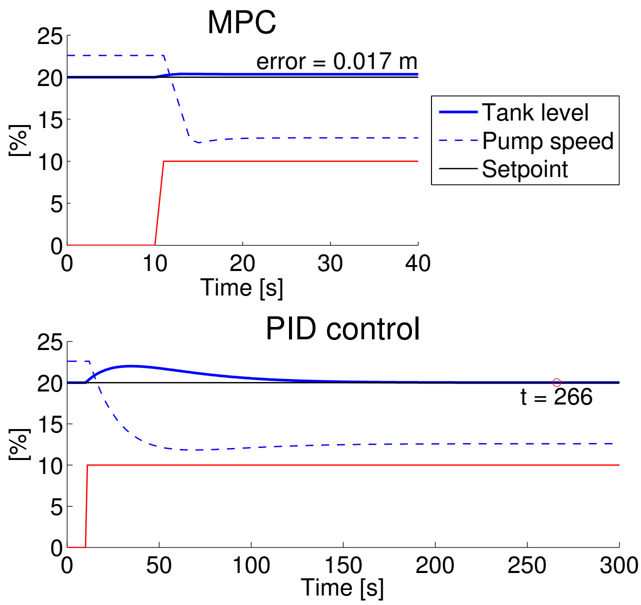


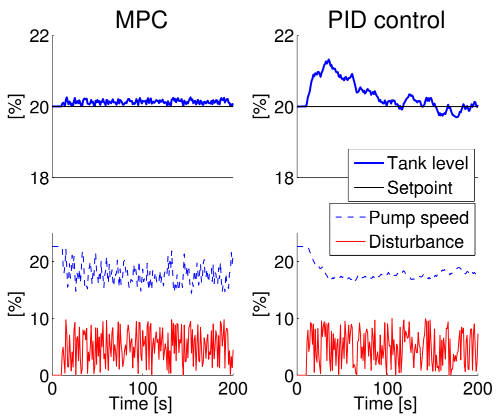
Figure 3.7: Single tank system with PID and Model Predictive Controller with constant disturbance from $t = 10$ s. The pump gives a 10 percentage points higher flow than the controller expects.

The system response when adding a white noise disturbance is a disturbed tank level. The noise rejection is better with the Model Predictive Controller than with the PID controller. This can be seen in Figure 3.8a, where the pump gives a $0-10$ percentage points higher flow than expected. The variance of the tank level is 1.2×10^{-5} with the Model Predictive Controller and 3.1×10^{-4} with the PID controller.

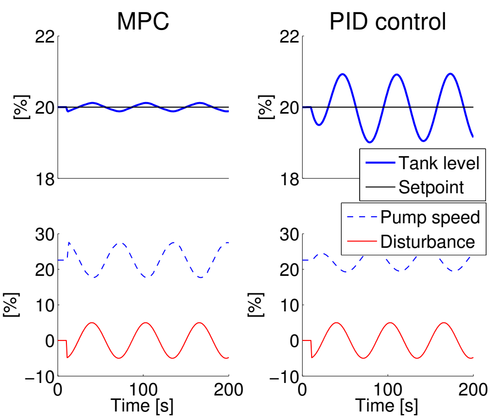
To better understand what to expect from the steam reformer in part two of this project a sine-shaped disturbance has also been applied, see Figure 3.8b. The disturbance varies between -5% and $+5\%$ of maximum pump flow. The Model Predictive Controller rejects the disturbance much better, and gives a tank level variance of 1.6×10^{-5} m while the PID controller gives a tank level variance of 1.0×10^{-3} m. It should also be noted that the mean of the system under PID control lies much closer to the setpoint then the system while controlled by the Model Predictive Controller.

3.4 Summary

In this chapter we have seen how a Model Predictive Controller of two different tank systems has be derived from a theoretical model. We have also seen how it can be tuned with respect to penalising of the control signal and horizons. Finally the performance of the Model Predictive Controller was compared to a PID controller, and it was seen to confirm what we know from the literature.



(a) The pump gives $0-10$ percentage points higher flow than the controller expects. The disturbance is a white noise.



(b) The pump gives -5 to $+5$ percentage points higher flow than the controller expects. The disturbance is shaped like a sine wave.

Figure 3.8: Single tank system under non-constant disturbance with PID contoller and Model Predictive Controller.

4 The Steam Reformer

Steam reforming is a very common method for producing hydrogen gas by heating methane-rich natural gas. When the gas has reached a temperature of 600 °C it is filtered with catalysts to extract the hydrogen gas. The remaining gas, or the *purge gas*, is collected in a buffer tank, and will be used in the heating of new natural gas. The purge gas consists of hydrogen, methane, carbon monoxide, and some water, and has an energy content of about 20% to that of the natural gas [11]. To heat new gas, it is not enough to use only the purge gas as fuel. Thus extra natural gas is added. A sketch of the process where the important parts in this context are shown in Figure 4.1.

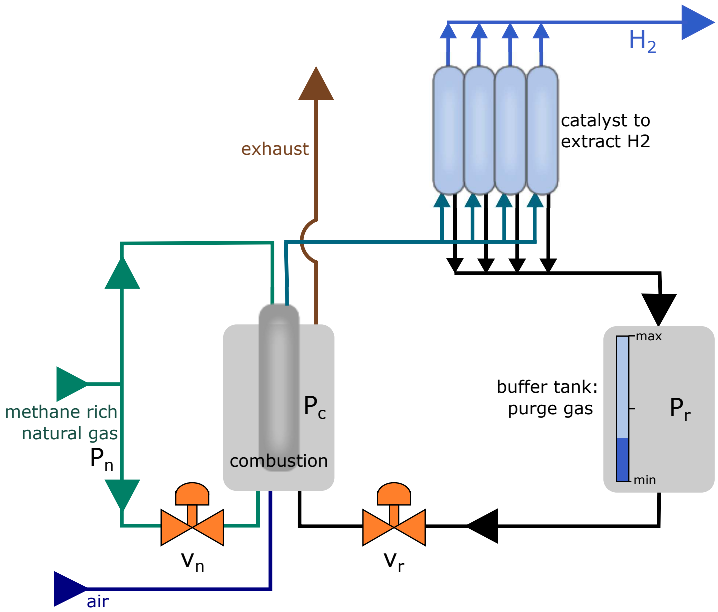


Figure 4.1: Simplified scheme of steam reformer process.

The transportation and filtering are executed by pressure differences. Therefore, the pressure in the purge gas tank can not be too high or the gas would not leave the catalysts, nor too low or the purge gas would not enter the combustion for heating new gas. The combustion is taking place in atmospheric pressure and the pressure in the tank may vary between 1.32 and 1.82 bar(a)[1].

The system has are four catalysts which let the hydrogen pass through and collects all the purge gas. After 15 minutes a catalyst is full and needs cleaning. The cleaning takes 5 minutes. Thus, there are always three catalysts working and one cleaning. More purge gas is leaving the catalyst in the beginning of the cleaning than in the end, which gives a uneven inflow into the buffer tank.

[1] bar(a) is the unit of absolute pressure, where 0 bar(a) is perfect vacuum and 1.013 bar(a) is atmospheric pressure.

The flow into the combustion chamber has to be constant to keep same combustion temperature at all times. This also allows the air flow to stay constant giving a complete combustion. The control goal is now, by controlling the valves v_r and v_n, to keep the pressure of the purge gas tank between the given values and keep the energy flow into the combustion chamber as constant as possible. For details on the chemical process see e.g. [8] or [3].

Thus the control goals are:

- Keep the energy flow into the combustion chamber constant to allow for a constant inflow of air.
- Keep the pressure in the purge gas tank, P_r, between 1.32 and 1.82 bar(a) to keep the process running.

This shall be achieved by setting the opening level of the control valves, v_r and v_n. The measured input is the pressure in the purge gas tank level P_r.

4.1 Performance and Control of Today

Since 2008, Emerson's DeltaV system is in operation which offers advanced control possibilities such as an MPC implementation, but this possibility is currently not in use. The controller currently implemented is a PID with feed forward and an on-line tuning. The energy inflow into the combustion varies and thus the air inflow is over-estimated at most times – with the present control, the exhaust has to have a set point at 1.5% oxygen. The exhaust can not contain less than 0.5% of oxygen to always ensure an efficient combustion.

A benefit with lower residual oxygen content is that less air has to be heated, which means that less natural gas needs to be combusted. Natural gas stands for the greater part of the production cost. If the residual oxygen content is lowered from 1.5% to 0.5%, the use of natural gas will decrease with 44 Nm^3/h (the unit refers to normal cubic meter, which is a cubic meter of gas at normal pressure and temperature). With a price for natural gas price of 3 $\mathrm{SEK}/\mathrm{Nm}^3$ the savings would be around 1 000 000 SEK (about €110 000) per year. [11]

The air inflow to combustion is currently controlled with respect to the oxygen content in the exhaust gas. With the MPC solution this project is aiming for this control to be kept but optimally it would be redundant.

The idea in this project was to compare the PID Controller's and the Model Predictive Controller's performances in a Hysys model. It was intended to be finished before the work of this thesis was started. Unfortunate this has not been as expected and the Hysys model do still not perform as the process in the factory. Hence it has not been possible to make a proper comparison of the two control types. In this part I will therefore mainly focus on the development and performance of a Model Predictive Controller for the Steam Reformer process.

4.2 Modelling

The system was modelled using MATLAB. The model of the system without any control can be seen in Figure 4.2, where the valve openings are the average of what is given with full control.

As noted previously, when using MPC it is very important that the model is as accurate as possible. In this case the main uncertainty is the inflow to the purge gas tank that depends on the emptying of the catalysts. This process works in cycles of about 5 minutes, if it is assumed that the four catalysts are identical, otherwise in cycles of 20 minutes. In the process model the flow from the catalysts was modelled as a rough sawtooth wave, with appropriate max and min values known from measurements[2]. When verifying the controller, this was exchanged to a recording of the inflow of one short cycle. It is recommended for future implementations in reality to use a full 20-minute record, as it has been seen that the catalyst varies somewhat in behaviour. I also suggested an investigation of their behaviour over time, as this controller is intended to work for several years with preserved high performance.

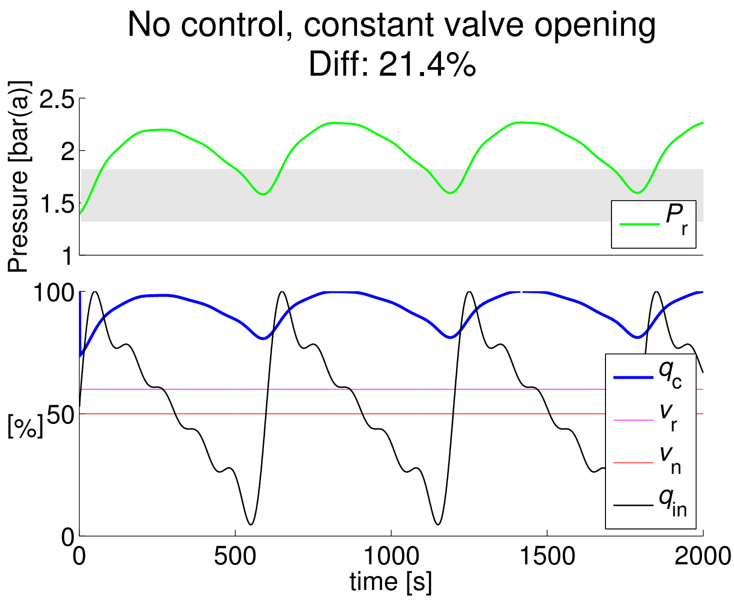


Figure 4.2: The steam reformer process without control. The inflow to the combustion chamber varies with 21.4%. The valve openings, v_r and v_n, are kept constant, and the inflows to both purge-gas tank, q_{in}, and to combustion, q_c, are in percent of maximum flow. P_r is the pressure in the purge-gas tank and the grey band indicates its desired span.

Both the control valves, v_r and v_n, are of the Equal Percentage type, but the first

[2]The details regarding the size of the flow are industrial secrets.

has a flow factor approximately 10 times larger than the second.[3] That a control valve is of Equal Percentage type means that the maximal flow is scaled with $\tau^{v_\mathrm{r}-1}$, where $v_\mathrm{r} \in [0, 1]$ is the opening of the valve and τ is a valve design parameter.

The maximal flow when the valve is fully opened is calculated as

$$q_\mathrm{max} = \frac{K_v N}{3600}\sqrt{\frac{(P - P_\mathrm{c})P_\mathrm{c}}{\rho_\mathrm{NTP} T}}, \tag{4.1}$$

where K_v is the flow factor of the valve, $N = 514$[4], P is the pressure before the valve, P_c is the pressure after the valve, ρ_NTP is the density of the gas at normal pressure and temperature, and T is the temperature of the gas. The constant 1/3600 is to get the result per second instead of per hour. The gas is assumed to keep constant temperature and composition due to good mixing.

The pressure in the purge gas tank, P_r, is measured but to control it we need to make an estimation of how it will change depending on the inflow:

$$\begin{aligned} \hat{P}_\mathrm{r}(k+1) &= P_\mathrm{r}(k) + c\left(q_\mathrm{in}(k) + q_\mathrm{max}\tau^{v_\mathrm{r}(k)-1}\right) \\ c &= \frac{\rho_\mathrm{NTP} R T}{M V_\mathrm{tank}} \cdot 10^{-5} \end{aligned} \tag{4.2}$$

where q_in is the inflow to the purge gas tank and v_r is the opening degree of the valve. c converts the flow from m^3/s to a pressure difference in bar and is assumed to be constant, R is the gas constant, M is the molar mass of the purge gas, and V_tank is the volume of the purge gas tank. The constant 10^{-5} is to convert from Pa, the SI unit for pressure, to bar, which is more convenient here.

This gives the flow $v_\mathrm{r} \cdot q_\mathrm{max,r}$ into the combustion. The flow from the natural gas grid is constant and expressed as $q_\mathrm{n} = v_\mathrm{n} \cdot q_\mathrm{max,n}$, where $q_\mathrm{max,n}$ according to Equation (4.1). What is interesting here is not the total gas inflow but the total *energy* inflow. The purge gas contains about 20 % of energy per mass compared to natural gas, and thus the flow of the purge gas has to be scaled. Hence the inflow to be controlled is calculated as

$$q_\mathrm{energy,\,c}(k) = 0.2 q_\mathrm{max,r}\tau^{v_\mathrm{r}(k)-1} + q_\mathrm{max,n}\tau^{v_\mathrm{n}(k)-1}. \tag{4.3}$$

4.3 Controller

The controller developed here consists of two coupled Model Predictive Controllers: the first to control the pressure in the purge gas tank, the other to adjust the flow from the natural gas grid to match the flow from the purge gas tank. The controllers are coupled in such that the predicted values of the first controller are used in the second.

[3] The details regarding the flow factors of the valves are industrial secrets.

[4] Known from working experience with control valves. [2]

An element in MPC theory is that constraints are implemented naturally in the solving process [13, ch. 2]. The control valves, v_r and v_n, have physical limitations, and can take values between 0 and 1, for fully closed to fully opened. Furthermore the pressure in the purge gas tank, P_r is limited to stay between 1.32 and 1.82 bar(a).

4.3.1 Linearisation

To use the model developed in Section 4.2 in the controller, Equations 4.2 and 4.3 had to be linearised. Linearisation of Equations (4.2) gives:

$$\hat{P}_\mathrm{r}(k+1) = P_\mathrm{r}(k) + cq_\mathrm{in}(k) + cq_\mathrm{r,max}\tau^{v_\mathrm{r}(k-1)-1}\left(1 + \ln\tau(v_\mathrm{r}(k) - v_\mathrm{r}(k-1))\right)$$

Since the valve by the purge gas tank has to be able to work in a large range of flows and pressures the linearisation point is the valve opening of the last iteration, $v_\mathrm{r}(k-1)$.

Linearisation of Equations 4.3 gives:

$$q_\mathrm{energy,c}(k+1) = 0.2q_\mathrm{max,r}\tau^{v_\mathrm{r}(k)-1} + q_\mathrm{n,max}\tau^{v_\mathrm{n}(k-1)-1}\left(1 + \ln\tau(v_\mathrm{n}(k) - v_\mathrm{n}(k-1))\right)$$

The valve for the natural gas has a much smaller working range, and like the purge gas tank valve it does not have a constant set point. Therefore the linearisation point is $v_\mathrm{n}(k-1)$. Here, $v_\mathrm{r}(k)$ is already known.

From above, equations on the form $\hat{P}(k+1) = \mathbf{A}_\mathrm{r}P(k) + \mathbf{B}_\mathrm{r}v_\mathrm{r}(k-1) + \text{constants}$ and $q_\mathrm{c}(k+1) = \mathbf{A}_\mathrm{n}q_\mathrm{c}(k) + \mathbf{B}_\mathrm{n}v_\mathrm{n}(k-1) + \text{constants}$ are found.

4.3.2 Verification

To verify the controller that is implemented in MATLAB a simulation of the whole steam reformer was used. This simulation is developed in HYSYS as a part of another master thesis at ÅF Group [11]. To use the data from the controller in the simulation during runtime, Microsoft Excel was used as an adapter. The Excel macros can retrieve data from and send data to HYSYS during runtime. It may also do so with text files, a capacity that also MATLAB has. The whole set up is illustrated in Figure 4.3. The usage of text files is necessary since MATLAB's reading from and writing to Excel files is very slow.

It was not possible to make MATLAB and HYSYS communicate, but by running them both in real-time, they at least get each other's data continuously. The MATLAB controller gets the present pressure in the tank and the HYSYS simulation gets the calculated best opening of the valves in each time step. The HYSYS simulation was updated twice every second. Retrieving and sending data was performed once every second in both directions.

For the verification, the inflow to the purge-gas tank was recorded and used in the model of the controller. Also, instead of using the tank pressure, the pressure just

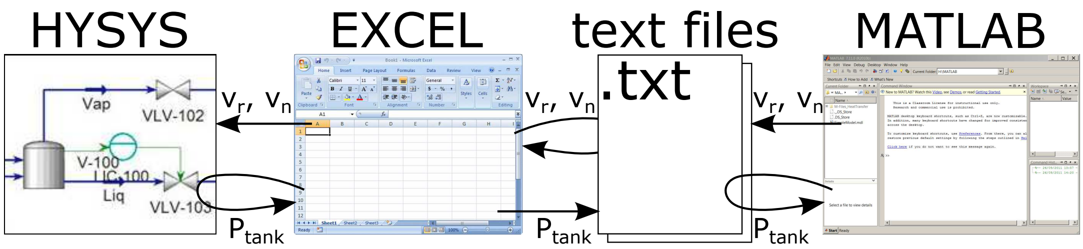


Figure 4.3: The work flow of the verification process.

before the valve was used. The discrepancy, between these pressures can be seen in Figure 4.4. The pressure decreases in the pipes between the tank and the valve. For the model of the controller to be correct and the Model Predictive Controller to work, a correlation needs to be found.

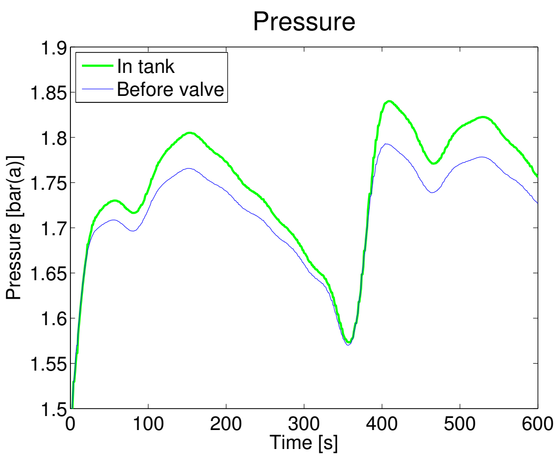


Figure 4.4: Pressure in purge-gas tank and just before the valve.

The correctness of the Model Predictive Controller is determined from the energy inflow into combustion. This can not be measured in the factory but only in the simulation model.

4.4 Results

The MPC keep the pressure of the tank within the limits and gives a close to constant inflow to the combustion, see Figure 4.5. The measured flow into combustion has a variation of 11 %.

The inflow to the combustion chamber seems to get an additional contribution from Valve 2, the valve controlling the natural gas. This contribution is not known by the controller.

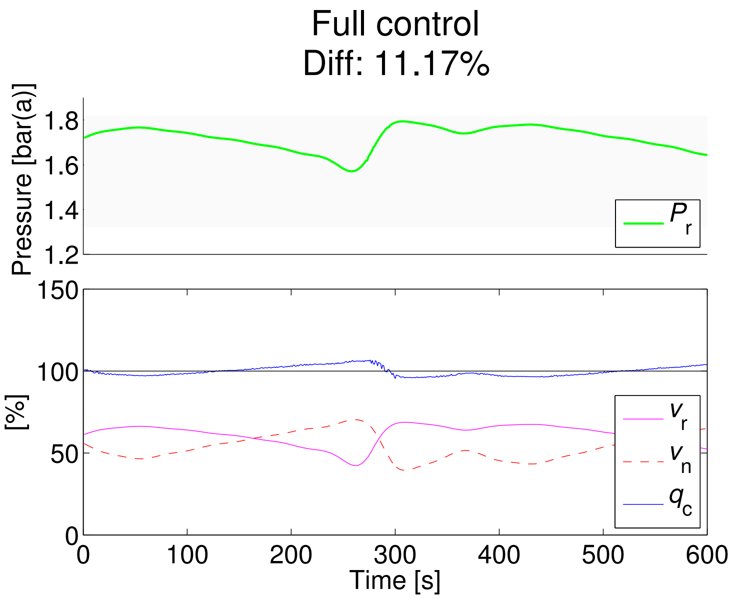


Figure 4.5: The performance of the Model Predictive Controllers used in the Hysys simulation. Valve openings, v_{r} and v_{n} are in percent of its maximum. The inflow to the combustion chamber, q_{in}, is in percentage of its mean. P_{r} is the pressure in the purge-gas tank and the grey band indicates its desired span.

4.5 Summary

In this chapter we have looked closer at the performance of a Steam Reformer process, especially at the feed back of purge gas into the combustion chamber. A Model Predictive Controller for the system has been developed from a theoretical model and then it's performance has been examined with the Hysys model. Here we have seen how important it is to have a model that matches the system well.

5 Discussion

As seen in the results of the tank systems, a Model Predictive Controller is not very good at handling unmodelled constant disturbances or disturbances that contain a constant element. They will then have a remaining error, an offset from the setpoint. In known from literature that propagation of measurement noise is lower with Model Predictive Controllers than PID [5, ch. 1], which can also be seen here. It has been verified in both parts of the project that the Model Predictive Controller is as good as its model, as the literature states.

Furthermore the Model Predictive Controller works very well with the given constraints and it is fast and accurate in comparison with the traditional PID controller. This has been seen theoretically in the tank systems, and is a great benefit in the steam reformer system discussed in Chapter 4 where it is so important to keep the constraints.

The Model Predictive Controller for the steam reformer, when verified with the Hysys simulation, does not work as well as expected. There as some possible sources of errors, which can be divided into parameter errors, unmodelled behaviour and errors in the Hysys simulation.

Regarding the parameter errors there is an uncertainty about the valve design parameter τ, in both the real system and in the simulation. Typically, the parameter should be between 20 and 50. Unfortunately it seems impossible to set this value for the Hysys valve element, and the parameter of the value in use cannot be found either. During simulations, different values of τ have been tried in the model, but it is difficult to tune it precisely without knowing the value of τ.

It has been assumed that the mixing of the gas in the purge gas tank is good, and therefore the composition, the temperature, and the density of the purge gas is constant. If the average time the gas stays in the tank is too short, this might not be true and to make a correct model it needs more investigation.

Another possible unmodelled behaviour is the speed of the control valves. They might have a short dead time when getting the signal to change until the valve actually moves. This was not modelled in my controller, nor the Hysys simulation at the time of verification, but of course it needs to be taken into account before implementing a controller onto the steam reformer system. If the dead time is implemented in the model of the Model Predictive Controller it should not be a problem, but it is necessary to investigate what is happening when the valve is moving constantly during several time steps.

As previously mentioned the Hysys simulation was not completely verified for the time of verifying the Model Predictive Controller of the steam reformer. This introduces a huge uncertainty. The fact that it was chosen to use two controllers, instead of one, does not seem to have affected the quality of the behaviour.

Very large amounts of gas flows in the steam-reformer system and a lot of money is spent on the production. With good system control, great savings can be made. This system has constraints which need to be obeyed. This would be well-handled using MPC, and the risk of failure with a subsequent halt of production would be decreased. It is therefore suggested that a Model Predictive Controller is used. Such an implementation also follows the practise used by the industry today. Nevertheless, the MPC developed in this project should not be implemented, since it has too bad performance.

6 Conclusion

The most important take-home message from this work is how well Model Predictive Controllers follows their model, with the consequence that errors in the model leads to pour results. In the model of the steam reformer, some parts – first appearing as mere details – were left out. This meant that the controller did not work as well as indicated. Nevertheless it has been seen that Model Predictive Controller is outstanding when comes to following a reference and stay within given constraints on several different parameters.

The model predictive controllers have not been thoroughly embraced by the industry, partly because they further shift the supervision of the process from the operator to the automatic controller. The educational material developed in this project can hopefully deal with some of this resistance. Another way to get around this problem, and to save computational power, MPC can be implemented on a higher level, giving reference signal to the actuators, but letting a traditional PID controller handle the details. This way an operator still may take over the system and handle it manually if needed.

Therefore, a better approach to this problem would probably be to have a more holistic view of the system, and to observe the flow and temperature of the exhaust. By specifying desired output, the expected exhaust can be calculated. From this, the desired flow from the purge-gas tank and natural-gas grid can be calculated. This may be more work than suitable for a M.Sc. thesis, but if it is possible to save 1 000 000 SEK it might be worthwhile hiring an engineer for the job.

Bibliography

[1] James B. Rawlings Rishi Amrit. *Nonlinear model predictive control tools (nmpc tools)*. Department of Chemical Engineering, University of Wisconsin, April 2008.

[2] Axel Larsson Maskinaffär AB, Upplands Väsby, Sweden. *Reglerteori Kortfakta.*

[3] L. Barelli, G. Bidini, F. Gallorini, and S. Servili. Hydrogen production through sorption-enhanced steam methane reforming and membrane technology: A review. *Energy*, 33(4):554 – 570, 2008.

[4] R. A. Bartlett, A. Wachter, and L. T. Biegler. Active set vs. interior point strategies for model predictive control. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, volume 6, pages 4229–4233 vol.6, 2000.

[5] Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control.* Springer, London, England, 2nd edition, 2007.

[6] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control - part i. the basic algorithm. *Automatica*, 23(2):137 – 148, 1987.

[7] J.F. Coales and A.R.M. Noton. An on-off servo mechanism with predicted change-over. In *Proc. Institution of Electrical Engineers, Part B, vol. 103*, pages 449–462, 1956.

[8] M. de Jong, A.H.M.E. Reinders, J.B.W. Kok, and G. Westendorp. Optimizing a steam-methane reformer for hydrogen production. *International Journal of Hydrogen Energy*, 34(1):285 – 292, 2009.

[9] Hans Joachim Ferreau, Hans Georg Bock, and Moritz Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[10] Miriam Hoekstra, Mathijs Vogelzang, Evgeny Verbitskiy, and Maarten W. N. Nijsten. Health technology assessment review: Computerized glucose regulation in the intensive care unit–how to create artificial control. *Critical Care (London, England)*, 13(5):223–223, 2009.

[11] Antun Hroh. Optimization of a steam reformer. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, in prep.

[12] Bengt Lennartsson. *Reglerteknikens grunder.* Studentliteratur, 4:7 edition, 2002.

[13] Jan Marian Maciejowski. *Predictive Control: with constraints.* Prentice Hall, Harlow, England, 2002.

[14] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer New York, New York, 2006.

A Appendix: Guide for Laboratory Practical on MPC on Tanks

A.1 Syfte och mål

Laborationens syfte är att visualisera modellbaserad reglering (Model Predictive Control, MPC) i jämförelse med traditionell PID-reglering.

Efter avslutad laboration ska du ha kunskaper om de grundläggande principer som MPC-reglering bygger på ur ett användarperspektiv. Du ska även ha kännedom om begrepp som styrsignalsstraff, prediktions- och reglerhorisont, och hur de används inom MPC-reglering.

Denna laboration förutsätter kännedom om PID-reglering t.ex. genom reglerteknikslaboration 1-4.

A.2 Introduktion till modellbaserad reglering

Modellbaserad reglering bygger på en teoretiskt eller praktiskt framtagen modell av systemet som ska regleras. Utifrån denna kan man förutsäga i vilket tillstånd systemet kommer att befinna sig i om en viss tid. Tiden mäts i *tidssteg*, alltså det intervall med vilket regulatorn uppdateras. Antalet sådana tidssteg som man vill förutse tillståndet kallas för *prediktionshorisont* och betecknas H_p. En algoritm hittar sen det optimala värdet på styrsignalen för framtiden. Det beräknade värdet för kommande samplingsintervall är det som används.

Om samplingsintervallet är för långt blir regulatorn långsam. Om samplingsintervallet är för kort finns en risk att regulatorn inte hinner genomföra alla beräkningar och ta fram en ny styrsignal innan den ska uppdateras, eftersom det tar mer datorkraft att ha ett kortare samplingsintervall.

Detsamma gäller prediktionshorisonten: är den för kort blir regulatorn dålig och långsam. Är den istället för lång tar det för mycket datorkraft att beräkna det bästa värdet. Det skapar en risk för att styrsignalen inte hinner beräknas innan den ska implementeras. Därför gäller det att hitta en bra avvägning både för samplingsintervallet och prediktionshorisonten.

Vidare finns även en parameter som kallas λ (lambda) som är ett straff på styrsignalen, alltså ett *styrsignalsstraff*. Denna parameter ska inte förväxlas med lambdametoden. Med ett högt värde på λ kommer regulatorn inte använda för stark styrsignal för att kontrollera systemet. Insvängningen blir då mjukare med risk för att bli långsammare. Omvänt ger ett litet lambda blir regulatorn mer aggressiv.

En av de främsta fördelarna med modellbaserad reglering är att det är möjligt att lägga in restriktioner när modellen byggs; så som att nivån i tanken inte får överstiga 100% och inte understiga 0%, att pumpen inte kan varva upp eller ner hur snabbt som helst och att ventiler inte kan öppnas eller stängas hur snabbt som helst. Detta är extra användbart i system där den optimala arbetspunkten ligger nära en gräns, t.ex. vid förbränning som det måste ha åtminstone en viss temperatur, men om temperaturen blir för hög kostar det mer än nödvändligt.

I industrin har den modellbaserade regulatorn framför allt fått genomslag inom processindustrin men den har också använts till allt från robotarmar till cementproduktion. Till skillnad från PID-regulatorn är den oftast krångligare att utveckla, men enklare att implementera. I många fall har att en välgjord modellbaserad regulator kunnat vara i bruk under lång tid utan nästan några åtgärder.

A.2.1 Viktiga begrepp

omodellerad störning: En, troligen okänd, störning som inte är modellerad i systemmodellen.

prediktionshorisont: Så många samplingsintervall in i framtiden som regulatorn ska förutse (prediktera) systemets beteende.

styrsignalsstraff: Betecknas oftast med den grekiska bokstaven λ (lambda). Gör det svårare för regulatorn att använda en aggressiv styrsignal.

systemmodell: Teoretisk beskrivning av systemets dynamik som regulatorn bygger på.

samplingsintervall: Tidsintervallet mellan uppdateringarna av styrsignalen.

stegsvar: Den tid som det tar för systemet att nå jämviktsläge efter ett steg.

A.3 Laboration

I de här tre laborationerna får du bekanta dig med modellbaserad reglering och jämföra dess egenskaper med PID-regulatorns. Först får du lära känna systemet och testa på att regulatorn naturligt implementerar gränser – en av de viktigaste egenskaperna hos modellbaserade regulatorer. Nästa steg är att göra en lämplig avvägning och ställa in regulatorns parameterar. Sista delen handlar om störningar och hur de kan hanteras av modellbaserade regulatorer jämfört med PID-regulatorer.

A.3.1 Uppgift 1: Lär känna systemet och testa gränserna.

I den här uppgiften ska du testa att låta regulatorn svänga in till ett börvärde som ligger nära en fysisk gräns, i det här fallet begränsningen att tanken inte får bli överfylld.

Gå in i “Enkeltank med MPC” och gör följande inställningar:

Inställningar, uppgift 1	
Pump 1	Auto
Horisont	100
λ	1
Tidssteg	1
Börvärde	75
Ventil 1	30%
Ventil 2	0%

När systemet har svängt in, ändra börvärdet i ett steg till 95%.

Vad händer?

Vad tror du hade hänt med en PID-regulator? (Om du känner dig osäker, gå gärna in och testa.)

A.3.2 Uppgift 2: Ställ in regulatorn.

I den här uppgiften ska du ställa in parametrarna för en så bra modellbaserad regulator som möjligt. Detta sker genom att undersöka hur snabbt systemet svarar på ett steg utan reglering, detta kallas stegsvar.

Gör följande inställningar:

Inställningar, Uppgift 2	
UT	300
Ventil 1	40%
Ventil 2	0 %

När tanken har nått jämviktsläge, alltså när utflödet är lika stort som inflödet och nivån i tanken hålls konstant, ändra ”UT” till 600. Klicka fram ”Trend”-grafen och se hur lång tid det tar innan systemet åter befinner sig inom 2% från jämviktsläge och kalla den tiden T.

Vad brukar man kalla termen T?

För enkelhetens skull kan vi till att börja med välja $\lambda = 1$ och $T_{\mathrm{s}} = 1\,\mathrm{s}$, där Ts är samplingsintervallet. Sätt sedan $H_{\mathrm{p}} = T/T_{\mathrm{s}}$. Sätt $H_{\mathrm{c}} \approx 0.1H_{\mathrm{p}}$, avrunda uppåt till närmaste heltal.

Vad skulle hända med T om du valde en annan storlek på ditt steg?

Gör om experimentet med steget från början av den här uppgiften, fast låt den här gången pumpen vara i ”Auto” med inställningarna ovan.

Vad är skillanden mot ett helt oreglerat system?

Prova några olika värden på λ och se hur det påverkar regleringen. Testa t.ex. $\lambda = 10$ och $\lambda = 0.1$.

Hur påverkas regleringen av ett stort respektive litet λ med avseende på

insvängningstid och aggressivitet?

Använd värdena från Laboration 2, ”Uppgift 7: Optimering” för PID-regulatorn.

Hur skiljer sig den modellbaserade regulatorn från PID-regulatorn?

A.3.3 Uppgift 3: MPC och störning

I den här uppgiften ska du se hur den modellbaserade regulatorn klarar störningar som inte finns i modellen och sedan jämföra detta med hur PID-regulatorn klarar samma sak.

Använd värdena som togs fram för den modellbaserade regulatorn i Uppgift 2. Använd värdena från Laboration 2, ”Uppgift 7: Optimering” för PID-regulatorn.

Utflödet genom ventil 2 är inte implementerat i modellen som regulatorn bygger på, utan kallas *omodellerad störning*. Du ska nu undersöka hur väl regulatorn kan hantera en sådan.

Ställ in systemet med de värden du kom fram till i förra uppgiften.

Inställningar, uppgift 3	
Pump 1	Auto
Horisont	*Enligt förra uppgiften*
λ	*Enligt förra uppgiften*
Tidssteg	*Enligt förra uppgiften*
Börvärde	60
Ventil 1	30%
Ventil 2	0%

Låt systemet svänga in. Sätt Ventil 2 till 10%.

Vad händer? Hur hanterar regulatorn störningar?

Vad tror du hade hänt med en PID-regulator? (Om du känner dig osäker, gå gärna in och testa.)

A.4 Sammanfattning

Som du har sett här har modellbaserade regulatorer många fördelar som PID-regulatorer saknar. En av fördelarna för operatören är att de är lättare att ställa in och inte har lika många driftstopp som traditionella regulatorer eftersom de kan hålla sig inom vissa givna gränser.

När den här laborationen är godkänd har du grundläggande kännedom om modellbaserade regulatorer när de dyker upp ute i verkligheten.