



NTNU

Department of  
Chemical Engineering

TKP4581 - Specialization project

Moving throughput manipulator

for improved bottleneck control

- with application to level control

for tanks in series

Student: David Alexander Lillevold Skaug

Main supervisor: Sigurd Skogestad

Co-supervisor: Cristina Zotica

December 2019

## **Abstract**

Inventory or material balance control is important in the process industry to ensure that the mass balance for a plant and the process units in it are fulfilled at all times. The aim of this project was to design a supervisory control layer that could change the location of the throughput manipulator (TPM) when the bottleneck of the process changes when different disturbances occur. This was done by use of split range control (SRC) and model predictive control (MPC). It is shown that the control structure with SRC have large delays because of the windup in the controllers when switching between different MVs. This is not the case for the MPC, because it uses the manipulated variables freely without have to "see" a change in the inventories beforehand. In the cases studied here, supervisory control with MPC seems to give the best solution.

# Contents

|   |           |
|---|-----------|
| <b>List of figures</b>                                      | <b>iv</b> |
| <b>List of tables</b>                                       | <b>vi</b> |
| <b>1 Introduction</b>                                       | <b>1</b>  |
| 1.1 Motivation . . . . .                                    | 1         |
| <b>2 Theory</b>   | <b>3</b>  |
| 2.1 Control hierarchy in a plant . . . . .                  | 3         |
| 2.1.1 Supervisory control layer . . . . .                   | 4         |
| 2.2 Split range control . . . . .                           | 6         |
| 2.3 Selectors . . . . .                                     | 8         |
| 2.4 Model predictive control . . . . .                      | 8         |
| 2.4.1 Sequential approach . . . . .                         | 9         |
| 2.4.2 Simultaneous approach . . . . .                       | 10        |
| 2.5 SIMC tuning . . . . .                                   | 11        |
| <b>3 Level control for tanks in series</b>                  | <b>13</b> |
| 3.1 Process description . . . . .                           | 13        |
| 3.2 Model . . . . .   | 14        |
| 3.3 Split range control . . . . .                           | 16        |
| 3.4 Improved split range control with bias update . . . . . | 18        |
| 3.5 Model predictive control . . . . .                      | 19        |
| <b>4 Simulation results</b>                                 | <b>22</b> |
| 4.1 Split range control . . . . .                           | 22        |
| 4.2 Split range control with bias update . . . . .          | 23        |
| 4.3 Model predictive control . . . . .                      | 25        |

|                                     |           |
|-------------------------------------|-----------|
| <b>5 Discussion</b>                 | <b>27</b> |
| <b>6 Conclusion and future work</b> | <b>28</b> |
| <b>Appendix</b>                     | <b>30</b> |
| <b>A Comments to MATLAB files</b>   | <b>31</b> |
| <b>B Tuning of level controller</b> | <b>32</b> |
| <b>C Detailed Simulink model</b>    | <b>34</b> |

# List of Figures

|    |  |    |
|----|--|----|
| 1  | Local consistency for inventory control. The control structure must radiate around the TPM [1]. . . . .  | 2  |
| 2  | Control hierarchy in a plant with time-scale separation [9]. . . .   | 4  |
| 3  | Switching between two MVs to maintain control of CV1 (y1) (MV-CV switching). C1 and SR in the figure makes up the "full" split range controller [8]. . . . . | 6  |
| 4  | Split range controller where the MV is denoted with u and CV by y [6]. . . . .   | 7  |
| 5  | Typical split range block with the split value $v^*$ at 50 % [6] . . .   | 7  |
| 6  | The process which is modeled in this project. The levels + TPM are the CVs while the four flows are the MVs. . . . .   | 14 |
| 7  | Control structure must radiate from the TPM to be locally consistent . . . . .   | 14 |
| 8  | Process in Simulink . . . . .  | 15 |
| 9  | Control structure with split range control. . . . .  | 16 |
| 10 | Split range block in the simulations conducted in this project. The split value is here at 4500. . . . .   | 18 |
| 11 | Split range block with bias update on MV2. . . . .   | 19 |
| 12 | Control structure with MPC. . . . .  | 21 |
| 13 | Simulation with step change on q1, without bias update . . . .   | 22 |
| 14 | Simulation with step change on q3, without bias update . . . .   | 23 |
| 15 | Simulation with step change on q1, with and without bias updates   | 24 |
| 16 | Simulation with step change on q3, with bias updates . . . . .   | 24 |

|    |  |    |
|----|--|----|
| 17 | Comparing simulation of MPC and SRC (with bias updates) with a step change on $q_3$ . Both the actual flow and the flow setpoint where changed in the simulation with MPC. . . . . | 25 |
| 18 | Close-up of the MPC solution in figure 17. Both the actual flow and the flow setpoint where changed for $q_3$ . . . . .  | 26 |
| 19 | Process in Simulink . . . . .  | 34 |
| 20 | Level controller for tank 1 . . . . .  | 35 |

# List of Tables

|   |  |    |
|---|--|----|
| 1 | Nominal operating conditions . . . . . | 15 |
|---|--|----|

# Chapter 1

## Introduction

This section will give a general background for the work in this project. The aim of the project is also described here.

### 1.1 Motivation

Inventory or material balance control is important in the process industry to ensure that the mass balance for a plant and the process units in it are fulfilled. In other words, the inventory control system is "consistent" by keeping the inventories (of total mass, components and phases) within an acceptable range, both dynamically and at steady-state. "Local consistency" is a stricter version of consistency, where the inventory of a unit is controlled only by using the in- or outflow of the unit [1].

In every flow network there must also be at least one degree of freedom, called the throughput manipulator (TPM), which sets the production through the system. For units in series, how the control structure for local consistency will look depends on where the TPM is placed, see figure 1 [1, 2]. It is often the case that optimal operation is the same as maximizing throughput, which means maximizing the flow through the bottleneck of the plant. The TPM should then be placed at the bottleneck to maintain optimal operation. Should the bottleneck move in the plant, then the control loops and TPM must be relocated to ensure optimal operation and consistency. This may however not be desirable or practical, because it can confuse the operators [2].

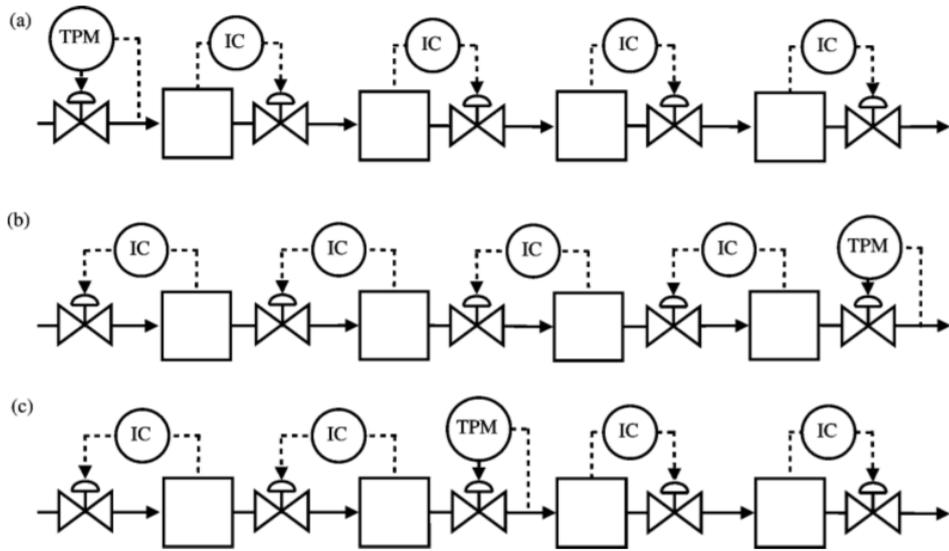


Figure 1: Local consistency for inventory control. The control structure must radiate around the TPM [1].

The aim of this project is to design a supervisory control layer that can change the location of the TPM when the bottleneck of the process changes. The equipment in the process can for instance be a distillation column, reactor or anything that contain an inventory. In this project these will be modeled as tanks with simple feedback control with a throughput manipulator originally at the inlet of the process (case a in figure 1), but this can change depending on the disturbance on the system.

# Chapter 2

## Theory

The theory needed for the report is described in this section, starting from a higher overview and later going into more detail about each topic.

### 2.1 Control hierarchy in a plant

In a chemical plant there may be a countless number of control loops in the regulatory or stabilizing level. There are several layers above that, all separated by a certain time scale as shown in figure 2. A procedure to make a complete control structure design for chemical plants is given by S. Skogestad ([10]) and consist of two parts: top-down and bottom up. The top-down approach looks at the economics of the plant, while bottom-up deals with obtaining a control structure to stabilize the plant under any (economic) operation. The role of the supervisory layer will be further discussed here.

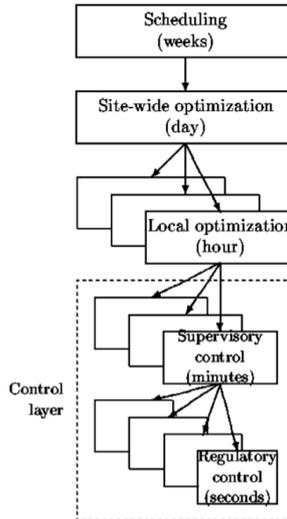


Figure 2: Control hierarchy in a plant with time-scale separation [9].

### 2.1.1 Supervisory control layer

The supervisory control layer has mainly three tasks [10]:

1. Control the primary controlled variables CV1, with their setpoints given from the layer above (these can be set by, for instance, an operator or Real-time optimization (RTO)). The supervisory layer uses the setpoints to the regulatory layer as primary manipulated variables (MV1) to keep the CV1 at their setpoints. The supervisory layer can also use other variables such as valves as manipulated variables, which are not used by the regulatory layer (extra valves) [10].
2. Make sure that no MV in the regulatory level saturates (reached min or max value), or else the CV2 will drift away from its setpoint [10].
3. Switch out controlled variables and control strategies when the active constraints in the process change due to disturbances to the process. This can be either done with PID control with some added "advanced" elements like feedforward, split range control or other logic or by use of model predictive control (MPC) [10].

A procedure is proposed for maintaining optimal operation when different disturbances occur [4, 8]:

1. Define all the constraints

2. Identify relevant active constraint combinations and switches.
3. Propose a control structure for the nominal point.
4. Propose switching schemes.
5. Design controllers for all cases (active constraint combinations)

The maximum number of active constraint combinations are given by  $2^{n_g}$ , where  $n_g$  is the number of constraints. The number of combinations can in practice be lower, as some constraints might be always active and not all combinations are physically possible.

There are three main cases in switching control schemes: CV-CV, MV-MV and MV-CV switching [4].

1. CV-CV switching: When there's one MV to control several CV and a selector must be used to switch between which CV to control.
2. In MV-MV switching one MV saturates and another MV must be used to further control the CV. Split range control, controllers with different setpoints or valve position control can be used in this case.
3. MV-CV switching occurs when there's two MVs and two CVs (MV1, MV2, CV1, CV2, not to be confused with the notation used above), and one MV, e.g. MV1, saturates leaving CV1 uncontrolled. If the input saturation rule was followed (meaning CV1 can be given up) then we do not have to do anything. If the rule was not followed (meaning CV1 can not be given up), then control of CV2 must be given up so MV2 can be used to control CV1. This can be implemented by use of split range control with a min/max selector, see figure 3 [4, 8]. MV-CV switching is the case studied in this project.

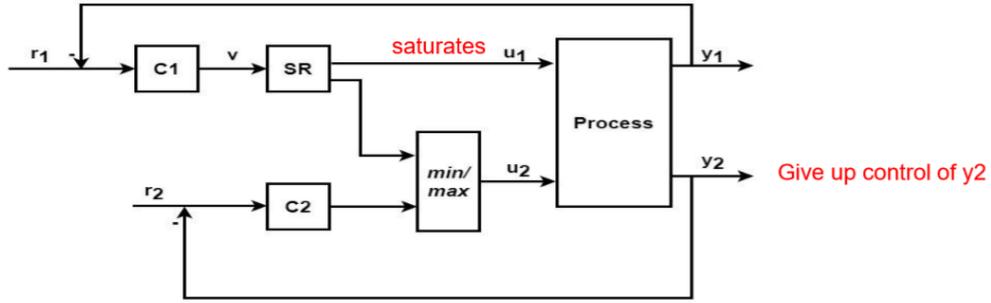


Figure 3: Switching between two MVs to maintain control of CV1 ( $y_1$ ) (MV-CV switching). C1 and SR in the figure makes up the "full" split range controller [8].

### Valve position control

Valve position control (VPC) can be used when there's two MV and one CV, where the MVs have a big difference in gain. An example of this is two valves placed in parallel, one small (MV1) and one large (MV2), where use of the large valve is slow compared to MV1 and does not give sufficient control by itself. MV2 is only used once MV1 goes beyond a desired position, to bring it back to the desired position. If optimal operation requires MV1 to be almost saturated then VPC gives a near-optimal operation, since MV1 must be within a desired range to handle disturbances (back-off from the optimal operation point) [5].

## 2.2 Split range control

Split range control (SRC) is used when there's several MV that are used control one CV, usually in the sense that one MV takes over when the first MV saturates. Figure 4 shows a SRC with two MV ( $u_1$  and  $u_2$ ) and one CV ( $y$ ). C is usually a PI controller, which outputs a internal value  $v$  that is sent to the split range (SR) block that outputs the physical manipulated variables [6]. SRC is also the control structure used in this work.

An example of how a SR block might look is shown in figure 5 with the split value  $v^*$ , where the controller switches between using either  $u_1$  or  $u_2$ , set

at 50%. This value might be different depending on the process where SRC is used, since the split value should be used as a design parameter to fit the dynamic response of each MV. Changing  $v^*$  will change the slope,  $\alpha$ , which is a measure of the gain (or effect) the internal value has on each MV [6].

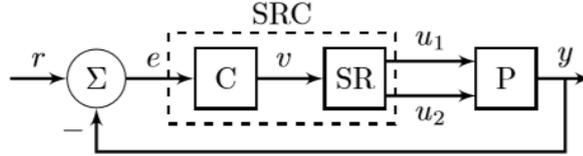


Figure 4: Split range controller where the MV is denoted with  $u$  and CV by  $y$  [6].

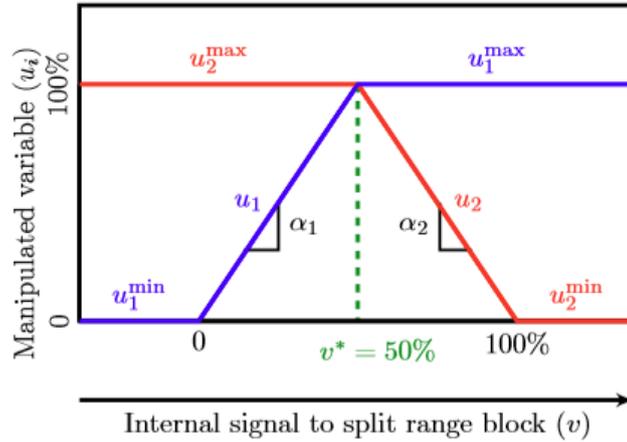


Figure 5: Typical split range block with the split value  $v^*$  at 50 % [6]

A systematic way to design the split range block is given in [6] and has seven steps:

1. Define the range of the internal signal  $v$ .
2. Define the physical bounds for each MV.
3. Get the desired controller tuning for each MV.
4. If using PI control, then the integral time for the SRC should be chosen depending on if the process is slow (integrating process) or fast. For slow processes a larger integral time should be used and small for fast processes for stability reasons.

5. Choose the order in which the MV are used based on economic reasons.
6. Find the slopes  $\alpha_i$  from equation 1 and either 2 (fast process) or 3 (slow process).  $K_c$  and  $K_l$  are the proportional and integral gain in the controllers, respectively.  $K_{C,i}$  and  $\tau_i$  are obtained from the tuning of each individual MV, while  $K_C$  and  $\tau_I$  are for the SRC.
7. Find the split values by using equation 4 for all MV.

$$v^{max} - v^{min} = \sum_{i=1}^N \frac{u_i^{max} - u_i^{min}}{|\alpha_i|} \quad (1)$$

$$K_{l,i} = \alpha_i K_l \quad (2)$$

$$K_{C,i} = \alpha_i K_C \quad (3)$$

$$\Delta v_i = v_i^* - v_{i-1}^* = \frac{u_i^{max} - u_i^{min}}{|\alpha_i|} \quad (4)$$

### 2.3 Selectors

In CV-CV switching one MV is used to control two different CVs where a selector is used to switch between which CV is controlled at any given time. Each CV can have their own controller and the selector (a logical switch) will either take the highest (max selector) or lowest (min selector) output from the controllers (MV) and send that value to the plant. This ensures that the CV that is constrained is always controlled. The CVs does not need to be of the same type, e.g. one can be temperature and the other flow [4].

### 2.4 Model predictive control

In model predictive control (MPC) an optimization problem, with initial conditions being the current operating point of a plant, is solved for a finite horizon (also called control horizon) and an optimal control sequence is found.

The first control in this sequence is implemented in the plant, and a new optimization problem is solved for the next sampling point [3].

A typical control objective, called the cost function is given in equation 5 [7]

$$\min \sum_{i=1}^N (x - x_{i, \text{setpoint}})^T Q (x - x_{i, \text{setpoint}}) + \sum_{i=1}^M (u_k - u_{k-1})^T R (u_k - u_{k-1}) \quad (5)$$

subject to

$$\frac{dx}{dt} = f(x(t), u(t), d(t)) \quad (6)$$

$$g(x(t), u(t)) \quad (7)$$

$$x(t_0) = x_0 \quad (8)$$

$$u_{min} \leq u \leq u_{max} \quad (9)$$

which the optimizer will then try to minimize by changing the MVs. Q and R are the tuning parameters, g is the operational constraints (such as temperatures that can not go above a certain limit), x is the differential state, z algebraic state and u the inputs (MVs). The constraints (including the model equations) are usually given in the form of differential (eq. 6) or differential algebraic equations (eq. 10 and 11) [7].

$$\frac{dx}{dt} = f(x(t), z(t), u(t), d(t)) \quad (10)$$

$$0 = g(x(t), z(t), u(t)) \quad (11)$$

The MPC problem formulation can be mainly set up in two ways: using the sequential approach or the simultaneous approach.

### 2.4.1 Sequential approach

The concept in the sequential approach, also called single shooting, is to discretize the control profile for a certain time period divided into intervals. The

ODEs are integrated based on an initial guess for all the inputs and the cost function is evaluated. This information is then given to the optimizer, that finds new values for the inputs that decreases the cost function. This is repeated until the solution is within the set tolerance. This is a simple and easy to implement approach with a small optimization problem, but is poorly suited for unstable or very nonlinear processes [7].

### 2.4.2 Simultaneous approach

Single shooting is not suited for nonlinear problems, and this becomes even worse if a longer integration time is needed. The way to solve this is to integrate over multiple time steps, called multiple shooting. Here also the state variables  $x$  are given to the optimizer, in addition to the inputs (MVs) used in single shooting. An initial guess must be given for the state variables, like the inputs, but the states are also calculated from the integrator. To get a meaningful solution then this constraint, shown in equation 12, is also given to the optimizer making the optimization problem larger. Here  $f$  is the integrator function while  $x_{i+1}$  is the variable in the optimizer [7].

$$f(x_i, u_i) - x_{i+1} = 0 \quad (12)$$

Another way to transform the differential equations into something the optimizer can solve is by using direct collocation. The concept used here is to also let the optimizer do the integration by approximating the solution  $x(t)$  by a polynomial of order  $K$  [7]

$$x(t) \approx A + Bt + \frac{1}{2}Ct^2 + \frac{1}{3}Dt^3 + \dots \quad (13)$$

and use this to find a relation between  $\frac{dx}{dt}$  and  $x(t)$ . There are a number of methods that can be used based on the number of collocation points (points between each state), position of the collocation points (time steps) and accuracy. Some examples are Gauss-Legendre, Gauss-Radau and Gauss-Lobatto [7].

As an example, if we want to solve a problem using three collocation points and a polynomial of order 3 (number of points and order must be equal) the equations that we must add to the nonlinear optimization problem as constraints is then [7]

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_0 \\ x_0 \end{bmatrix} + M \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} \quad (14)$$

where  $x_0$  is our initial guess and  $x_1$ ,  $x_2$  and  $x_3$  is the collocation points. If we insert our expressions for  $x(t)$  and  $\frac{dx}{dt}$  and rearrange, then we get the M matrix [7]

$$M = \begin{bmatrix} t_1 & \frac{1}{2}t_1^2 & \frac{1}{3}t_1^3 \\ t_2 & \frac{1}{2}t_2^2 & \frac{1}{3}t_2^3 \\ t_3 & \frac{1}{2}t_3^2 & \frac{1}{3}t_3^3 \end{bmatrix} \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix}^{-1} \quad (15)$$

The values of  $t_1$ ,  $t_2$  and  $t_3$  will depend on the method used (Gauss-Legendre, Gauss-Radau or Gauss-Lobatto). This can be done for the whole prediction horizon, but will result in very poorly performance/accuracy. Instead it's possible to divide the horizon into smaller elements and do collocation on each element, called orthogonal collocation on finite elements. This is an efficient way to solve the differential equations, with a trade-off between accuracy and performance; shorter elements increases the accuracy, but makes the problem larger [7].

## 2.5 SIMC tuning

A systematic way to tune a PID controller is by use of the SIMC rules [11]. The procedure here is to obtain a first- or second-order plus time delay model of the process, and obtain controller parameters based on this model. A first order model in the Laplace-domain is

$$g_1(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} \quad (16)$$

with tuning parameters for a series-form controller

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} \quad (17)$$

$$\tau_I = \min(\tau_1, 4(\tau_c + \theta)) \quad (18)$$

Here  $k$  is the process gain,  $\tau_1$  is the open loop time constant,  $\theta$  is the time delay,  $K_c$  is the controller gain and  $\tau_I$  is the integral time. With these systematic tuning rules, we have only one tuning parameter  $\tau_c$  [11].

The transfer-function model can either be derived from the differential equations for the states or by doing a step-change in the input to the process (output from controller,  $u$ ) and measuring the CV  $y$ . If using the step-change approach, then the plant gain is calculated as  $k = \frac{\Delta y}{\Delta u}$  at steady-state. If the process uses a very long time to reach steady-state ( $\tau > 8\theta$ ) then the gain can be calculated as  $k' = \frac{\Delta y}{\Delta t \Delta u}$  (integrating process). The model then becomes

$$g_1(s) = \frac{k'}{s} e^{-\theta s} \quad (19)$$

with tuning parameters [11]

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta} \quad (20)$$

$$\tau_I = 4(\tau_c + \theta) \quad (21)$$

## Chapter 3

# Level control for tanks in series

A general description is given here for the process studied in this work, starting from the basis and then moving into more detail for each of the control structures used.

### 3.1 Process description

In this report the process consists of three tanks in series shown in figure 6, with the TPM being  $q_0$ . The main control objective is to keep the level in the tanks at their setpoint, while it is also desirable to control the TPM at a given value. The MVs are the flows  $q_i$  for  $i = 0, 1, 2, 3$ . It is assumed a fast cascade on the valve position, so the flows are set directly. One MV is used to control the TPM, while the other three are used to control the levels. Controlling the TPM can however be given up, in favor of controlling the levels when one of the flows saturate. To obtain local consistency the control structure must radiate from the TPM as shown in figure 1. This is shown in figure 7.



Figure 6: The process which is modeled in this project. The levels + TPM are the CVs while the four flows are the MVs.

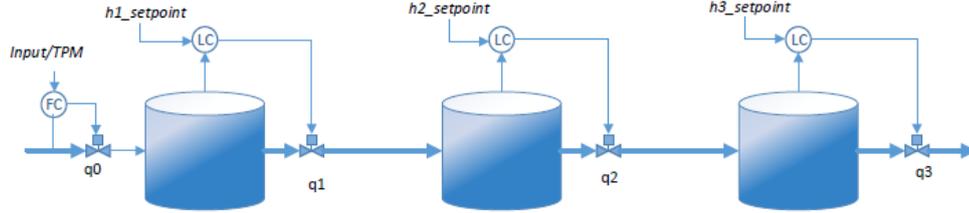


Figure 7: Control structure must radiate from the TPM to be locally consistent

Here the TPM is placed at the inlet of the process, but when a disturbance occur (e.g. the flow out of a tank is suddenly reduced due to plugging in the pipe) then the bottleneck changes place. The aim of this project is to design a supervisory control layer that can change the location of the TPM when the bottleneck of the process changes. This will be done using SRC with min selectors and MPC.

## 3.2 Model

Starting out from the mass balance for one tank

$$\frac{dm_i}{dt} = q_{i-1} - q_i \quad (22)$$

where  $q$  has units  $\text{kg}/\text{min}$ . Then introducing  $m = \rho V_i$ ,  $V_i = A_i h_i$  and rearranging the differential equations telling how the level of the tanks change with the flows are obtained is equation 23

$$\frac{dh_i}{dt} = \frac{1}{\rho A_i} (q_{i-1} - q_i) \quad (23)$$

Here  $\rho$  is the density of the liquid (constant),  $A$  is the area of the bottom of the tank (constant, equal for all tanks),  $h_i$  is the level in the tanks and  $q_i$  is the flow, for  $i = 0, 1, 2, 3$ . The liquid is modeled as water with  $\rho = 1000 [kg/m^3]$ , tanks having a diameter of  $8 m$  and the flow is bound to a min/max of  $0/9000 kg/min$  for  $q_1, q_2$  and  $q_3$ . Figure 8 shows how this is done in Matlab/Simulink, while the nominal operating conditions for the simulations are given in table 1.

Table 1: Nominal operating conditions

| Variable | Value | Unit   |
|----------|-------|--------|
| $h_1^*$  | 4     | m      |
| $h_2^*$  | 4     | m      |
| $h_3^*$  | 4     | m      |
| $q_0^*$  | 4500  | kg/min |
| $q_1^*$  | 4500  | kg/min |
| $q_2^*$  | 4500  | kg/min |
| $q_3^*$  | 4500  | kg/min |

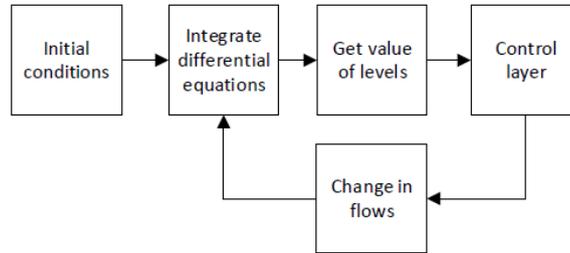


Figure 8: Process in Simulink

The SIMC procedure for tuning the level controllers have been used here by derivation of the transfer function, see appendix B. Since the model of the tanks are all equal, only one tuning is necessary and the same parameters are then given to each controller in figure 7.

### 3.3 Split range control

We want to improve the control structure from figure 7 such that the level controller can also manipulate the flow into the tank, in case the outlet flow saturates. The case could be that a plug form in the pipe out of the tank, and the level controller tries to open the valve more but nothing happens to the flow. This is here done by using split range control and min selectors, as shown in figure 9. The case here is then that we have MV-CV switching (figure 3), with MV1 being  $q_1$  and MV2 being  $q_0$ . The CV that can be given up is the TPM to maintain control of the level in tank 1. Anti-windup is implemented for the controllers, by stopping the integral-action when MV2 saturates for all SRCs.

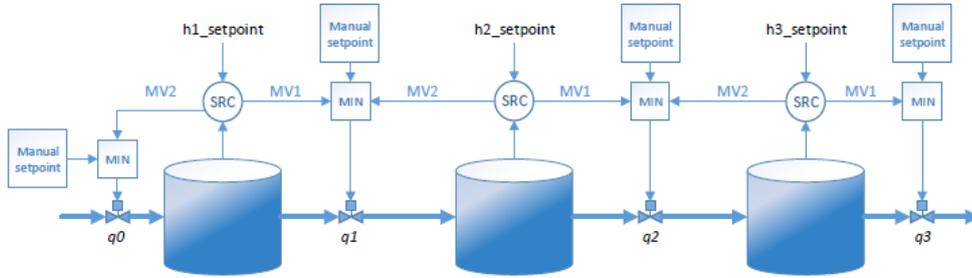


Figure 9: Control structure with split range control.

For the split range block the seven-step procedure described in [6] is used. The resulting split range block is shown in figure 10.

1. Range of  $v$ :  $-4500 \leq v \leq 13500$
2. Bounds for MV:  $0 \leq u \leq 9000$
3. Controller tuning for each MV: Shown in appendix B. Since the MVs are the outlet and inlet flow, they will have equal gain but with different sign.
4. Integral time: As both MVs have the same impact on the state (just different gain), the same integral time as the individual MV was chosen for the overall SRC.
5. Want to maximize throughput: Use outlet flow of tank first, inlet flow second.

6. Using equation 1

$$13500 - (-4500) = \frac{9000 - 0}{|\alpha_1|} + \frac{9000 - 0}{|\alpha_2|} \quad (24)$$

and equation 3

$$K_{C,1} = \alpha_1 K_C \quad (25)$$

$$K_{C,2} = \alpha_2 K_C \quad (26)$$

we get that  $\alpha_1 = -\alpha_2$  from equation 25 and 26. Inserting this into equation 24

$$18000 = \frac{9000}{|\alpha_1|} + \frac{9000}{|\alpha_1|} \quad (27)$$

yields that  $\alpha_1 = 1$  and  $\alpha_2 = -1$ .

7. Find split value by use of equation 4

$$\Delta v_1 = \frac{9000 - 0}{1} = 9000 \quad (28)$$

which puts the split value at

$$v^* = -4500 + 9000 = 4500 \quad (29)$$

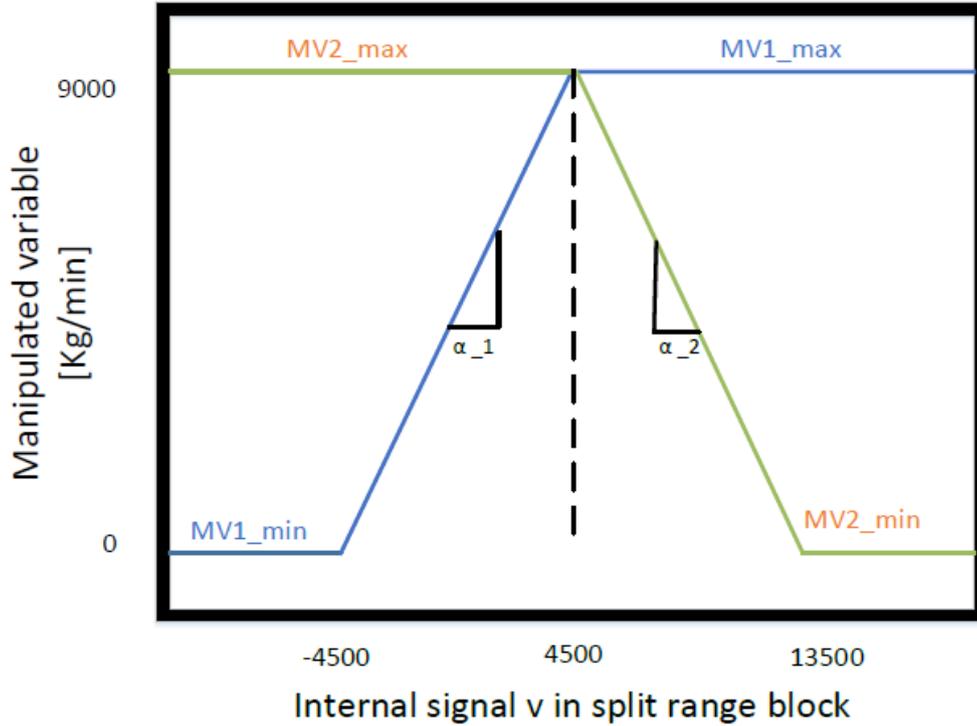


Figure 10: Split range block in the simulations conducted in this project. The split value is here at 4500.

### 3.4 Improved split range control with bias update

To further improve the split range control structure in figure 9 the addition of updating the bias for each controller is implemented. The idea behind this is to remove the wind-up on MV2 for the controllers. For the SRC on tank 1 this is done by setting  $MV2_{max}$  in figure 10 equal the manual setpoint on  $q_0$ . This will change the whole graph for MV2, as the slope  $\alpha$  is held constant, see figure 11.

The bias updates becomes a bit more complex for the SRC on tank two and three, since the inlet valve can here be manipulated by two other sources; either the manual setpoint or from MV1 of the upstream SRC. The bias for the second tank is set as the minimum (with another min selector, not the one shown in figure 9) of either the manual setpoint or the output from tank 1. The same approach is done for tank 3.

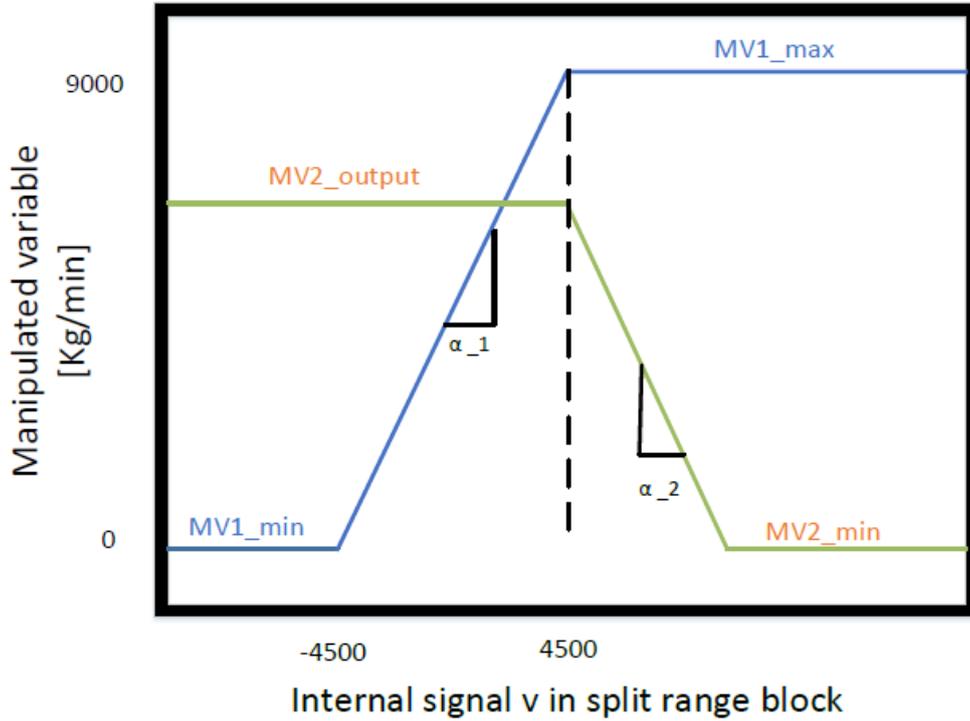


Figure 11: Split range block with bias update on MV2.

### 3.5 Model predictive control

The control structure with MPC is shown in figure 12. The MPC problem is here formulated by use of Gauss-Radau orthogonal collocation with three collocation points and a sampling time of ten seconds, with CasADi in Matlab. The horizon is 60 minutes with  $N = 360$  control intervals and the optimization problem is solved by use of IPOPT. The collocation points is given at times  $t_1 = 0.155051$ ,  $t_2 = 0.644949$  and  $t_3 = 1$  for a time interval of 0-1, so the times needs to be scaled for a sampling time of 10 seconds. For a level  $h_i$  and use of equation 14, the collocation points become

$$\begin{bmatrix} h_{i,1} \\ h_{i,2} \\ h_{i,3} \end{bmatrix} = \begin{bmatrix} h_{i,0} \\ h_{i,0} \\ h_{i,0} \end{bmatrix} + bM \begin{bmatrix} \frac{dh_{i,1}}{dt} \\ \frac{dh_{i,2}}{dt} \\ \frac{dh_{i,3}}{dt} \end{bmatrix} \quad (30)$$

with the scaling  $b = 1/6$  (model is in minutes, so one minute times 1/6 is ten

seconds) and M given by equation 31

$$M = \begin{bmatrix} 0.1968 & -0.0655 & 0.0238 \\ 0.3944 & 0.2921 & -0.0415 \\ 0.3764 & 0.5125 & 0.1111 \end{bmatrix} \quad (31)$$

The cost function J is given by equation 32

$$J = \min \sum_j^N ((h_j - h_s)^T Q (h_j - h_s) + (u_j - u_s)^T W (u_j - u_s) + \Delta u_j^T R \Delta u_j) \quad (32)$$

where N is the number of time steps in the horizon and  $h_j - h_s$ ,  $u_j - u_s$  and  $\Delta u_j$  is given by equations 33 to 35. The flow setpoint  $u_s$  is in the MPC given as the lowest of the individual flow's setpoints. If it were possible to set several flow setpoints, eg.  $q_{1,s} = 4500$  and  $q_{3,s} = 3000$ , then the solution would not be unique. The weights Q, W and R are given in equations 36 to 38 and were found by trial and error, so the MPC would not change the flows much faster or slower than with SRC.

$$h_j - h_s = \begin{bmatrix} h_{1,j} - h_{1,s} \\ h_{2,j} - h_{2,s} \\ h_{3,j} - h_{3,s} \end{bmatrix} \quad (33)$$

$$u_j - u_s = \begin{bmatrix} u_{1,j} - u_s \\ u_{2,j} - u_s \\ u_{3,j} - u_s \end{bmatrix} \quad (34)$$

$$\Delta u_j = \begin{bmatrix} u_{1,j} - u_{1,j-1} \\ u_{2,j} - u_{2,j-1} \\ u_{3,j} - u_{3,j-1} \end{bmatrix} \quad (35)$$

$$Q = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix} \quad (36)$$

$$W = \begin{bmatrix} 1/10000 & 0 & 0 & 0 \\ 0 & 1/10000 & 0 & 0 \\ 0 & 0 & 1/10000 & 0 \\ 0 & 0 & 0 & 1/10000 \end{bmatrix} \quad (37)$$

$$R = \begin{bmatrix} 1/100 & 0 & 0 & 0 \\ 0 & 1/100 & 0 & 0 \\ 0 & 0 & 1/100 & 0 \\ 0 & 0 & 0 & 1/100 \end{bmatrix} \quad (38)$$

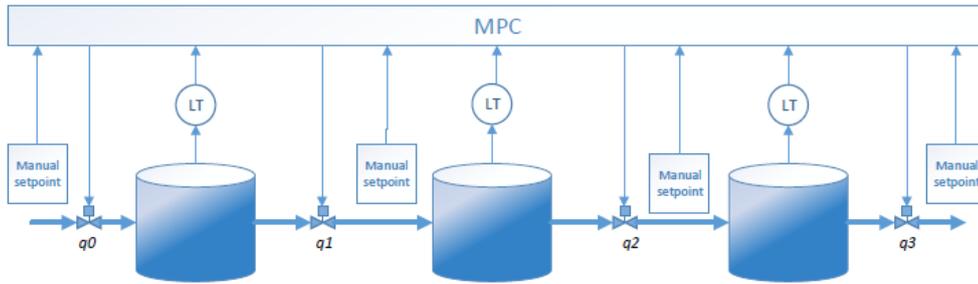


Figure 12: Control structure with MPC.

# Chapter 4

## Simulation results

Simulations was here conducted for disturbances on  $q_1$  and  $q_3$ , to see how the system responds to a disturbance at the inlet and outlet and compare them.

### 4.1 Split range control

Figure 13 shows the response for a change of  $q_1$  from 4500 kg/min to 2250 kg/min (manual setpoint on  $q_1$  sat to 2250 kg/min to simulate that a plug has formed in the pipe) without bias updates. The delay that can be seen on  $q_0$  is the time it takes for the controller to unwind both outputs (first  $q_1$ , then unwind  $q_0$ ).

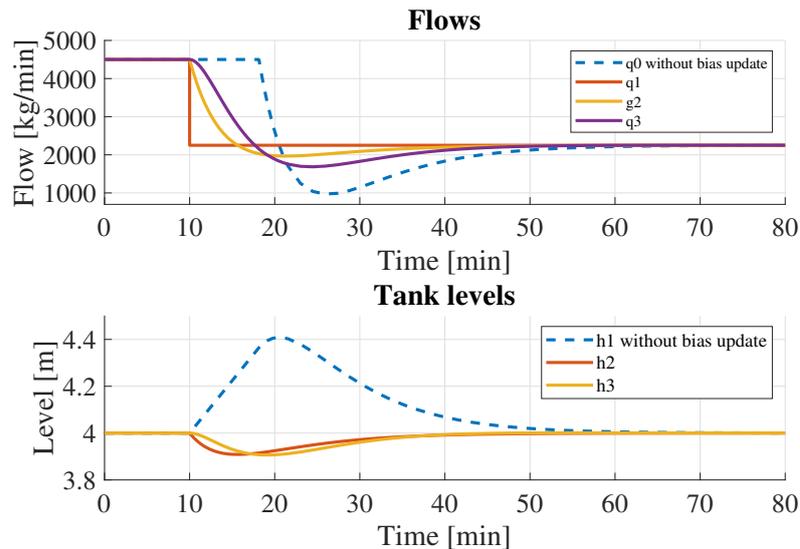


Figure 13: Simulation with step change on  $q_1$ , without bias update

Figure 14 shows a disturbance on  $q_3$  (manual setpoint change), with the flow changed from 4500 kg/min to 3500 kg/min. Here it is observed that there is a

much longer time delay, because all the controllers need to unwind, and thus a smaller disturbance was implemented here than in figure 13. Still, because of the long time delay the controllers need to use the MVs more, as seen by comparing figure 13 and 14.

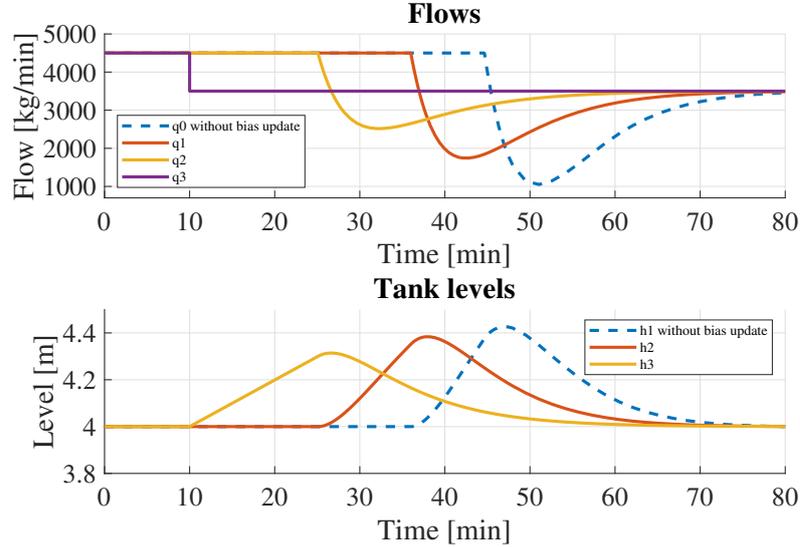


Figure 14: Simulation with step change on  $q_3$ , without bias update

## 4.2 Split range control with bias update

A disturbance on  $q_1$  from 4500 kg/min to 2250 kg/min was conducted and is shown in figure 15 with bias updates, and plotted together with the flows from figure 13. The bias update remove some of the time delay before the controller change  $q_0$ , but not all since the controller still need to unwind the output to  $q_1$ . The flows  $q_2$  and  $q_3$  are the same with our without bias updates for this disturbance.

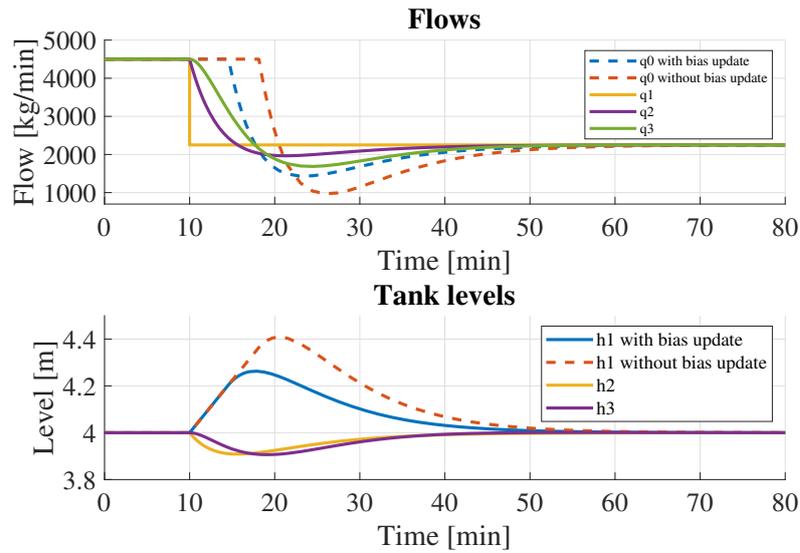


Figure 15: Simulation with step change on  $q_1$ , with and without bias updates

Figure 16 shows a disturbance on  $q_3$  (manual setpoint change) with the flow changed from 4500 kg/min to 3500 kg/min, same case as in figure 14, except that bias update are used here. The "flattening" of the curves for  $q_1$  and  $q_2$  is happening because of the bias is updated continuously.

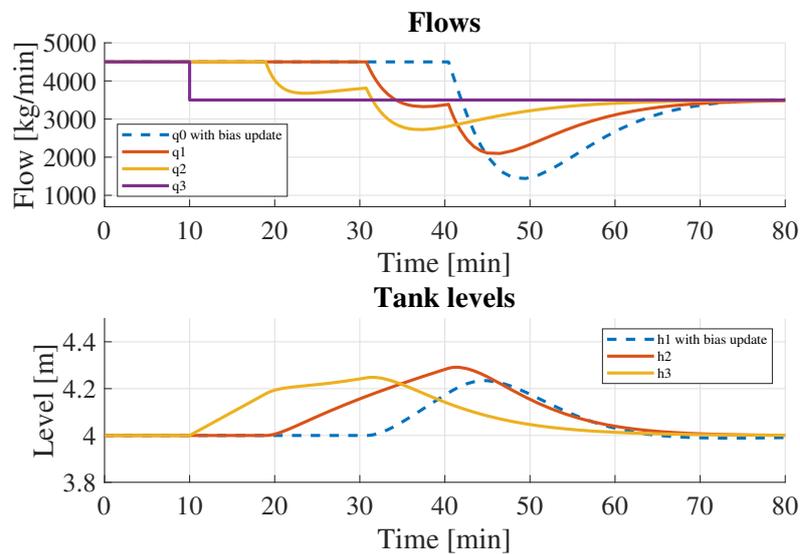


Figure 16: Simulation with step change on  $q_3$ , with bias updates

### 4.3 Model predictive control

A simulation with the same disturbance on  $q_3$  as in figure 16 was also done for the MPC. The setpoint (in the cost function) for the flows was also changed, to simulate that a plug had formed in the pipe. The resulting graphs is plotted together with the solution for SRC with bias updates in figure 17. It's clearly seen that because the MPC doesn't need to wait for the MVs to unwind it can manipulate the other flows immediately. A better graph of the MPC solution is shown in figure 18.

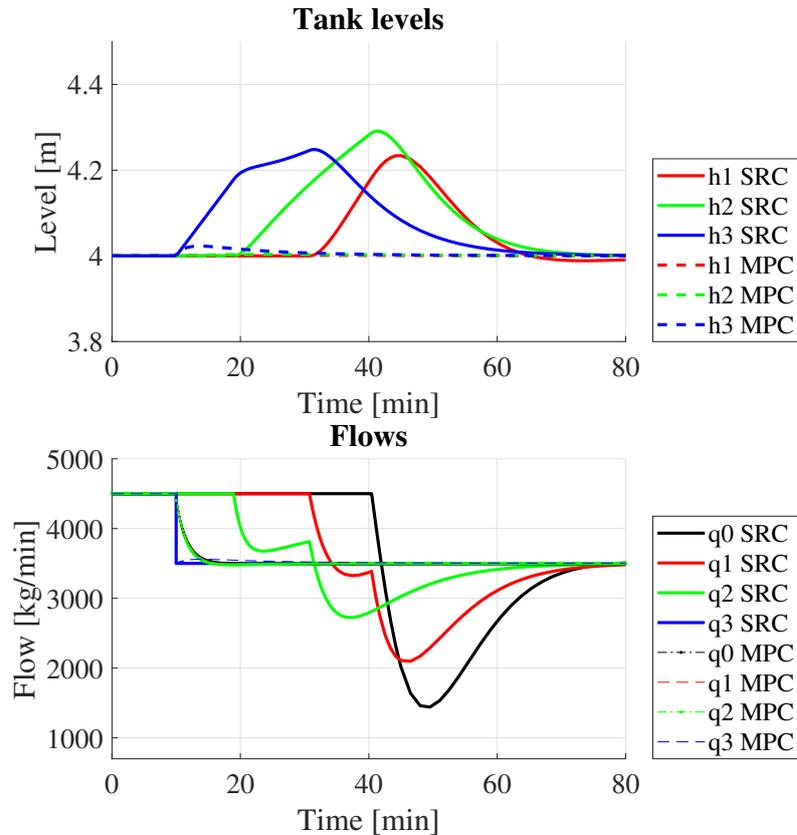


Figure 17: Comparing simulation of MPC and SRC (with bias updates) with a step change on  $q_3$ . Both the actual flow and the flow setpoint where changed in the simulation with MPC.

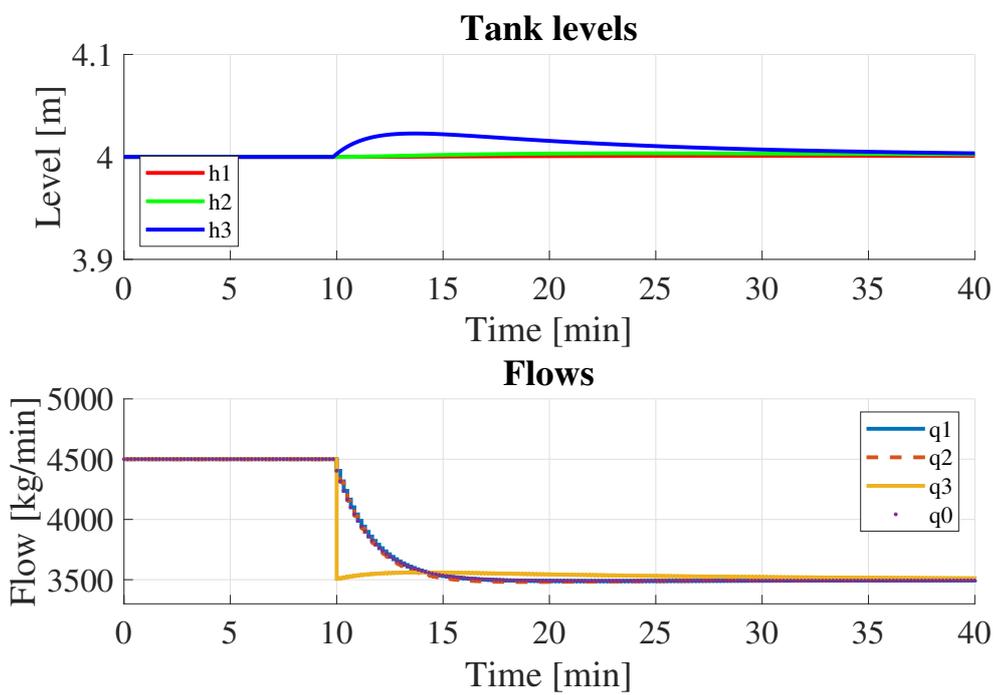


Figure 18: Close-up of the MPC solution in figure 17. Both the actual flow and the flow setpoint were changed for  $q_3$ .

## Chapter 5

# Discussion

Simulations were here conducted for a disturbance on a flow at the inlet and one at the outlet of the process, where use of split range control and model predictive control were compared. It is shown that a disturbance at the inlet have a very different impact on the process than a disturbance at the outlet, because of the windup in the split range controllers. A possible fix to this could be to not update the bias on the output, but rather the internal signal  $v$  in the controllers.

Since the process studied in this report contained fairly simple tanks, the model was easily derived. If the process were to consist of different equipment where a lot more dynamics come into play, then both the model and MPC (with satisfactory tuning) could be much more difficult to obtain.

## Chapter 6

# Conclusion and future work

Inventory or material balance control is important in the process industry to ensure that the mass balance for a plant and the process units in it are fulfilled, both dynamically and at steady-state. In every flow network there needs to be a throughput manipulator (TPM) that sets the production through the system. For optimal operation, the TPM should be placed at the bottleneck in the plant to maintain optimal operation.

The aim of this project was to design a supervisory control layer that could change the location of the TPM when the bottleneck of the process changes when different disturbances occur. This was done by use of split range control (SRC) and model predictive control (MPC). It is shown that the control structure with SRC have large delays because of the windup in the controllers when switching between different MVs. This is not the case for the MPC, because it uses the manipulated variables freely without have to "see" a change in the levels beforehand. In the cases studied here, supervisory control with MPC seems to give the best solution, because of the low deviations in the levels when disturbances occur.

A possible fix for the control structure with SRC, something that would require further work, is to not update the bias on the output from the SRC, but rather the internal signal  $v$ . This might again make the system too complex, or so much logic is required that it becomes a mess.

# Bibliography

- [1] Elvira Marie B. Aske and Sigurd Skogestad. “Consistent Inventory Control”. In: *Industrial & Engineering Chemistry Research* 48.24 (2009), pp. 10892–10902. DOI: 10.1021/ie801603j.
- [2] Elvira Marie B. Aske, Sigurd Skogestad, and Stig Strand. “THROUGHPUT MAXIMIZATION BY IMPROVED BOTTLENECK CONTROL”. In: *IFAC Proceedings Volumes* 40.5 (2007). 8th IFAC Symposium on Dynamics and Control of Process Systems, pp. 63–68. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20070606-3-MX-2915.00009>.
- [3] D.Q. Mayne et al. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814. ISSN: 0005-1098. DOI: [https://doi.org/10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9).
- [4] Adriana Reyes-Lúa and Sigurd Skogestad. “Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures”. In: *Industrial & Engineering Chemistry Research* 0.0 (0), null. DOI: 10.1021/acs.iecr.9b04511.
- [5] Adriana Reyes-Lúa, Cristina Zotică, and Sigurd Skogestad. “Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control”. In: *IFAC-PapersOnLine* 51.18 (2018). 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018, pp. 440–445. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2018.09.340>. URL: <http://www.sciencedirect.com/science/article/pii/S2405896318320238>.
- [6] Adriana Reyes-Lúa et al. “Systematic Design of Split Range Controllers”. In: *IFAC-PapersOnLine* 52.1 (2019). 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS

2019, pp. 898–903. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2019.06.176>.

- [7] Johannes Jäschke. Norwegian University of Science and Technology. *Lecture notes in model predictive control*. Unpublished. 2019.
- [8] Sigurd Skogestad. Norwegian University of Science and Technology. Lecture notes in advanced process control. *kyb2-advanced-control*. Retrieved from <https://folk.ntnu.no/skoge/vgprosessregulering/lectures/kyb-course/> on 25.11.19. 2019.
- [9] Sigurd Skogestad. “Control structure design for complete chemical plants”. In: *Computers & Chemical Engineering* 28.1 (2004). Escape 12, pp. 219–234. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2003.08.002>.
- [10] Sigurd Skogestad. “Economic Plantwide Control”. In: *Plantwide control: recent developments and applications*. Ed. by Gade Pandu Rangaiah and Vinay Kariwala. Chichester, West Sussex: Wiley, 2012. Chap. 11, pp. 229–251. ISBN: 9780470980149.
- [11] Sigurd Skogestad and Chriss Grimholt. “The SIMC Method for Smooth PID Controller Tuning”. eng. In: *PID Control in the Third Millennium: Lessons Learned and New Approaches*. 2012th ed. Advances in Industrial Control. London: Springer London, 2012, pp. 147–175. ISBN: 9781447124245.

# Appendix A

## Comments to MATLAB files

Here are some comments/description of the Matlab files that will be delivered together with the report.

The files that will be delivered are:

1. Project\_three\_tanks.slx : The Simulink file. The level controllers have been put in their own sub-system, to avoid having everything on one "page".
2. Project\_3\_tanks.m : The main file to start the SRC-simulation.
3. SysThreeTanks.m : The model equations for the process, put in the "Interpreted MATLAB Function" block in Simulink.
4. SRC\_withbias.m : Split range block function used in Simulink.
5. MPC.m : The script to run for the MPC simulations. This became very hard-coded as I recently took a module-course about MPC where I also hard-coded a lot, so I just continued from there. I've tried my best to comment in the file itself, but I can see that it would require a lot of reading and studying for anyone else than me.
6. Plant\_three\_tanks : The function that represents the "real process" for the MPC simulation.

## Appendix B

# Tuning of level controller

Here the tuning parameters are obtained by deriving the transfer function for a tank.

Starting from the general differential equation for a tank

$$\frac{dh_i}{dt} = \frac{1}{\rho A_i}(q_{i-1} - q_i) \quad (39)$$

At the nominal point we have

$$\rho A_i \frac{dh_i}{dt} = \bar{q}_{i-1} - \bar{q}_i = 0 \quad (40)$$

where the bar denotes the nominal point variable (constant). We now define the deviation variable  $q'_i$  and  $h'_i$  in equation 41 and 42

$$q'_i = q_i - \bar{q}_i \quad (41)$$

$$h'_i = h_i - \bar{h}_i \quad (42)$$

and inserting these into equation 39 and some rearranging yields equation 43

$$\rho A_i \frac{dh'_i}{dt} = q'_{i-1} - q'_i \quad (43)$$

This is turned into the Laplace-domain function in equation 44

$$\rho A_i s H'_i(s) = Q'_{i-1}(s) - Q'_i(s) \quad (44)$$

where  $s$  is the Laplace variable and  $H$  and  $Q$  are the transformed functions.

We now define the function  $g_1(s)$  in equation 45

$$g_1(s) = \frac{H'_i(s)}{Q'_{i-1}(s)} = \frac{1}{\rho A s} \quad (45)$$

which is an integrating process (equation 19). The plant gain  $k'$  is then given in equation 46

$$k'_1 = \frac{1}{\rho A} = \frac{1}{1000 \text{ kg/m}^3 \cdot 55.2655 \text{ m}^2} = 1.81 \times 10^{-5} \text{ m/kg} \quad (46)$$

The gain for  $g_2(s)$  defined in equation 47

$$g_2(s) = \frac{H'_i(s)}{-Q'_{i-1}(s)} = \frac{-1}{\rho A s} \quad (47)$$

is then

$$k'_2 = \frac{-1}{\rho A} = \frac{-1}{1000 \text{ kg/m}^3 \cdot 55.2655 \text{ m}^2} = -1.81 \times 10^{-5} \text{ m/kg} \quad (48)$$

We then see that the gains are equal in magnitude but with different sign, which makes sense since the outlet and inlet pipes of the tank are equal in size.

The tuning parameters for the controllers in figure 7 is then given in equation 49 and 50

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta} = -1.84 \times 10^4 \quad (49)$$

$$\tau_I = 4(\tau_c + \theta) = 12 \text{ min} \quad (50)$$

with  $\theta = 0$  and chosen  $\tau_c = 3$  minutes.

# Appendix C

## Detailed Simulink model

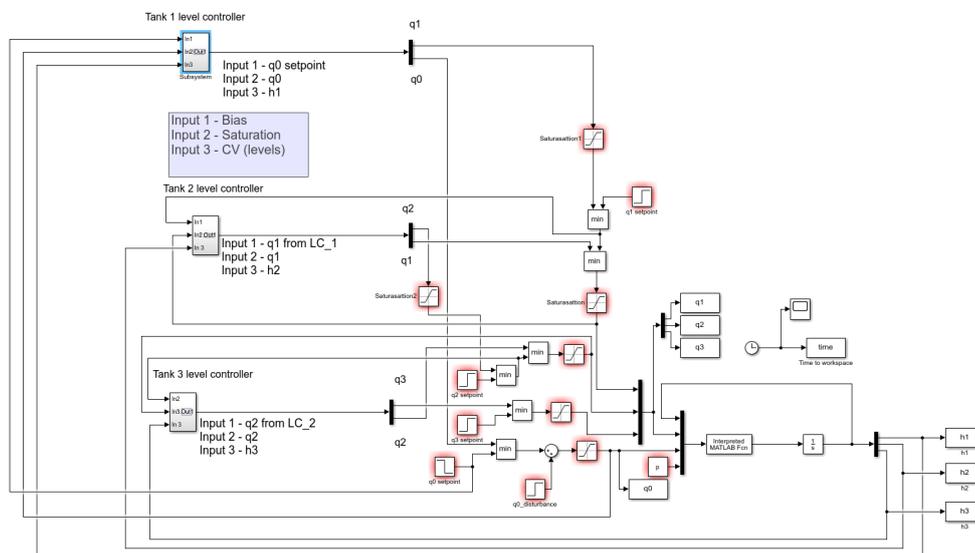


Figure 19: Process in Simulink

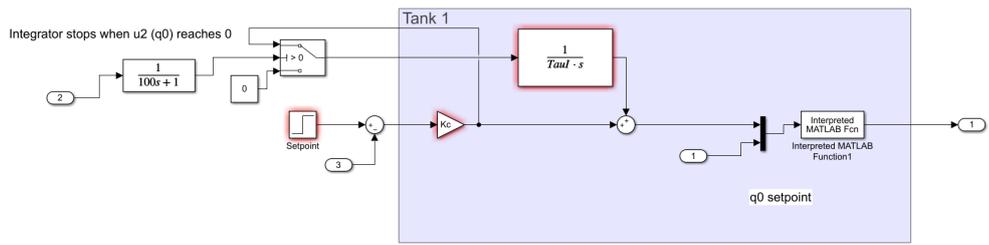


Figure 20: Level controller for tank 1