Master’s Thesis in System and Control Engineering 2016

Jianfeng Zhang

Modeling, simulation and control of the inflow basin of a wastewater treatment plant

University College of Southeast Norway
Faculty of Technology

University College of Southeast Norway
Faculty of Technology
Department of System and Control Engineering
Kjølnes ring 56
3918 Porsgrunn

http://www.usn.no

Abstract

Inflow basin is an important component for wastewater treatment plant, which can equalize the hydraulic and composition variations of wastewater inflow and stabilize the performance of nutrition and toxicants removal and energy removal in subsequent processes when discharge and component of sewage varies. In this thesis, the simulations with different MPC's parameters setting are designed, run and analyzed. The PI controller is tuned by the Skogestad's tuning method. Both the MPC and PI controller are implemented in a dynamic simulator that is based on a discretized mathematical model of the inflow basin. As the result of well-designed modeling, simulation and control, the MPC with the optimized parameters setting and the Gain schedule of PI controller are verified and established. The simulation data is logged and analyzed graphically and numerically, which shows that the MPC can serve to maintain a good balance between the stability of the basin level[1] and the smoothness of the pumps flow.

Keywords: Inflow basin; MPC; PI controller; modelling and simulation; data analysis.

[1] Basin level means the level of the stored wastewater in inflow basin

Figures and tables

Abbreviation

MPC	Model Predictive Control
CNMPC	Continuous Nonlinear Model Predictive Control
COST	European Co-operation in the field of Scientific and Technical Research
BSM1	Benchmark Simulation Model 1
GS	Gain Scheduling

Nomenclature

Symbol	Unit	Description
d_t	[min]	Time interval of simulation horizon
Kp	[liter/(s • m)]	Proportional gain
Nc		Number of calculated control signals for predictive horizon
N_k	[min]	Length of simulation horizon
Np		Predictive horizon of MPC
P		Weighting of error between level set point and level
R		Weighting of pumps flow change.
Tc	[s]	Time constant
tf	[min]	Finish simulation time
Ti	[s]	Integrating time
T_span	[min]	Simulation horizon
ts	[min]	Initial simulation time
∆ pumps_flow	[liter/s]	Pumps flow change for each time interval

Contents

Preface

This thesis aims to find the control strategy alternatives for the inflow basin of wastewater treatment plant. The inflow measurement from Vækerø and the pumps flow have been logged and provided by VEAS, which need to be utilized comprehensively in this thesis. Knowledge about the control theory should be applied proficiently. Matlab is the main programming language for all of the simulations in this thesis, and needs to be utilized skillfully.

By implementing the different control strategies on the simulator of the inflow basin, one strategy with the optimized control performance on stabilizing the pumps flow is expected to be established.

Here, I would like to thank my supervisor Finn Haugen who gave me this topic, provided me the suggestion and help patiently.

Besides, I feel lucky that, in Norway, I meet some important friends in my life who are always trying to encourage me and help me to open the doors of new life. Here, I want to give my special thanks to Alexander Jonsaas who teaches me snowboard and longboard, Lily Eikehaug who takes me to kickboxing course, Sondre Mogård who is the first Norwegian I talked to and trust so much, Abdalla Habib who I work out with and Peter Alexander Smestad who likes my jokes. Thank you for all.

Porsgrunn, 1. June 2016

Jianfeng Zhang

1 Introduction

1.1 Background

Wastewater treatment plants are operated in dynamic situations where inflow rate varies due to weather condition and industrial and household water consumption. Designed capacity of plants is limited and needs to be optimized and counteract negative effect triggered by variation of inflow rate. Inflow basin is usually designed and built to attenuate the instability of inflow and furthermore helps to meet a balance between utilization percentage and performance of wastewater treatment plants.

Water system closely connected with control technology is more effective to satisfy societal needs. Many works and research have been done in order to improve wastewater treatment performance(van Overloop et al., 2013). This section will mainly feature the control methods and strategies for equalization basins.

Equalization basin can improve the performance of wastewater plants by smoothing the inflow rate, attenuating the variations of inflow concentration and increasing wastewater load, which ultimately refine the effluent quality indexes (Jeppsson and Pons, 2004). Bolmstedt and Olsson (2005) extends the BSM1 to include an equalization basin and implements the different control strategies on the BSM1 based on the COST simulation benchmark (Jeppsson and Pons, 2004). The averaging control is used as the baseline control which utilizes the historical inflow data and provides the effluent at the average outflow rate for the wastewater treatment, as a result, the averaging control decreases the effluent concentration as the inflow rate to the plant is lower and more load can be treated (Bolmstedt, 2004). The feedback control uses two strategies first of which is to control the volume in the basin and second of which is to use the effluent ammonium concentration as a feedback, which causes the sedimentation in the basin and effluent ammonium concentration are both reduced (Bolmstedt, 2004). Early emptying is implemented as another control strategy that aims to empty the basin before the elevated flow arrives. Additional rain events are simulated to testify this control strategy (Bolmstedt, 2004). In Bolmstedt and Olsson (2005)'s paper, the BSM model is extended with the basin which is modelled as an ASM1 reactor and a new settler model with a small cross-sectional area (400m^2) is implemented without other related parameters changed in order to study the effect of storm; the evaluation period is chosen to cover the time before and after a short rain to illustrate the effect of the pre-emptive action and lingering effect of a washout; the information about the elevated flow rates is also used for reserving space of basin to cope with the first flush and calculating the optimal pumps inflow rates to a plant. In addition, the emptying control is evaluated both for a large settler and a small settler with the different preparation periods (1.5h and 9h); the suspend solids are considerably decreased by using the emptying control. However, it is hard to drop the conclusion that a longer preparation period is always better.

MPC is of interest in water system due to its ability of handling multi-variables interaction, constraints, system states and optimization requirements. With a well-developed model, MPC can provide the optimal control option for water system. Nederkoorn et al. (2013) develop the continuous nonlinear MPC and optimized pumps schedule for a hybrid water system in order to minimize the cost due to pumps' energy consumption. In (Nederkoorn et al., 2013)'s work, the water level is kept staying in the predetermined boundaries; the requirements of the pumps operation are developed as the equality and inequality constraints, e.g. capacity, operation order and pumps actions time; the discontinuity of the pumps' features e.g. capacity and jump due to the operating status switch is removed. In addition, the continuous Pumps flow function is substituted in the original model; the CNMPC is formed by integrating the efforts which have been done and an optimal pump schedule is established.

Maestre et al. (2013) propose a way of modeling the weather forecast and use the tree-based MPC to optimize the valuable variables in a drainage system. In his paper, the weather forecast is treated as a disturbance; the model is formed as a disturbance tree; the optimization problem is solved by using dual decomposition method.

Fuzzy logic is suitable for high order and non-linear systems and can be considered as a potential way for equalization basin level control. Malki and Umeh (2000) introduces the fuzzy logic control as an alternative for the level control of a water tank, which can attenuate the overshoot and reduce the settling time effectively.

PI controller is widely used and practical for basin level control and the parameters of PI controllers need to be tuned with proper tuning methods. Wang and Crusca (2002) develops and tunes a relevant PI controller for the water level control.

In this thesis, the alternative control methods and strategies are simulated and analyzed.

1.2 Thesis outline

In the introduction chapter, the research background and outline of the work is presented.

In the control system chapter, the mathematical discretized model of the inflow basin is introduced, the theory and background of the MPC and PI controller is illustrated and a structure chart of the control system is created to show the interaction between the different parts of control system.

In the MPC design and application chapter, the measurement from VEAS is resampled; the simulations with the different MPC setting are implemented and compared in order to identify the optimized parameters for the MPC; the control performance indexes e.g. computing time, variance etc. are logged and presented as part of the judging criteria. The simulations on the different simulation horizons are also implemented to test the overall control performance of MPC.

In the PI controller application chapter, the basin model is further analyzed, the Skogestad's tuning method which is explained in Skogestad (2003)'s paper is introduced and implemented. By reference to Cho and Lazaro (2010)'s work and Sung et al. (2009, p. 225)'s book, a Gain schedule is established as a tuning result for the PI controller and tested in the basin simulator at different time steps.

In the result analysis chapter, the control performance of MPC is discussed and analyzed numerically; several different judging standards are developed both for the pumps flow and basin level. The control performance from VEAS is also illustrated as the baseline for comparison.

In the discussion and conclusion chapter, some critical situations are discussed and analyzed, some general conclusions are made.

In the future work chapter, some suggestions on the future work are presented.

1.3 Contribution

The contribution of this thesis is mainly presented in the MPC design and application chapter, the PI controller chapter and the result analysis chapter and summarized in the discussion and conclusion chapter.

The implementation of a dynamic simulator of the inflow basin in an appropriate programming environment is the general goal for this whole thesis. It is achieved by the modeling, simulation and control as follows:

- Implementation of the discretized inflow basin model and simulation.
- Resampling of the measurement data from VEAS for the MPC and PI controller simulations
- Design and application of the MPC with the optimized MPC parameters setting.
- Design and application of the PI controller with the properly tuned parameters by using the Skogestad's tuning method.
- Comparison and analysis of the MPC and measurement data from VEAS.
- Analysis of the PI controller's control performance on the inflow basin simulator.
- Logging simulations' data.

2 Control system

2.1 Mathematical model

According to the mass balance equation (1), the mass of the stored wastewater in the inflow basin is determined by the wastewater inflow and pumps flow.

$$dm/dt = \dot{m_{inflow}} - \dot{m_{pumps\ flow}} \tag{1}$$

Based on the mathematical calculation (Haugen, 2015), a final equation (2) which shows the relation of the derivative of the basin level and the flow rates is deducted.

$$dh/dt = [1/A(h)] * (F\_inn\_total - F\_pumpe) \tag{2}$$

Where $A(h)$ is the surface area of the stored wastewater; F_inn_total is the total inflow rate to the inflow basin and F_pumpe is the working pumps flow rate.

The basin model needs to be further discretized in order to be simulated for the work in following chapters. The discretized model is shown in equation (3).

$$h_{k+1} = [1/A(h_k)] \cdot \left(F_{inn_{total_k}} - F_{pump_k}\right) \cdot \Delta t + h_k \tag{3}$$

Where h_{k+1} is the basin level at time step k+1; h_k is the basin level at time step k; $F_{inn_{total_k}}$ is the total inflow rate at time step k; F_{pump_k} is the working pumps flow rate at time step k; Δt is the length of time step.

2.2 Control system structure

The total inflow can be calculated from the measurement flow taken from Vækerø, before being used in the simulations of the basin model. The MPC and PI controller are chosen to take in charge of controlling the simulator of the inflow basin. The basin level is the feedback to the PI controller and the only variable the PI controller is designed to control. For the MPC, both of the pumps flow change and basin level are of interest.

The main control system structure is shown in Figure 2.1.

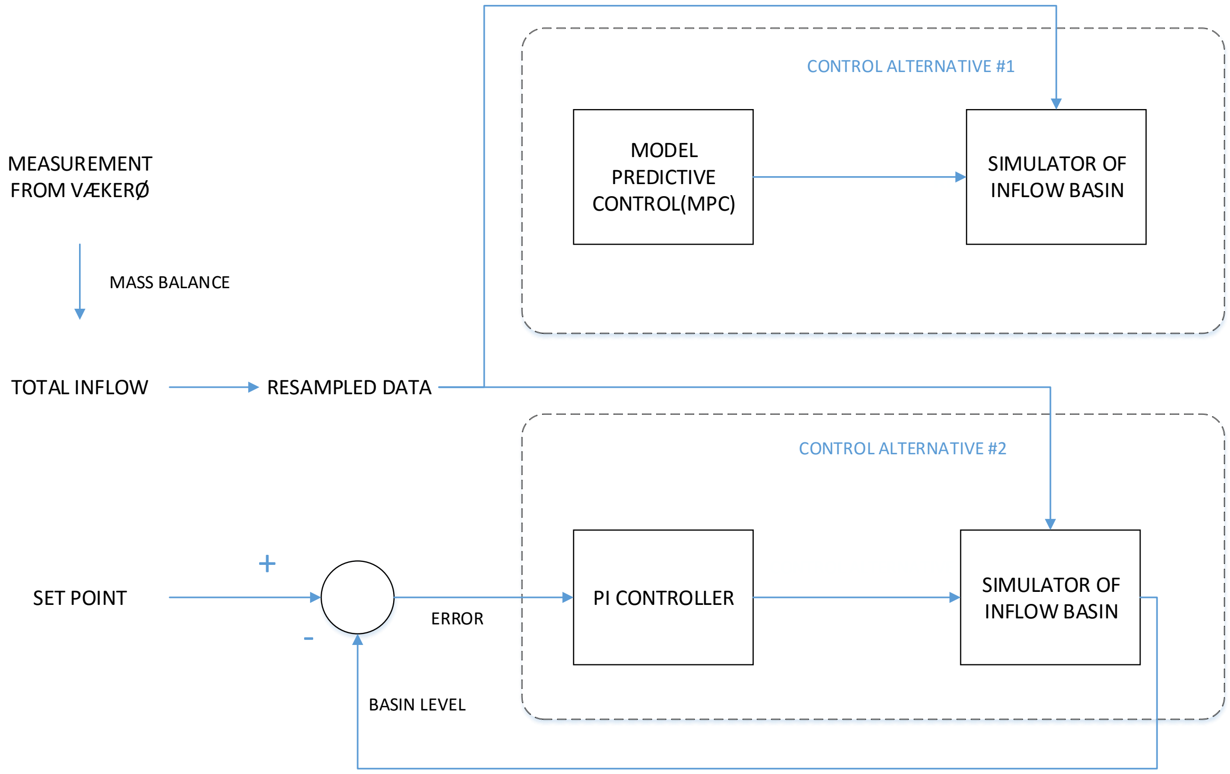

Figure 2.1 Control system structure

2.3 MPC

Model Predictive Control (MPC) can be taken into consideration as a control alternative when the mathematical model of the process is provided. The system response of interest can be acquired when the control inputs are implemented into the process model. The objective function can be customized according to the different practical requirements by mathematically combining the model states, response and control inputs together. The optimized control inputs can be estimated for the predictive horizon. All of the variables of interest should subject to the constraints based on the real condition of the process (Angelo and M.Kramer.(eds), 2012, pp. 185-200).

For the predictive horizon, only the first element from the optimized control inputs sequence will be implemented, based on which, in order to get the optimized control input at each time step of the simulation horizon, the receding horizon needs to be used (Angelo and M.Kramer.(eds), 2012, p. 185)

2.4 PI controller

PI controller is widely used in industry due to its feasibility and simplicity. PI controller is composed of two parts, proportional term P and integral term I.

In frequency domain:

$$u(s) = Kp \cdot e(s) + \frac{Kp}{Ti \cdot s} \cdot e(s) \tag{4}$$

Where P equals to $Kp \cdot e(s)$; I equals to $\frac{Kp}{Ti \cdot s} \cdot e(s)$.

P term is in proportion to error between process response and set point and used to reduce error.

I term will continuously take action as error remains and eliminate the offset between stable process response and set point (Sung et al., 2009, pp. 124-126).

3 MPC design and application

3.1 Measured data

The inflow rate in Vækerø, pumps flow rate and basin level in inflow basin were measured during the period from 1st August to 14th December 2015 by VEAS.

The total inflow rate from the wastewater transport tunnels can be calculated according to the inflow rate in Vækerø, which makes up 65% of the total inflow rate. The time for the wastewater flow to travel from Vækerø to the inflow basin is approximately 0.15 day, which should be treated as a time delay when the inflow measurement is implemented into the basin simulator.

3.2 MPC design

MPC features predicting the futures' control inputs and corresponding plants' response based on the mathematical model developed from the real process. For each sampling instant, predicted control inputs sequence is implemented into the process's model in order to generate the corresponding predicted states and response sequence when the model is simulated. Optimized predictive control can be achieved by minimizing the objective function that is built by mathematically structuring predicted states, response and control inputs sequence together based on specific control requirements. In addition, all the calculation should subject to the inputs and states constraints that can be either equality constraints or inequality ones. Only the first element from control inputs sequence will be implemented into the model to get the calculated states that will be used to substitute the initial states for the predictive horizon of next sampling time, the length of predictive horizon should be kept unchanged in the meanwhile. This procedure will be repeated until it covers the whole simulation horizon, which is known as horizon strategy. Due to the limitation of predictive horizon and the variation of process dynamics, receding horizon strategy is always implemented in MPC (Angelo and M.Kramer.(eds), 2012, pp. 185-120).

In this thesis, the control purpose is to find a balance between the smoothness of the pumps flow and the stability of the basin level, based on which, a linear quadratic function is chosen as the objective function that is shown in equation (5).

$$J_{min} = \sum_{i=1}^{Np} \frac{1}{2}(h_{basin}(i) - ref) \cdot P \cdot (h_{basin}(i) - ref) + \sum_{i=1}^{Np} \frac{1}{2} \Delta pump_{flow}(i) \cdot R \cdot \Delta pump\_flow(i) \tag{5}$$

This objective function is composed of two parts, the first part is computed from the difference between basin level and level set point $(h_{basin}(i) - ref)$and the second part is from the pumps flow rate change$\Delta pump_flow(i)$.

According to VEAS requirements, the pumps flow rate change should be kept less than 300 liters per 15 minutes. The inequality constraints are designed as:

$$-20 \cdot d\_t(liters/s) \leq \Delta pump\_flow(i) \leq 20 \cdot d\_t(liter/s) \tag{6}$$

Where d_t is the length of time step.

The objective function uses the weightings in order to balance two separate goals, e.g. tracking the basin level reference and smoothing the pumps flow. R and P are used as the weighting numbers herein, of which the final values are decided based on the performance of simulations with the different setting. It is important to note that tracking the set point of the basin level and the change of the pumps flow per time step is of significantly different orders of magnitude. According to the practical need from the wastewater treatment plant, keeping the pumps flow as smooth as possible should be prioritized.

The last MPC parameters that need to be decided are two horizon parameters. The first is the length of prediction horizon Np that is also the number of the time steps through which both tracking error of the basin level and the pumps flow change is computed in the objective function. Moreover, the prediction horizon can be viewed as the length of time through which the MPC algorithm predicts the future behavior of the system under control for each time step. In this work, the predictive horizon Np =10 and time step d_t =21.6 minutes are chosen for the MPC, of which the purpose is to make fully utilization of the time delay of the inflow measurement from Vækerø, which is 216 minutes. The second parameter Nc is the number of values used as the control inputs.

In order to find the optimized parameters setting for MPC, 6 different simulations are finished. Setting for each simulation is listed in Table 3.1.

Table 3.1 The parameters setting for MPC

	d_t	Np	Nc	P	R	N_k	T_delay	Resampled data [2]
#1 (baseline)	21.6mins	10	10	1	1	600	216mins	NO
#2 (tuning Nc)	21.6mins	10	2	1	1	600	216mins	NO
#3 (resampled data)	21.6mins	10	2	1	1	600	216mins	YES
#4 (R=10)	21.6mins	10	2	1	10	600	216mins	YES
# 5 (R=100)	21.6mins	10	2	1	100	600	216mins	YES
# 6 (R=1000)	21.6mins	10	2	1	1000	600	216mins	YES

3.2.1 Simulation #1 (baseline)

Simulation #1 (Nc =Np=10, d_t =21.6 mins, P=R=1) is set as the baseline for the other following simulations. There is a 1.6 minutes' time difference between the time step d_t and the sampling time (20 mins) used by VEAS. However, in this simulation, the total inflow for each time step is assumed equal as the calculated inflow for each sampling time.

[2] Resampled data means using the method of interpolation to estimate the total inflow inside each sampling time.

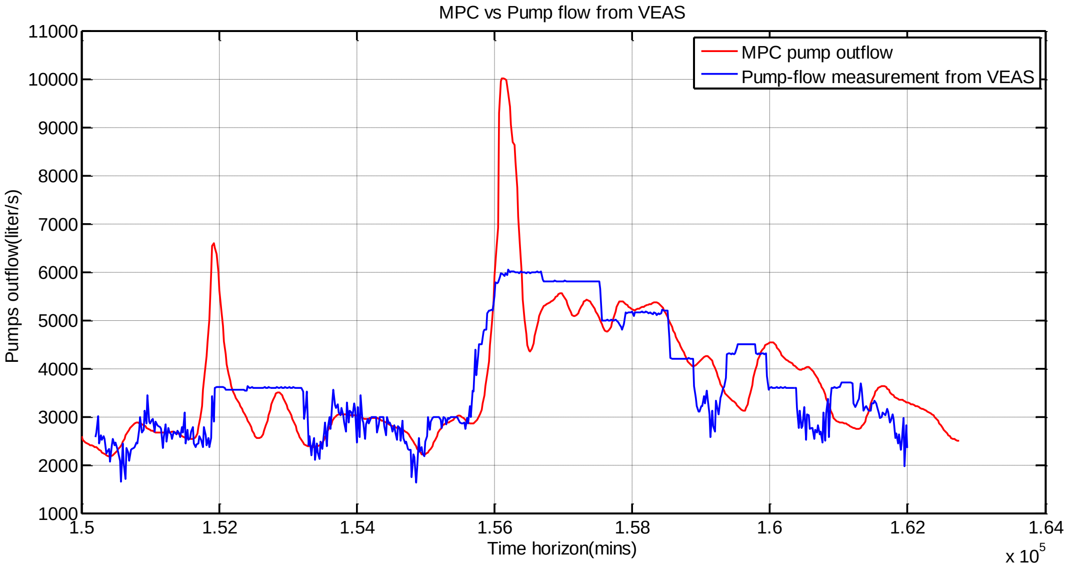

Figure 3.1 MPC vs the pumps flow from VEAS, Np=Nc=10, P=R=1, no resampled measurement data

Table 3.2 The numeric result of simulation #1 for the pumps flow

Computing time(s)	**Pumps flow variance**[3]	**Total variation**[4]
6.2755e+03	1.9218e+04	4.8499e+04

[3] Pumps flow variance is the variance of pumps flow rate change at each time step.

[4] Total variation is the sum of the absolute value of pumps flow rate change at each time step on the simulation horizon.

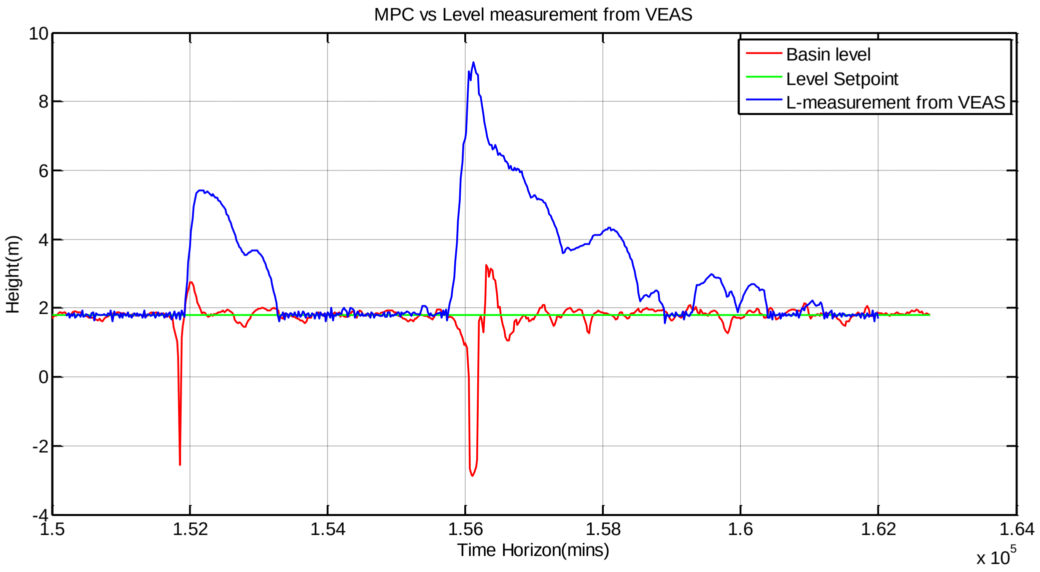

Figure 3.2 MPC vs the level measurement from VEAS, Np=Nc=10, P=R=1, no resampled measurement data

Table 3.3 The numeric result of simulation #1 for the basin level

Computing time (s)	**Highest level (m)**	**Lowest level (m)**	**Violating time[5] (mins)**
6.2755e+03	3.5200	-2.8870	63 • 21.6

In Figure 3.1, the red curve represents the pumps flow controlled by MPC and the blue curve represents the pumps flow controlled and logged by VEAS. It can be seen that the red curve appears to be smoother when compared with the blue one, which means the smoothness of the pumps flow is improved by MPC. In Figure 3.2, the red curve represents the basin level controlled by MPC and the blue curve represents the basin level controlled and logged by VEAS. It can be seen that the red curve is closer to the green curve that represents the level set point to some extent when compared with the blue curve, which means the stability of the basin level is improved by MPC.

In Table 3.2 and Table 3.3, the computing time is 6.2755e+03s, which shows the computing process is time-consuming to some extent. The reason is that for each predictive horizon Np, Matlab needs to find N_c=Np optimized control inputs. A bigger Nc will complicate the process of optimization and cost much more computing time for MPC to find the optimal control inputs.

[5] Violating time is the time when the basin level is higher than the upper limit (2.1m) or lower than the lower limit (1.5m).

As a result, decreasing Nc, in some cases, will affect the control performance positively (Angelo and M.Kramer.(eds), 2012, p. 194). In the Simulation #2(tuning Nc), the improvement of the MPC performance and the reduction of the computing time will be evaluated to verify the feasibility of reducing Nc.

The inflow measurement from Vækerø can be taken into the basin model in order to predict the control inputs and model response as soon as being measured. According to the time delay that is explained in 3.1, the pumps flow calculated by MPC is supposed to react ahead of VEAS. However, the result only meets this expectation partly, instead of being ahead; the pumps flow from MPC becomes to react slower than the pumps flow from VEAS after the time step around 4000 mins. The reason will be analyzed and discussed in the Simulation #3(resampled data).

3.2.2 Simulation #2(tuning Nc)

In this simulation, Nc is reset as 2, which means for each predictive horizon, only 2 optimized pumps flow will be calculated. Therefore, instead of having optimized pumps flow for each time interval, the basin model will implement each pump flow for half-length of the predictive horizon. This approach is called Nc tuning(Angelo and M.Kramer.(eds), 2012, p. 194). The purpose of tuning Nc is to not only reduce the computing time but improve the smoothness of the pumps flow as well.

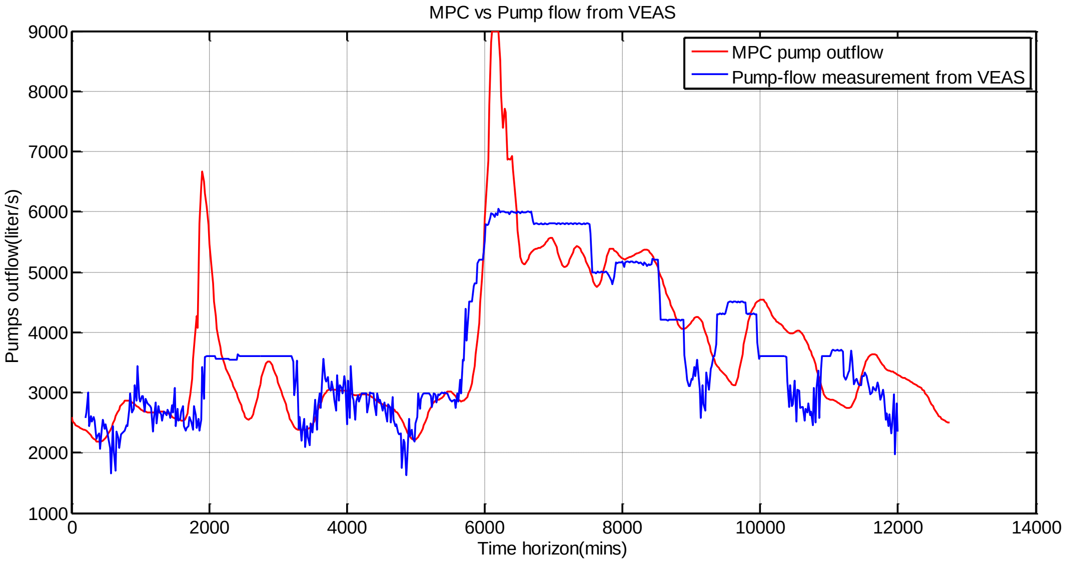

Figure 3.3 MPC vs pumps flow from VEAS, Np=10, Nc=2, P=R=1, no resampled measurement data

Table 3.4 The numeric result of simulation #2 for the pumps flow

Computing time(s)	Pumps flow variance	Total variation
1223.294942	1.8195e+04	4.6727e+04

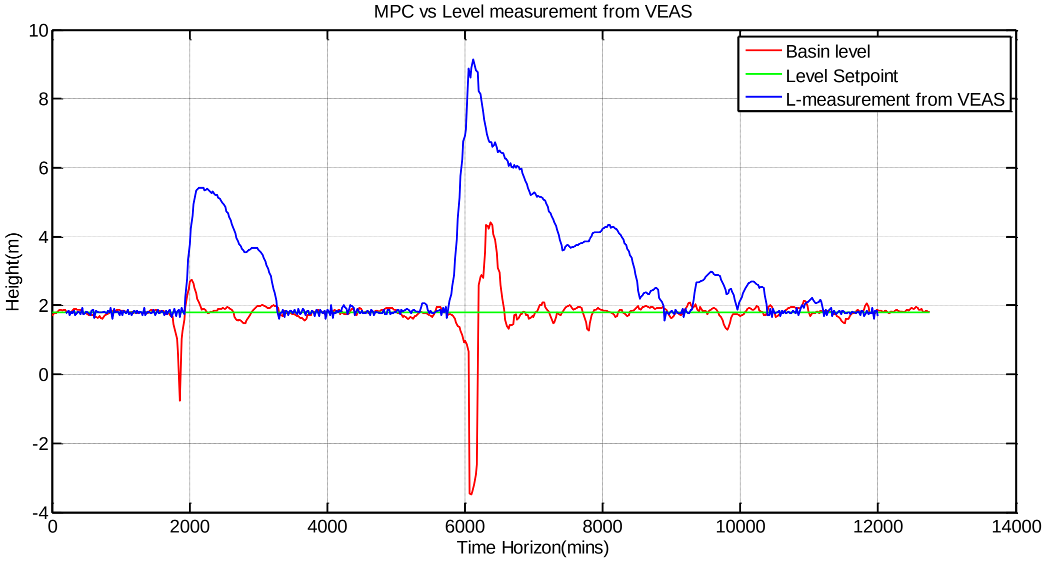

Figure 3.4 MPC vs the level measurement from VEAS, Np=10, Nc=2, P=R=1, no resampled measurement data

Table 3.5 The numeric result of simulation #2 for the basin level

Computing time (s)	Highest level (m)	Lowest level (m)	Violating time (mins)
1223.294942	4.4237	-3.4774	66 • 21.6

As can be seen in Figure 3.3, the curve of the pumps flow controlled by MPC follows the similar shape of the corresponding one in Figure 3.1. In Table 3.4, after reducing Nc, the pumps flow variance decreases from 1.9218e+04 to 1.8195e+04 and the total variation from 4.8499e+04 to 4.6727e+04, which means that the smoothness of the pumps flow is improved. Besides, the computing time is shortened to 5 times less than in the simulation #1 approximately, which drops from 6.2755e+03 s to 1223.294942 s. In Figure 3.4, the curve of the basin level controlled by MPC shows the similarity with the corresponding one in Figure 3.2.

In Table 3.5, the highest level is 0.9037m higher and the lowest level is 0.5904m lower than the corresponding ones in Table 3.3. The reason is that the improvement of the smoothness of the pumps flow is achieved at the cost of sacrificing the stability of the basin level.

In Figure 3.3, the lag phenomenon[6] of the pumps flow described in Simulation #1 (baseline) still remains.

3.2.3 Simulation #3(resampled data)

In order to solve the lag phenomenon of the pumps flow which is illustrated in Figure 3.3. The measurement from Vækerø should be resampled. The reason is that the error derived from approximating the total inflow for each time step equal to the measurement flow of each sampling time is accumulated when the simulation is running. In this work, the sampling time is further divided into 20 parts evenly; after resampling, the total inflow for each 1minute can be estimated. In this simulation, the total inflow at each time step (21.6mins) is approximated to equal to the total inflow for each 22 minutes which is estimated by using the method of interpolation. By doing this resampling, the error from the process of approximation should be decreased and the lag phenomenon is expected to be solved.

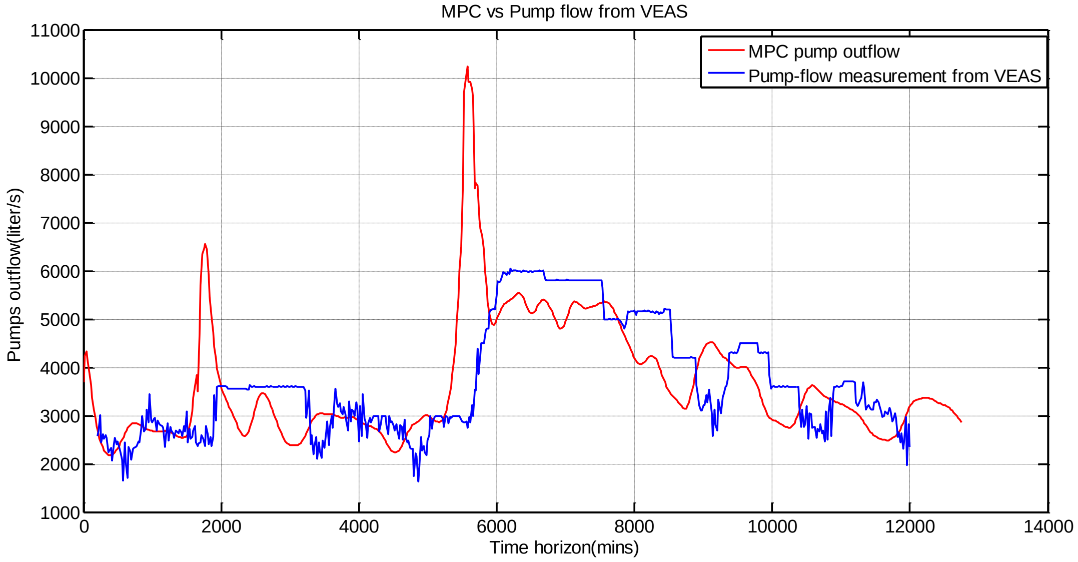

Figure 3.5 MPC vs the pumps flow from VEAS, Np=10, Nc=2, P=R=1, resampled measurement data

Table 3.6 The numeric result of simulation #3 for the pumps flow

Computing time(s)	**Pumps flow variance**	**Total variance**
939.691741	1.1438e+04	4.1856e+04

[6] Lag phenomenon means that the pumps flow from MPC becomes to react slower than the pumps flow from VEAS

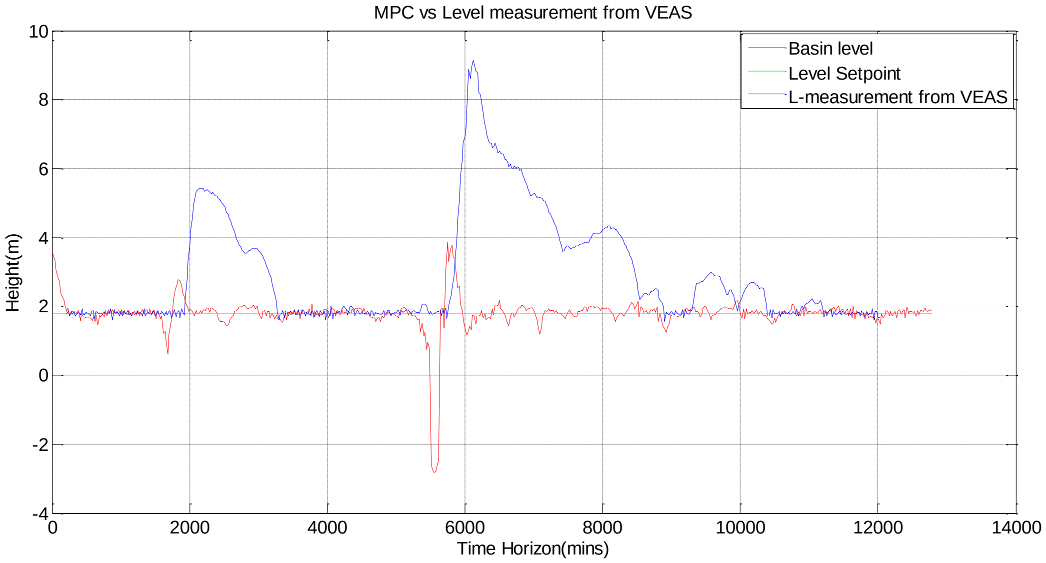

Figure 3.6 MPC vs the level measurement from VEAS, Np=10, Nc=2, P=R=1, resampled measurement data

Table 3.7 The numeric result of simulation #3 for the basin level

Computing time (s)	**Highest level (m)**	**Lowest level (m)**	**Violating time (mins)**
939.691741	3.8709	-2.8371	74 • 21.6

In Figure 3.5, it shows that the pumps flow controlled by MPC appears to react ahead of VEAS pumps flow; as a result, the basin level controlled by MPC is adjusted ahead of VEAS level control, which is shown in Figure 3.6. In Table 3.6, the pumps variance and the total variation decrease by 0.6757e+04 and 0.4871e+05 correspondingly when compared with Table 3.4.

In Table 3.7, the highest level is 0.5528m lower and the lowest level is 0.6403m higher than the corresponding ones in Table 3.5.

3.2.4 Simulation # 4(R=10)

P and R are the weighting values in the objective function (5) , which need to be tuned to meet those practical requirements. In this thesis, the smoothness of pumps flow is of most interest. It means R should be set to be bigger than P if more smoothness of the pumps flow is expected to appear. However, it does not mean that R should be as big as possible; it is because the stability of the basin level will be negatively influenced when R keeps increasing. Therefore, finding a proper value for R to balance the basin level and pumps flow should be taken into consideration. In this simulation, R is chosen to be 10.

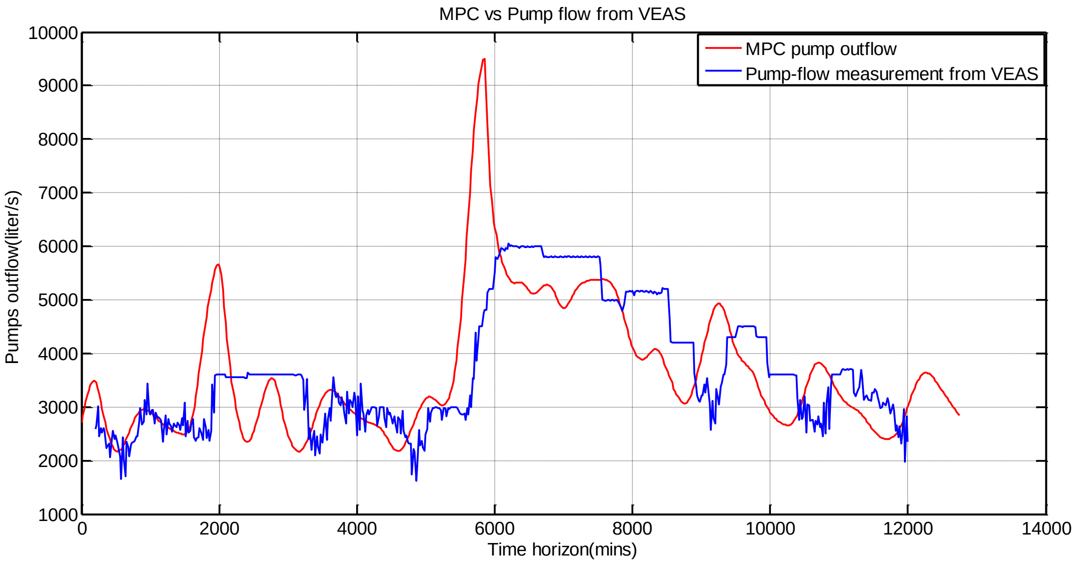

Figure 3.7 MPC vs the pumps flow from VEAS, Np=10, Nc=2, P=1, R=10, resampled measurement data

Table 3.8 The numeric result of simulation #4 for the pumps flow

Computing time(s)	**Pumps flow variance**	**Total variance**
704.240325	1.0079e+04	3.9375e+04

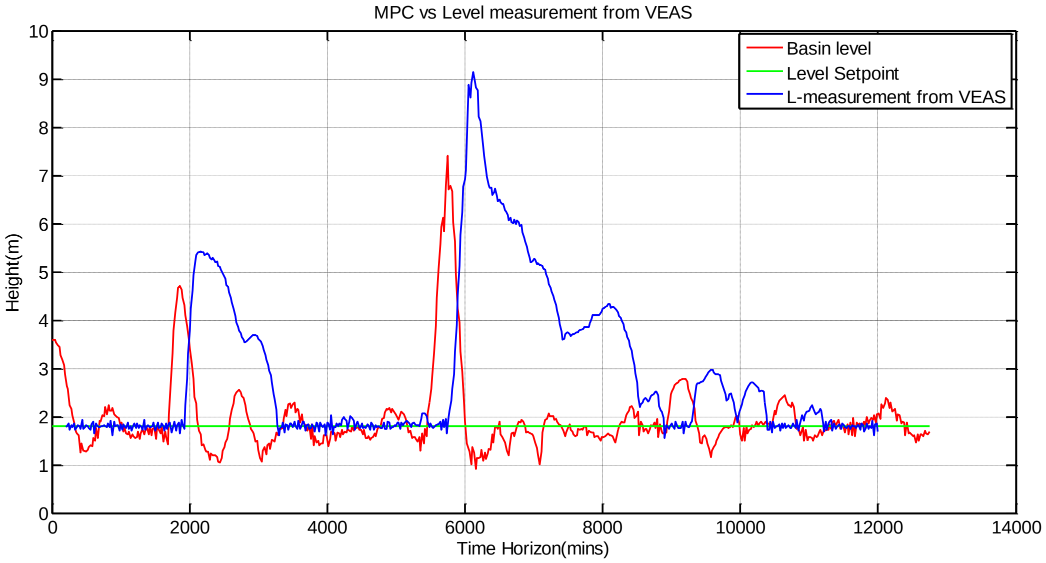

Figure 3.8 MPC vs the level measurement from VEAS, Np=10, Nc=2, P=1, R=10, resampled measurement data

Table 3.9 The numeric result of simulation #4 for the basin level

Computing time (s)	Highest level (m)	Lowest level (m)	Violating time (mins)
704.240325	7.4036	0.9128	224 • 21.6

From Figure 3.7 ,it can be seen that the fluctuation of the pumps flow becomes mild when compared with Figure 3.5 in Simulation #3(resampled data).

In Table 3.8, the pumps flow variance is decreased from 1.1438e+04 to 1.0079e+04 and the total variation is decreased from 4.1856e+04 to 3.9375e+04. The smoothness of the pumps flow is further improved.

However, in Figure 3.8, the stability of the basin level is negatively affected when R is increased; it is because the target of MPC leans to maintaining more stable pumps flow. As can be seen from Table 3.9, the highest level is 7.4036m which is much bigger than 3.8709m and the violating time becomes approximately 3 times longer than 74 • 21.6 mins in Simulation #3(resampled data) .

3.2.5 Simulation #5 (R=100)

In order to observe the influence of increasing weighting R on both the pumps flow and basin level, in this simulation, R is reset as 100.

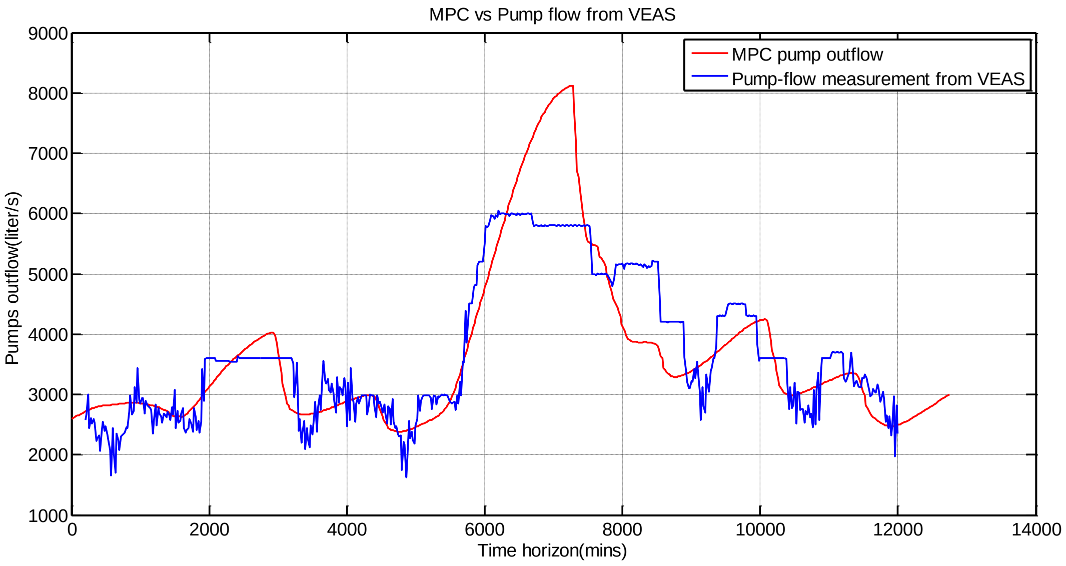

Figure 3.9 MPC vs the pumps flow from VEAS, Np=10, Nc=2, P=1, R=100, resampled measurement data

Table 3.10 The numeric result of simulation #5 for the pumps flow

Computing time(s)	**Pumps flow variance**	**Total variation**
854.440728	3.3674e+03	1.8723e+04

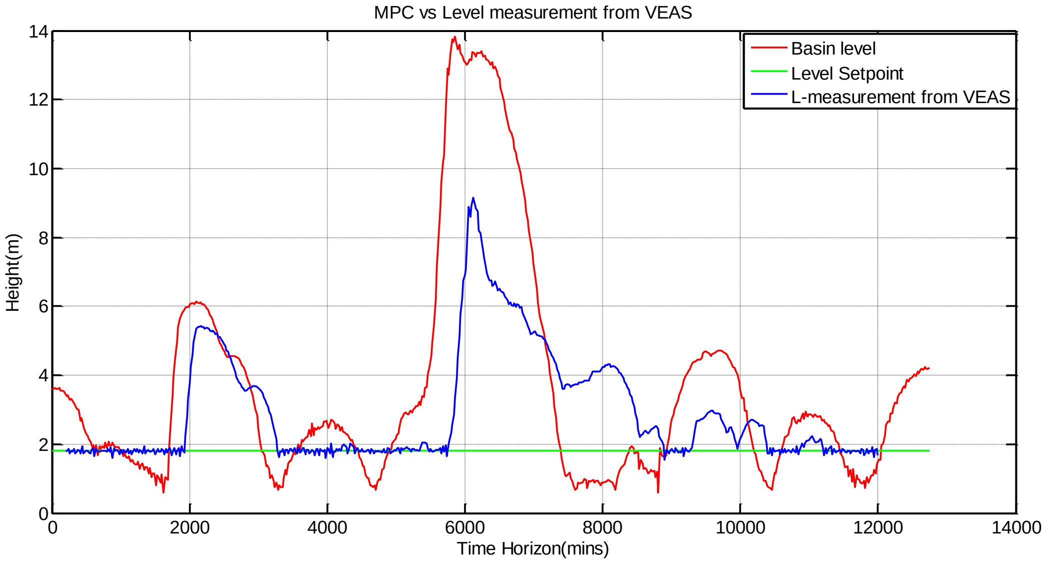

Figure 3.10 MPC vs the level measurement from VEAS, Np=10, Nc=2, P=1, R=100, resampled measurement data

Table 3.11 The numeric result of simulation #5 for the basin level

Computing time (s)	**Highest level (m)**	**Lowest level (m)**	**Violating time (mins)**
854.440728	13.8221	0.5981	501 • 21.6

In Figure 3.9, the curve representing pumps flow becomes much more smooth, in Table 3.10, both the pumps flow variance and total variation decrease when compared with Table 3.8. However, the stability of the basin level remains degraded in Figure 3.10. The violating time in Table 3.11 keeps increasing from 224 • 21.6 mins to 501 • 21.6 mins.

3.2.6 Simulation #6 (R=1000)

In this simulation, R is reset as 1000.

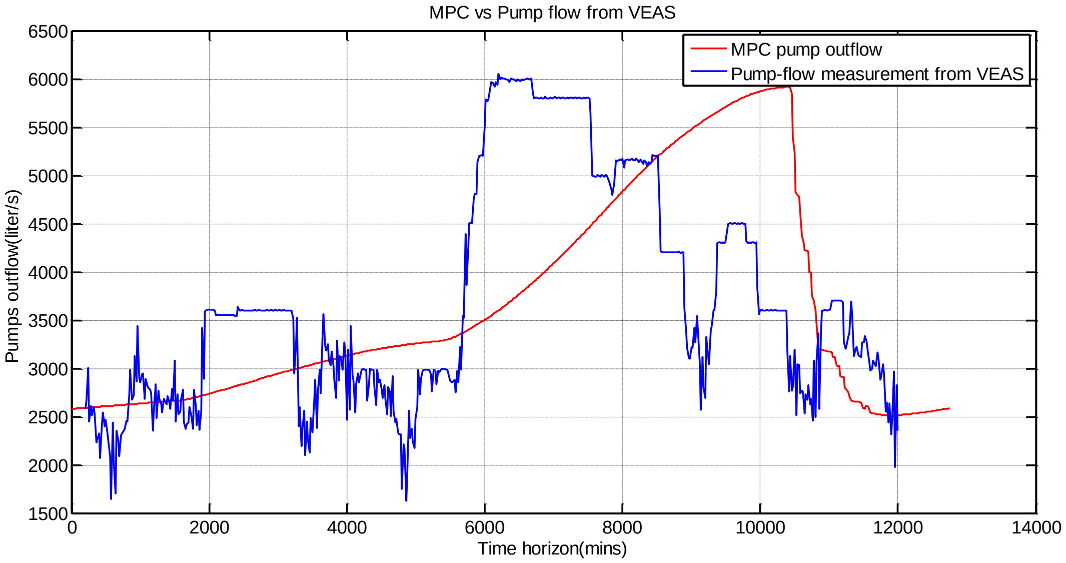

Figure 3.11 MPC vs the pumps flow from VEAS, Np=10, Nc=2, P=1, R=1000, resampled measurement data

Table 3.12 The numeric result of simulation #6 for the pumps flow

Computing time(s)	Pumps flow variance	Total variation
901.241346	1.3517e+03	6.8838e+03

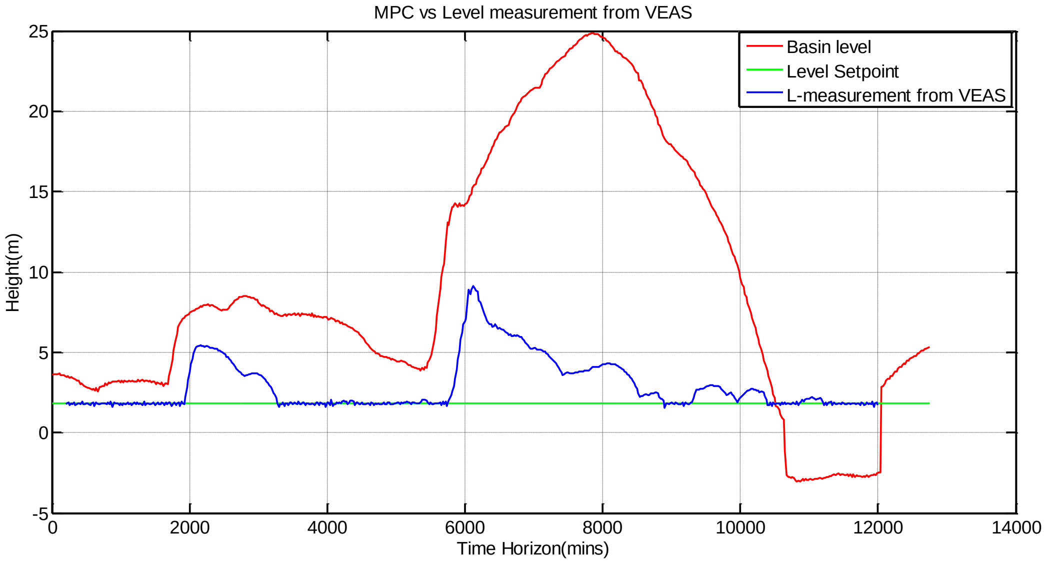

Figure 3.12 MPC vs the level measurement from VEAS, Np=10, Nc=2, P=1, R=1000, resampled measurement data

Table 3.13 The numeric result of simulation #6 for the basin level

Computing time (s)	**Highest level (m)**	**Lowest level (m)**	**Violating time (mins)**
901.241346	24.8549	-3.0453	590 • 21.6

In Figure 3.11 and Figure 3.12, the smoothness of the pumps flow is the best but the basin level is totally out of control; in Table 3.12, the total variation is furthermore decreased when compared with other simulations; however, in Table 3.13, the violating time is almost as long as the simulation horizon. There is no balance between the smoothness of the pumps flow and the stability of the basin level being seen in this simulation.

3.2.7 Optimized setting

By comparing the control performance in those simulations above, the parameters in Simulation # 4(R=10) are tested to be the optimized setting with which the smoothness of the pumps flow and the stability of the basin level can achieve a satisfying balance.

The optimized setting is listed in Table 3.14.

Table 3.14 The optimized setting for MPC

d_t	Np	Nc	P	R	Resampled data
21.6 mins	10	2	1	10	YES

Application of MPC will be based on this optimized setting.

3.3 MPC application

The measurement from VEAS is logged from 01st August to 14th December 2015; which means the performance of MPC covering this period is of interest. Therefore, MPC is chosen to be implemented on three different simulation horizons that are shown in Table 3.15.

Table 3.15 The simulation horizons for MPC application

Simulation horizon #1(min)	ts=0; d_t=21.6; tf= 600 • 21.6.
	Simulation horizon#1= ts:d_t:tf;

Simulation horizon #2(min)	ts=10^5; d_t=21.6; tf= 600 • 21.6 +10^5.
	Simulation horizon#2= ts:d_t:tf;

Simulation horizon #3(min)	ts=1.5 • 10^5; d_t=21.6; tf=
	600 • 21.6+1.5 • 10^5.
	Simulation horizon#3= ts:d_t:tf;

The inflows are shown in Figure 3.13, Figure 3.14 and Figure 3.15, which are related to the corresponding simulation horizons. The red curve represents the measured inflow in Vækerø and the blue curve represents the calculated total inflow. Because of the time delay, the time when the total inflow arrives the inflow basin is 0.15d later than the time when the inflow is measured in Vækerø.

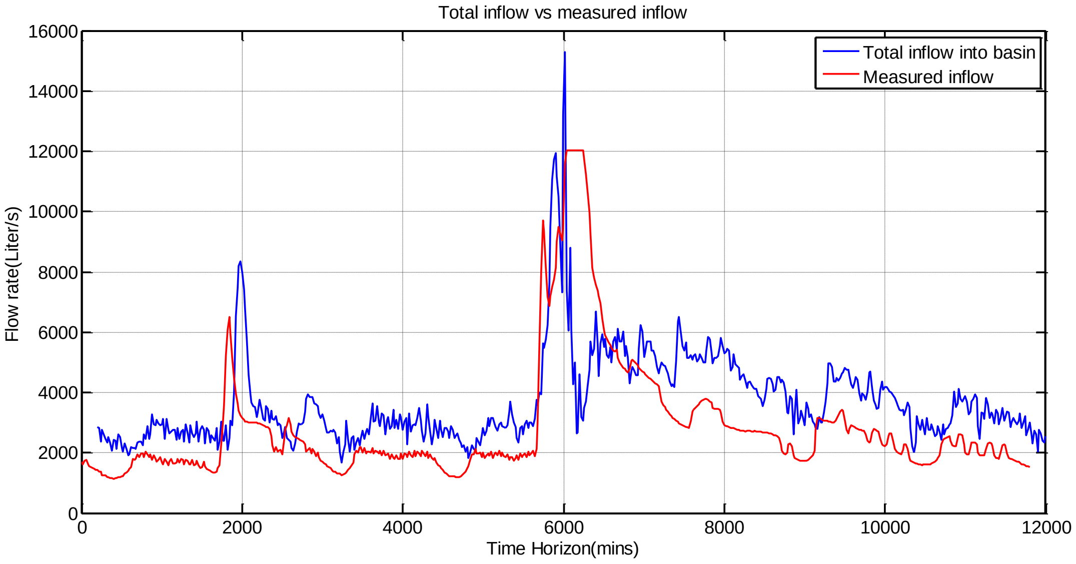

Figure 3.13 The total inflow into the inflow basin vs the measured inflow from Vækerø for simulation horizon #1

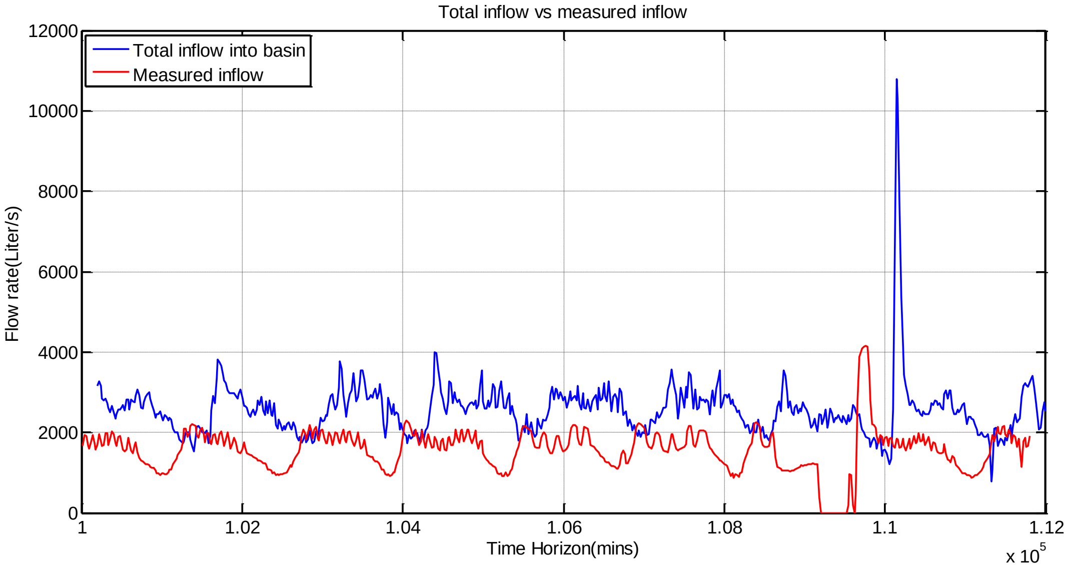

Figure 3.14 The total inflow into the inflow basin vs the measured inflow from Vækerø for simulation horizon #2

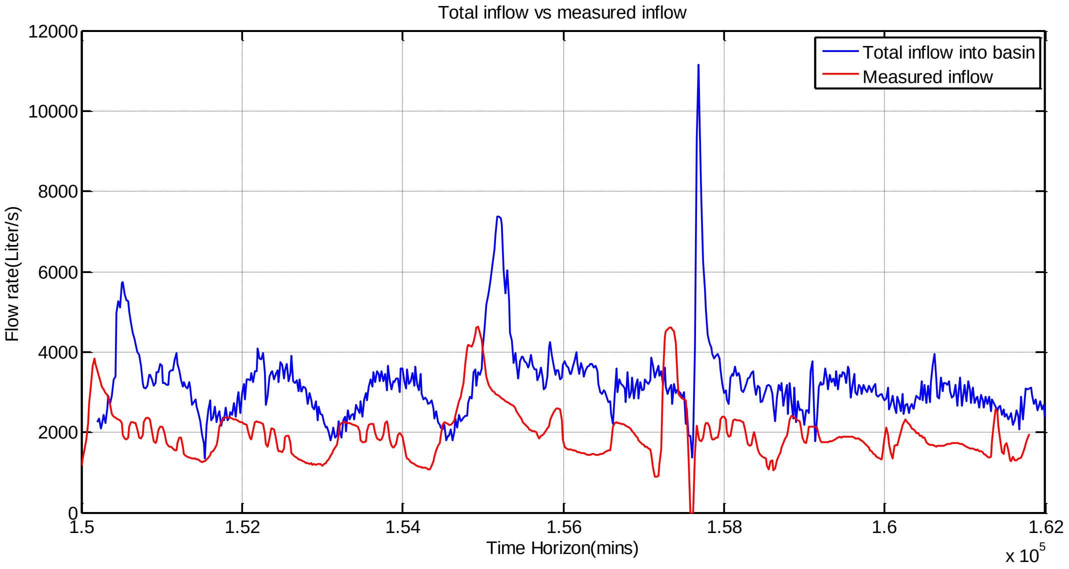

Figure 3.15 The total inflow into the inflow basin vs the measured inflow from Vækerø for simulation horizon #3

3.3.1 Influence of the internal inflow

When it is raining season, the inflow will be much bigger than dry weather, which means massive wastewater will flow into wastewater treatment plants. Because of the physical limitation, those plants are not able to handle the excessive flow, which causes the retention time of wastewater to be shorted in settlers and reaction tanks. This effect will damage the treatment procedures e.g. the bacteria in aeration tanks will be washed out and the active sludge will have not enough time to settle down and recycle. etc.. Besides, the effluent quality index of the outflow will be much lower. In addition, the level in the inflow basin will have the risk of exceeding the level limits that will cause the overflow and damage the environment.

Internal inflow is the wastewater flow that is transported back to the inflow basin instead of sending to the water treatment plants. It can protect the water treatment process and prevent the overflow when the wastewater inflow is too big.

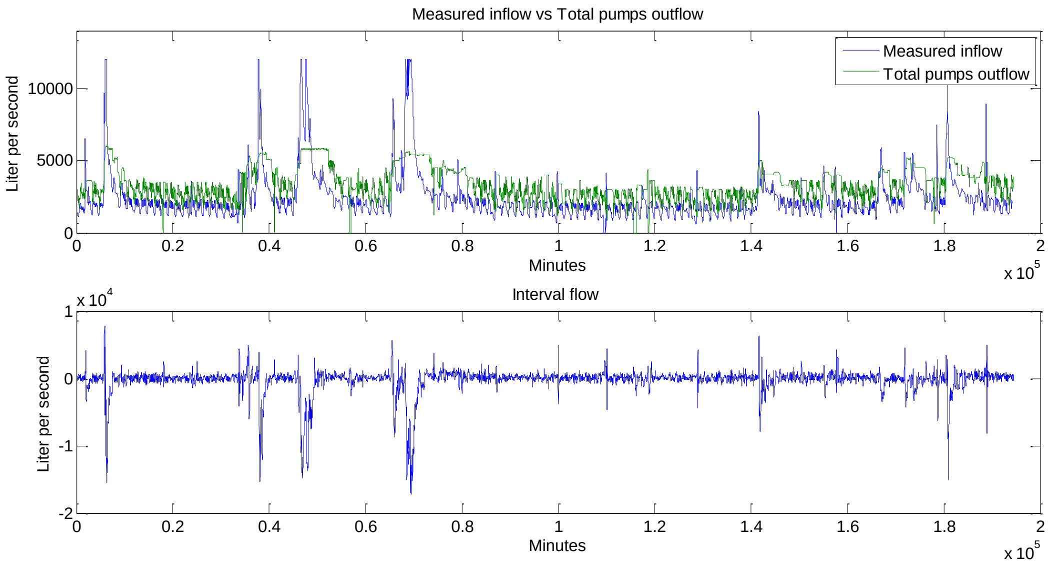

Figure 3.16 The measured inflow vs the total pumps flow and calculated internal flow

As can be seen in Figure 3.16, when the measured inflow increases drastically either in the raining weather or rush hour of the water consumption, those pumps will be open more to make sure that there is more wastewater flow being treated by the wastewater treatment plant. However, the excessive wastewater supply to the plant will disrupt the wastewater treatment process and deteriorate the quality of the treated wastewater. In order to keep the wastewater treatment at a qualified level, the internal flow should be increased in order to limit the wastewater flow flowing into wastewater treatment plant. An extra pump that is operated by VEAS in order to pump the wastewater back to the inflow basin creates the internal flow. Because the flow direction of the internal flow is opposite to the normal wastewater flow, the internal flow in Figure 3.16 shows as negative value.

3.3.2 MPC application on simulation horizon #2

In Figure 3.17, the pumps flow controlled by MPC appears to be smoother than the pumps flow from VEAS. In Table 3.16, the relative numeric result is shown.

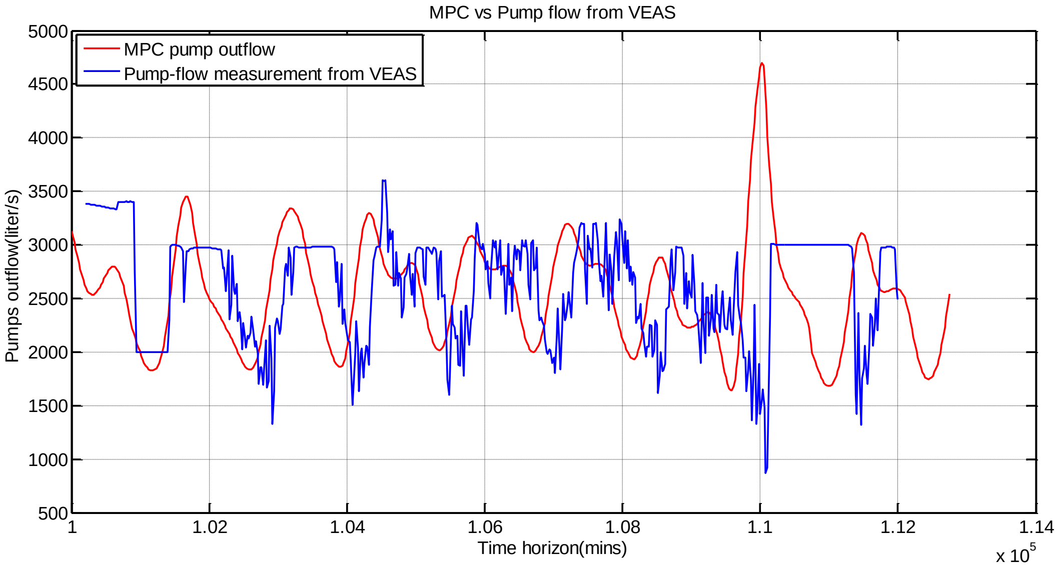

Figure 3.17 MPC with the optimized setting vs the pumps flow from VEAS for simulation horizon #2

Table 3.16 The numeric result of simulation horizon #2 for the pumps flow with the optimized setting

Computing time(s)	**Pumps flow variance**	**Total variation**
801.546446	3.9734e+03	2.7995e+04

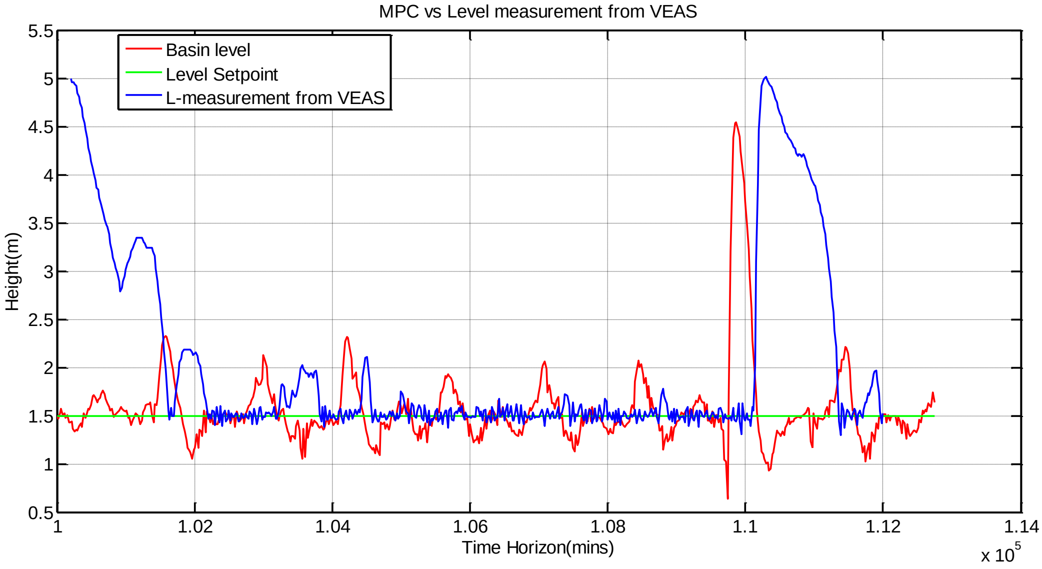

Figure 3.18 MPC with the optimized setting vs the level measurement from VEAS for simulation horizon #2

Table 3.17 The numeric result of simulation horizon #2 for the basin level with the optimized setting

Computing time (s)	Highest level (m)	Lowest level (m)	Violating time (mins)
801.546446	4.5402	0.6382	352 • 26.1

3.3.3 MPC application on simulation horizon #3

In Figure 3.19, the pumps flow controlled by MPC is compared with the pumps flow by VEAS. In Table 3.18, the relative numeric result is shown.

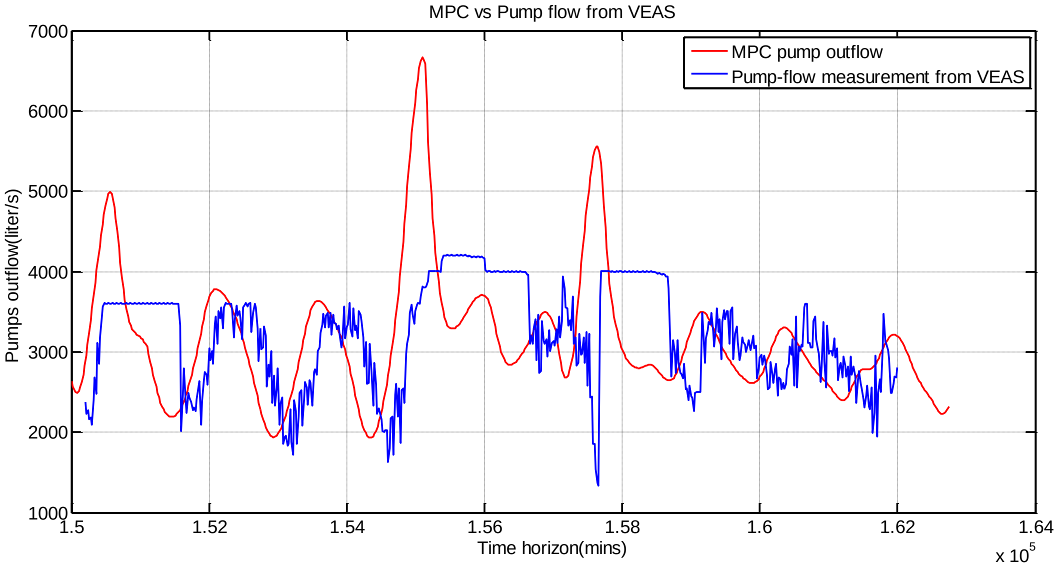

Figure 3.19 MPC with the optimized setting vs the pumps flow from VEAS for simulation horizon #3

Table 3.18 The numeric result of simulation horizon #3 for the pumps flow with the optimized setting

Computing time(s)	Pumps flow variance	Total variation
907.916540	7.2164e+03	3.4282e+04;

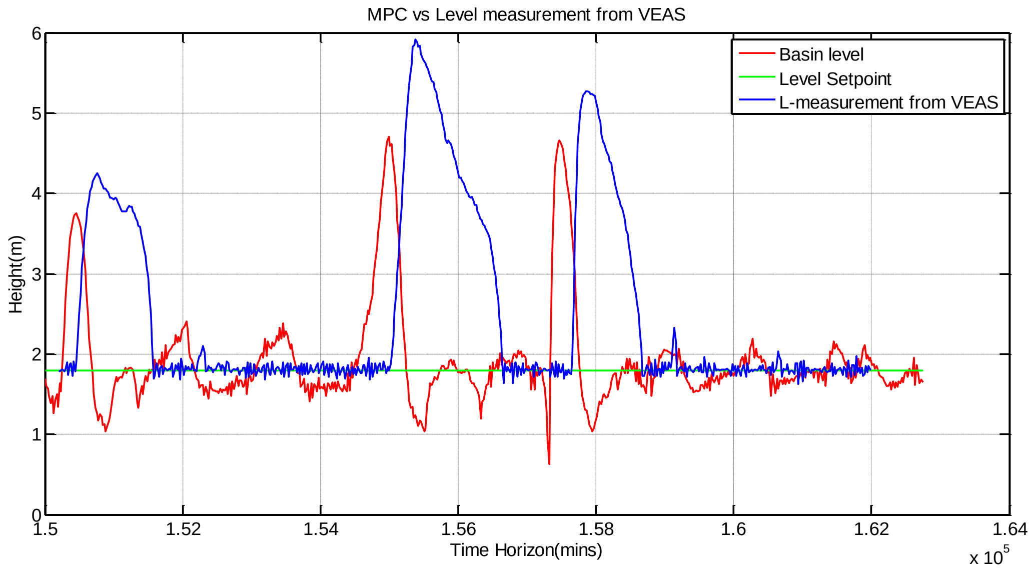

Figure 3.20 MPC with the optimized setting vs the level measurement from VEAS for simulation horizon #3

Table 3.19 The numeric result of simulation horizon #3 for the basin level with the optimized setting

Computing time (s)	**Highest level (m)**	**Lowest level (m)**	**Violating time (mins)**
907.916540	4.7083	0.6311	164 • 21.6

In Figure 3.18 and Figure 3.20, compared with the level measurement by VEAS, the basin level controlled by MPC has a lower maximum value and shows more fluctuation.

Pumps flow variance and total variation in Table 3.16 are smaller than in Table 3.18, which means MPC has a better performance of keeping the smoothness of the pumps flow on simulation horizon #2 than simulation horizon #3.

Violating time in Table 3.19 is less than in Table 3.17, which means MPC has a better performance of keeping the basin level in the height limits on simulation horizon #3 than simulation horizon #2.

4 PI controller application

In this thesis, the inflow basin is modelled as an integrator system and controlled by PI controller. The inflow basin model has been discretized in the previous stage. PI controller is described in a continuous form in time domain and needs to be discretized.

In time domain:

$$u(t) = Kp \cdot e(t) + \frac{Kp}{Ti}\int e(t)dt \tag{7}$$

$$e(t) = ref - basin\_level(t) \tag{8}$$

Where $u(t)$ is the control input (pumps flow); Kp is the proportional gain; Ti is the integral time (s) and $e(t)$ is the error between the set point and basin level at time t.

Discrete PI controller:

$$Z_{k+1} = \ Z_k + \Delta t \cdot \frac{Kp}{Ti} \cdot e(k) \tag{9}$$

$$u_k = Z_k + \ Kp \cdot \ e(k) \tag{10}$$

Where Z_k is a state of basin model; Δt is the time interval.

The sampling time is chosen to be 20 minutes that is too large for PI controller; therefore, a resampling process needs to be executed in order to get the estimated values in the sampling time based on the measurement data from Vækerø. The resampling interval is chosen as 1 minute. This work has been already discussed and done in Simulation #3(resampled data).

The pumps flow is limited to only be able to change 300 liters per second per 15 minutes, which means anti-windup should also be considered in PI controller (Sung et al., 2009, p. 130)

4.1 Controller tuning

4.1.1 Skogestad’s tuning method

The inflow basin can be treated as an integrator system but with varying process dynamics. The surface area of the inflow basin is a function of the basin level. When the basin level changes, the process parameters will change correspondingly, because of which, the PI parameters adjustment with the Skogestad’s tuning method should be implemented.

The mathematic model of the inflow basin is shown in equation (11):

$$\dot{H} = (\mathrm{F\_inn\_total} - \mathrm{F\_pump})/\mathrm{Area(H)} \quad (11)$$

Where H is the basin level; F_inn_total is the total inflow including the measured inflow and the internal flow and treated as the input disturbance of the simulator of the inflow basin; F_pump is the total pumps flow; Area (H) is the surface area of the inflow basin and the function of H.

The simulation of the surface area is implemented to show the dynamic property of the inflow basin.

The basin level is assumed to change from -2.4 to 5 m, and the change of the surface area (Area (H)) is shown in Figure 4.1.

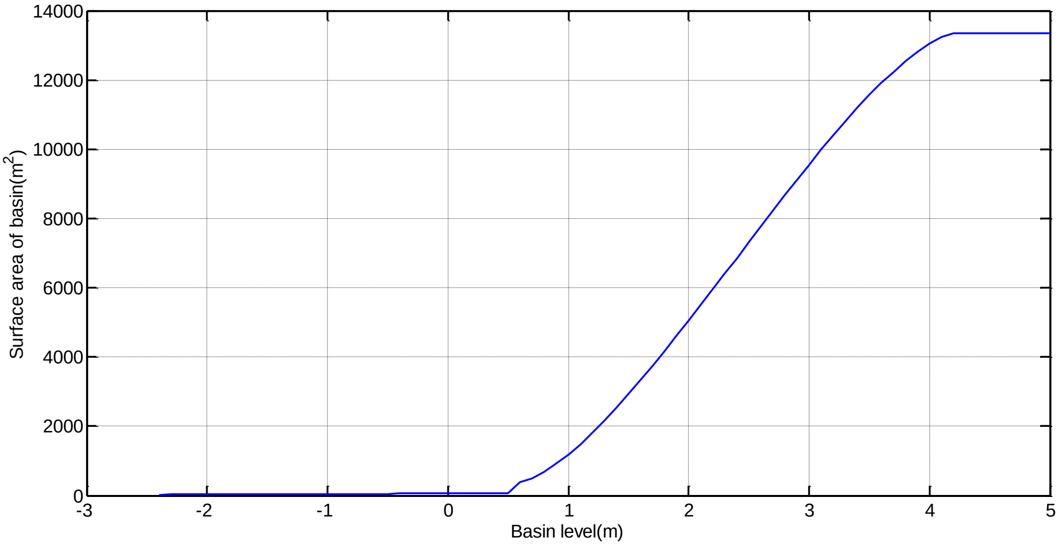

Figure 4.1 The surface area of the inflow basin varies when the basin level changes

As can be seen in Figure 4.1, the surface area remains low and stable when the basin level is lower than 0.6 m; but when the basin level continues to rise, the surface area begins to change much sharply, especially in the range from 1 m to 3 m. As a result, the PI parameters should be adjusted.

4.1.2 Gain schedule with PI parameters

The basin level is designed as the Gain scheduling variable GS, which expresses the dynamic property of the process at every instant of time.

The level range from 1 m to 3 m should be sliced into several parts due to the rapid change of the surface area; the length of each part is designed as 0.5m.

The Gain schedule of the PI controller parameters by using the Skogestad’s tuning method is shown in Table 4.1.

Table 4.1The Gain schedule table

GS	Basin level	Surface_area	Process Gain	Tc	Kp	Ti
GS1	(3 m –upper limit)	$1.3336 \bullet 10^{4}$ m^{2}	Approximately $7.5 \bullet 10^{-8}$ m/liter	9000 s	-1481.5 liter/(s • m)	36000 s
GS2	(2.5 m – 3 m)	$9.5550 \bullet 10^{3}$ m^{2}	Approximately $1.0467 \bullet 10^{-7}$ m/liter	9000 s	-1061.5 liter/(s • m)	36000 s
GS3	(2 m – 2.5 m)	$7.3219 \bullet 10^{3}$ m^{2}	Approximately $1.3650 \bullet 10^{-7}$ m/liter	9000 s	-814 liter/(s • m)	36000 s
GS4	(1.5 m – 2 m)	$5.0531 \bullet 10^{3}$ m^{2}	Approximately $1.9783 \bullet 10^{-7}$ m/liter	9000 s	-561 liter/(s • m)	36000 s
GS5	(1 m – 1.5 m)	$2.9311 \bullet 10^{3}$ m^{2}	Approximately $3.4117 \bullet 10^{-7}$ m/liter	9000 s	-325 liter/(s • m)	36000 s
GS6	(0.6 m – 1 m)	$1.1867 \bullet 10^{3}$ m^{2}	Approximately $8.4267 \bullet 10^{-7}$ m/liter	9000 s	-131 liter/(s • m)	36000 s
GS7	(-2m— 0.6 m)	66.408 m^{2}	Approximately $1.5 \bullet 10^{-5}$ m/liter	9000 s	-7.4 liter/(s • m)	36000 s

As can be seen from Table 4.1, the surface area of the inflow basin changes dramatically when the basin level drops from 5 to -2 m, which furthermore shows that the PI parameters adjustment is of need.

4.2 PI controller application

Because of the variation of the process dynamics, the basin model at some critical points of level is chosen in order to prove the feasibility of the Gain schedule.

Two different basin levels are chosen and tested in Table 4.2.

Table 4.2 Gain schedule testing for critical points of the basin level

Test	#1	#2
Basin level(m)	3(GS1)	1.8(GS4)
Surface area (m^2)	9.5550e+03	4.1757e+03
Kp	-1481.5	-562
Ti(s)	9000	9000

4.2.1 Test #1(GS1 is applied)

When the basin level is 3m that almost equals to the height of the basin roof, the full buffering capacity of the inflow basin is activated and GS1 should be applied. If the level keeps rising and higher than 3 m, part of the stored wastewater will remain and stay in wastewater transporting tunnel.

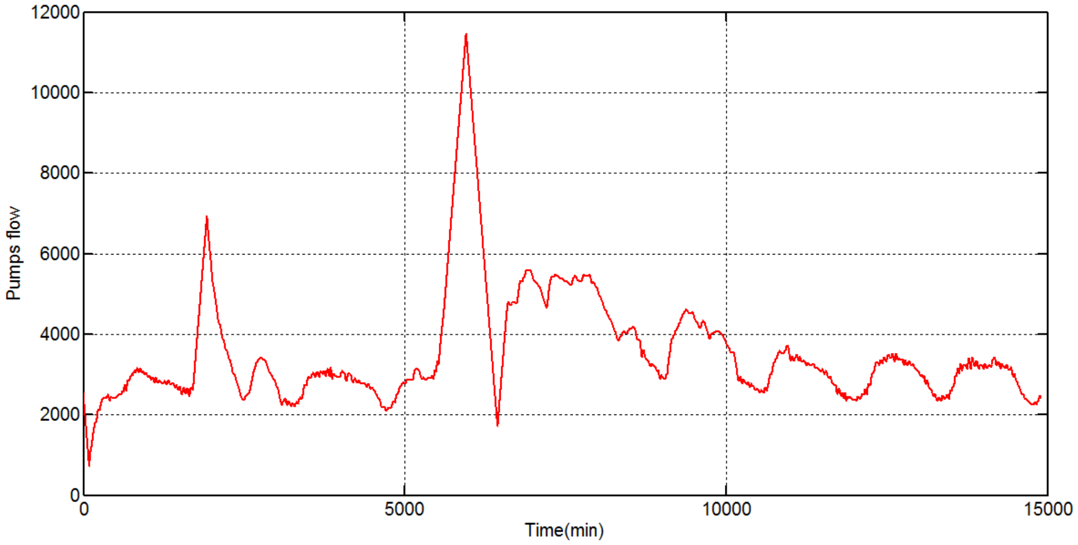

Figure 4.2 The pumps flow of PI controller when GS1 is activated

Table 4.3 The total variance of the pumps flow controlled by PI controller when GS1 is activated

Test	**Total variance**
#1	9.3081e+04

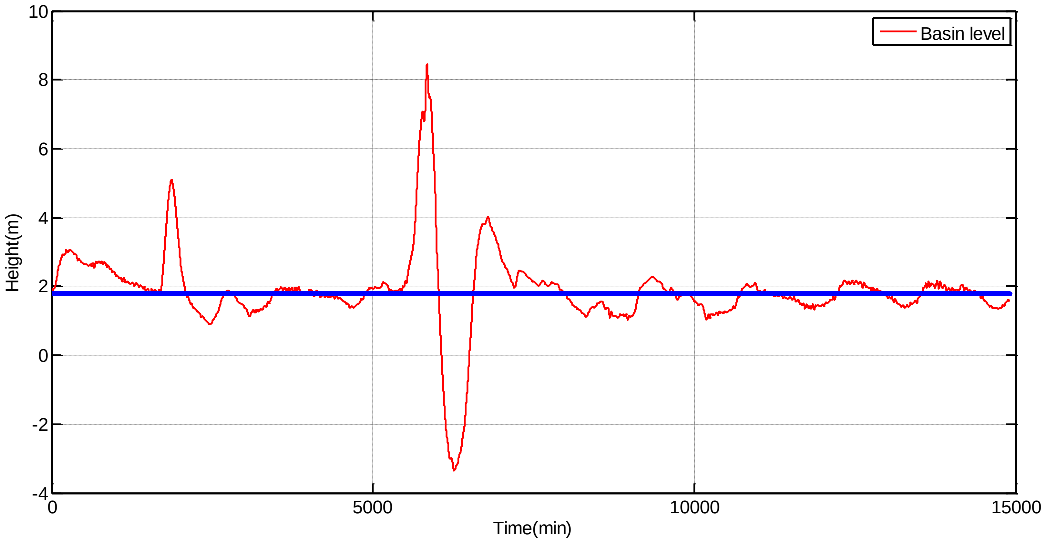

Figure 4.3 The basin level controlled by PI controller when GS1 is activated

Table 4.4 The numeric result of the basin level controlled by PI controller when GS1 is activated

Highest level (m)	Lowest level (m)	Violating time (mins)
8.4439	-3.3572	7527

In Figure 4.2 and Table 4.3, the fluctuation of the pumps flow and the total variation of the pumps flow are much bigger when compared with Figure 3.7 and Table 3.8. In addition, Figure 4.3 and Table 4.4 show that the stability of the basin level is also degraded.

4.2.2 Test #2 (GS4 is applied)

GS4 is the range where VEAS desires to keep the basin level for most of time. The basin level 1.8 m is the one of set points for the inflow basin and chosen for this simulation.

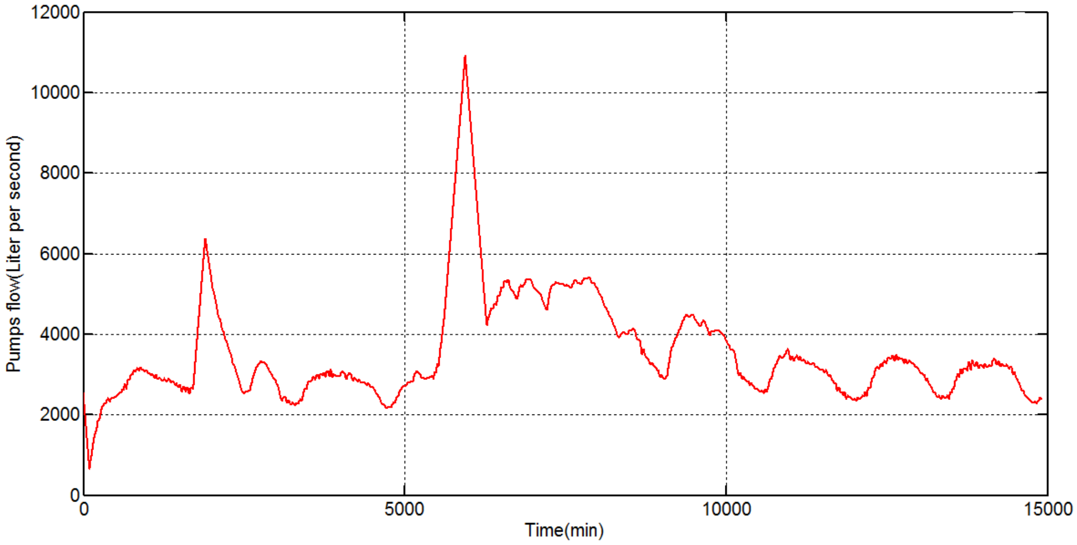

Figure 4.4 The pumps flow of PI controller when GS4 is activated

Table 4.5 The total variance of the pumps flow controlled by PI controller when GS4 is activated

Test	Total variation
#2	8.1984e+04

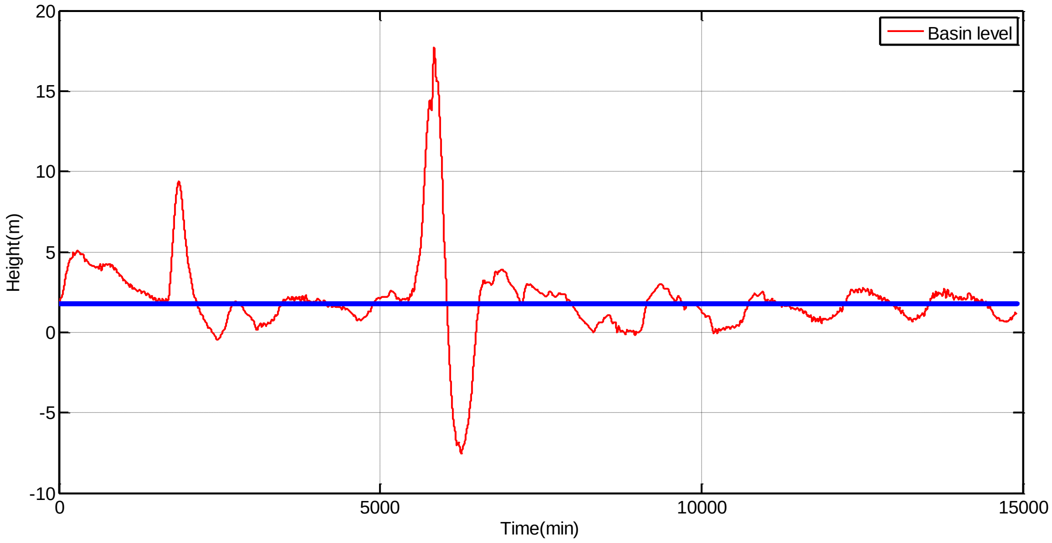

Figure 4.5 The basin level controlled by PI controller when GS4 is activated

Table 4.6 The numeric result of the basin level controlled by PI controller when GS4 is activated

Highest level (m)	**Lowest level (m)**	**Violating time (mins)**
17.7281	-7.5263	12053

In Figure 4.4 and Table 4.5, the fluctuation of the pumps flow and the total variance of the pumps flow are bigger when compared with Test #1(GS1 is applied). In addition, Figure 4.5 and Table 4.6 show the stability of the basin level keeps being degraded furthermore.

5 Results analysis

Results analysis is mainly designed for analyzing the simulation results by using different data analyzing methods.

5.1 Simulations analysis

5.1.1 Pumps flow

The stability of the pumps flow is a big concern when it comes to choose the control methods. VEAS requires that the pumps flow rate change for each 15mins should be less than 300 liters/s. Too big pumps flow rate change will bring the disturbance to the further wastewater treatment process. In this chapter, the stability of the pumps flow will be analyzed and compared with VEAS data.

5.1.1.1 Pumps flow rate variation

As can be seen from Figure 5.1, VEAS pumps flow rate variation for every 20minutes looks much more ‘noisy’ than MPC pumps flows’, which means that MPC can achieve a better stability to some extent. Simulation horizon #1 is the time from 0 to 12000 mins approximately. Simulation horizon #2 is the time from 100000mins to 112000mins; Simulation horizon #3 is the time from 150000mins to 162000 mins. More details about the simulation horizons can be seen in Table 3.15. Figure 5.1, Figure 5.2 and Figure 5.3 show the pumps flow rate variation for the corresponding simulation horizon.

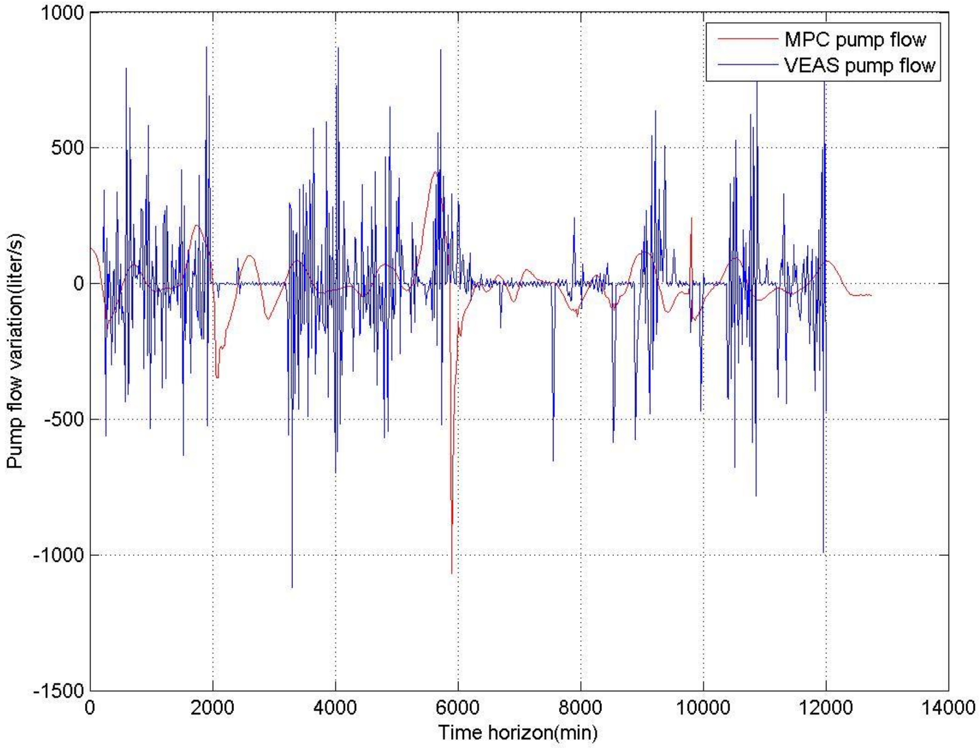

Figure 5.1 Pumps flow variation (MPC) vs pumps flow variation (VEAS) for simulation horizon #1

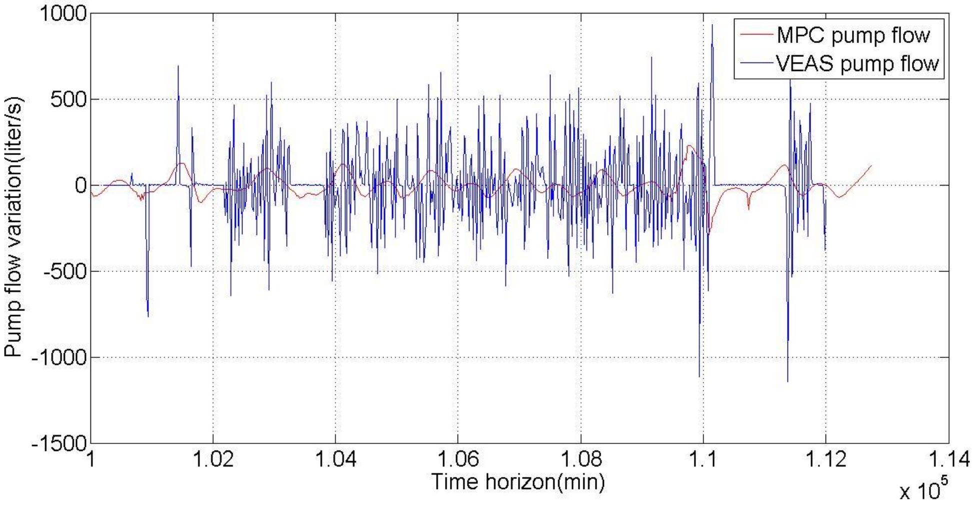

Figure 5.2 Pumps flow variation (MPC) vs pumps flow variation(VEAS) for simulation horizon #2

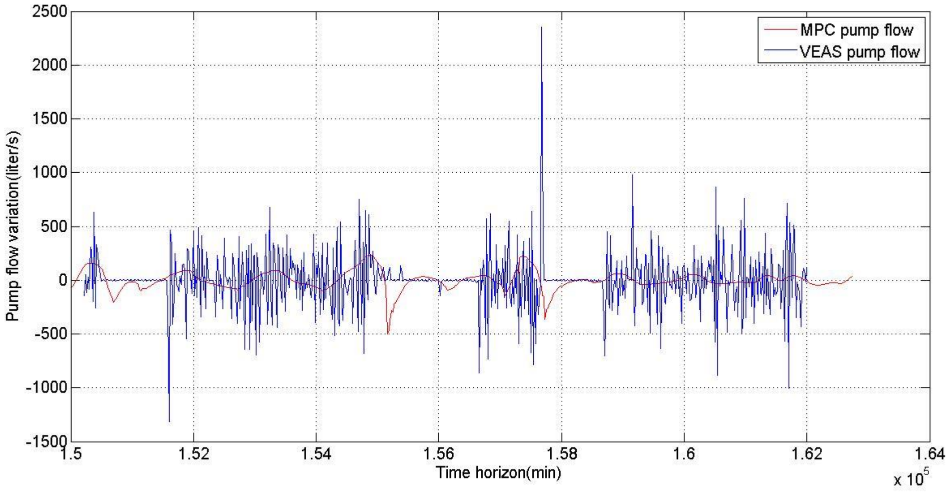

Figure 5.3 Pumps flow variation (MPC) vs pumps flow variation (VEAS) for simulation horizon #3

5.1.1.2 Variance and standard deviation

Variance and standard deviation are the other two indexes to describe the stability of the pumps flow in this thesis.

Figure 5.4 shows the pumps flow rate variance both for MPC and VEAS pumps flow, Figure 5.5 shows the pumps flow rate standard deviation for them.

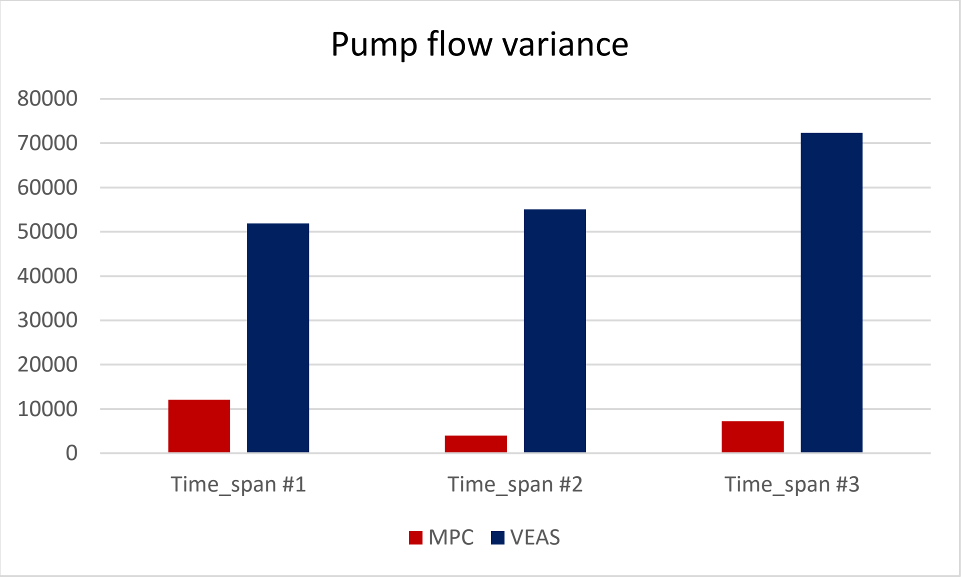

Figure 5.4 Pumps flow variance(MPC) vs pumps flow variance(VEAS)

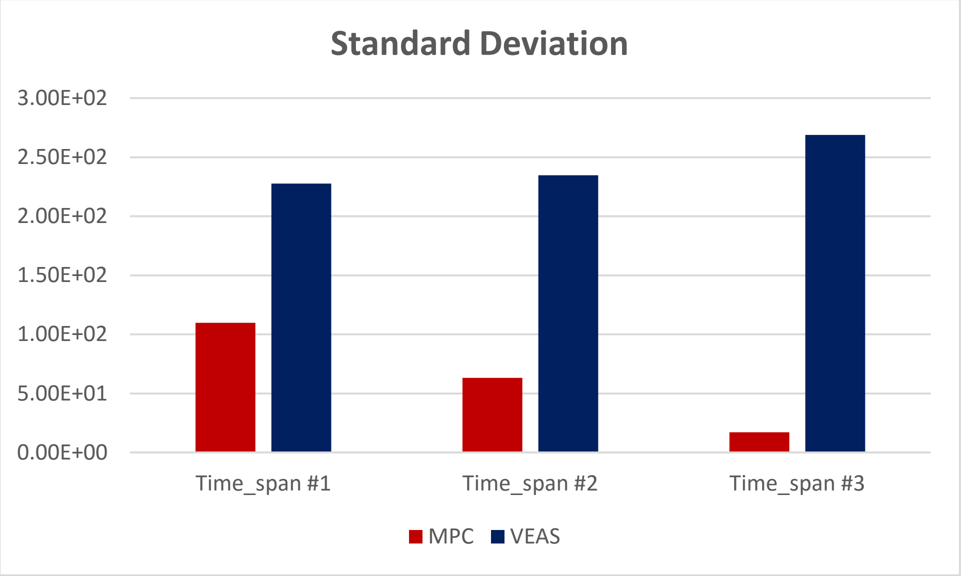

Figure 5.5 Pumps flow standard deviation (MPC) vs pumps flow standard deviation (VEAS)

From Figure 5.4 and Figure 5.5, both the variance and standard deviation of pumps flow rate from MPC are much less than those from VEAS.

Table 5.1 Variance and standard deviation calculation for MPC and VEAS

	Time span #1	Time span #2	Time span #3
Variance (MPC)	1.2040e+04	3.9734e+03	7.2164e+03
Variance(VEAS)	5.1859e+04	5.5063e+04	7.2348e+04
Standard deviation(MPC)	109.7259	63.0348	84.9496
Standard deviation(VEAS)	227.7267	234.6545	268.9751

According to Table 5.1, the average variance (MPC) and standard deviation (MPC) are 7.7433e+03 and 85.9034, which are approximately 7.5 times less than the average variance (VEAS) which is 5.8689e+04 and 3 times less than the standard deviation (VEAS) which is 243.7854.

5.1.1.3 Total variation

Total variation is used to measure the sum of the pumps flow variation throughout the whole time span, which is an efficient way for calculating the overall stability of the pumps flow.

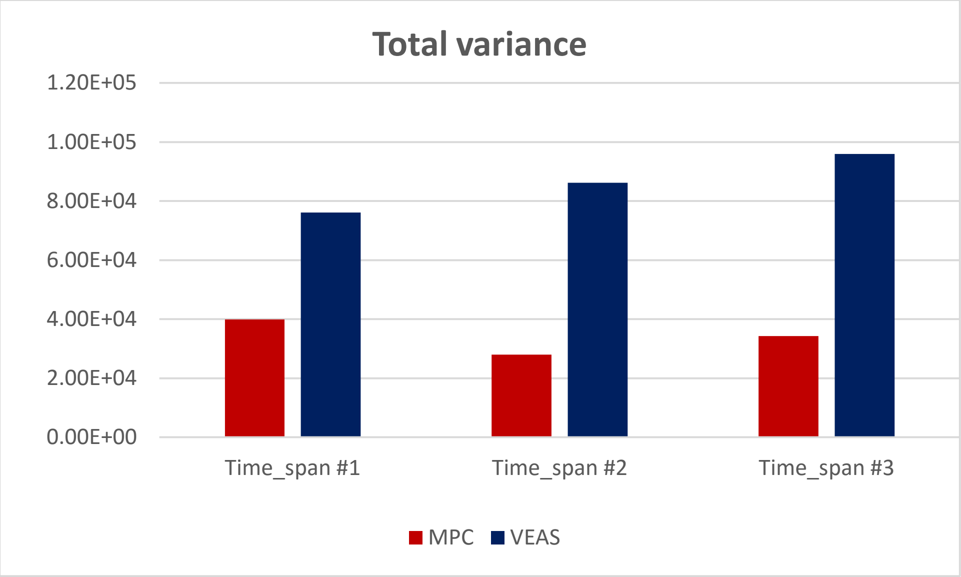

Figure 5.6 Total variance (MPC) vs total variance (VEAS)

Table 5.2 Total variance calculation for MPC and VEAS

	Time span #1	Time span #2	Time span #3
Total variation (MPC)	3.9375e+04	2.7995e+04	3.4282e+04
Total variation(VEAS)	7.6073e+04	8.6229e+04	9.5898e+04

In Table 5.2, the total variation (MPC) is 3.4073e+04, which is approximately 2 times less than the total variation (VEAS) which is 5.8689e+04.

According to the graphical result in Figure 5.6 and numerical analysis in Table 5.2, it is obviously to conclude that the stability of the pumps flow is improved by implementing MPC into the control system.

5.1.2 Basin level

5.1.2.1 Violation of height limits

The basin level needs to be kept in the range from 1.5 m to 2.1 m as much as possible; In order to measure and compare the violation of height limits, the areas beyond the limits, which are shown in Figure 5.7, will be calculated. The less the calculated areas are, the more stable the control system will be.

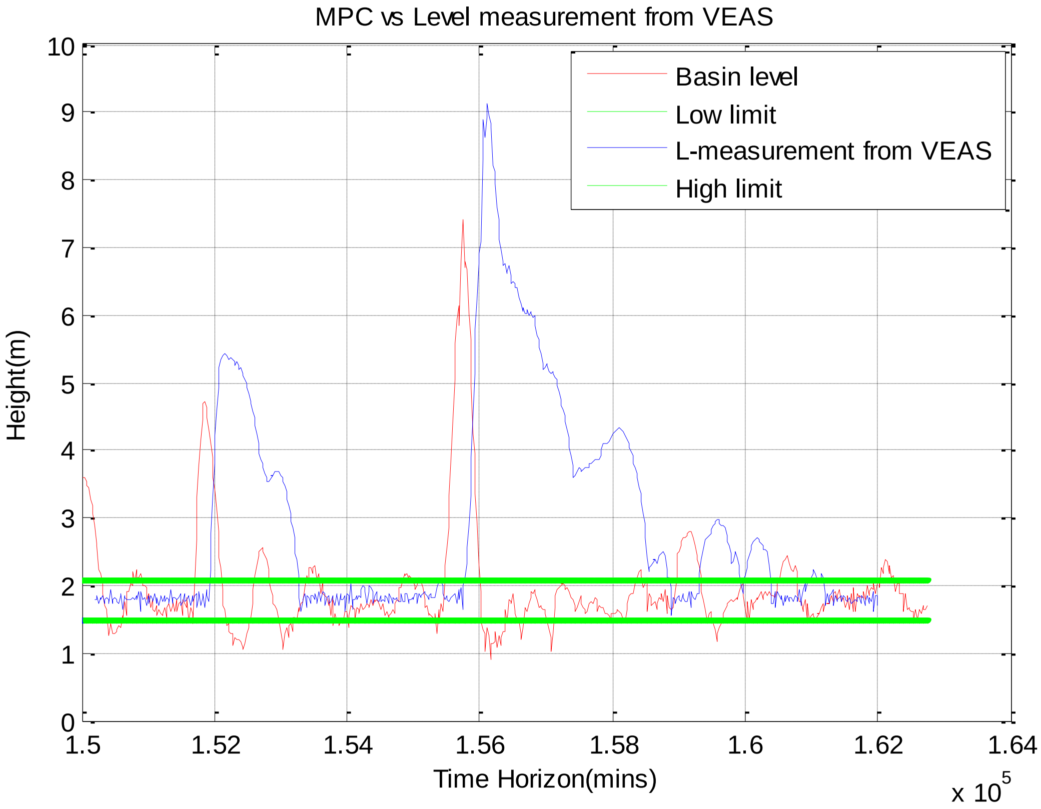

Figure 5.7 Mechanism of calculating the violation areas

The overall areas throughout three chosen time spans are shown in Figure 5.8.

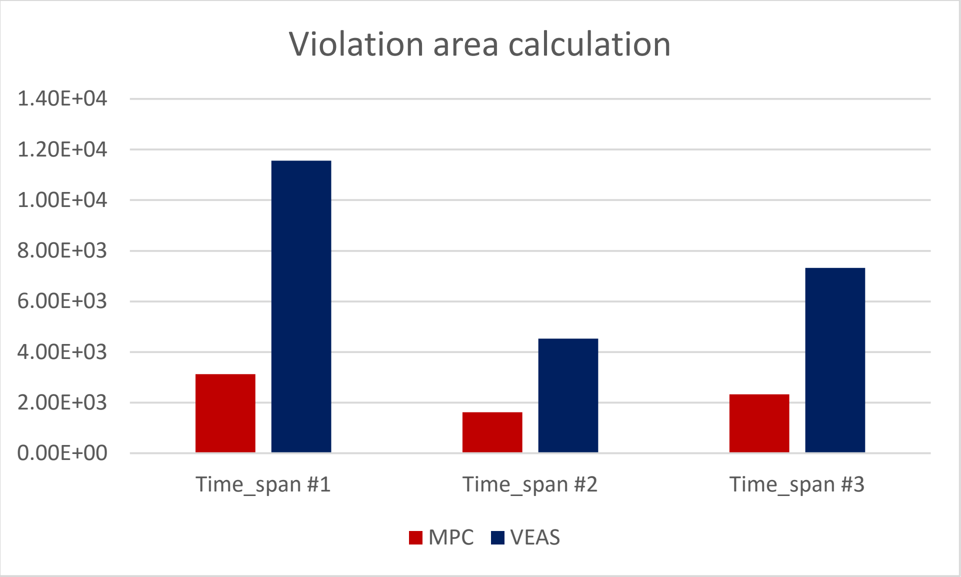

Figure 5.8 Violation area (MPC) vs violation area (VEAS)

Table 5.3 Violation area calculation for MPC and VEAS

	Time span #1	Time span #2	Time span #3
Violation of area(VEAS)	3.118e+03	1.622e+03	2.330e+03
Violation of area(MPC)	1.155e+04	4.531e+03	7.327e+03

In Figure 5.7, the total areas that are out of the range delineated by the green lines will be calculated both for blue and red curves and compared. As can be seen from Figure 5.8 and Table 5.3, the violation areas of MPC are much smaller than the violation areas of VEAS.

The percentage of the violating time to the whole time spans is also calculated and shown both for MPC and VEAS in Figure 5.9 and Figure 5.10.

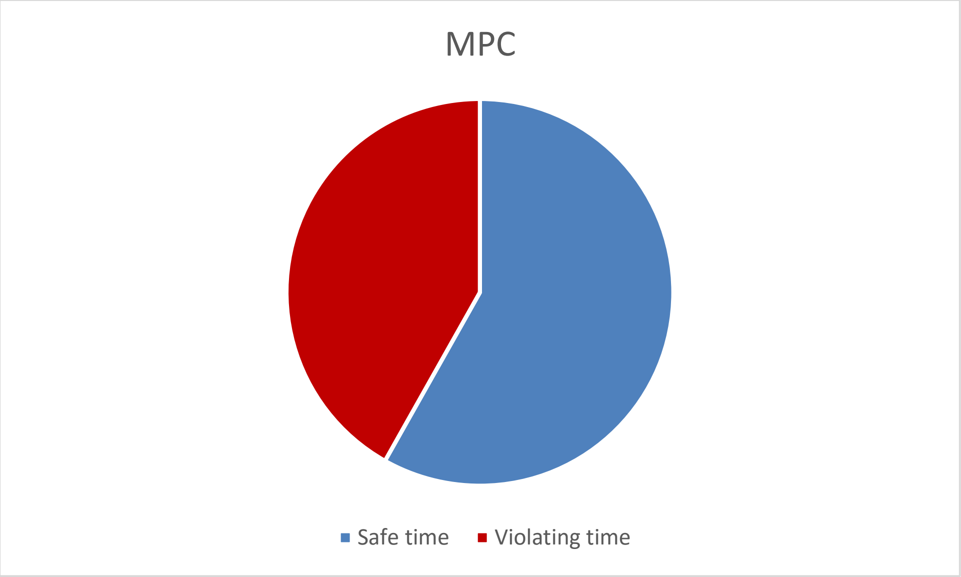

Figure 5.9 Violating time for MPC

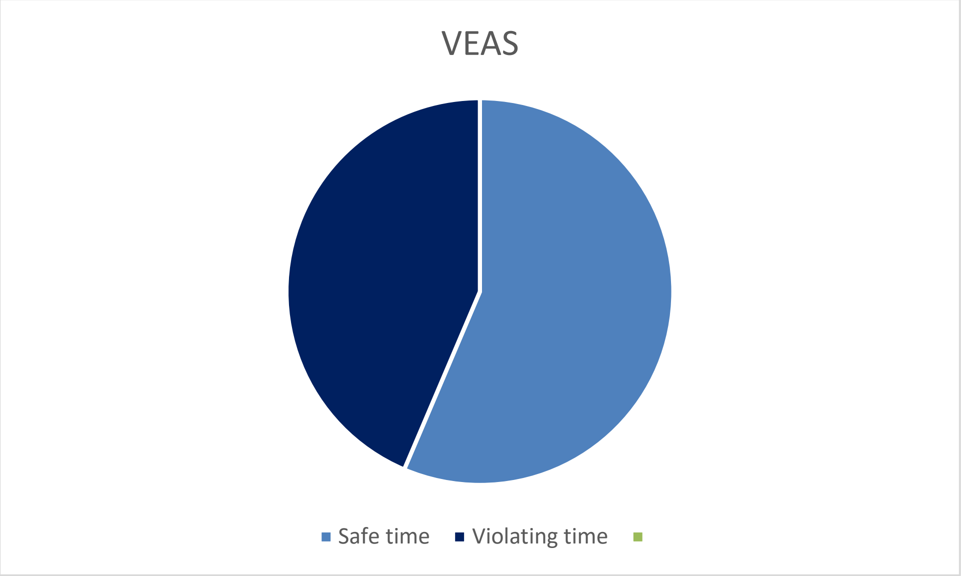

Figure 5.10 Violating time for VEAS

Table 5.4 Violation time for MPC and VEAS

	Violating time(mins)
MPC	237 • 21.6 + 352 • 21.6+164 • 21.6
VEAS	279 • 21.6 + 310 • 21.6+187 • 21.6

As can be seen in Table 5.4, the total violation time of MPC is 1.6265e+04 mins , and the total violation time of VEAS is 1.6762e+04 mins, according to which, the stability of the basin level is slightly improved.

5.1.2.2Highest value and lowest value

The highest value and lowest value of the basin level are shown in Table 5.5.

Table 5.5 Lowest and highest value of the basin level controlled by MPC and VEAS

Time span	MPC basin highest level	VEAS basin highest level	MPC basin lowest level	VEAS basin lowest level
#1	7.4111m	9.1400m	0.8768m	1.5500 m
#2	4.5402m	5.0100m	0.6382m	1.3000 m
#3	4.7083m	5.9100m	0.6311m	1.6300 m

Thanks to MPC, the smoothness of the pumps flow has been improved considerably without scarifying the stability of the basin level.

6 Discussion and conclusion

6.1 Discussion

In this thesis, the time delay is set as 0.15d, which means the difference from the time when the inflow of wastewater is measured from Vækerø and the time when the inflow of the wastewater flows into the inflow basin. According to the level measurement provided by VEAS, during the measuring period from 1st August to 14th December, the wastewater level in the inflow basin can even go up to higher than 8.0 m, which is much higher than the roof of the inflow basin (4.2 m). The reason is that the inflow will increase largely along with the raining weather or the increase of the water consumption. Due to the limitation of the volume of inflow basin, part of the wastewater will be stored in the wastewater transport tunnel. As a result, the time the inflow costs to travel from Vækerø to the stored wastewater will be less than 0.15d and varies accompanied by the level change of the stored wastewater. Besides, the inflow basin is modeled mathematically under the condition when the wastewater level is not higher than the roof of the inflow basin, which means, the performance of the MPC and PI controller developed based on this mathematic model will be degraded if the level of the stored wastewater rises higher than the roof of the inflow basin.

In another case, if the level of the stored wastewater drops too low, the PI controller tuned in this thesis is not able to provide a satisfying control performance. When the level of the stored wastewater is 0.6 m, the usage of the inflow basin is too little and the surface area of the stored wastewater surface is too small.

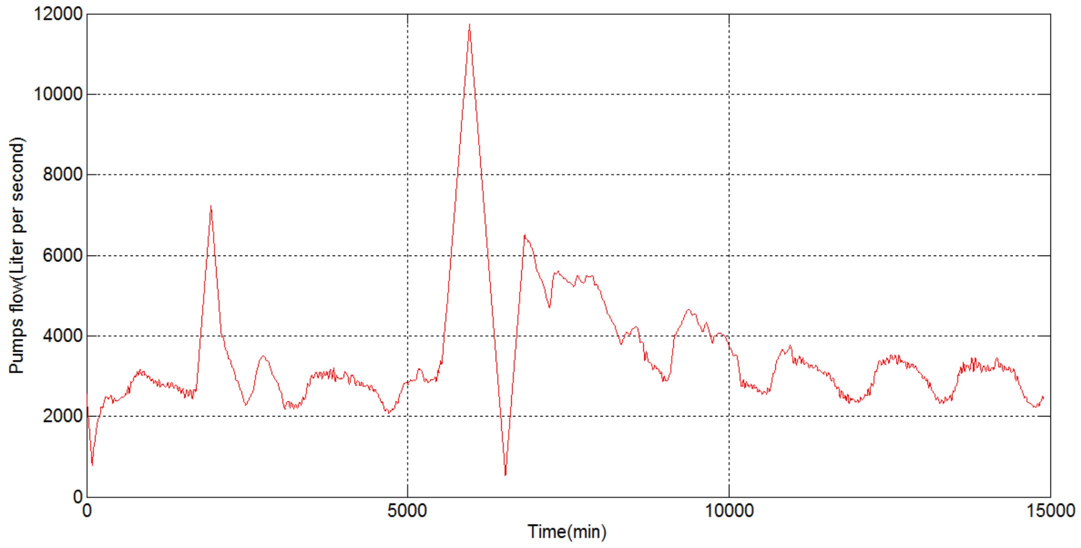

Figure 6.1 Pumps flow of PI controller when GS6 is activated

Table 6.1 Total variance of pumps flow controlled by PI controller when GS6 is activated

Test	Total variation
#3	6.6439e+04

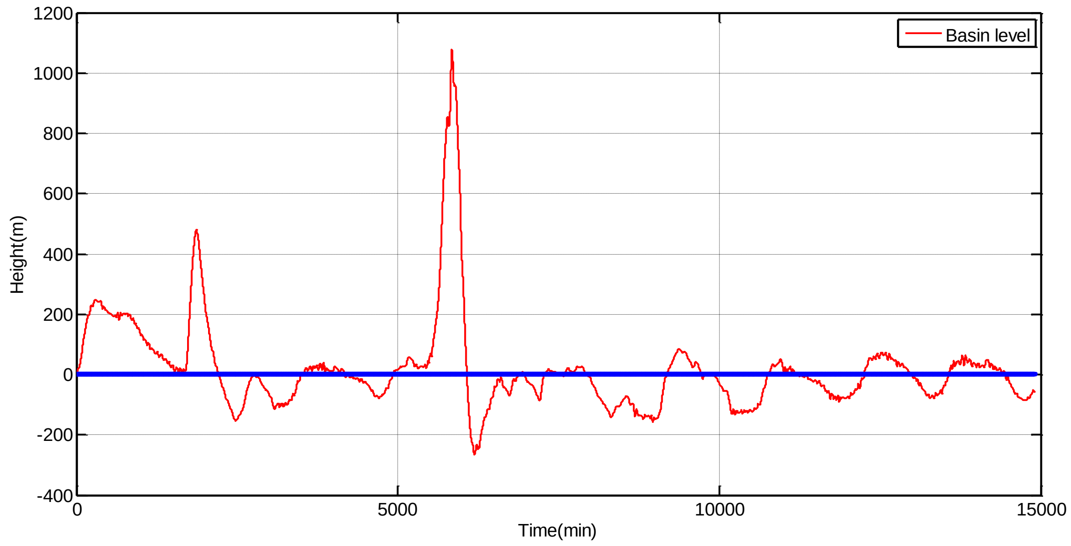

Figure 6.2 Basin level controlled by PI controller when GS6 is activated

Table 6.2 The numeric result of basin level controlled by PI controller when GS6 is activated

Highest level (m)	Lowest level (m)	Violating time (mins)
1.0787e+03	-266.5018	14861*1

In Figure 6.2, PI controller can barely maintain the stability of the basin level in a very limited way when compared with either the simulation results from MPC or VEAS, the magnitude of the level fluctuation is too big and cannot be endured by the inflow basin. Besides, in Figure 6.1and Table 6.1, the pumps flow is also not stable enough.

In Table 6.2, the calculated highest level and lowest level are not achievable in reality. When the basin level is too low, the surface area in equation (3) is too small. It will make the basin model slow in reacting in terms of level adjustment. In other words, PI controller is not able to maintain the stability of the basin level and the smoothness of the pumps flow in an acceptable way if the level of the stored wastewater is too low.

In Figure 3.7, there are two spikes shown on the red curve that represents the pumps flow controlled by MPC. The reason is that the full capacity of the pumps flow is treated as a soft constraint of MPC and allowed to be violated if the violating time is not too long. In VEAS, the operators can manually control the pumps flow and keep them in the capacity limits, which is one of the reasons that the blue curve that represents the pumps flow controlled by VEAS is flat during some period and differs from the red curve. In Figure 3.8, two spikes can be seen at the same time when the spikes in Figure 3.7 happen, the reason is that the basin level will decrease correspondingly when the pumps flow increases. The spikes of the pumps flow that appear in those figures in 3.2 all can be explained with the same reason.

6.2 Conclusion

There are three main conclusions summarized in this thesis:

The mathematical model of the inflow basin is able to simulate and present the varying dynamic properties of the inflow basin largely. In addition, Inflow basin can be simplified as an integrator system for tuning PI controller.

MPC has a better control performance in terms of maintaining the balance between the smoothness of the pumps flow and the stability of the basin level. The simulation results show that the pumps flow controlled by MPC can be kept smoother when compared with the pumps flow that is provided by VEAS. The optimized setting for the MPC is testified to be able to provide a satisfying control performance.

The Gain schedule of the PI controller parameters that are tuned by the Skogestad’s tuning method is utilized and proved a good method for controlling the inflow basin with varying process dynamics in the situation where the basin level is not either much lower or higher than the level limits of the basin level.

7 Future work

Some potential work can be done in the future.

Weather condition plays an important role in deciding the volume and flow rate of the total inflow, based on which, a weather forecast model can be developed and used to predict the inflow to the inflow basin in the future.

The dynamic simulator of the inflow basin developed in this thesis can serve as a reference base on which another theoretical optimized design for new inflow basin can be made in the future.

8 References

ANGELO, C. & M.KRAMER.(EDS), H. J. 2012. *Industrial Crystallization Process Monitoring and Control (1),* Hoboken, DE, Wiley-VCH.

BOLMSTEDT, J. 2004. *Controlling the influent load to wastewater treatment plants.* Licentiate Thesis, Lund University,Lund,Sweden.

BOLMSTEDT, J. & OLSSON, G. 2005. A benchmark study of controlled emptying of equalization basins. *Water Science and Technology,* 52**,** 113-121.

CHO, S. & LAZARO, A. 2010. Control theoretic model using PID controller for just-in-time production scheduling. *The International Journal of Advanced Manufacturing Technology,* 51**,** 699-709.

HAUGEN, F. 2015. Matematisk modellering, simulering og nivåregulering av innløpsavsnittet (Utdrag av foreløpig versjon av dokumentet).

JEPPSSON, U. & PONS, M.-N. 2004. The COST benchmark simulation model—current state and future perspective. *Control Engineering Practice,* 12**,** 299-304.

MAESTRE, J. M., RASO, L., VAN OVERLOOP, P. J. & DE SCHUTTER, B. 2013. Distributed tree-based model predictive control on a drainage water system. *Journal of Hydroinformatics,* 15**,** 335-347.

MALKI, H. A. & UMEH, C. G. 2000. Design of a fuzzy logic-based level controller. *Journal of Engineering Technology,* 17**,** 32-38.

NEDERKOORN, E., SCHUURMANS, J., GRISPEN, J. & SCHUURMANS, W. 2013. Continuous nonlinear model predictive control of a hybrid water system. *Journal of Hydroinformatics,* 15**,** 246-257.

SKOGESTAD, S. 2003. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control,* 13**,** 291-309.

SUNG, S. W., LEE, J. & LEE, I.-B. 2009. *Process Identification and PID Control (1),* Hoboken, SG, Wiley-IEEE Press.

VAN OVERLOOP, P. J., NEGENBORN, R. R. & SCHWANENBERG, D. 2013. Water prediction and control technology Introduction. *Journal of Hydroinformatics,* 15**,** 229-231.

WANG, M. & CRUSCA, F. 2002. Design and implementation of a gain scheduling controller for a water level control system. *Isa Transactions,* 41**,** 323-331.

Appendix A

HSN University College of Southeast Norway

FMH606 Master's Thesis

Title: Modeling, simulation and control of the inflow basin of a wastewater treatment plant

TUC supervisor: Finn Aakre Haugen, docent

External partner: VEAS (Norway's largest wastewater treatment plant, situated in Slemmestad south of Oslo)

Task background: The inflow basin of the VEAS wastewater treatment plant (WWTP) serves two main purposes: It equalizes hydraulic and composition variations of the wastewater inflow so that the flow into the subsequent processes for nutritions and toxicants removal and energy removal from the sedimented sludge show less variations. The level control of the basin must meet the following two contradictory requirements: (1) Compliant level control to obtain an hydraulic and compositional smoothing. (2) Stiff level control to prevent the level from passing critical limits in the case of inflow slugs. The optimal compromise between these two requirements is dependent on the level, and on other conditions.

Task description:

- Literature survey of control of inflow basins.
- Implementation of a dynamic simulator of the inflow basin including alternative level control systems in an appropriate programming environment. Real data from the VEAS WWTP are available.
- Candidates for level controller functions are PI (Proportional plus Integral) and MPC (Model-based Predictive Control).

Student category: Jianfeng Zhang

Practical arrangements: The student will work at the campus in Porsgrunn.

Signatures:

Student (date and signature): 31/5 - 16. Jianfeng Zhang

Supervisor (date and signature): 31/5 - 16 Finn Haugen

Appendix B (MPC)

Main_MPC_basin.m

```
% Model predictive con trol should based on measured inflow from vækerø and
internal flow, in order to get both stable pumps flow and level of basin.
last version (5.19)
clc, clear
%% PARAMETERS (Parameters should be called here including the measured inflow
and internal outflow)
Parameter;
% measured inflow and internal flow.
%% Initialization
N_p=10;
N_c=2;% change N_c in order to increase computation speed.
load F_inn_total  % loading F_inn_total
load F_pump_total(VEAS) % in order to compare the result of MPC and PI
contoller.
load L_measurement_MPC(no resampling) % in order to compare the result of MPC
and PI controller
load F_v(no resampling VEAS) % for plotting
%% Assume the simulating F_inn_total and F_measurement from V?kere to see if
the MPC can work well or not (for test run)
d_T=60*21.6; % d_T is designed to be 21.6mins which is correspond to
Time_delay 216 mins
tspan = ts:d_T:tf;% define the tspan for simulation
ts_o=0; tf_o=(600-10)*d_T; % change the tima_span_10^5
tspan_o=ts_o:d_T:tf_o;% define the tspan for ploting Flow.
ts_c=10*20*60; d_T_c=60*20; tf_c= 600*d_T_c;
tspan_c=ts_c:d_T_c:tf_c;% define the tspan for pump measurement from VEAS.
ts_c_o=0; tf_c_o=(600-10)*d_T_c;
tspan_c_o=ts_c_o:d_T_c:tf_c_o;
N_k=length(tspan);% no_samples=length(tspan);
ref=-13.2*ones(N_k,1);% the reference height for basin level(m)

dF_o=zeros(N_c,1);% Make the u_voltage initial value as 0.35;
F_init=2711*ones(N_c,1)*1;% the initial Pumpflow is defined ;
H_init=-13.31; %define the initial height as -11.42m;
F_pump_store=zeros(N_k,1); %here is to store the F_pump outflow
H_store=zeros(N_k,1);% here is to store the basin level;

%% Receding horizon strategy
% for loop should be built here in order to achive sliding
tic
for i=1:N_k
    tk = tspan_o(i); %automatically updated as i change, this is for the
sliding ;
    H_store(i)=H_init;
    [Opt_dF,fval,solutions] =
Optimize_Basin(dF_o,H_init,ref(i),F_init,N_p,N_c,d_T,tk,F_inn_total(i:i-
1+N_p));

    F_init=Opt_dF+F_init; % here is to use use the control signal for the
first step of horizon;(updating u_init)
    %take the first control action
F_con = F_init(1);% the FIRST PUMP flow will be taken action
F_pump_store(i)=F_con;% store the pump outflow for ploting ;
%warm start SO fmincon dont need to start everytime from 0; to save computing
time
dF_o = Opt_dF;
F(1)=F_inn_total(i);
F(2)=F_con;
H_updated=Diff_solver(H_init,d_T,tk,F);
H_init= H_updated(end,:)';
```

```
fprintf('Loop index:%d\n',i)
end
toc
%% Graghing
figure
subplot(3,1,1)
plot(tspan_o/60,F_pump_store,'r', tspan_c/60, F_pump_total(11:601),'b')
title('MPC vs Pump flow from VEAS')
xlabel('Time horizon(mins)')
ylabel('Pumps outflow(liter/s)  ')
legend('MPC pump outflow','Pump-flow measurement from VEAS')
grid on
subplot(3,1,2)
plot(tspan_o/60,H_store,'b',tspan_o/60,ref,'r',tspan_c/60,L_measurement(11:60
1),'g')
title('MPC vs Level measurement from VEAS')
xlabel('Time Horizon(mins)')
ylabel('Height(m)')
legend('Basin level','Level Setpoint','L-measurement from VEAS')
grid on
subplot(3,1,3)
plot(tspan_c/60,F_inn_total(11:N_k),'b',tspan_c_o/60,F_v(11:N_k),'r')
title('Total inflow vs measured inflow')
xlabel('Time Horizon(mins)')
ylabel('Flow rate(Liter/s)')
legend('Total inflow into basin','Measured inflow ')
grid on
```

Optimize_Basin.m

```
%% this file is used to get the optimized pump outlfow change
function [Opt_dF,fval,solutions] =
Optimize_Basin(dF_o,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total) % change
the parameters arguments
% %options for chosing the type of method for optimization and other options
% %ops =
optimset('Algorithm','sqp','Display','off','MaxIter',300,'UseParallel','alway
s');
% %other methods can also be used
 ops =
optimset('Algorithm','sqp','Display','off','MaxIter',300,'UseParallel','alway
s');
% %ops = optimset('Algorithm','active-
set','Display','off','MaxIter',300,'UseParallel','always')
% %define the function (separate .m file) where the objective function will
be calculated
dFLast = [];% Last place compute_both was called
myJ = [];% Use for objective at xLast
myG = [];% Use for nonlinear inequality constraint
myHeq = [];% Use for nonlinear equality constraint
Obj_fun =
@(dF)Obj_fun_MPC_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total);
% define the constraint function
Con_fun
=@(dF)Con_fun_MPC_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total);
%use the fmincon solver
[Opt_dF,fval,solutions] =
fmincon(Obj_fun,dF_o,[],[],[],[],[],[],Con_fun,ops);
function J =
Obj_fun_MPC_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total)
    if ~isequal(dF,dFLast) %check if computation is necessary
          [myJ myG myHeq] =
Compute_Both_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total);
          dFLast = dF;
      end
       %now compute objective function
       J = myJ;
   end

 function [G Heq] =
Con_fun_MPC_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total)
       if ~isequal(dF,dFLast) %check if computation is necessary
          [myJ myG myHeq]
=Compute_Both_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total);
          dFLast = dF;
       end
       %now compute constraints
       G = myG;
       Heq = myHeq;
 end
end
```

Compute_Both_Basin.m

```
%% version 1.0
function [myJ myG
myHeq]=Compute_Both_Basin(dF,H_init,ref,F_init,N_p,N_c,d_t,tk,F_inn_total)
% dF is the increased pump outflow each time inteval
% ref is the reference level height of basin
% F_init is the intial pump outflow which is calculated
% H_init is the initial height
% d_t is the time interval
% tk is current time
%Np is chosen as 10 and Nc can be chosen as 10, but can be changed and
%optimized in the future
Parameter;
R=1;
P=1;
J=0;
%calculate J, J comprises of two terms ,first term is the error between
%reference basin level and measured basin level; second time is the
%pump flow change.
for i = 1:N_p  % because the first sample is the initial values
   if i<=ceil(N_p/2)
      F_pump=dF(1)+F_init(1); % F is the pump outflow (total)
      second_term = dF(1)'*R*dF(1)/unit^2;
   else
       F_pump=dF(2)+F_init(2);
       second_term= dF(2)'*R*dF(2)/unit^2;
   end

      err = ref - H_init;
      first_term = err'*P*err;

      J = J+first_term + second_term;
      F(2)=F_pump;
      F(1)=F_inn_total(i);
      H_updated=Diff_solver(H_init,d_t,tk,F);
%        [Time,H_updated] = ode45(@Diff_basin,[tk+(i-1)*d_t
tk+i*d_t],H_init,[],F);
      % H_updated is the updated basin level for each iteration
      %now use the last value returned by the ode solver
      H_init = H_updated(end,:)';% update the basin level
end
myJ=J/2;

myHeq=[];% the equality constraint
myG=[dF-d_t*dF_limit;
   -dF-d_t*dF_limit;
   -H_init+H_pumpeinntak;
   ];% the inequality constraint
```

Diff_solver.m

```
% Diff_solver is used to solve discretized differential equation
function  H=Diff_solver(H_init,d_t,tk,F)
% here H_init is the initial value for Height, d_t is the time interval, tk
% is the starting time, F including F(1)= F_inn_total (total inflow
% calculated),F(2)=F_pump_flow (total pump flow calculated) (4.13)
% Euler forward method is taken here
Unit=1000;% from liter to cubic
N_l=round(d_t/60);
T_interval= 60;% time interval is 1 mins
for i=1:N_l
    H_updated=T_interval*(F(1)-F(2))/(Area_calculation(H_init)*Unit)+H_init;
    H_init=H_updated;
end
H=H_updated;
```

Appendix C (data re-processing)

Resample.m

```
%% Resample.m is used to get the resampled data.
Parameter;
fileID=fopen('logfile1.txt');% here is to the the file identifier
DATA_CELL=textscan(fileID,'%s %s %f %f %f %f %f %f');% here is to get the
data from the text file
fclose(fileID);
F_v=cell2mat(DATA_CELL(1,3));% F_v is the inflow from V?ker?
L_measurement=cell2mat(DATA_CELL(1,4))-H_0;% L_measurement equals IPU_lc01_er
which is level measurement.
F_pump_7=cell2mat(DATA_CELL(1,7));%F_pump_7 equals IPU_FB07 is flow
measurement of pump no. 7.
F_pump_8=cell2mat(DATA_CELL(1,8));%F_pump_8 equals IPU_FB07 is flow
measurement of pump no. 8.
fileID=fopen('logfile2.txt');
DATA_CELL=textscan(fileID,'%s %s %f %f %f %f %f %f');
fclose(fileID);
F_pump_1=cell2mat(DATA_CELL(1,3));%F_pump_1 equals IPU_FB01 is flow
measurement of pump no. 1.

F_pump_2=cell2mat(DATA_CELL(1,4));%F_pump_2 equals IPU_FB02 is flow
measurement of pump no. 2.
F_pump_3=cell2mat(DATA_CELL(1,5));%F_pump_3 equals IPU_FB03 is flow
measurement of pump no. 3.
F_pump_4=cell2mat(DATA_CELL(1,6));%F_pump_4 equals IPU_FB04 is flow
measurement of pump no. 4.
F_pump_5=cell2mat(DATA_CELL(1,7));%F_pump_5 equals IPU_FB05 is flow
measurement of pump no. 5.
F_pump_6=cell2mat(DATA_CELL(1,8));%F_pump_6 equals IPU_FB07 is flow
measurement of pump no. 6
N=length(L_measurement);% Get the length of L_measurement vector
X=1:20:20*N;
X_new=1:20*N;
F_v_New=interp1(X,F_v,X_new,'spline');
L_measurement_New=interp1(X,L_measurement,X_new,'spline');% get the resampled
level measurement.
F_pump_7_New=interp1(X,F_pump_7,X_new,'spline');
F_pump_8_New=interp1(X,F_pump_8,X_new,'spline');
F_pump_1_New=interp1(X,F_pump_1,X_new,'spline');
F_pump_2_New=interp1(X,F_pump_2,X_new,'spline');
F_pump_3_New=interp1(X,F_pump_3,X_new,'spline');
F_pump_4_New=interp1(X,F_pump_4,X_new,'spline');
F_pump_5_New=interp1(X,F_pump_5,X_new,'spline');
F_pump_6_New=interp1(X,F_pump_6,X_new,'spline');
F_pump_total_New=F_pump_1_New+F_pump_2_New+F_pump_3_New+F_pump_4_New+F_pump_5
_New+F_pump_6_New+F_pump_7_New+F_pump_8_New;
```

Extract_for_inflow.m

```
%% this m.file is created to make extract the total flow from F_inn_total_new
for each 21.6 mins
load F_inn_total_new(resampling);
dt=22;% dt is the length of time step
 for i=1:N_k_new
    F_inn_total_extracted(i)=F_inn_total_new(1+(i-1)*dt);
end
%% this m.file is created for MPC to use the F_inn_total for another time
span 10^5 to 10^5 +21.6*600;
% N_shift=10^5;% new time span starts from time 10^5 mins
% N_k=length(F_inn_total_new)-N_shift;
% N_k_new=floor(N_k/22);
% F_inn_total_extracted=zeros(N_k_new,1);
% for i=1:N_k_new
%    F_inn_total_extracted(i)=F_inn_total_new(1+(i-1)*dt+N_shift);
% end
%% this m.file is created for MPC to use the F_inn_total for another time
span 1.5^5 to 1.5^5 +21.6*600;
N_shift=1.5*10^5;% new time span starts from time 10^5 mins
N_k=length(F_inn_total_new)-N_shift;
N_k_new=floor(N_k/22);
F_inn_total_extracted=zeros(N_k_new,1);
for i=1:N_k_new
    F_inn_total_extracted(i)=F_inn_total_new(1+(i-1)*dt+N_shift);
end
```

Appendix D (basin simulator)

Basin_simulator_v2.m

```
%% This section is to extract the related data from logfiles
Parameter;
fileID=fopen('logfile1.txt');% here is to the the file identifier
DATA_CELL=textscan(fileID,'%s %s %f %f %f %f %f %f');% here is to get the
data from the text file
fclose(fileID);
Reference=cell2mat(DATA_CELL(1,5));% Reference is the level reference for
basin.
F_v=cell2mat(DATA_CELL(1,3));% F_v is the measured inflow
L_measurement=cell2mat(DATA_CELL(1,4))+H_0;% L_measurement equals IPU_lc01_er
which is level measurement.
F_pump_7=cell2mat(DATA_CELL(1,7));%F_pump_7 equals IPU_FB07 is flow
measurement of pump no. 7.
F_pump_8=cell2mat(DATA_CELL(1,8));%F_pump_8 equals IPU_FB07 is flow
measurement of pump no. 8.
fileID=fopen('logfile2.txt');
DATA_CELL=textscan(fileID,'%s %s %f %f %f %f %f %f');
fclose(fileID);

F_pump_1=cell2mat(DATA_CELL(1,3));%F_pump_1 equals IPU_FB01 is flow
measurement of pump no. 1.

F_pump_2=cell2mat(DATA_CELL(1,4));%F_pump_2 equals IPU_FB02 is flow
measurement of pump no. 2.
F_pump_3=cell2mat(DATA_CELL(1,5));%F_pump_3 equals IPU_FB03 is flow
measurement of pump no. 3.
F_pump_4=cell2mat(DATA_CELL(1,6));%F_pump_4 equals IPU_FB04 is flow
measurement of pump no. 4.
F_pump_5=cell2mat(DATA_CELL(1,7));%F_pump_5 equals IPU_FB05 is flow
measurement of pump no. 5.
F_pump_6=cell2mat(DATA_CELL(1,8));%F_pump_6 equals IPU_FB07 is flow
measurement of pump no. 6
N=length(L_measurement);% Get the length of L_measurement vector
X=1:20:20*N;
F_pump_total=F_pump_1+F_pump_2+F_pump_3+F_pump_4+F_pump_5+F_pump_6+F_pump_7+F
_pump_8;% F_pump_total is the total pump outflow based on data given.

%% this section is to use the data acquired above and implement simulation of
inflow basin
N_delay=round(T_delay/T_interval); % calculate the numbers of time_intervals
in time delay;
Tspan_N=N_delay*20:20:(N-2)*20;% Define time horizon for interval inflow
simulation.
Tspan_O=0:20:(N-2-N_delay)*20;% Define time horizon for measured inflow.
for i=1:N-1-N_delay
dhdt=L_measurement(1+i+N_delay)-L_measurement(i-1+N_delay);
Area=Area_calculation(L_measurement(i+N_delay));
F_inn_total(i)=dhdt*unit*Area/(2*dt)+F_pump_total(i+N_delay);%the total
F_inn_total is calculated from time 2+T_d
F_interval(i)=F_inn_total(i)-F_v(i)/Ratio_v2t;%the  F_feedback is calculated
from time 2+T_delay
end

subplot(2,1,1)
 plot(Tspan_O,F_v(1:N-1-N_delay),Tspan_N,F_pump_total(1+N_delay:N-1));
 title('Measured inflow vs Total pumps outflow')
 xlabel('Minutes')
 ylabel('Liter per second')
```

```
legend('Measured inflow ','Total pumps outflow')
subplot(2,1,2)
plot(Tspan_N,F_interval);

title('Interval flow ')
xlabel('Minutes')
ylabel('Liter per second')
% save the F_interval,Measured inflow and total pumps outflow from
% workspace.
```

Parameters.m

```
H_tak=-10.8; % the ceiling of channel
H_sp_hoy=-12; % the absolute high setpoint
H_sp_h=3;% the related high setpoint
H_sp_lav=-13.2;% the absolute low setpoint
H_sp_l=1.8;% the related low setpoint
H_kanalgulv=-14.4;% the floor of channel
H_sensor=-14.6; % the height of sensor
H_0= -15; % baseline height
H_sp_0=0; % the related baseline setpoint
H_overkant_sug_pump=-15.4;%
H_pumpeinntak=-17; % the height of pump intake
H_sumpgulv=-17.4; % the height of the floor of basin
u=1.12;% unit is rad % in the document it s written 0.112
R_1=15.2;  % the radius of the basin
L_sumpbakke=3; % the length of the back of basin
L_sumpgulv=1.6; %  the floor of basin
L_bue=17;
L_sidekanal_maks=30;
S_sidekanal=1/6;% this is the stig angle of sidekanal;
B_sidekanal=5.2;%
B_kanal=4.7;% the width of channel
Z_1=4.4;
L_kanal=58.4;% the length of channel
D_0=3.5;
a=7.5e-4;% the angle of tunnel(rad)
Area_part= 1.2828e4;% the area of tunnel when level is higher than -10.8;

dt=60*20; % dt is the measurement time for the change of height (s);setting
the time interval
dt_new=60;% dt_new is for
Ratio_v2t=0.65;% this is the ratio of flow of V?ker? and total inflow from
tunnel
T_delay=216;% time delay(minutes) that wasterwater flows from V?ker? to
basin.
T_interval=20;% sampling time (minutes)


unit=1000;
dF_limit=300/15;% the change of outflow rate from pump is 300(L/s)/15mins
```

Area_calculation.m

```
function Area=Area_calculation(H)
Parameter;
if H>=H_sumpgulv && H<=H_kanalgulv
   Area=(pi*R_1^2-pi*(R_1-(L_sumpgulv+H-H_sumpgulv))^2)*u/(2*pi);
else
   if H>H_kanalgulv && H<=H_tak
      Area=Area_calculation_H(H);
```

```
    else
        Area=Area_part;
        %%fprintf('the level of inflow basin is not in regular range');
    end
end
end
```

Area_calculation_H.m

```
%% here is the function of calculating the area when level >H_kanalgulv and <H_tak
function Area=Area_calculation_H(H)
Parameter;
Z_2=Z_1*cos(u/2);
A_kanal=L_kanal*B_kanal;% the area of channel
L_sidekanal=(H-H_kanalgulv)/S_sidekanal;
A_sidekanal=L_sidekanal*B_sidekanal;

A_sump=pi*R_1^2*u/(2*pi)-(Z_2*B_kanal)/2;%the area of basin
D_1=D_0/sin(a);
r=0.5*D_1;
A_sirk=pi*r^2;

x_1=2*r; x_2=0;
h_1=H_kanalgulv;h_2=H_tak;
x=x_1+((x_2-x_1)/(h_2-h_1))*(H-h_1);
v=acos((r-x)/r);
y=sqrt(r^2-(r-x)^2);
A_triangel=abs(r-x)*y/2;

A_sekant=r^2*v/2;
A_segment=A_sekant-A_triangel;
A_tunnel_part_sirk=A_sirk-2*A_segment;
b=0.5*D_0;
A_tunnel_part_ellipse=A_tunnel_part_sirk*(b/r);
Area= A_tunnel_part_ellipse+A_sidekanal+A_kanal+A_sump;


End
```

Appendix E (PI controller)

PI controller.m

```
%% PI controller
load F_inn_total_new(resampling);
    Parameter; % loading the parameters

    z_k=0; % Initialization of state;
    e_k=0; % Initialization of error;


    K_p=-1/(T_c*K_gain);% equation from the table of skogestad tuning method
    T_c=9000; %last time found the good tuning parameter

    T_i=4*T_c;
    T_step=60;% T_interval is time step 1 mins,60s
    T_interval=21.6;% T_interval is the time interval 21.6 mins
% Tspan=0:590*T_interval;% Tspan is the simulation time span for PI
controller
    Tspan=0:(700-10)*21.6;
    N_k=length(Tspan);
    Ref=-13.2;
    F_pump_store=zeros(N_k,1);

   F_pump=2560;

    H_store=zeros(N_k,1);
    H_init=-13.2;
    H_filted=-13.2;
    T_f=60*5;

for i=1:N_k
 z_k1= z_k+T_step*(K_p/T_i*e_k);  %PI controller state;

 F_pump_updated=K_p*e_k+z_k1;
  %PI controller;
   dF_pump=F_pump_updated-F_pump;
z_k=z_k1;% update the state for next loop;
%% anti wind-up
 if dF_pump> dF_limit %anti-windup
      dF_pump =dF_limit; %Passing the controller output to system
F_pump_updated=F_pump+dF_pump;
 else
     if dF_pump<-dF_limit    % anti-windup
      dF_pump=-dF_limit; %Passing the controller output to system
 F_pump_updated=F_pump+dF_pump;
       else
          F_pump_updated=F_pump+dF_pump; %Passing the controller output to
system
    end;
 end
 F_pump=F_pump_updated;
 F_inn_total_new(i);

% %% Simulator (Basin)
    H_updated=T_step*(F_inn_total_new(i)-
    F_pump)/(Area_calculation(H_init)*unit)+H_init;
    H_init=H_updated;% H_init will be taken by controller;
  % update the error for PI controller;
    F_pump_store(i)=F_pump;

 %% Lowpass Filter
```

```
A=T_step/(T_step+T_f);
H_filted_new= (1-A)*H_filted+A*H_init;
H_filted=H_filted_new;
%    e_k=Ref-H_filted;
   H_store(i)=H_filted;
e_k=Ref-H_filted;
H_store(i)=H_filted;
end
plot(Tspan,Ref,'r',Tspan,H_store,'b');
xlabel('Time(min)');
ylabel('Height(m)')
legend('Level setpoint','Basin level')
grid on
```

Skogestad_tuning.m

```
load F_inn_total_new(resampling);
    Parameter; % loading the parameters
    T_step=60;% T_interval is time step 1 mins,60s
    T_interval=21.6;% T_interval is the time interval 21.6 mins
Tspan=0:590*T_interval;% Tspan is the simulation time span for PI controller

    N_k=length(Tspan);
    F_pump(1:N_k)=2500;
    H_store=zeros(N_k,1);
    H_init=-14.3;
    H_filted=-14.3;
    T_f=60*5;

for i=1:N_k
%% Simulator (Basin)

  H_updated=T_step*(2600-F_pump(i))/(Area_calculation(H_init)*unit)+H_init;
    H_init=H_updated;% H_init will be taken by controller;
   H_store(i)=H_init;
   A_store(i)=Area_calculation(H_init);
 %% Lowpass Filter

A=T_step/(T_step+T_f);
H_filted_new= (1-A)*H_filted+A*H_init;
H_filted=H_filted_new;
H_store(i)=H_filted;
end
subplot(2,1,1)
plot(Tspan,H_store);
subplot(2,1,2)
plot(Tspan,A_store);
```

Appendix F (data analysis)

Delta_pump.m

```
ts=10*21.6*60+1.5*10^5*60; d_T=60*21.6; tf=600*d_T+1.5*10^5*60;% d_t is
designed to be 21.6mins which is correspond to Time_delay 216 mins
tspan = ts:d_T:tf;% define the tspan for simulation
ts_o=0+1.5*10^5*60; tf_o=(600-10)*d_T+1.5*10^5*60; % change the
tima_span_10^5
tspan_o=ts_o:d_T:tf_o;% define the tspan for ploting Flow.
ts_c=10*20*60+1.5*10^5*60; d_T_c=60*20; tf_c= 600*d_T_c+1.5*10^5*60;
tspan_c=ts_c:d_T_c:tf_c;% define the tspan for pump measurement from VEAS.
N_k=length(tspan_o);
for i=1:590
  delta_pump_MPC_tspan_3(i)= F_pump_store_tspan_3rd(i+1)-
F_pump_store_tspan_3rd(i);

  delta_pump_VEAS_tspan_3(i)=F_pump_total(7511+i)-F_pump_total(7510+i);
end
plot(tspan_o(2:N_k)/60,delta_pump_MPC_tspan_3,'r', tspan_c(2:N_k)/60,
delta_pump_VEAS_tspan_3,'b')
xlabel('Time horizon(min)');
ylabel('Pump flow variation(liter/s)');
legend('MPC pump flow','VEAS pump flow');
grid on
```

Violation_of_limit.m

```
load('H_store(best run MPC with real data).mat')
load('L_measurement_MPC(no resampling).mat')
load('H_store_tspan_2nd.mat')
load('H_store_tspan_3rd.mat')
load('F_pump_store_baseline(1st run MPC with real data)')
load('H_store_baseline(1st run MPC with real data)')

j=1;
k=1;
violate_area_mpc=0;
violate_area_VEAS=0;
for i=1:591
    if H_store_tspan_3rd(i)<-13.5;
       H_violate_mpc(j) =-13.5-H_store_tspan_3rd(i);

       violate_area_mpc=violate_area_mpc+H_violate_mpc(j)*21.6;

       j=j+1;
    else if H_store_tspan_3rd(i)>-12.9
         H_violate_mpc(j)= 12.9+H_store_tspan_3rd(i);
         violate_area_mpc=violate_area_mpc+H_violate_mpc(j)*21.6;
         j=j+1;
        end
    end
end
 for  i=1:591
    if L_measurement(7510+i)<-13.5
    H_violate_VEAS(k)=-13.5-L_measurement(7510+i);
    violate_area_VEAS=violate_area_VEAS+H_violate_VEAS(k)*20;
    k=k+1;
    else if L_measurement(7510+i)>-12.9
           H_violate_VEAS(k)=12.9+L_measurement(7510+i);
```

```
            violate_area_VEAS=violate_area_VEAS+H_violate_VEAS(k)*20;
            k=k+1;
        end
    end
end
```

Total_variance.m

```
load delta_pump_MPC_tspan_1;
load delta_pump_VEAS_tspan_1;
load delta_pump_MPC_tspan_2;
load delta_pump_VEAS_tspan_2;
for i=1:590;
    total_variance_MPC_tspan_3(i)=abs(delta_pump_MPC_tspan_3(i));
    total_variance_VEAS_tspan_3(i)=abs(delta_pump_VEAS_tspan_3(i));
end
% total_variance_mpc_tspan_1=   3.9941e+04;
% total_variance_VEAS_tspan_1 = 7.6073e+04;
% total_variance_mpc_tspan_2 =   2.7995e+04;
% total_variance_VEAS_tspan_2 = 8.6229e+04;
% total_variance_mpc_tspan_3 = 3.4282e+04;
% total_variance_VEAS_tspan_3 = 9.5898e+04;
```