

Master's Thesis 2014

Candidate: Bhuwan Dhakal (120289)

Title: Stochastic Flood Control of Kragerø
Waterways using MPC

TelemarkUniversityCollege



Faculty of Technology

Kjølnes

3914 Porsgrunn

Norway

Lower Degree Programmes – M.Sc. Programmes – Ph.D. Programmes

TFver. 0.9



Telemark University College

Faculty of Technology
M.Sc. Programme

MASTER'S THESIS, COURSE CODE FMH606

Student: Bhuwan Dhakal (120289)

Thesis title: Stochastic Flood Control using MPC of Kragerø Waterways

Signature:

Number of pages: 80

Keywords: Model Predictive Control, Model Prediction, Simulation, Linearization, Quadratic Programming, Optimization, Stochastic Process, Lake Toke, State, Validation

Supervisor: Prof. Bernt Lie sign.:

2nd Supervisor: sign.:

Censor: sign.:

External partner: Skagerak Energi, Ingvar Andreassen sign.:

Availability:

Archive approval (supervisor signature): sign.: **Date :**

Abstract:

The model predictive control (MPC) have been a remarkable benchmark in many of the control process application over past few decades because of its ability to handle constraints and multivariable nonlinear systems with time delay and the use of feedback from measurement using latest information. A MPC application is developed and implemented predicting flood gate opening in this thesis work to control the flooding situation that normally occurs in Kragerø Waterways.

The system we have considered herein is the model of lake Toke where we have developed a dynamic model for lake Toke using general mass balance equation. The necessary measurements were described and predicted model for current turbine flow was developed. The model was linearized and MATLAB simulations were performed for validation using historic data provided from Skagerak Energi. The sensitivity of the parameters used in the model was analyzed. A quadratic Programming is developed in order to solve the optimization problem as well as constraints and slack variables were defined to satisfy the system requirements.

Both linear and nonlinear MPC were performed using MATLAB. Finally, a simple stochastic MPC analysis was done using real values and assumed values by adding random Gaussian noise.

MPC application was simulated introducing model error where the future inflow of water was unknown gave almost desired reference level without any violation of system boundary indicates MPC can handle the unknown inflow predictions or big uncertain inputs can be handled by MPC.

Telemark University College accepts no responsibility for results and conclusions presented in this report.

Table of contents

1	INTRODUCTION	9
1.1	BACKGROUND	9
1.2	PREVIOUS WORK.....	9
1.3	OBJECTIVES	10
1.4	OVERVIEW OF REPORT	11
2	SYSTEM DESCRIPTION	12
2.1	LAKE TOKE OVERVIEW	12
2.2	DALSFOSS DAM DATA MEASUREMENTS	14
2.2.1	<i>Level Description of Dalsfoss Dam</i>	14
2.3	FUNCTIONAL DESCRIPTION.....	17
2.3.1	<i>Inlet flow V_i</i>	18
2.3.2	<i>Turbine Flow V_t</i>	18
2.3.3	<i>Prediction Models</i>	20
	PROBLEM DESCRIPTION	21
2.3.4	<i>System Requirements</i>	21
3	MODEL DEVELOPMENT & VALIDATION	23
3.1	INTRODUCTION.....	23
3.2	MODEL DESCRIPTION	23
3.2.1	<i>Model Summary & Model Parameters</i>	24
3.3	MODEL DEVELOPMENT	25
3.3.1	<i>Assumptions and Clarifications</i>	27
3.3.2	<i>Model for h_1 & h_2</i>	27
3.3.3	<i>Model for V12</i>	31
3.3.4	<i>Free Flood gate & Model for V_g</i>	33
3.3.5	<i>Surface Area of Lake Toke</i>	36
3.4	LINEARIZATION	37
3.4.1	<i>Linear Analysis</i>	37
3.4.2	<i>Discretization</i>	39
3.5	SENSITIVITY OF PARAMETERS	40
4	MODEL PREDICTIVE CONTROL (MPC)	42
4.1	OPTIMAL CONTROL AND QUADRATIC PROGRAMMING (QP)	44
4.2	CONSTRAINTS HANDLING	46
4.2.1	<i>Equality Constraints</i>	46
4.2.2	<i>Inequality constraints & slack variables</i>	48
4.2.3	<i>MPC Implementation</i>	50
4.2.4	<i>Result Intrepretation</i>	51
5	STOCHASTIC MPC	57
5.1	BACKGROUND	57
5.2	MARKOWITZ PORTFOLIO OPTIMIZATION	57

5.3 STOCHASTIC ANALYSIS.....	58
6 CONCLUSION AND FUTURE WORK	60
REFERENCES.....	62
APPENDICES	63
APPENDIX 1: TASK DESCRIPTION	63
APPENDIX 2: LINEARIZATION AND DISCRETIZATION OF THE MODEL	66
APPENDIX 3: MATLAB FILE FOR SIMULATING THE DATA (x_o, x_q, Vt, Vg and We).....	69
APPENDIX 4: MATLAB CODE FOR MODEL VALIDATION AND SIMULATION	71
APPENDIX 5: MATLAB FILE FOR LINEARIZATION AND DISCRETIZATION	73
APPENDIX 6: MATLAB FILE FOR MPC	74
APPENDIX 7: MATLAB FILE TO RUN MPC.....	78
APPENDIX 8: MATLAB FILE FOR NON-LINEAR MODEL	79

Preface

This is a Masters thesis work titled “Stochastic Flood Control of Kragerø Waterways using MPC” carried out in Telemark University College (TUC), Porsgrunn, Norway for the course fulfillment of Master of Science in systems and Control Engineering. This report is a detail of thesis work carried under the supervision of Professor Bernt Lie on Stochastic Flood Control of Kragerø Waterways using MPC.

I have tried to fulfill all the tasks specified in task description of thesis work, but due to my part time work I had to go for work, so some of the tasks may not be discussed in depth. The task requires the knowledge on MATLAB programming, MPC and modelling and simulation of dynamic systems. The report has separate Appendix at the end of this report which includes the necessary mathematical solutions and MATLAB codes.

I would like to express our gratitude to my supervisor Prof. Bernt Lie for his constant support, incredible guidance and straight forward direction in need. He was always there for us to rectify my work and provided his valuable suggestions too. He spent many of his valuable hours to go through my document and corrected.

I could not stay without saying thanks to the external partners from Skagerak Energi, Ingvar Andreassen for all the information provided to the success of this thesis. I would especially like to show my appreciation and thank to the group members of the project work ‘flood control using MPC of Kragerø waterways, 2013’ performed by group of students of SCE in Telemark University College for their relevant work and assistance.

Finally, I am pleased to Telemark University College for providing the opportunity to perform this thesis work and I am grateful to my colleagues for their help and support for carrying this work in to this level.

Porsgrunn ,3 June 2014

Bhuwan Dhakal

Nomenclature

Lists of symbols, abbreviations, and subscripts used in the thesis are included here.

Symbol	SI unit	Description
AHRV	m	Limit for highest regulated water level due to operating Procedure throughout a year
ALRV	m	Limit for lowest regulated water level due to operating Procedure throughout a year
HRV	m	Limit for highest regulated water level
LRV	m	Limit for lowest regulated water level
MPC	-	Model Predictive Control
TUC	-	Telemark University College
NVE	-	Norwegian Water Resources and Energy Directorate
MATLAB	-	Matrix Laboratory
QP	-	Quadratic Programming

Overview of tables and figures

Table 2-1: Operating Procedure of Lake Toke (Thorsen, 2011)	22
Table 3-1: Parameters for Lake Toke	24
Table 3-2: Operating conditions for validating model.....	24
Figure 2-1: Map of Lake Toke displaying upper and lower Toke along with available measurement points (Lie, 2013)	12
Figure 2-2: Kragerø waterways from Lake Toke to Kilsfjorden(Lie, 2013)	13
Figure 2-3: General overview of water level of Dalsfoss dam	14
Figure 2-4: Presentation of actual Level in front of Dalsfoss Dam	15
Figure 2-5: Level in front of Dalsfoss Dam (x_o), and at after turbine(x_u) and at Quay (x_q)..	16
Figure 2-6: Downstream flow from the dam V_o relating Quay level x_q and Quadratic fitting of V_o with x_q	17
Figure 2-7: Schematic process diagram for the model of Kragerø waterways.....	18
Figure 2-8: Turbine flow (V_t) compared with total power production (We) and level difference $ht=(x_o - x_q)$	19
Figure 2-9 Correlation of power prediction between measured data (o,blue),linear prediction of We from correlation $We= 0.0075 \cdot x_o - x_q \cdot V_t - 0.2477$ (black) and theoretical value $We = F ((x_o - x_q)V_t)$ (red)	20
Figure 2-10: Requirement for water level of Lake Toke and Lake level (2001, 2002 & 2009) ((Gøthesen et al., 2013).....	22
Figure 3-1: Possible layout of model of Lake Toke	26
Figure 3-2: Levels h_1, h_2 in two compartments, hg_{max} with given parameters and operational conditions in initial states ($h_1(t=0)=2.5, h_2(t=0)=2.5$) at time period 40000 sec	28
Figure 3-3: Levels h_1, h_2 in two compartments, hg & hg_{max} with given parameters and operational conditions in steady state ($h_1(t=0)=5.4, h_2(t=0)=4.75$) at time period 40000 sec	29
Figure 3-4: Levels h_1, h_2 in two compartments, hg & hg_{max} with given parameters and operational conditions at time period 100000 sec. i.e. 28.7hr	30
Figure 3-5: Levels h_1, h_2 in two compartments, hg & hg_{max} with given parameters and operational conditions at time period 1000000 sec. i.e. 11.6 days	31
Figure 3-6: Simulation of inter compartmental flow V_{12} , along with gate flow V_g and disturbance flow V_{in} in steady state. Time span: 40000 seconds, i.e. 11.1 hr	32
Figure 3-7: Simulation of inter compartmental flow V_{12} , along with gate flow V_g and disturbance flow V_{in} in steady state. Time span: 100000 seconds, i.e. 28.7hr	32

Figure 3-8: Simulation of inter compartmental flow V_{12} , along with gate flow V_g and disturbance flow V_{in} in steady state. Time span:1000000 seconds, i.e. 11.6 days	33
Figure 3-9: Sketch of free flood gate (Lie, 2014)	34
Figure 3-10: Simulation of gate opening hg at time span 40000 sec.	34
Figure 3-11: Simulation of gate opening hg at time span 28.7 hours	35
Figure 3-12:Simulation of gate opening hg at time span 11.6 days.....	35
Figure 3-13: Simulation of surface area (A) as function of level h	36
Figure 3-14: Levels h_1, h_2 in two compartments with 5% (0.0525)increased α parameter and operational conditions in steady state ($h_1(t=0)=5.4, h_2(t=0)=4.75$) at time period 40000 sec	40
Figure 3-15: Levels h_1, h_2 in two compartments with 5% (0.021)increased β parameter and operational conditions in steady state ($h_1(t=0)=5.4, h_2(t=0)=4.75$) at time period 40000 sec	41
Figure 3-16: Levels h_1, h_2 in two compartments with 5% (0.021)increased K_{12} parameter and operational conditions in steady state ($h_1(t=0)=5.4, h_2(t=0)=4.75$) at time period 40000 sec	41
Figure 4-1:Block diagram of MPC algorithm.....	42
Figure 4-2: Illustration of receding horizon principle along with computed input sequence over a defined prediction horizon.	43
Figure 4-3:MPC simulation of linear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 5.5 days over a time stamp of 10 days.	52
Figure 4-4:MPC simulation of nonlinear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 2.5 days over a time stamp of 10 days	53
Figure 4-5:MPC simulation of linear model with predicted inflow $500 \text{ m}^3/\text{s}$ for approximately 5.5 days over a time stamp of 10 days	54
Figure 4-6:MPC simulation of nonlinear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 2 days over a time stamp of 10 days	55
Figure 4-7:MPC simulation of nonlinear model with predicted inflow $900 \text{ m}^3/\text{s}$ for approximately 2 days showing the extreme flooding condition over a time stamp of 10 days	56
Figure 5-1:MPC simulation with assumed inflow of $300 \text{ m}^3/\text{s}$ in linear model.....	58
Figure 5-2:MPC simulation with random inflow in linear model	59

1 Introduction

Flood controls are one of the major challenges among the natural disasters that occur over the world as they are affected any time without any notifications and may create severe damages for the creatures and environment itself. Meanwhile, the process control advancement in technology had made it efficient to some extent and possible to control and predicts the flooding environments and troubleshoots these types of natural disasters. This thesis is all about the controlling of flood using MPC of the Kragerø waterways. This thesis is carried out in TUC in cooperation with a Norwegian utility company Skagerak Energi.

The advanced control strategies such as Model Predictive Control (MPC), more specifically Deterministic MPC and/or Stochastic MPC may be used to predict and control in many control and process applications for the current and future state. In this project, a model predictive control approach for random flooding environment is assumed and MPC control strategy is applied in the flood gates. The flood gates opening are operated by the controller measuring the water level. The controller receives the current state information of water level and operates each time continuously, minimizing the probability of occurrence of flood.

1.1 Background

The Kragerø waterway is located in south-east of Norway. The waterways usually experience flooding environment every year in April. The flooding is normally caused by the snow melting and rain. The water levels of the lake Toke are strictly regulated by the authorities during the different time annually, which may avoid the floods and helps to maintain the ecological and economical values. The Skagerak Energi which operates hydropower plants on the waterways and is currently handling this project for better operation and fulfillment of those requirements with optimum power generation for less wastage of water.

The aim of the company is to construct a new floodgate for the controlling of floods that arises in the waterways. There are already two existing floodgates that are manually operated by a specialist operator who is nearly to get retirement and the main emphasis of this work is to purpose a new automatic control system that is implemented for a trial period before the retirement of operator.

1.2 Previous work

It is of great importance to deal with some literatures that have been done previously relating to the flood control using MPC of Kragerø waterways. Numerous research have been done on the field of flood control using advanced control strategies like MPC and been implemented, some of the relevant research work to this thesis are described shortly.

Optimization and control of Kragerø waterways was carried out using MPC and PI algorithm in MATLAB. A hydrology model had been investigated and model for one state had been developed in 2011 by Hege Marie Thorsen in TUC(Thorsen, 2011).A group project work ‘flood control using MPC of Kragerø waterways’ was carried out by M. Sc. Students in Telemark University College, Norway and developed a control system using MPC. The linearized dynamic models for one state and two states were developed and validated. The developed models were implemented using MPC for the control purpose and found worked well for the given requirements(Gøthesen et al., 2013). In my thesis work the future prediction of flooding environment and controlling using Stochastic MPC will be purposed; the models will be developed, deployed and tested for using MPC for future data models as well.

Moreover, an adaptive multi model predictive control (AMPC) method was purposed as an alternative method for existing flood control technique using multiple models at the flood of Wivenhoe Dam, Queensland, Australia in 2011, which used independent MPC controller Kalman filter. The set of state space models were developed using disturbances and MPC calculated the control in each time sequence on the other hand, Kalmanfilter was used to predict the system state at each time stamp(Delgoda et al., 2012).

The approximate mathematical model instead of hydrodynamic equations of Saint-Venantwas developed for the flood control of rivers to reduce the computational complexity by approximating the dynamics of every reach with a linear model together with nonlinear model of gate equation with high accuracy to use directly the approximate models in the design of controller. The set point control was achieved by minimizing the deviation of the most important water levels from their set points(Breckpot, 2013). I will use some of these mentioned strategies to predict the system state for MPC part and model development.

1.3 Objectives

Based on the facts obtained from above discussion, we are going to use the Stochastic MPC for the flood control of Kragerø waterways. We will use some of these used methods to find the dynamic models and also use more advanced control technique such as Stochastic MPC algorithm for the system. Another most important part of this project was development and validation of the model of Lake Toke which are actively involved to influence the system. The main objectives of this project work can be summed up to following points:

- Analyzing the level measurements, current and future inflow predictions, current and future turbine production flow for managing floods in Lake Toke.
- Deduce the dynamic model of relevant water levels at Lake Toke and validation against experimental/historical data.
- Development and testing of MPC solution based on purposed deterministic inflow from a hydrological model and production flow through the turbine.
- Use Markowitz Portfolio Optimization for stochastic MPC.
- Comparison between the use of Stochastic MPC and use of deterministic MPC.

1.4 Overview of Report

The report has been organized according to the objectives of this thesis and work flow. The main work starts with the development of dynamic model and ends with the implementation of stochastic MPC. The core report is of 52 pages excluding some formal pages (like abstract, table of contents, preface, etc.) at the beginning. The main report has been divided in to 6 chapters and each chapter includes specific task.

In *Chapter 1* I will discuss about the *Introduction* of the thesis and importance of this thesis work, *Chapter 2* deals with the *System Description*, *Chapter 3* includes *Model development and Validation*, in *Chapter 4* I will discuss about *Model Predictive Control (MPC)* similarly, *Chapter 5* includes *Stochastic MPC optimization* and finally *Chapter 6* finishes with *Discussion and Conclusions*.

At the end of this thesis, appendix of relevant mathematical expressions and MATLAB codes are included.

2 System Description

The functional description of the Kragerø waterways illustrates the overview of the system that we are going to develop and implement for controlling of flooding situations. We will define a system along with inputs, outputs, disturbances and system state from the available data/information, level measurements and production plans that are provided by Skagerak Enegr and based on these information we further do analysis on the nature of system and future predictions and make some assumptions for the development of dynamic model of Lake Toke. We will furthermore use some of this relevant information in MPC too.

In this section we will deal on the details of available data from the authority and functional description of the Lake Toke.

2.1 Lake Toke Overview

Kragerø waterway also known as Tokevassdraget lies in the south-west of Telemark County of Norway covering the catchment area of approximately 1156 km² and surface area of approximately 32 km² and can hold about 150 millions m³ of water. The average annual water inflow to the lake is approx. 24 m²/s and the water residence time is about 72 days. For the simplicity the lake Toke can be divided into two parts: Upper Toke and Lower Toke as shown in Figure 2-1(Lie, 2013a).

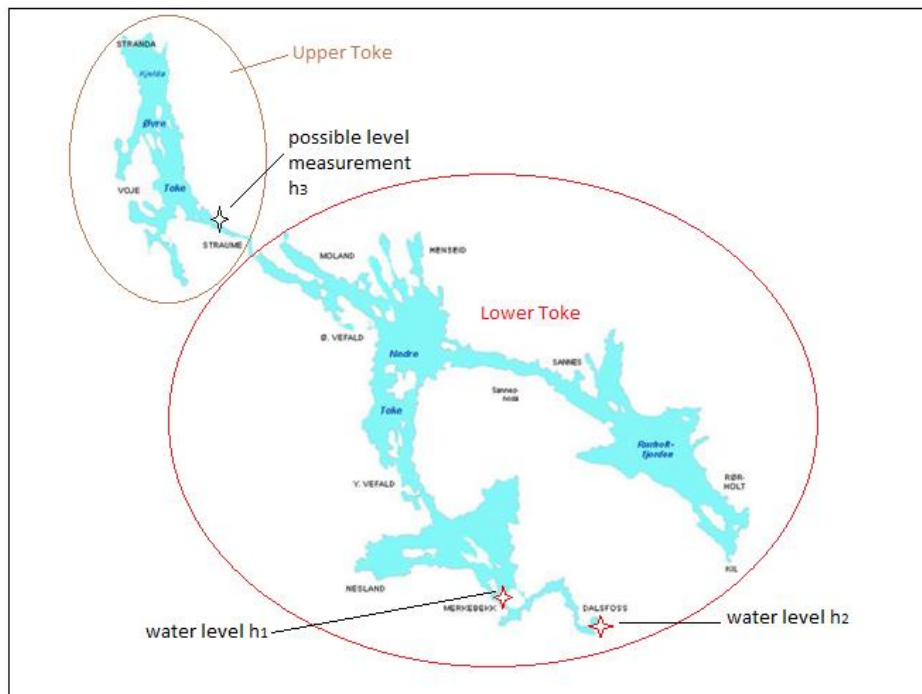


Figure 2-1: Map of Lake Toke displaying upper and lower Toke along with available measurement points (Lie, 2013)

As shown in the Figure 2-1, the Lake Toke can be splitted into two parts where the water levels are currently being measured are shown with red starred points: the water level h_1 at Merkebekk and h_2 at Dalsfoss in lower Toke. Also, it is possible to take third level measurement h_3 from the upper Toke in Straume ‘starred with black’ in future and can be modeled the lake more efficiently for the control purpose of flooding environment. These levels h_1 and h_2 are used to describe the model of the Lake in later chapters.

Due to the large diversity of water noticed in Lake Toke, an electric power is efficiently being produced since 1960 A.D. Skagerak Energi operates 5 hydropower plants in Lake Toke. The hydropower plants are located at the water-stream down from the main lake starting from Dalsfoss power plant and downwards to the Tveitereidfoss, Langfoss, Vafoss and Kammerfoss power plants respectively. Figure 2-2 shows the location of 5 different hydropower plants in Kragerø waterways.

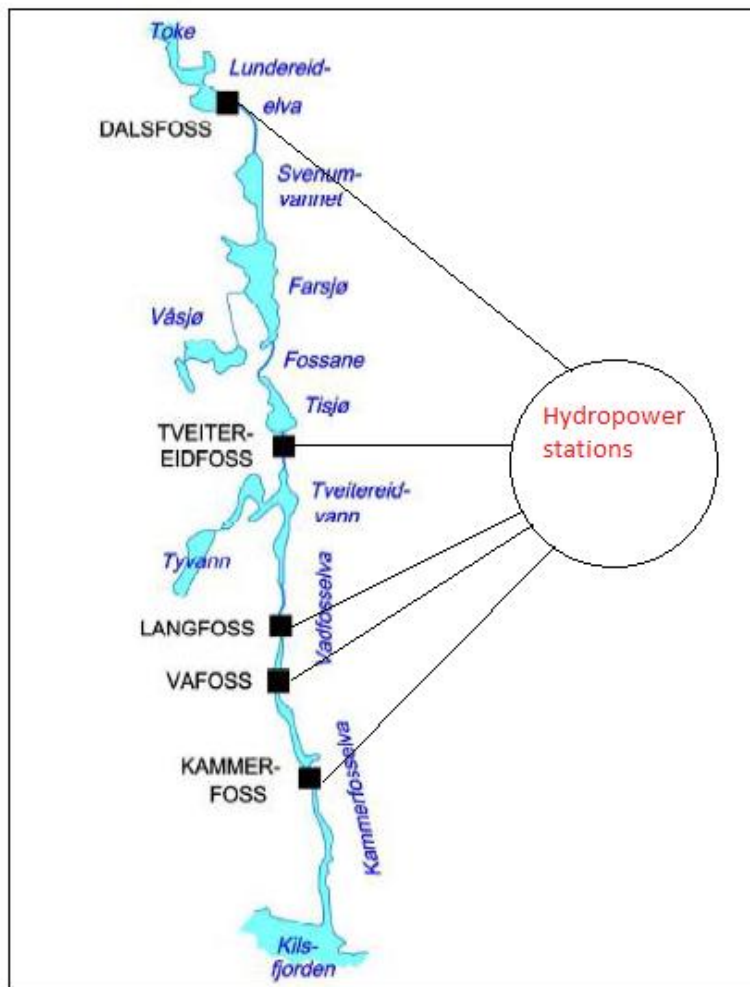


Figure 2-2: Kragerø waterways from Lake Toke to Kilsfjorden (Lie, 2013)

2.2 Dalsfoss Dam Data Measurements

Skagerak Energi have provided us data that explains the water level measurement in front of turbine (x_0) and level after turbine (x_u) of Dalsfoss dam in meters (overvann and undervann), water level of quay in meters (structure on the bank of lake Toke) (x_q), down streams flow from the dam (\dot{V}_o) in m^3/s and the power produced in three different turbines in megawatts. We further add the produced power of all three turbines and use total power production to find the turbine flow (\dot{V}_t) which is governed by theoretical/ mathematical formula.

All these data are provided by Skagerak Energi in the excel file in an hourly format. These data are collected for the year 2008-2009 and some of the measurements (**from** 1.25.2008,6:00- 1.26.2008, 22:00; 2.2.2008,12:00-2.3.2008,2:00; 2.5.2008,3:00-2.6.2008,21:00; 2.9.2008,15:00-2.14.2008,14:00; 3.6.2008,3:00-3.6.2008,20:00; 4.14.2008,10:00-4.15.2008,19:00 **to** 4.23.2008,0:00) of water level in front of turbine (x_0) and few for x_u are missing in the original excel data sheet and in MATLAB these are replaced by NaN (not-a-number).

2.2.1 Level Description of Dalsfoss Dam

According to the provided data by Skagerak Energi, a general sketch of water level of Dalsfoss dam is shown in Figure 2-3. The water level measurement in front of turbine (x_0) and level after turbine (x_u) of Dalsfoss dam in meters (overvann and undervann), water level of quay in meters (structure on the bank of lake Toke) (x_q) is shown in Figure 2-3. The downstream flow from the dam (\dot{V}_o) is also shown in Figure 2-3. Moreover, we have given total power production in terms of megawatts and we can predict the turbine flow \dot{V}_t from these input data.

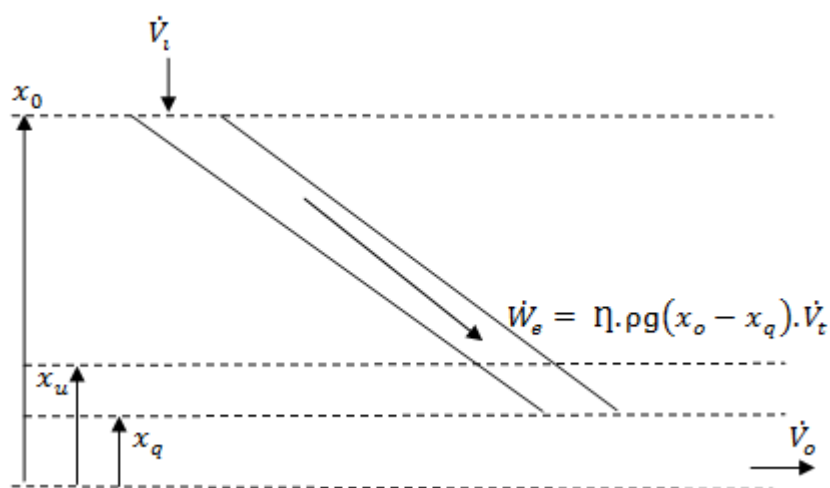


Figure 2-3: General overview of water level of Dalsfoss dam

Focusing on the trend of data how they are organized, Figure 2-4 shows the nature of height in front of turbine (x_o) following variation of 6 meters approximately between 57 meters to 63 meters somehow except with two outliers (values 3 times higher than expected level) around timestamp of approximately 7000 hours and 15000 hours. During MATLAB simulation, these outliers are replaced by averaging two neighboring values for all three levels x_o , x_u and x_q .

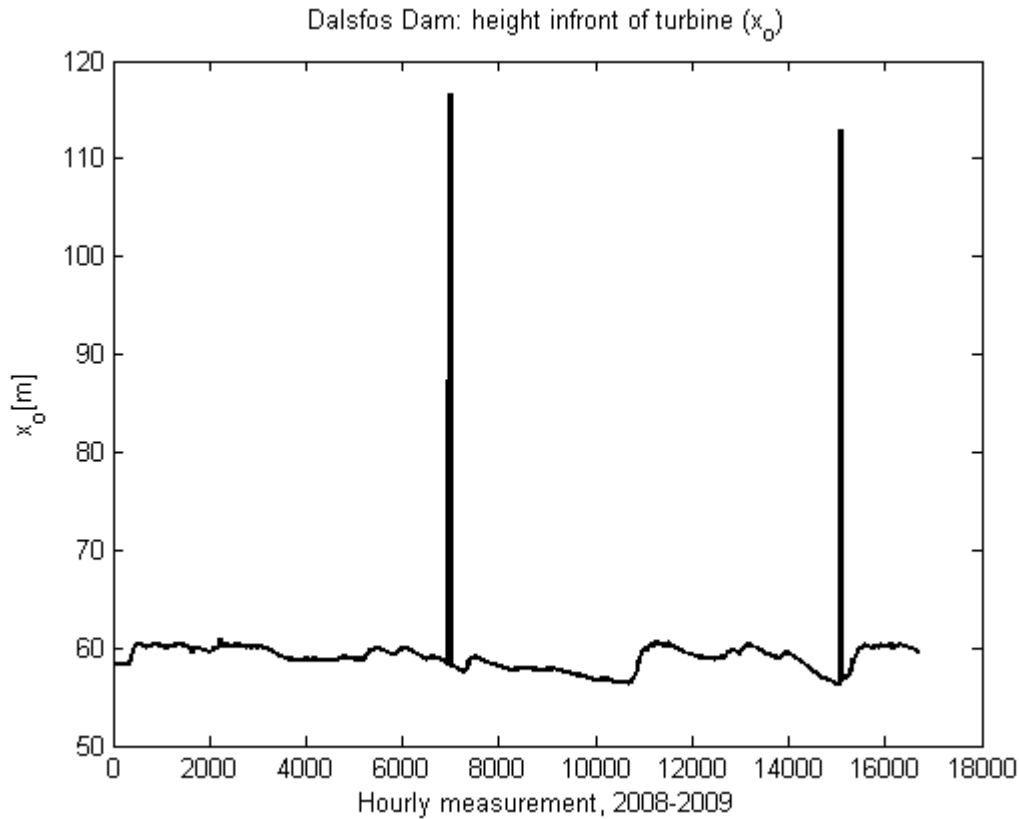


Figure 2-4: Presentation of actual Level in front of Dalsfoss Dam

Figure 2-5 represents the water level x_q , x_u and x_q without outliers measured in an hourly basis for the year 2008-2009 in meters. In the lower graph the red line showing the quay level x_q and blue line presenting the water level after the turbine x_u , we found an unusual behaviour of x_u that the level suddenly falls approx. 2 m at time between 6-7000 and then again climbs approx. 2 m around time span 12-13000. We already discussed in Figure 2-3, that x_u cannot be lower than x_q ($x_q \geq x_u$) this could be due to sensor/measurement error. Also, from discussion with operator/ data provider found that the level after turbine x_u is not in use and checked so far, so we won't use x_u in this project. Because of inaccuracy/ uncertainty of x_u and being similar to x_u we further use x_q instead.

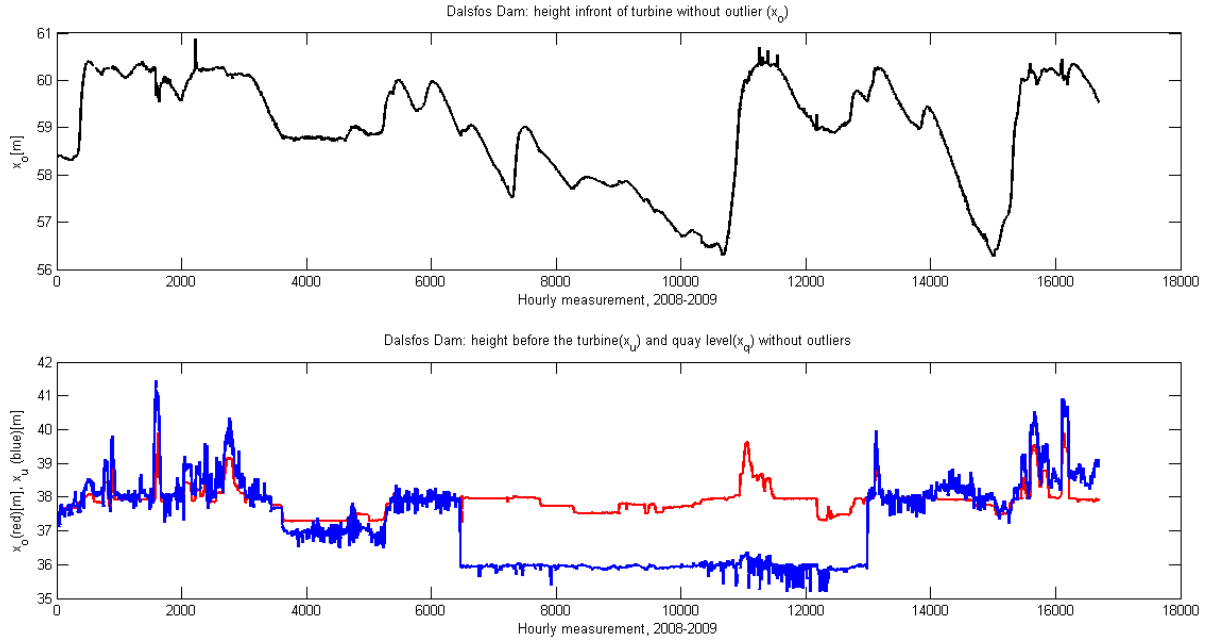


Figure 2-5: Level in front of Dalsfoss Dam (x_o), and at after turbine(x_u) and at Quay (x_q)

On the other hand, Figure 2-6 shows the downstream flow from the dam (\dot{V}_o). In Figure 2-6 we can see downstream flow \dot{V}_o relates Quay level x_q . We have considered only two outliers and the curve of \dot{V}_o is almost fitted with x_q . The 2nd order polynomial for \dot{V}_o is considered and the roots of the polynomial are found to be 18 , -1.297×10^3 and 2.3486×10^4 . According to figure, a very regular relation between \dot{V}_o and x_q can be observed can be said that there are no possible independent measurements. Moreover, we can say that measuring level is easier than measuring flow of a river. Hence, \dot{V}_o can be calculated using simple formula including x_q multiplied with the roots of quadratic polynomial:

$$\dot{V}_o = 18. x_q^2 - 1.297 \times 10^3 . x_q + 2.3486 \times 10^4 \quad (2.1)$$

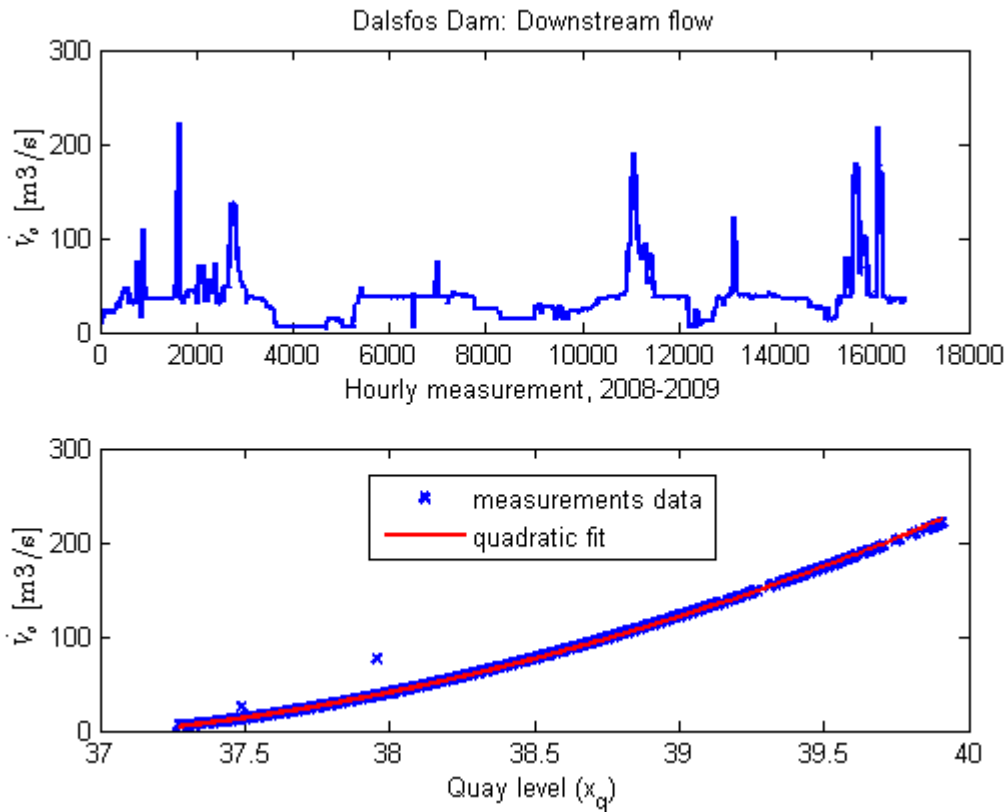


Figure 2-6: Downstream flow from the dam \dot{V}_o relating Quay level x_q and Quadratic fitting of \dot{V}_o with x_q

2.3 Functional Description

The schematic description for the Kragerø waterways includes all the measurements/information and can be illustrated as a system/process with inputs, outputs, states, disturbances etc. We have assumed inlet flow \dot{V}_i [m^3/s] and turbine flow \dot{V}_t [m^3/s] as system disturbance, height of gate h_g as control input, water level at Tokevatn h_1 and water level at Merkebekk h_2 in meters as system states/measurement and finally flow from the dam \dot{V}_o [m^3/s] and h_1 are the controlled outputs. Figure 2-7 shows the schematic process diagram for the model of Kragerø waterways. Each of the measurements available in the excel data set that are actively present in the system are described in details in the next section.

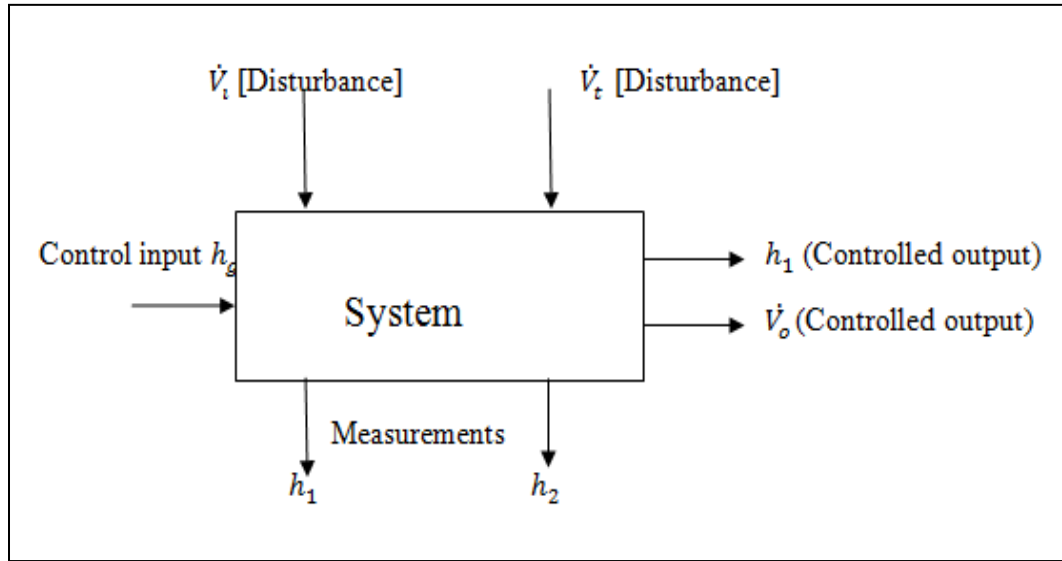


Figure 2-7: Schematic process diagram for the model of Kragerø waterways

2.3.1 Inlet flow \dot{V}_i

The total water flow from the catchment region and into the lake Toke is defined as inlet flow \dot{V}_i [m^3/s] and is based on the meteorological predictions of precipitation (of rain and snow) and temperature. Temperature is important for calculating the melting rates of snow. Generally we need to predict the inlet flow into the lake by considering the meteorological predictions based on the hydrology model of the catchment region of the lake. The meteorological predictions needed for Skagerak Energi are done by the weather forecasting company on a daily basis to compute \dot{V}_i and provides the most likely flow for 10 days prior. These predictions are quite uncertain and thus the use of possible numbers of predictions (ensemble) may be considerably used for the prediction. In this project, we will only use these predicted data (15 days prior) for the inlet flow. However, a more direct approach for minimizing the uncertainty of prediction can be used in future work.

2.3.2 Turbine Flow \dot{V}_t

As we have already mentioned that measuring the flow is not an easier task, hence to predict the flow through the turbine, we assume that the flow through the gate of Dalsfoss Dam is made only when the downstream flow \dot{V}_o becomes more than $36 m^3/s$ ($\dot{V}_o \geq 36 m^3/s$) and there should be minimum turbine flow equal to more than $4 m^3/s$. Thus we can relate \dot{V}_t as,

$$\dot{V}_t = \max(\dot{V}_o, 36) m^3/s$$

Also, the assumption that the flow due to production is equal to $36 \text{ m}^3/\text{s}$ and all the rest of the flow is through the gate leads to deduce gate flow \dot{V}_g as,

$$\dot{V}_o = \dot{V}_t + \dot{V}_g \quad (2.2)$$

In addition, the turbine flow \dot{V}_t is compared with total power production \dot{W}_e and level difference $h_t = (x_o - x_q)$ as shown in Figure 2-8. This assumption leads to develop a model for \dot{V}_t using \dot{W}_e , x_o and x_q . We have already mentioned that we will not use x_u due to inaccuracy/uncertainty.

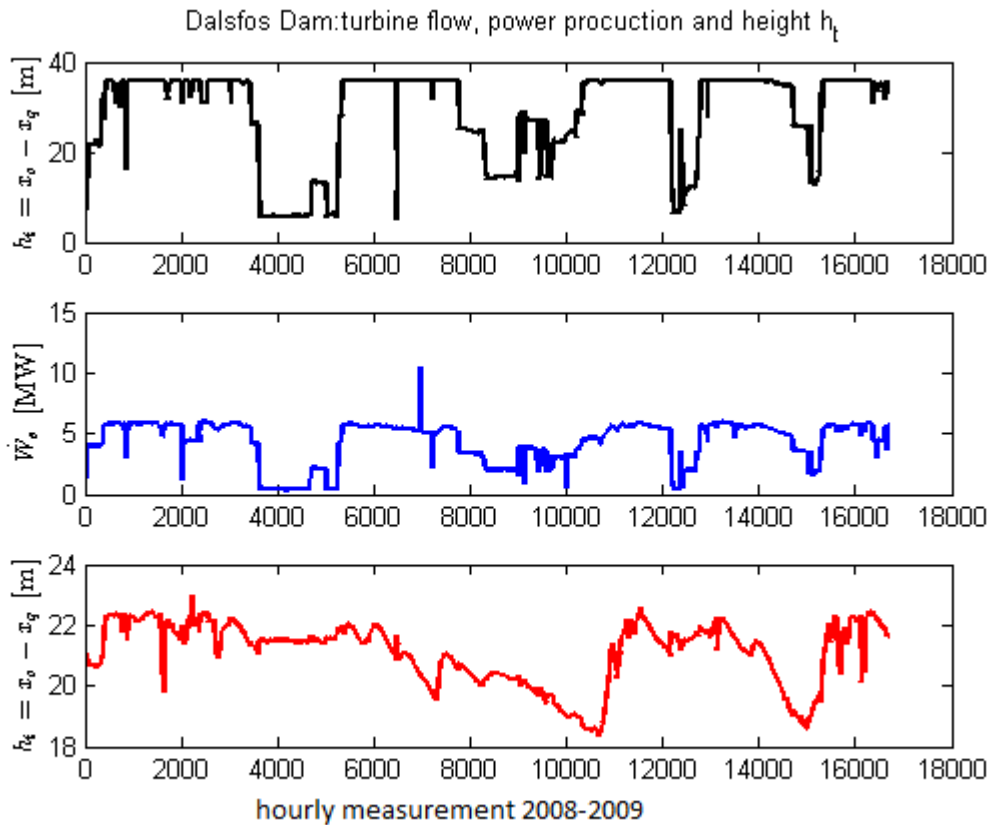


Figure 2-8: Turbine flow (\dot{V}_t) compared with total power production (\dot{W}_e) and level difference $h_t = (x_o - x_q)$.

Using concept of energy balance(Lie, 2013a), we have

$$\dot{W}_e = \eta \rho g. (x_o - x_u) \dot{V}_t. \quad (2.3)$$

Where,

η = efficiency factor of the power

ρ = density of water (1000 m^3)

g = gravitational constant (9.81 m/s)

We have already mentioned that we will not use x_u due to inaccuracy/uncertainty and thus use x_q , so (2.3) now becomes,

$$\dot{W}_e = F ((x_o - x_q) \dot{V}_t). \quad (2.4)$$

Where $F(\cdot)$ is a function which should be linear when η is constant.

Now our interest is to find the model for \dot{W}_e with a linear fit to $(x_o - x_q)\dot{V}_t$ and we have all the data required. Using “Polyfit”- in MATLAB, we can fit the linear equation and find the values of polynomial using “Polyval”. Thus we find simply 2 polynomial values: 0.075 and -0.02477, hence using these values (to relate \dot{W}_e , \dot{V}_t , x_o and x_q) we achieved relation for \dot{W}_e as,

$$\dot{W}_e = 0.0075 \cdot (x_o - x_q) \cdot \dot{V}_t - 0.2477 \quad (2.5)$$

We cannot guarantee that the developed model for is perfect because we have assumed that $\dot{V}_t = \max(\dot{V}_o, 36) m^3/s$ and \dot{V}_o is not certain. In spite of this uncertainty we will take a close look for the correlation between \dot{W}_e , \dot{V}_t , x_o and x_q as shown in Figure 2-9.

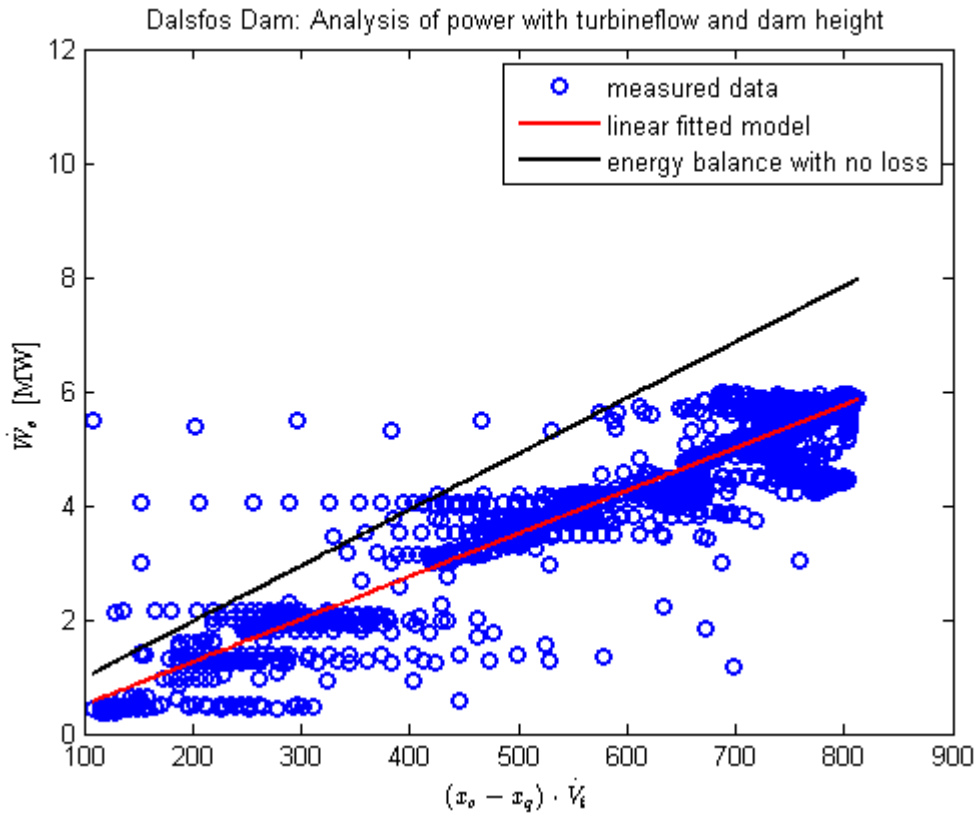


Figure 2-9 Correlation of power prediction between measured data (o,blue),linear prediction of \dot{W}_e from correlation $\dot{W}_e = 0.0075 \cdot (x_o - x_q) \cdot \dot{V}_t - 0.2477$ (black) and theoretical value $\dot{W}_e = F((x_o - x_q)\dot{V}_t)$ (red)

2.3.3 Prediction Models

Now if we assume we have known values of \dot{W}_e , \dot{V}_g and x_o we can compute \dot{V}_o , \dot{V}_t and x_q from the developed models as,

$$\dot{V}_o = 18 \cdot x_q^2 - 1.297 \times 10^3 \cdot x_q + 2.3486 \times 10^4$$

$$\dot{W}_e = 0.0075 \cdot (x_o - x_q) \cdot \dot{V}_t - 0.2477$$

$$\dot{V}_o = \dot{V}_t + \dot{V}_g$$

So, to guarantee that the proposed model is good enough we will predict the model parameters by re-parameterising as follows:

$$\dot{V}_o = \hat{c}_1 x_q^2 - \hat{c}_2 x_q + \hat{c}_3$$

And from (2.5), we can say that,

$$\dot{V}_t = a \cdot \frac{\dot{W}_e}{(x_o - x_q)} + b$$

So, (2.2) becomes,

$$\hat{c}_1 x_q^2 - \hat{c}_2 x_q + \hat{c}_3 = C \cdot \frac{\dot{W}_e}{(x_o - x_q)} + b + \dot{V}_g$$

Where,

$$\hat{c}_1 = 18, \hat{c}_2 = 1.297 \times 10^3, \hat{c}_3 = 2.3486 \times 10^4, a = 0.0075 \text{ And } b = 0.2477$$

Or

$$\dot{W}_e = \underbrace{\hat{c}_1}_{c_1} x_q^2 \cdot (x_o - x_q) + \underbrace{\hat{c}_2}_{c_2} x_q (x_o - x_q) + \underbrace{\hat{c}_3 - b}_{c_3} (x_o - x_q) + \underbrace{\frac{1}{a}}_{c_4} \dot{V}_g (x_o - x_q)$$

So, after renaming the parameters and adding the possible bias c_5 , we may rewrite the parameter model as,

$$\dot{W}_e = c_1 x_q^2 \cdot (x_o - x_q) + c_2 x_q (x_o - x_q) + c_3 \cdot (x_o - x_q) + c_4 \dot{V}_g (x_o - x_q) + c_5$$

Problem Description

2.3.4 System Requirements

The Norwegian government authority Norwegian Water Resources and Energy Directorate (NVE) provides strict regulations in order to maintain consistent and environmentally prospective management of water resources and contribute to the economic utilization of energy in Norway¹; hence Lake Toke is also strictly constrained by NVE with certain operating procedures throughout the year to protect the environment and population. The operating procedures are the minimum and maximum water levels in meter that should be maintained in Lake throughout the year. These procedures are revised with different water levels for summer and autumn. The restrictions are governed by highest and lowest regulated water level as well as limit for highest and lowest regulated water level throughout a year.

¹<http://www.nve.no/en/About-NVE/>

Moreover, there should always minimum of 4 m³/s of water flow in to a Dalsfoss water stream even if the power production is stopped either.

Table 2-1 shows the operating procedure for Lake Toke for a year. There are different water level requirements according to the seasons.

Table 2-1: Operating Procedure of Lake Toke (Thorsen, 2011)

Date	LRV	HRV	ALRV	AHRV
1 Jan- 1 May	55,75	60,35	55,75	60,35
1 May- 1 Sep	55,75	60,35	58,85	59,85
1 Sep- 15 Sep	55,75	60,35	58,35	59,35
15 Sep- 1 Nov	55,75	60,35	55,75	59,35
1 Nov- 15 Nov	55,75	60,35	55,75	59,85
15 Nov- 31 Des	55,75	60,35	55,75	60,35

From the Table 2-1, we can conclude that as soon as the winter draining starts the ALRV increases and the AHRV decreases as the spring starts indicating flooding environment.

Figure 2-10 shows the water level requirement from NVE for a year along with water level of Lake Toke in three different years (2001, 2002 & 2009).

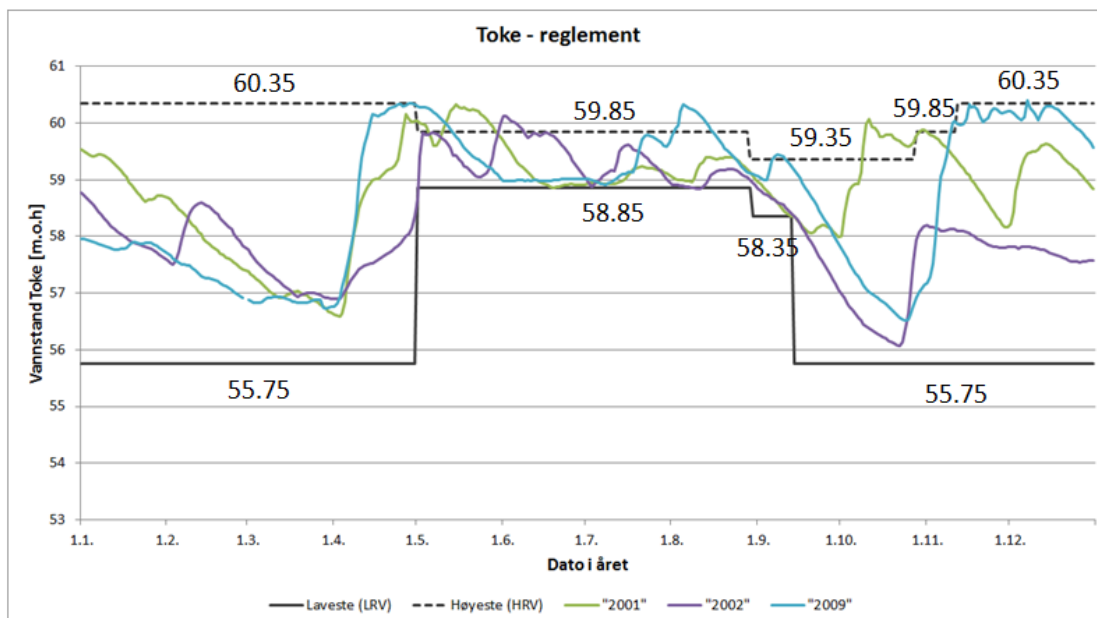


Figure 2-10: Requirement for water level of Lake Toke and Lake level (2001, 2002 & 2009) ((Gøthesen et al., 2013))

3 Model Development & Validation

After the exploration of information we extracted from the data and analysis of Skagerak Energi, our goal is to develop the model of Lake Toke and validate/calibrate the model against the real data obtained. In this part of my report the detailed explanation and development of mathematical model of Lake Toke is done. In the second part of this chapter, the developed model is validated/calibrated against the experimental/historical data and finally the chapter ends up with the accuracy measurement and precision of model.

3.1 Introduction

A model is a simplified system representing how the real system behaves. The model of a system describes how the input and output variables interact with the system and the surroundings. The dynamic model of a system represents the behavior and a mathematical model of a system that makes easier to understand and solve the model complexity. The mathematical model includes the mathematical equations (such as: integral equations, algebraic equations, ordinary and partial differential equation etc.) that reflects the overall idea of a system.

I will develop mathematical model based on mechanistic model followed by simple mass balance/ mass conservation theory that will describe the property of dam. The model of dam is so developed that is based on mass balance and first principle approach. Further, we need to develop the system model to implement MPC algorithm: which is the core part of this work.

3.2 Model Description

We have already mentioned that we have considered two level measurements h_1 and h_2 for Lake Toke thus we consider two compartments (upper compartment having index 1 with water level h_1 and lower compartment having index 2 with height h_2); so we will have two models for the lake. The main water inflow is supplied to the upper compartment h_1 and rest of the minimal flow is channeled to the lower compartment h_2 as shown in the Figure 3-1. The flood gate is used to control the out flow of lower compartment h_2 which ultimately passes through the hydropower turbine (\dot{V}_t) and radial flood gate flow (\dot{V}_g) when the turbine flow $\dot{V}_t = \max(\dot{V}_o, 36) m^3/s$. The gate is only opened when this condition is achieved; it is obvious that we are working on the flood control so we generally expect \dot{V}_g and hopefully we can develop the model for flood gate as well as for the whole system and then implement MPC algorithm for control purpose.

3.2.1 Model Summary & Model Parameters

Working on the model development, Skagerak Energi has provided us the various parameters of Lake Toke model. Using these parameters and operating conditions to validate the model, we will develop and validate the models. The parameters for the Lake Toke is presented in Table 3-1 and the operating conditions for validating model is given in Table 3-2. The operating conditions are set to validate the developed models and based on the simulation results obtained by (Lie, 2014).

Table 3-1: Parameters for Lake Toke

Parameter	Value	Unit	Comment
α	0.05	-	Fraction of surface area in compartment 2
β	0.02	-	Fraction of inflow to compartment 2
K_{12}	800	-	Flow coefficient at Merkebekk
C_d	1	-	Discharge coefficient, Dalsfoss gate
w	11.2	m	Width of Dalsfoss gate
h_g^{max}	5.6	m	Maximum opening height of Dalsfoss gate
x_{HRV}^{max}	60.35	m	Maximum high regulated level value
x_{LRV}^{min}	55.75	m	Minimum low regulated level value

Table 3-2: Operating conditions for validating model

Quantity	Value	Unit	Comment
$h_1(t=0)$	2.5	m	Initial level, compartment 1
$h_2(t=0)$	2.5	m	Initial level, compartment 2
\dot{V}_i	-	m ³ /s	Inlet flow jumps from 400 m ³ /s to 24 m ³ /s after 25×10^3 s
h_g	-	m	Gate opening jumps from $0.5h_g^{max}$ to h_g^{max} after 25×10^3 s
\dot{V}_t	36	m ³ /s	Volumetric flow through turbines

In brief, we sum up the models for the Lake Toke²:

We have already discussed that; the best model can be obtained by including the third compartment flow through straueme. However, in this work we couldn't include third compartment and the developed model may not be perfect but we find the solution based on these two models. Hence the possible model for two-compartment of Lake Toke is,

$$\frac{dh_1}{dt} = \frac{1}{(1-\alpha)A(h_1)} [(1-\beta)\dot{V}_i - \dot{V}_{12}]$$

$$\frac{dh_2}{dt} = \frac{1}{\alpha A(h_2)} [\beta\dot{V}_i - \dot{V}_t - \dot{V}_g + \dot{V}_{12}]$$

The area of the Lake is given by a filling curve $A(h)$,

$$A(h) = 28 \times 10^6 \cdot 1.1 \cdot h^{\frac{1}{10}},$$

The inter compartmental flow \dot{V}_{12} is given by,

$$\dot{V}_{12} = K_{12}(h_1 - h_2)\sqrt{|(h_1 - h_2)|},$$

The gate flow \dot{V}_g is given by,

$$\dot{V}_g = C_d \cdot w \cdot \min(h_g, h_2)\sqrt{2g \max(h_2, 0)},$$

And the turbine flow \dot{V}_t is given by,

$$\dot{V}_t = \max(\dot{V}_o, 36) \text{ m}^3/\text{s}$$

3.3 Model Development

Figure 3-1 is a geometry of possible layout of model of lake Toke developed by (Lie, 2013a). The water inflow (\dot{V}_i) in a lake is divided in to two compartments. The main portion of inflow $(1-\beta)\dot{V}_i$ flows in the upper compartment and the minor portion of inflow $(\beta\dot{V}_i)$ flows to the lower compartment. As indicated in the figure 3.1, the flow between two compartments is \dot{V}_{12} . The input to the compartment 2 is \dot{V}_{12} and minor inflow $(\beta\dot{V}_i)$ resulting the gate flow (\dot{V}_g) and

²Model based on a SkagerakEnergi report developed by Bjørn Glemmestad from TUC with necessary modification

turbine flow(\dot{V}_t). We assume α as the fraction of surface area in compartment 2 and β as the fraction of inflow to compartment 2 whereas $V_{h=0}$ is the water volume below $h = 0$ in $[m^3]$. The lowest regulated value is $x_{LRV}^{min} = 55.75 m$ above the sea level and the highest regulated value is $x_{HRV}^{max} = 60.35 m$ above sea level. $h [m]$ Is the water level above the datum line i.e $x = x_{LRV}^{min}$ and $h \in [0, h^{max}] [m]$.

Hence, during the simulation of models, we deal with the actual height (relative values) of h_1 and h_2 . Skagerak have provided us values of water level at Merkebekk (x_o), water level at Dalsfoss (x_u) and Quay Level (x_q), hence it is a good idea to use relative value for water level:

Water level at Merkebekk, $h_1 = x_o - x_{LRV}^{min}$,

Water level at Dalsfoss Dam, $h_2 = x_q - x_{LRV}^{min}$, and

Maximum regulated level, $h^{max} = x_{HRV}^{max} - x_{LRV}^{min} = 4.6 m$

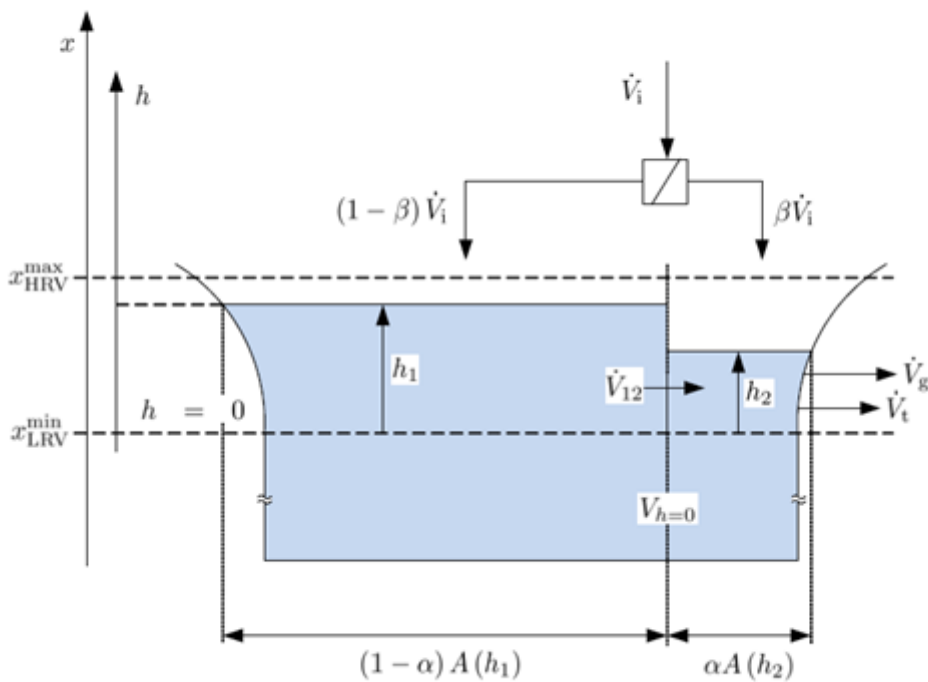


Figure 3-1: Possible layout of model of Lake Toke

3.3.1 Assumptions and Clarifications

- The developed models are based on general mass balance equation and are valid for standard chemical reactions. Lake Toke is assumed to be perfectly stirred tank having plane surface.
- The model assumes that the mass flow \dot{m} is slowly varying quantity.
- Density and temperature are assumed to be constant and is homogenous in the volume.
- The states of the system are $x = (h_1, h_2)$, inputs are $u = (h_g, \dot{V}_i, \dot{V}_t)$ and parameters are $\theta = (\alpha, \beta, w, C_d)$.
- The obtained models are non linear models; linearization and discretization are done further in later.

3.3.2 Model for h_1 & h_2

We start to develop the model based on simple mass balance. The general mass balance is given by,

$$\frac{dm}{dt} = \dot{m}_i - \dot{m}_o$$

Assuming constant density with respect to time,

$$m = \rho V \text{ And } \dot{m} = \rho \dot{V}$$

So, we obtain,

$$\frac{dV}{dt} = \dot{V}_i - \dot{V}_o$$

We know, the relation between volume and level is,

$$V = A \cdot h,$$

$$\text{Or, } \frac{dV}{dh} = A(h)$$

Hence,

$$dV = A(h)dh$$

Finally we get,

$$\frac{dh}{dt} = \frac{1}{A(h)} (\dot{V}_i - \dot{V}_o)$$

According to Figure 3-1, the input (\dot{V}_i) to the compartment 1 is $(1 - \beta)\dot{V}_i$, output is \dot{V}_{12} and the surface area to be $(1 - \alpha)A(h_1)$. So, we find the model for compartment 1,

$$\frac{dh_1}{dt} = \frac{1}{(1 - \alpha)A(h_1)} [(1 - \beta)\dot{V}_i - \dot{V}_{12}]$$

Similarly for compartment 2, the input is $(\beta\dot{V}_i+V_{12})$, output is $(\dot{V}_t + \dot{V}_g)$ and the surface area to be $\alpha A(h_2)$ hence the model becomes,

$$\frac{dh_2}{dt} = \frac{1}{\alpha A(h_2)} [\beta\dot{V}_i - \dot{V}_t - \dot{V}_g + V_{12}]$$

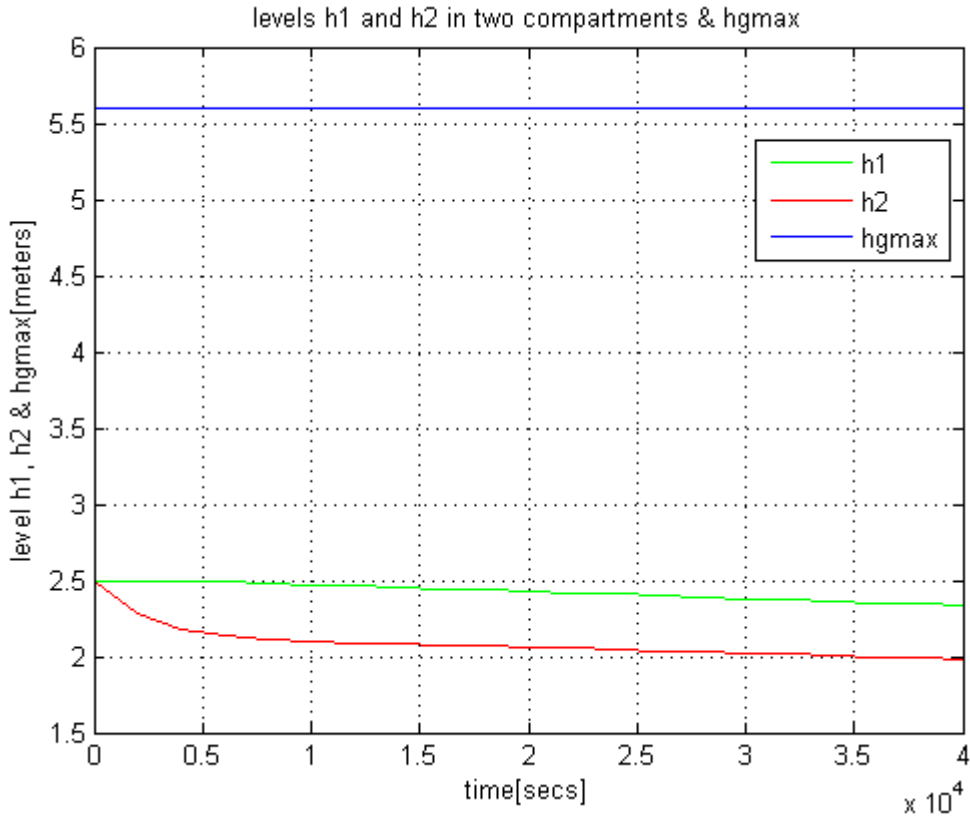


Figure 3-2: Levels h_1 , h_2 in two compartments, h_{gmax} with given parameters and operational conditions in initial states ($h_1(t=0)=2.5$, $h_2(t=0)=2.5$) at time period 40000 sec

We have done the simulation of nominal model under the given operational conditions for validating the model of Lake Toke. We started the system in initial states as in Table 3-2, $h_1(t=0) = 2.5\text{ m}$ and $h_2(t=0) = 2.5\text{ m}$ and shows somehow reasonable effect to follow the steady state as in figure 3-3. Figure 3-2 is the level variations of h_1, h_2 and h_g^{max} with given parameters and operational conditions. Time used to simulate the models is 40,000 seconds \cong 11.1 hours. Analyzing Figure 3-2, the output of the system, h_1 i.e. level at compartment 1, the level at Merkebekk seeks the same pattern as in past and seems controlled enough.

Figure 3-3 is the level variation of h_1, h_2 and h_g^{max} with given parameters and operational conditions in steady state with in the same time i.e. 40,000 seconds.

The steady state values for h_1 and h_2 are found by equating the derivatives $\frac{dh_1}{dt}$ and $\frac{dh_2}{dt}$ to zero and simplifying the equation. We find the steady state values around, $h_{1,ss} = 5.4\text{ m}$ and

$h_{2,ss} = 4.75 \text{ m}$ by mathematical calculations. The simulation of steady state seems to have quiet similar behavior as in initial state except some more variations in h_1 and h_2 .

The simulation time really differs the system dynamics resulting slow system behavior, thus we can use e.g. hours or days as time unit. Though we have not done this in this simulation, we only used seconds as time.

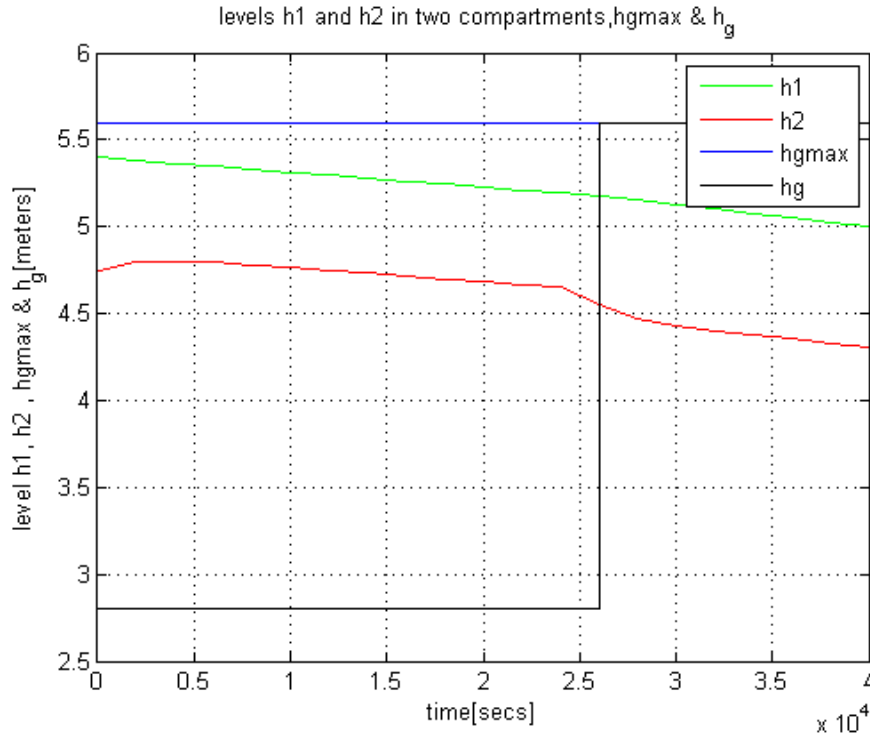


Figure 3-3: Levels h_1 , h_2 in two compartments, h_g & h_{gmax} with given parameters and operational conditions in steady state ($h_1(t=0)=5.4$, $h_2(t=0)=4.75$) at time period 40000 sec

It is of great importance that the simulation time really differs the behavior of system. Hence, we need to re-simulate the system with more longer time span of 10^5 or even more longer time span 10^6 to see more specific behavior of the models. Using these different simulation times, we can estimate the time constants for the models. Figure 3-4 is the simulation of levels variations of h_1 , h_2 and h_g^{max} with given parameters and operational conditions. Time used to simulate the models is 10^5 seconds $\cong 28.7$ hours.

Similarly, Figure 3-5 is the simulation of levels variations of h_1 , h_2 and h_g^{max} with given parameters and operational conditions. Time used to simulate the models is 10^6 seconds $\cong 11.6$ days.

The time constant can be found by calculating the pure real, negative Eigen value (λ). For a system with single, negative Eigen value, the time constant is the length of time from a change in the system until new the system reaches to the new steady state making a tangent to the initial response (Lie, 2013b).

In Figure 3-3, we can approximate the time constant in level h_2 (red line). The time constant of around 1600-200 seconds \cong 28-30 minutes can be seen around 24500-26500 seconds. Similarly, in figure 3-5, we can see the time constants in both of the levels h_1 and h_2 (because both of the curves follows same pattern) at around 50000- 400000 seconds i.e. 350000 seconds \cong 4 days. These estimations are quiet uncertain and require further analysis to find more accurate values.

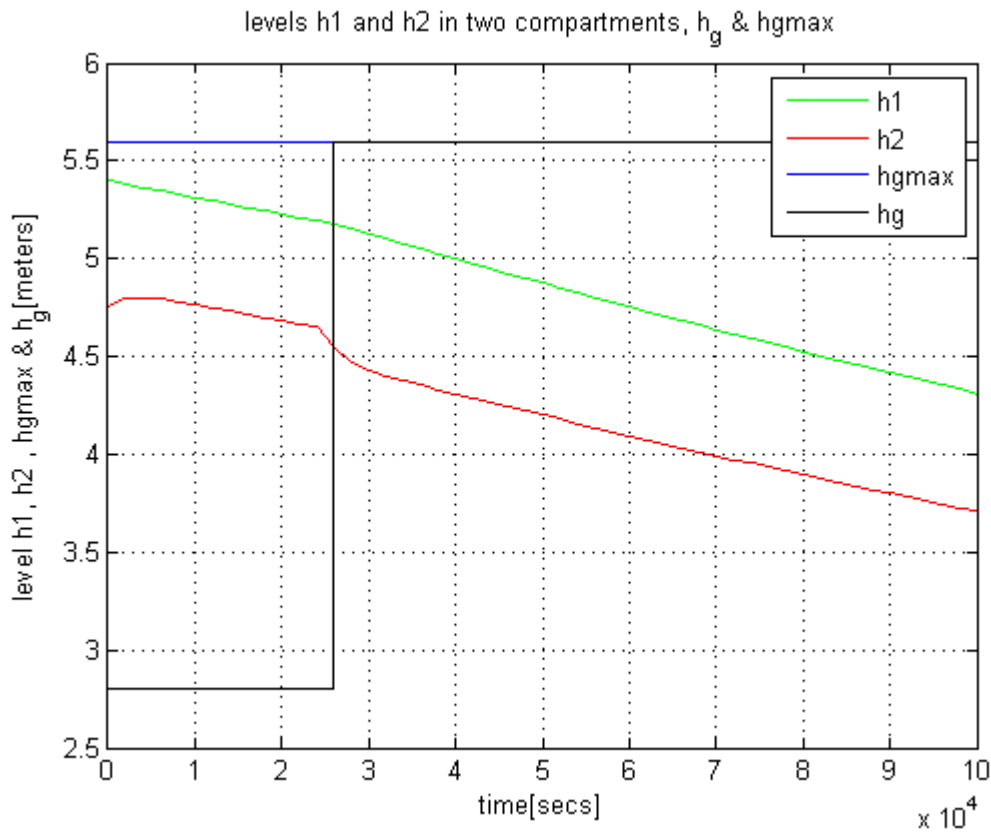


Figure 3-4: Levels h_1 , h_2 in two compartments, h_g & h_{gmax} with given parameters and operational conditions at time period 100000 sec. i.e. 28.7hr

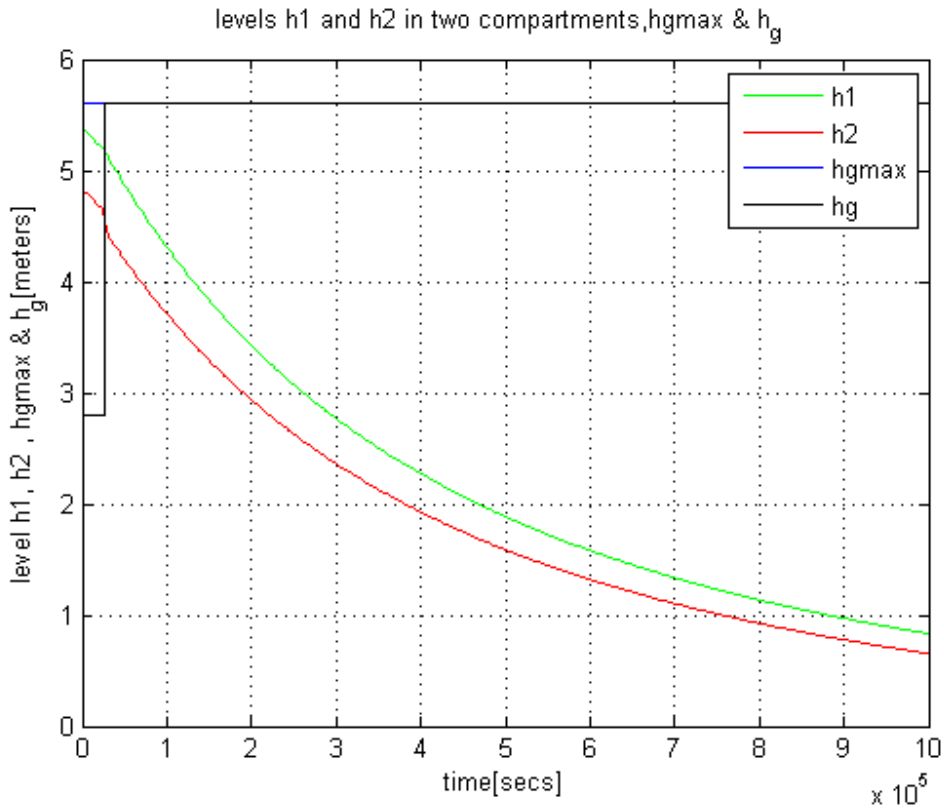


Figure 3-5: Levels h_1 , h_2 in two compartments, h_g & h_{gmax} with given parameters and operational conditions at time period 1000000 sec. i.e. 11.6 days

3.3.3 Model for \dot{V}_{12}

The flow between compartment 1 and compartment 2 is given by \dot{V}_{12} :

$$\dot{V}_{12} = K_{12}(h_1 - h_2)\sqrt{|(h_1 - h_2)|},$$

The expression to \dot{V}_{12} is based on the model to some extent uncertain data.

Figure 3-6 shows the change in input/disturbance \dot{V}_{in} , intercompartmental flow \dot{V}_{12} and gate flow \dot{V}_g at time span of 40000 seconds i.e. 11.1 hours. In steady state $(1 - \beta)\dot{V}_{in} = \dot{V}_{12}$ and $\dot{V}_{12} = \dot{V}_t + \dot{V}_g$. According to Figure 3-6, we can see that in steady state \dot{V}_g is much lower than \dot{V}_{in} because of the constant turbine flow \dot{V}_t and the inflow $\beta\dot{V}_{in}$ directly flows to the compartment 2. Similarly, the inter-compartmental flow \dot{V}_{12} is less than \dot{V}_{in} because fraction of inflow $\beta\dot{V}_{in}$ is diverted to compartment 2 directly. The inflow \dot{V}_{in} is constant with value 400 m³/s throughout the simulation.

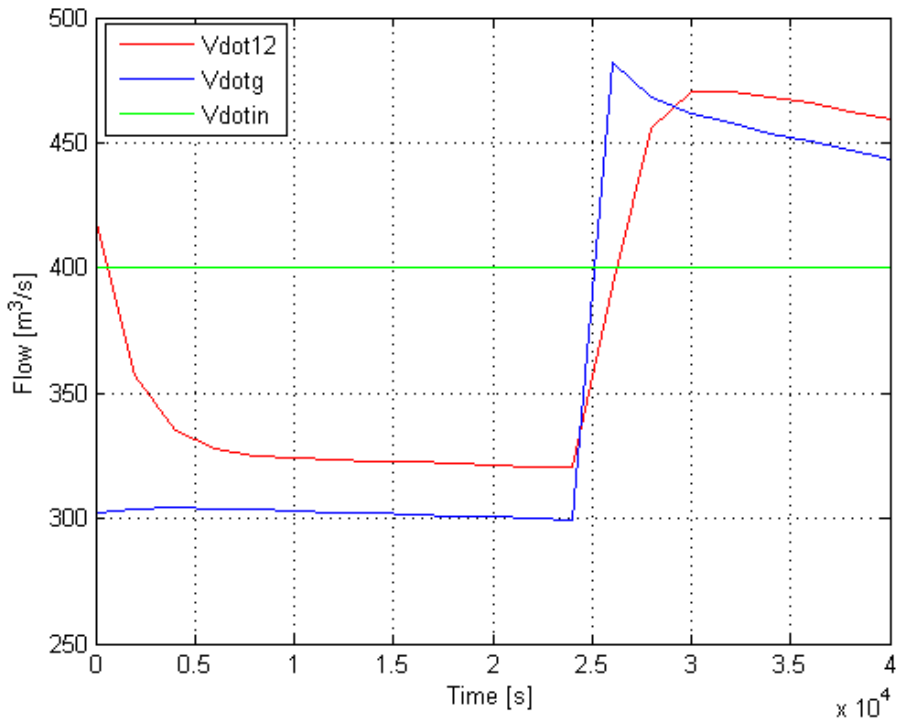


Figure 3-6: Simulation of inter compartmental flow \dot{V}_{12} , along with gate flow \dot{V}_g and disturbance flow \dot{V}_{in} in steady state. Time span: 40000 seconds, i.e. 11.1 hr

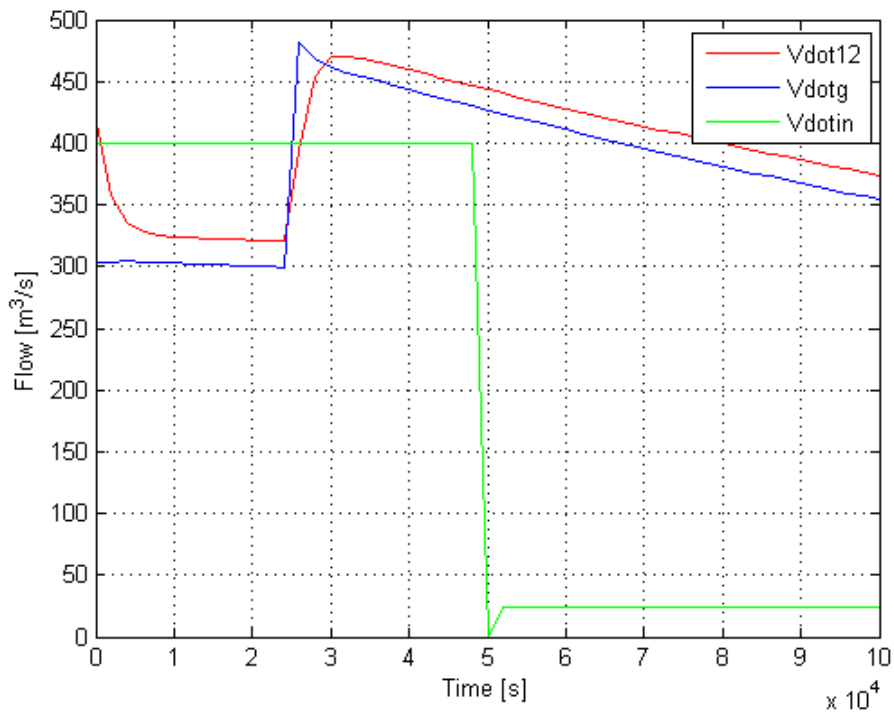


Figure 3-7: Simulation of inter compartmental flow \dot{V}_{12} , along with gate flow \dot{V}_g and disturbance flow \dot{V}_{in} in steady state. Time span: 100000 seconds, i.e. 28.7hr

Figure 3-7 shows the step change in input/disturbance \dot{V}_{in} , intercompartmental flow \dot{V}_{12} and gate flow \dot{V}_g at time span of 100000 seconds i.e. 28.7 hours. We already stated that in steady state $(1 - \beta)\dot{V}_{in} = \dot{V}_{12}$ and $\dot{V}_{12} = \dot{V}_t + \dot{V}_g$. According to Figure 3-8, we can see that flow \dot{V}_g and \dot{V}_{12} are lower than \dot{V}_{in} until 25000 seconds because of the constant turbine flow \dot{V}_t and the inflow $\beta\dot{V}_{in}$ directly flows to the compartment 2 in steady state but after 25000 seconds, both flows \dot{V}_g and \dot{V}_{12} suddenly jumps for a while and then decreases slowly. The inflow \dot{V}_{in} is constant at 400 m³/s until 50000 seconds and then jumps down around 24 m³/s after 50000 seconds.

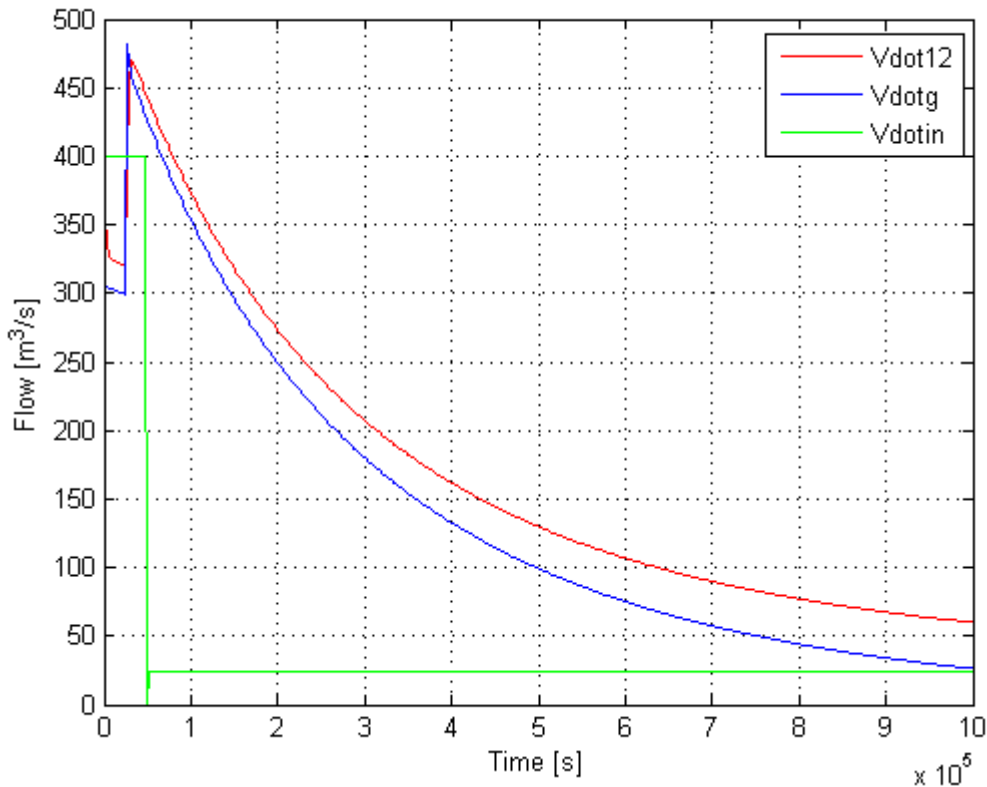


Figure 3-8: Simulation of inter compartmental flow \dot{V}_{12} , along with gate flow \dot{V}_g and disturbance flow \dot{V}_{in} in steady state. Time span: 1000000 seconds, i.e. 11.6 days

3.3.4 Free Flood gate & Model for \dot{V}_g

The outflow from compartment 2 partially goes through hydro power turbine (\dot{V}_t) and through a radial flood gate (\dot{V}_g) when needed. The sketch of radial flood gate is shown in Figure 3-9.

The flow through a gate is assumed to be 'free flow' since the downstream surface from gate is not higher than the level of gate.

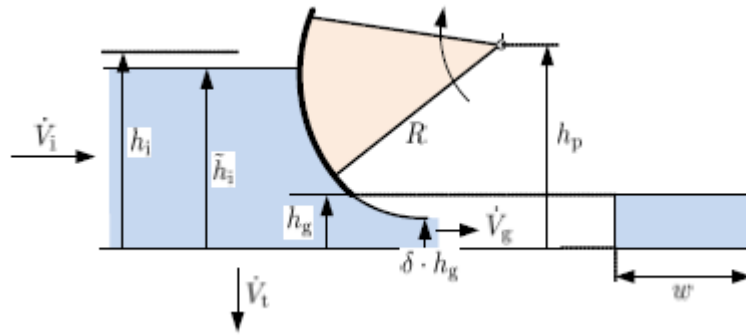


Figure 3-9: Sketch of free flood gate (Lie, 2014)

A model for the free flow gate is based on Bernoulli's law. We can find the model as (Lie, 2014),

$$\dot{V}_g = C_d \cdot w \cdot \min(h_g, h_2) \sqrt{2g \max(h_2, 0)}$$

Where C_d is the discharge coefficient and may varies with operating conditions. In our case the discharge coefficient becomes $C_d \rightarrow 1$.

Figure 3-10 is the gate opening h_g at time stamp 40000 seconds \cong 11.1 hours. The gate opening h_g remains stable at around 2.8 m until 25000 seconds and then jumps to the maximum opening of gate i.e 5.6 meters.

Figure 3-11 is the applied gate opening h_g at time stamp 100000 seconds \cong 28.7 hours.

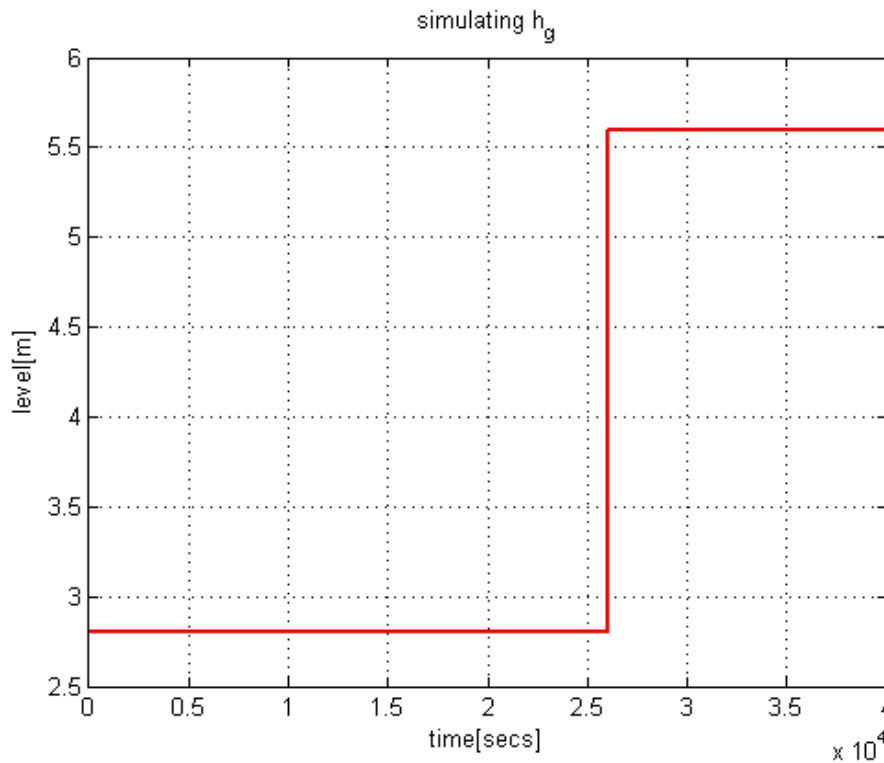


Figure 3-10: Simulation of gate opening h_g at time span 40000 sec.

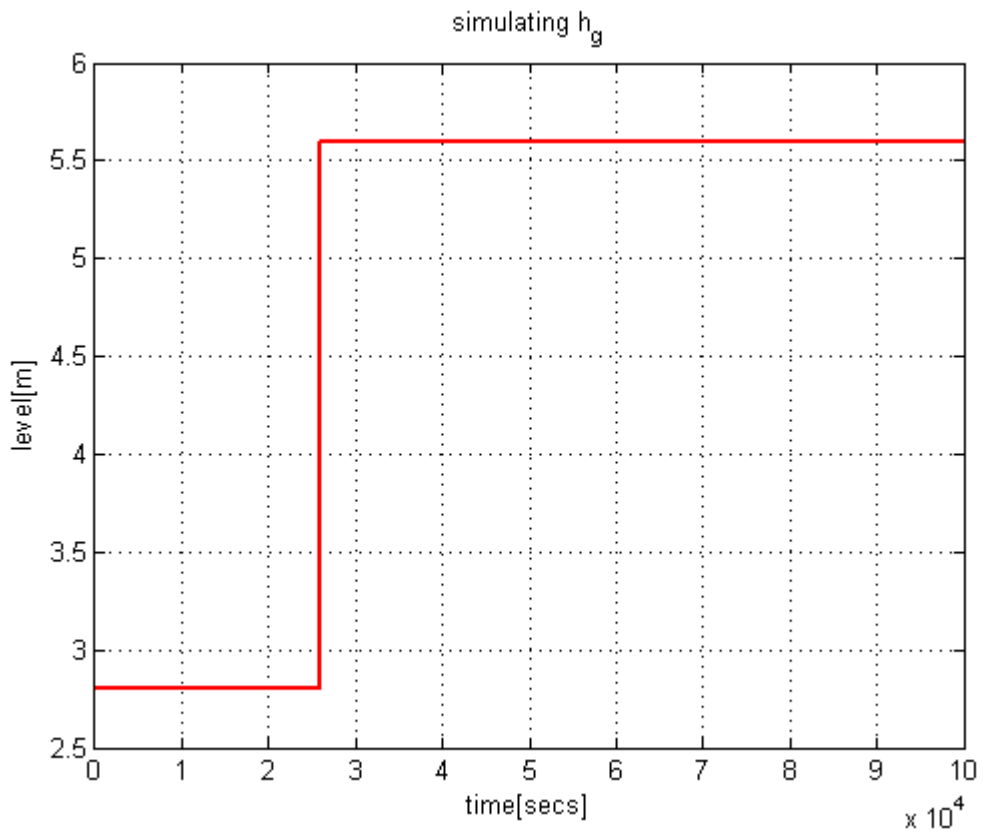


Figure 3-11: Simulation of gate opening h_g at time span 28.7 hours

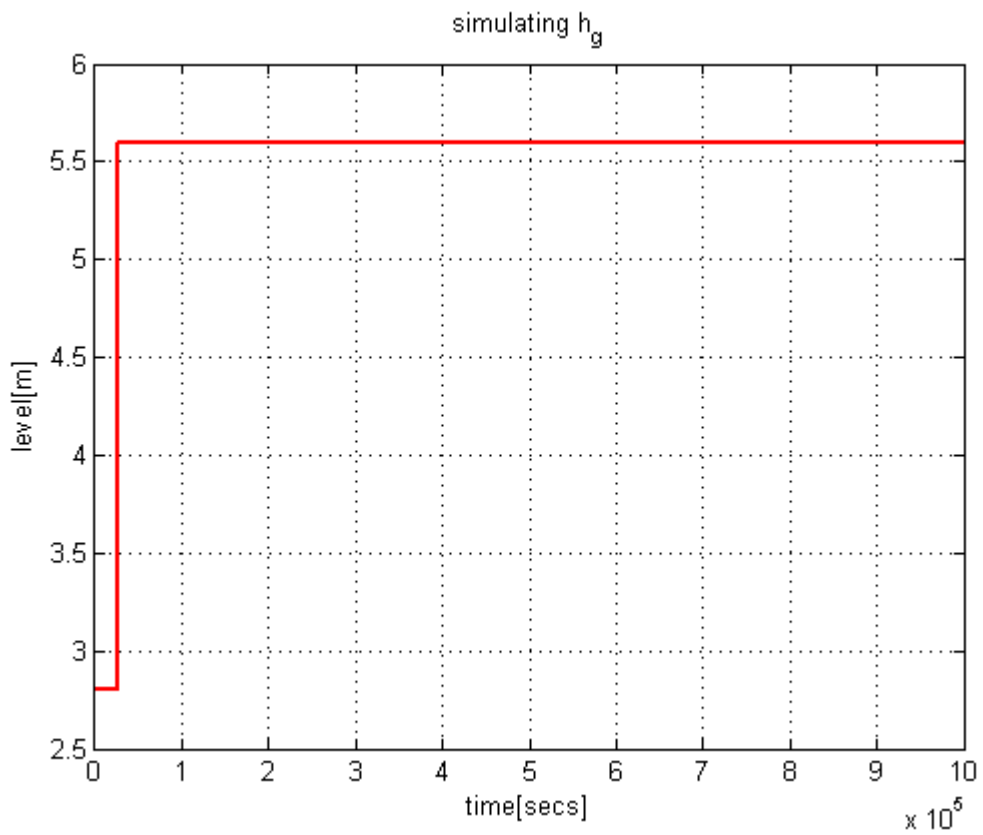


Figure 3-12: Simulation of gate opening h_g at time span 11.6 days

Figure 3-12 is the applied gate opening h_g at time span 11.6 days.

3.3.5 Surface Area of Lake Toke

The surface area (A) for Lake Toke is scaled between height and volume of lake as shown in Figure 3-13, normally known as filling curve. We have given,

$$V(h) = 28 \times 10^6 \cdot h^{1.1}$$

We know, the relation between Volume, Area and level is,

$$V = A \cdot h,$$

$$\text{or, } \frac{dV}{dh} = A(h)$$

So, from filling curve, we find the surface area $A(h)$,

$$A(h) = \frac{dV}{dh} = 28 \times 10^6 \cdot 1.1 \cdot h^{\frac{1}{10}},$$

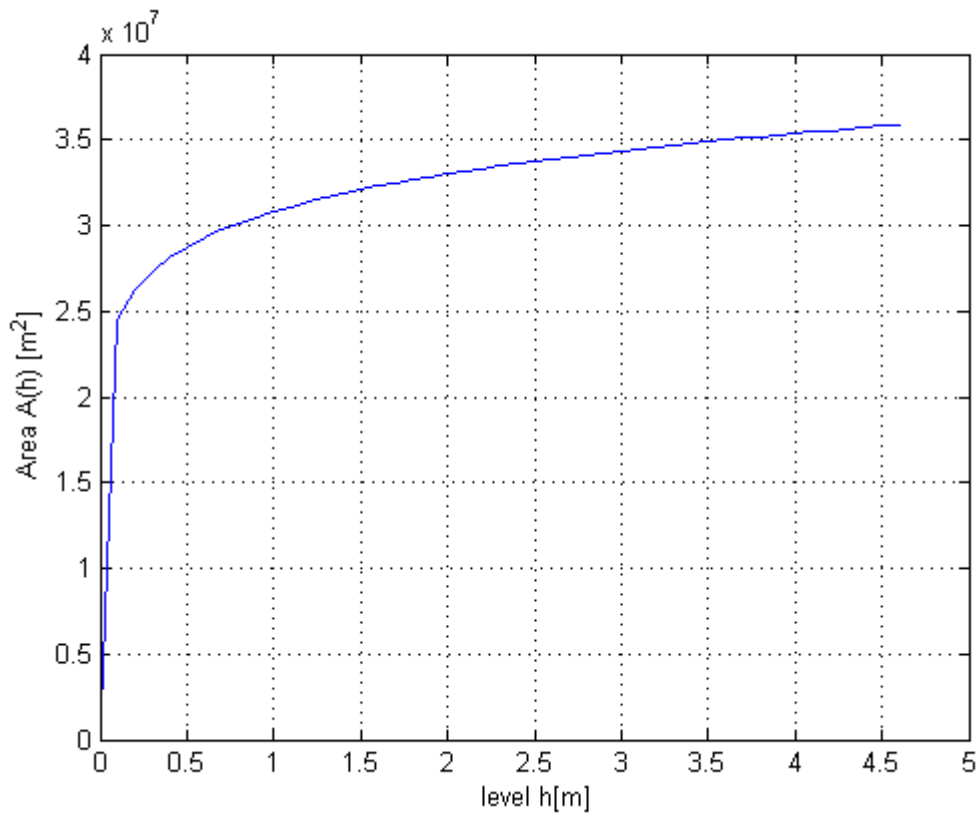


Figure 3-13: Simulation of surface area (A) as function of level h .

3.4 Linearization

The models until now we have discussed in above sections are nonlinear model that contains the function of dependent variables. The developed nonlinear model of our system describes the behavior of the lake Toke model but may not find the exact solution to predict the future behavior of the system and might be complex to interpret how well our system interacts with the nonlinear model whereas, in linear system it is relatively easier to predict the future behavior and nature of the system.

To find a linear model, we will in this section perform a linear analysis from the nonlinear model. Around the nominal value (operating point), we need a linear model to describe the nature of the system.

3.4.1 Linear Analysis

For a nonlinear system, the operating point $x_o(t)$, $u_o(t)$ and $y_o(t)$ must satisfy the nonlinear differential equation,

$$\frac{dx_o}{dt} = f(x_o, u_o; \theta)$$

$$y_o = g(x_o, u_o; \theta)$$

The operating point can normally be find by choosing the value of $u_o(t)$ and then simplifying differential equation to find $x_o(t)$ and $y_o(t)$. If we have a constant input (steady input), the analysis becomes much more simpler because at infinite time, the nonlinear model reaches to a steady state which gives constant output y_s and then we can find the solution to steady state x_s equating differential term $\frac{dx_o}{dt} = 0 = f(x_o, u_o; \theta)$

Then, expanding the differential term using Talyor's series expansion around operating point, the differential nonlinear equations becomes,

$$\begin{aligned} \frac{dh_1}{dt} &= f_1(h_1, h_2, \dot{V}_i) \\ &\approx f_1(h_1^o, h_2^o, \dot{V}_i^o) + \left. \frac{\partial f_1}{\partial h_1} \right|_o (h_1 - h_1^o) + \left. \frac{\partial f_1}{\partial h_2} \right|_o (h_2 - h_2^o) + \left. \frac{\partial f_1}{\partial \dot{V}_i} \right|_o (\dot{V}_i - \dot{V}_i^o) \\ &+ \text{higher ordered terms(neglected)} \end{aligned}$$

And

$$\begin{aligned}
\frac{dh_2}{dt} &= f_2(h_1, h_2, \dot{V}_i, \dot{V}_t, h_g) \\
&\approx f_2(h_1^o, h_2^o, \dot{V}_i^o, \dot{V}_t^o, h_g^o) + \left. \frac{\partial f_2}{\partial h_1} \right|_o (h_1 - h_1^o) + \left. \frac{\partial f_2}{\partial h_2} \right|_o (h_2 - h_2^o) \\
&\quad + \left. \frac{\partial f_2}{\partial \dot{V}_i} \right|_o (\dot{V}_i - \dot{V}_i^o) + \left. \frac{\partial f_2}{\partial \dot{V}_t} \right|_o (\dot{V}_t - \dot{V}_t^o) + \left. \frac{\partial f_2}{\partial h_g} \right|_o (h_g - h_g^o) \\
&\quad + \text{higher ordered terms (neglected)}
\end{aligned}$$

We assume, the operating point satisfies the nonlinear model, then

$$f_1(h_1, h_2, \dot{V}_i) = \frac{dh_1^o}{dt}$$

And,

$$f_2(h_1, h_2, \dot{V}_i, \dot{V}_t, h_g) = \frac{dh_2^o}{dt}$$

So, now rearranging the Taylor's series approximation we have,

$$\frac{d}{dt}(h_1 - h_1^o) = \left. \frac{\partial f_1}{\partial h_1} \right|_o (h_1 - h_1^o) + \left. \frac{\partial f_1}{\partial h_2} \right|_o (h_2 - h_2^o) + \left. \frac{\partial f_1}{\partial \dot{V}_i} \right|_o (\dot{V}_i - \dot{V}_i^o),$$

and

$$\begin{aligned}
\frac{d}{dt}(h_2 - h_2^o) &= \left. \frac{\partial f_2}{\partial h_1} \right|_o (h_1 - h_1^o) + \left. \frac{\partial f_2}{\partial h_2} \right|_o (h_2 - h_2^o) + \left. \frac{\partial f_2}{\partial \dot{V}_i} \right|_o (\dot{V}_i - \dot{V}_i^o) + \left. \frac{\partial f_2}{\partial \dot{V}_t} \right|_o (\dot{V}_t - \dot{V}_t^o) \\
&\quad + \left. \frac{\partial f_2}{\partial h_g} \right|_o (h_g - h_g^o)
\end{aligned}$$

Let us define the deviations from the operating points δx , δu and δy as,

$\delta x \triangleq x - x_o$, $\delta u \triangleq u - u_o$ and $\delta y \triangleq y - y_o$ and similarly we define the deviation of heights h_1 and h_2 form the operating point as well as for h_g , \dot{V}_i and \dot{V}_t as δh_1 , δh_2 , δh_g , $\delta \dot{V}_i$ and $\delta \dot{V}_t$.

Then we find,

$$\begin{aligned}
\frac{d}{dt} \delta h_1 &= \left. \frac{\partial f_1}{\partial h_1} \right|_o \delta h_1 + \left. \frac{\partial f_1}{\partial h_2} \right|_o \delta h_2 + \left. \frac{\partial f_1}{\partial \dot{V}_i} \right|_o \delta \dot{V}_i \\
\frac{d}{dt} \delta h_2 &= \left. \frac{\partial f_2}{\partial h_1} \right|_o \delta h_1 + \left. \frac{\partial f_2}{\partial h_2} \right|_o \delta h_2 + \left. \frac{\partial f_2}{\partial \dot{V}_i} \right|_o \delta \dot{V}_i + \left. \frac{\partial f_2}{\partial \dot{V}_t} \right|_o \delta \dot{V}_t + \left. \frac{\partial f_2}{\partial h_g} \right|_o \delta h_g
\end{aligned}$$

Finally, we can find the linear approximation as in the form of matrix and vectors (state-space form) that we will use in discretization and MPC application as follows,

$$\frac{d}{dt} \begin{pmatrix} \delta h_1 \\ \delta h_2 \end{pmatrix} = \begin{pmatrix} \left. \frac{\partial f_1}{\partial h_1} \right|_o & \left. \frac{\partial f_1}{\partial h_2} \right|_o \\ \left. \frac{\partial f_2}{\partial h_1} \right|_o & \left. \frac{\partial f_2}{\partial h_2} \right|_o \end{pmatrix} \begin{pmatrix} \delta h_1 \\ \delta h_2 \end{pmatrix} + \begin{pmatrix} \left. \frac{\partial f_1}{\partial \dot{V}_i} \right|_o & 0 & 0 \\ \left. \frac{\partial f_2}{\partial \dot{V}_i} \right|_o & \left. \frac{\partial f_2}{\partial \dot{V}_t} \right|_o & \left. \frac{\partial f_2}{\partial h_g} \right|_o \end{pmatrix} \begin{pmatrix} \delta \dot{V}_i \\ \delta \dot{V}_t \\ \delta h_g \end{pmatrix}$$

The above equation is in the form of standard state space equation,

$$\frac{d}{dt} (\delta x) = A_c \cdot \delta x + B_c \cdot \delta u$$

and,

$$y = C_c \delta x$$

Where,

$$x = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}, \text{ states of the system}$$

$$u = \begin{pmatrix} \dot{V}_i \\ \dot{V}_t \\ h_g \end{pmatrix}, \text{ inputs to the system}$$

$\delta x = x - x_o$, state deviation from operating point

$\delta u = u - u_o$, control deviation from operating point

$$A_c = \begin{pmatrix} \left. \frac{\partial f_1}{\partial h_1} \right|_o & \left. \frac{\partial f_1}{\partial h_2} \right|_o \\ \left. \frac{\partial f_2}{\partial h_1} \right|_o & \left. \frac{\partial f_2}{\partial h_2} \right|_o \end{pmatrix}$$

$$B_c = \begin{pmatrix} \left. \frac{\partial f_1}{\partial \dot{V}_i} \right|_o & 0 & 0 \\ \left. \frac{\partial f_2}{\partial \dot{V}_i} \right|_o & \left. \frac{\partial f_2}{\partial \dot{V}_t} \right|_o & \left. \frac{\partial f_2}{\partial h_g} \right|_o \end{pmatrix}$$

Using our developed model for linearization, we can find the linearized model matrices A_c and B_c in continuous form. The detailed mathematical calculations are shown in the appendix 2 in this report along with MATLAB code in appendix 5.

3.4.2 Discretization

A linearized discrete time model can be found out by discretizing the continuous time linear model that we have obtained in section 3.4.1 as follows:

$$\delta x_{k+1} = A \delta x_k + B \delta u_k$$

$$\delta y_k = C \delta x_k + D \delta u_k$$

Where A, B, C and D are now discrete-time model parameters. Considering discretization time T_s (in hours) and solving by MATLAB command using `c2d()` function we find discretized model matrix. One thing is noted during the discretization in MATLAB, that the discretization time T_s (in hours) plays an important role to decide the value of matrix; less the T_s , larger the values of discretized matrix.

3.5 Sensitivity of Parameters

The parameters we have used during the simulation for model are α, β, K_{12} and C_d . Using these parameters, the models are simulated and validated and the model will further used in MPC algorithm, so these parameters plays an important role in the model accuracy. These parameters are quiet uncertain so we can have a sensitivity test to find the most sensitive parameter. The most sensitive parameter is the most important parameter that contributes in our model.

In order to find the most important parameters among the 3 parameters, we change the parameters value by 5% of each and simulate the model one after another. The graph with increased parameters values for α, β, K_{12} are 0.0525, 0.021 & 840 respectively and the originally specified parameters values for α, β, K_{12} are 0.0525, 0.021 & 840 respectively are then compared. We choose order of sensitivity by observing the response of plot associated with each parameters; plot with more change is the most important one.

Let us compare the response of original plot of level h_1 and h_2 as in Figure 3-3 with same simulation time i.e. 40000 seconds with the increased parameter values shown in Figure 3-14, Figure 3-15 and Figure 3-16. Comparing those plots, we found there is no significant change in levels compared with the original plot, so we consider all 3 parameters contribute equally to the system model and are equally important. But, doing the analysis in more details we may find the most sensitive parameter.

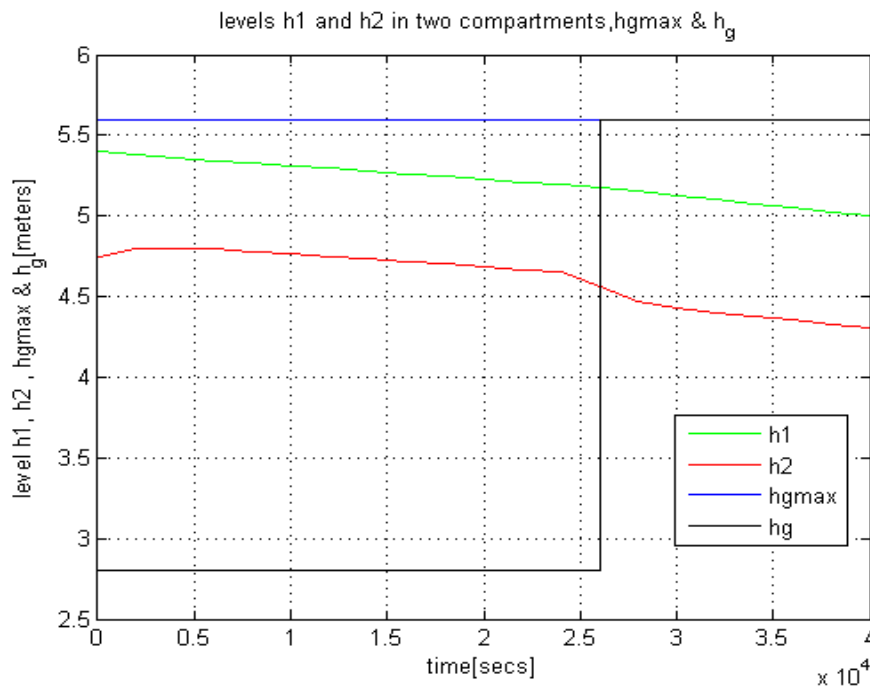


Figure 3-14: Levels h_1, h_2 in two compartments with 5% (0.0525) increased α parameter and operational conditions in steady state ($h1(t=0)=5.4, h2(t=0)=4.75$) at time period 40000 sec

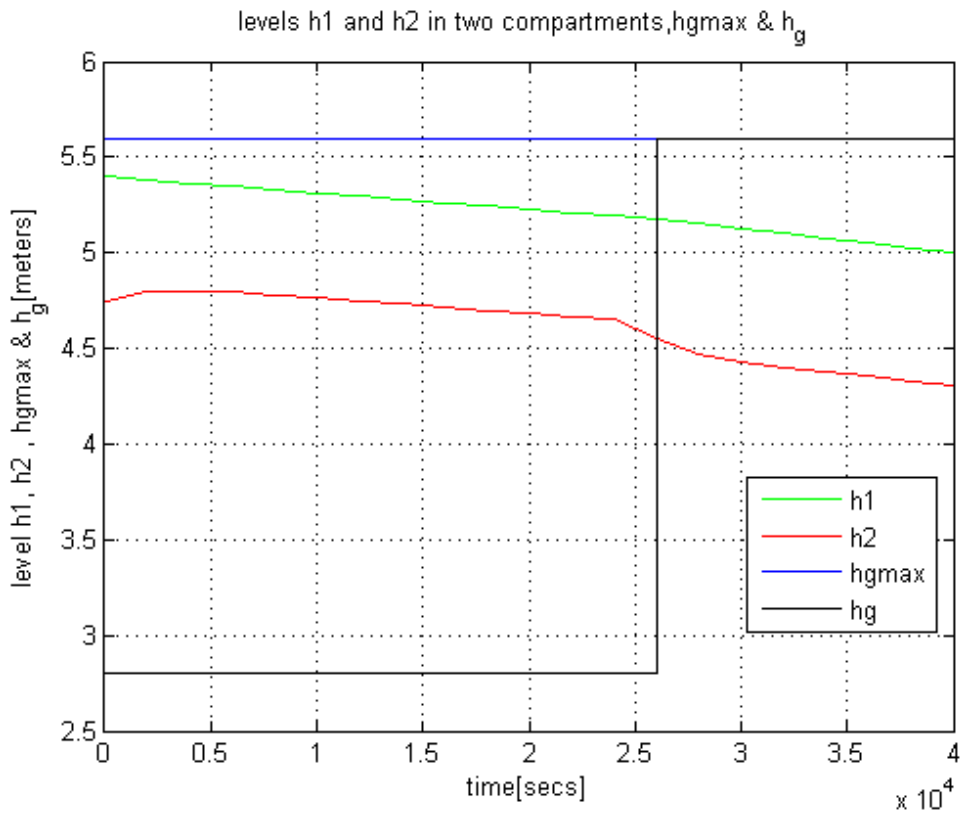


Figure 3-15: Levels h_1 , h_2 in two compartments with 5% (0.021) increased β parameter and operational conditions in steady state ($h_1(t=0)=5.4$, $h_2(t=0)=4.75$) at time period 40000 sec

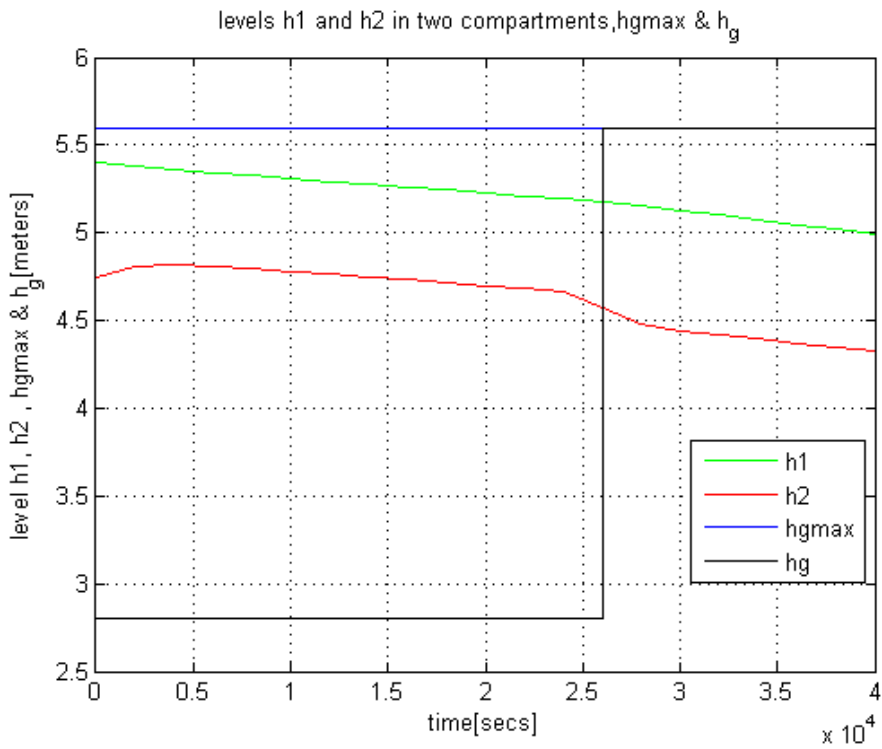


Figure 3-16: Levels h_1 , h_2 in two compartments with 5% (0.021) increased K_{12} parameter and operational conditions in steady state ($h_1(t=0)=5.4$, $h_2(t=0)=4.75$) at time period 40000 sec

4 Model Predictive Control (MPC)

The approach of Model Predictive Control (MPC) has been a new benchmark in the field of optimal control applications since 1980's. Explicitly, we can say that MPC is an especial case of optimal control theory. A new control input vector is calculated and supplied to the system every time in an optimization based problem that is governed by the constraints and performance index.

MPC is also known as Receding Horizon Control (RCH) because it is based on the iterative, finite horizon optimization of the process model and each time we get the new optimized value from optimization, the new control input u_0^* is applied to the process model. Hence, the prediction horizon (time length) keeps being shifted towards the final horizon (the horizon recedes). The basic principle of MPC strategy is that the current state of the process is sampled at time t , and the performance index is minimized using optimization at future time $t + T$ which gives a new control input and then we repeat the same process until the final time t_{final} . We only apply the first value of the computed control sequence and at the next time step, we get the system state and the process is repeated. Figure 4-1 shows the standard MPC principle.

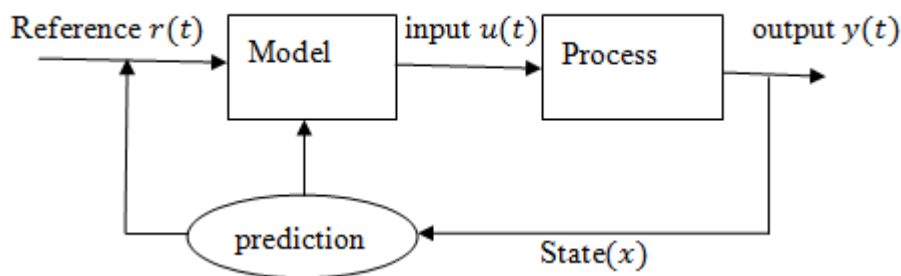


Figure 4-1:Block diagram of MPC algorithm

In the very first step of developing MPC algorithm, we need to define the prediction horizon and is normally denoted as N , and we will predict the next output using finite horizon. The system state is measured at time t , and the prediction through optimization solution gives the sequence of input $U_T(x(t))$ and fed to the dynamic model. The first element of the input sequence is applied to the process and finds the controlled output. This step is repeated until the desired output is achieved. Figure 4-2 shows the basic idea of MPC receding horizon. Then a good performance index (J) (cost function) is needed to define what we are controlling. Generally performance index is composed of error signal, control input and system constraints with their corresponding weighting matrices.

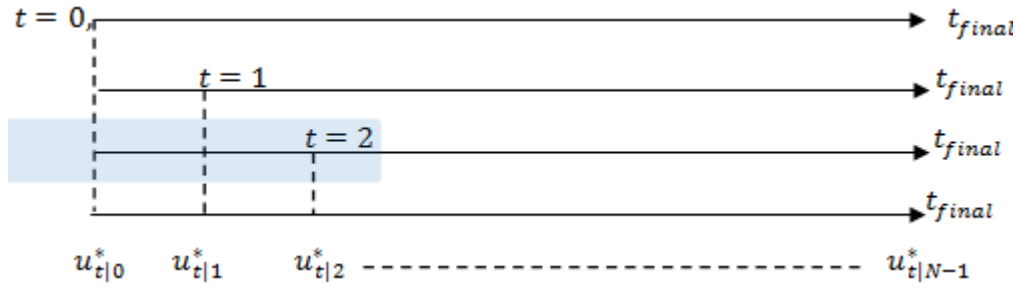


Figure 4-2: Illustration of receding horizon principle along with computed input sequence over a defined prediction horizon.

In our project, our main objective is to control flooding situations, so that the system state h_1 is to be minimized using MPC controller, alternatively we can say that the best control is achieved when we are subjected to minimize the flow through gate i.e. \dot{V}_g is minimum. So, in our performance index, height of gate is kept as much as minimum because we will minimize control input u_i , which contains h_g too. In our MPC solution, the performance index is limited by the dynamic model of the system. Also, the MPC controller is bounded by the system constraints i.e. upper and lower regulated water levels HRV and LRV. More over, our MPC controller considers the possible actuator limits and inflow prediction.

We have obtained a discretized linear model as in the form of,

$$\delta x_{k+1} = A\delta x_k + B\delta u_k + G\delta w_k \quad (4-1)$$

$$\delta y_k = C\delta x_k + H\delta w_k \quad (4-2)$$

Where, A matrix is a system state matrix, B is a control input matrix G and H are disturbance matrices in the system.

A performance tells us how well the system behaves over the prediction horizon. We want to minimize the performance index as much as possible with respect to the future input control vector. The purposed performance index for our MPC application is based on the performance index suggested by (Gøthesen et al., 2013) and is defined as,

$$I = \frac{1}{2} \sum_{i=1}^N [e_i^T P_i e_i + u_{i-1}^T Q_{i-1} u_{i-1} + \delta u_{i-1}^T R_{i-1} \delta u_{i-1} + (s_v)_i^T S_i (s_v)_i] \quad (4-3)$$

We are applying MPC algorithm in flood control of realistics data, hence we assume that the control input changes slowly. So, we replace $u_{i-1}^T Q_{i-1} u_{i-1}$ with $(u_{i-1} - u_{i-2})^T Q_{i-1} (u_{i-1} - u_{i-2})$ during the calculations.

The performance index consists the optimization parameters of error between reference signal and predicted output, control input, rate of change of control input and constraints (soft constraint with slack variable) and is always positive. According to our performance index in (4-3), the first element e_i , is subjected to minimize the error as,

$$e_i = r_i - y_i \quad (4-4)$$

Where,

r is a reference signal and y is predicted output. The reference signal is a value between the allowed regulated water level i.e HRV and LRV,

$r = \text{LRV} + a((\text{HRV}) - (\text{LRV}))$, a is a parameter for selecting r and is a positive value between 0 and 1.

Similarly, the second term u_{i-1} , reflects the input control to the process, the third term reflects the rate of change of control input vector as,

$$\delta u_{i-1} = u_{i-1} - u_{i-2} \quad (4-5)$$

The control input and rate of change of control input (control deviation) must be in same state.

Finally, the last term $(s_v)_i$, in our performance index illustrates the system soft constraints with slack variable that is to be optimized during the MPC application. The constraints we have applied in our application are equality and inequality constraints and will be discussed in next section.

P, Q, R and S are the weighting matrices of corresponding parameters and are users defined values and normally obtained by tuning. The values of weighing matrices must be in control between robustness and control performance. We have used the numerical values 10, 10, 10 and 1 of P, Q, R and S respectively in MATLAB solution.

4.1 Optimal control and Quadratic Programming (QP)

Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost function that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function (Wikipedia, 2014a).

The proposed performance index I in our case has quadratic terms, and hence we require quadratic programming to solve the performance index. QP has computational capability of controlled variable from a linear model to minimize quadratic performance index subjected to constraints over a defined prediction horizon.

So, rephrasing the above performance index in equation (4-3) to a standard form for QP,

$$I = \frac{1}{2} z^T H z + c^T z \quad (4-6)$$

Subjected to the constraints,

$$\begin{cases} A_e z = b_e \\ A_i z \leq b_i \\ z^l \leq z \leq z^u \end{cases} \quad (4-7)$$

Where,

H is known as Hessian matrix and is symmetric positive definite ($H > 0$), constant diagonal matrix.

c is a column matrix of independent sequence of unknown present and future inputs and outputs and is defined by the model with known inputs and outputs. We define c matrix as,

$$c^T = (c_1^T, c_2^T, c_3^T, c_4^T, c_5^T, c_6^T, c_7^T), \quad c \in \mathbb{0}^{N.(3n_u+3n_y+n_x)}$$

c Matrix is zero in our particular condition because the performance index does not include any of the linear terms.

z is a vector of unknowns for QP of the model and is assumed as,

$$z^T = \begin{cases} u^T = (u_1^T, \dots, u_N^T), & u \in \mathbb{R}^{N.n_u \times 1} \\ \delta u^T = (\delta u_1^T, \dots, \delta u_N^T), & \delta u \in \mathbb{R}^{N.n_u \times 1} \\ e^T = (e_1^T, \dots, e_N^T), & e \in \mathbb{R}^{N.n_y \times 1} \\ y^T = (y_1^T, \dots, y_N^T), & y \in \mathbb{R}^{N.n_y \times 1} \\ x^T = (x_1^T, \dots, x_N^T), & x \in \mathbb{R}^{N.n_x \times 1} \\ s_v^T = (s_{v1}^T, \dots, s_{vN}^T), & s_v \in \mathbb{R}^{N.n_y \times 1} \\ v_t^T = (v_{t1}^T, \dots, v_{tN}^T), & v_t \in \mathbb{R}^{N.n_u \times 1} \end{cases} \quad (4-8)$$

We can represent z^T as,

$$z^T = (u^T, \delta u^T, e^T, y^T, x^T, s_v^T, v_t^T), \quad z \in \mathbb{R}^{N.(3n_u+3n_y+n_x)} \quad (4-9)$$

Where,

u^T is a sequence of control input over a prediction horizon N . Similarly, δu^T is a sequence of deviation of control input, e^T is a sequence of error, y^T is a sequence of predicted output, x^T is a sequence of states, s_v^T is a sequence of soft constraints and v_t^T is a sequence of values of predicted turbine flow \dot{V}_t . Though we have developed a prediction model for \dot{V}_t in earlier chapters (and may not need to compute again in MPC), we are still interested to find \dot{V}_t because QP give more precise value for \dot{V}_t in this context.

Let us assume that the initial values $u_0, x_1, r_1, \dots, r_N$ are known, then the performance index in equation (4-6) can be in the following form,

$$I = \frac{1}{2} \underbrace{\begin{bmatrix} u \\ \delta u \\ e \\ y \\ x \\ s_v \\ v_t \end{bmatrix}^T}_{z} \underbrace{\begin{bmatrix} H_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & H_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & H_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & H_{44} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & H_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & H_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & H_{77} \end{bmatrix}}_H \underbrace{\begin{bmatrix} u \\ \delta u \\ e \\ y \\ x \\ s_v \\ v_t \end{bmatrix}}_z + \underbrace{\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix}^T}_c \underbrace{\begin{bmatrix} u \\ \delta u \\ e \\ y \\ x \\ s_v \\ v_t \end{bmatrix}}_z \quad (4-10)$$

Expanding equation (4-10) and putting $c = 0$, we get

$$z^T H z = u^T H_{11} u + \delta u^T H_{22} \delta u + e^T H_{33} e + y^T H_{44} y + x^T H_{55} x + s_v^T H_{66} s_v + v_t^T H_{77} v_t \quad (4-11)$$

Hence after simplification we define the elements of Hessian matrix,

$$\left. \begin{aligned} u^T H_{11} u &= \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}^T \begin{bmatrix} Q & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Q \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} &\Rightarrow H_{11} = I_N \otimes Q \\ \delta u^T H_{22} \delta u &= \begin{bmatrix} \delta u_1 \\ \vdots \\ \delta u_N \end{bmatrix}^T \begin{bmatrix} R & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & R \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \vdots \\ \delta u_N \end{bmatrix} &\Rightarrow H_{22} = I_N \otimes R \\ e^T H_{33} e &= \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix}^T \begin{bmatrix} P & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & P \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix} &\Rightarrow H_{33} = I_N \otimes P \\ H_{44} &= I_N \otimes 0_{ny} &\Rightarrow H_{44} = 0_{N.ny} \\ H_{55} &= I_N \otimes 0_{nx} &\Rightarrow H_{55} = 0_{N.nx} \\ e s_v^T H_{66} s_v &= \begin{bmatrix} s_v \\ \vdots \\ s_{vN} \end{bmatrix}^T \begin{bmatrix} S & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & S \end{bmatrix} \begin{bmatrix} s_v \\ \vdots \\ s_{vN} \end{bmatrix} &\Rightarrow H_{66} = I_N \otimes S \\ H_{77} &= I_N \otimes 0_{nu} &\Rightarrow H_{77} = 0_{N.nu} \end{aligned} \right\} \quad (4-13)$$

In equation (4-13), the output ($y^T H_{44} y$) and input ($x^T H_{55} x$) terms becomes zero, so H_{44} and H_{55} elements are zero.

Finally, the Hessian matrix is represented as,

$$H = \text{diag}(H_{11}, H_{22}, H_{33}, H_{44}, H_{55}, H_{66}, H_{77}) \quad (4-14)$$

4.2 Constraints handling

While developing the MPC solution, the system model is bounded by constraints. We have already discussed that the cost function is subjected the constraints either equality constraint or inequality constraint.

4.2.1 Equality Constraints

The equality constraint is applied to the linear model of the system, which includes the input and output. We have developed equality constraints according to the unknown we have defined in equation (4-8).

The equality constraint is represented as,

$$A_e z = b_e$$

Rearranging equation (4-2), up to N terms, we find the kronical form as in equation (4-16),

$$\left. \begin{aligned} \delta y_1 &= C\delta x_1 + H\delta w_1 \\ \delta y_2 - C\delta x_2 &= H\delta w_2 \\ &\vdots \\ \delta y_N - C\delta x_N &= H\delta w_N \end{aligned} \right\} \quad (4-15)$$

Now, we arrange the above expanded equation in the form of standard equality constraint,

$$\left. \begin{aligned} A_{e,11} &= -I_N \otimes H \\ A_{e,14} &= I_{N.ny} \\ A_{e,15} &= -I_{N,-1} \otimes C \\ b_{e,1} &= [C\delta x_1 + H\delta w_1 \quad H\delta w_2 \quad \dots \quad H\delta w_N]^T \end{aligned} \right\} \quad (4-16)$$

The elements presented in equation (4-16) are only the elements of first row of A_e and b_e , the other elements in first row of A_e are zeros.

Similarly rearranging equation (4-1), up to $i = 1, \dots, N$ terms, we find the kronical form as in equation (4-18),

$$\left. \begin{aligned} \delta x_2 &= A\delta x_1 + B\delta u_1 + G\delta w_1 \\ \delta x_3 &= A\delta x_2 + B\delta u_2 + G\delta w_2 \\ &\vdots \\ \delta x_N &= A\delta x_{N-1} + B\delta u_{N-1} + G\delta w_{N-1} \end{aligned} \right\} \Rightarrow \begin{aligned} -B\delta u_1 + \delta x_2 &= A\delta x_1 + G\delta w_1 \\ -B\delta u_2 - A\delta x_2 + \delta x_3 &= G\delta w_2 \\ &\vdots \\ -B\delta u_{N-1} - A\delta x_{N-1} + \delta x_N &= G\delta w_{N-1} \end{aligned} \quad (4-17)$$

Expanding equation (4-17) in the form of standard equality constraint,

$$\left. \begin{aligned} A_{e,21} &= -I_N \otimes B \\ A_{e,25} &= I_N \otimes I_{nx} - I_{N,-1} \otimes A = I_{N.nx} - I_{N,-1} \otimes A \\ b_{e,2} &= [A\delta x_1 + B\delta w_1 \quad B\delta w_2 \quad \dots \quad B\delta w_{N-1}]^T \end{aligned} \right\} \quad (4-18)$$

The elements presented in equation (4-18) are only the elements of second row of A_e and b_e , the other elements in second row of A_e are zeros.

Again, rearranging equation (4-4), up to $i = 1, \dots, N$ terms, we find the kronical form as in equation (4-20),

$$\left. \begin{aligned} e_1 + y_1 &= r_1 \\ &\vdots \\ e_N + y_N &= r_N \end{aligned} \right\} \quad (4-19)$$

Expanding equation (4-19) in the form of standard equality constraint,

$$\left. \begin{aligned} A_{e,33} &= I_{N.ny} \\ A_{e,34} &= I_{N.ny} \\ b_{e,3} &= [r_1 \quad r_2 \quad \dots \quad r_N]^T \end{aligned} \right\} \quad (4-20)$$

The elements presented in equation (4-20) are only the elements of third row of A_e and b_e , the other elements in third row of A_e are zeros.

Finally, rearranging equation (4-5), up to $i = 1, \dots, N$ terms, we find the kronical form as in equation (4-22),

$$\left. \begin{array}{l} -u_1 + \delta u_1 = -u_0 \\ u_1 - u_2 + \delta u_2 = 0 \\ \vdots \\ u_{N-1} - u_N + \delta u_N = 0 \end{array} \right\} \quad (4-21)$$

Expanding equation (4-21) in the form of standard equality constraint,

$$\left. \begin{array}{l} A_{e,41} = I_{N,-1} \otimes I_{nu} - I_N \otimes I_{nu} = I_{N,-1} \otimes I_n - I_{N,nu} \\ A_{e,42} = I_N \otimes I_{nu} = I_{N,nu} \\ b_{e,4} = [-u_0 \quad 0 \quad \dots \quad 0_{(N-1),nu}] \end{array} \right\} \quad (4-22)$$

The elements presented in equation (4-22) are only the elements of fourth row of A_e and b_e , the other elements in fourth row of A_e are zeros.

Now, we have obtained all the values of A_e and b_e matrix, so putting all the sub matrices from equation (4-16), (4-18), (4-20) and (4-22), we find the A_e and b_e matrix as follows:

$$A_e = \begin{bmatrix} 0_{N,nx \times N,nu} & -I_N \otimes H & 0_{N,ny \times N,nu} & I_{N \times ny} & -I_{N,-1} \otimes C & 0_{N,ny \times N,ny} & 0_{N,ny \times N,nu} \\ -I_N \otimes B & 0_{N,ny \times N,nu} & 0_{N,nx \times N,ny} & 0_{N,nx \times N,ny} & I_{N,nx} - I_{N,-1} \otimes A & 0_{N,nx \times N,nu} & 0_{N,nx \times N,ny} \\ 0_{N,ny \times N,nu} & 0_{N,ny \times N,nu} & I_{N,ny} & I_{N,ny} & 0_{N,ny \times N,nx} & 0_{N,ny \times N,ny} & 0_{N,nu \times N,nu} \\ I_{N,-1} \otimes I_n - I_{N,nu} & I_{N,nu} & 0_{N,nu \times N,xy} & 0_{N,nu \times N,xy} & 0_{N,nu \times N,ny} & 0_{N,nu \times N,xy} & 0_{N,xu \times N,nu} \end{bmatrix}$$

$$b_e = [C\delta x_1 \quad 0_{(N-1) \times 1} \quad A\delta x_1 \quad 0_{(N-1) \times 1} \quad r_1 \quad \dots \quad r_N \quad -u_0 \quad 0_{(N-1) \times 1}]^T$$

4.2.2 Inequality constraints & slack variables

The MPC solution must satisfy the inequality constraint as well. We will in this section define the inequality constraints that the system must satisfy.

We have a limited level of gate opening which rangers from 0 to 5.6 meters. The value of gate opening hg_i must lie in between the upper bound $hg^u = 5.6 \text{ m}$ and lower bound $hg^l = 0 \text{ m}$.

$$\text{i.e. } hg^l \leq hg_i \leq hg^u \quad (4-23)$$

we have already discussed in previous section that the water level at Merkebekk h_1 is regulated by NVE, and is vary from time to time. Thus, we define the water level of Merkebekk y_i , as output constraint with in a boundary of lower limit y^l and upper limit y^u ,

$$y^l \leq y_i \leq y^u \quad (4-24)$$

If there is a access of rain fall or the extreme condition when there is long period of rain fall occured, then we should extend the output constraint by using slack variable in order to handle these situations. Now the constraint becomes soft constraint and the added slack variable must be positive value, $S_v > 0$. We can represent the soft constraint as,

$$y^l - S_v \leq y_i \leq y^u + S_v \quad (4-25)$$

Rearranging equation (4-25), up to $i = 1, \dots, N$ terms we get,

$$\left. \begin{array}{l} y^l - S_i \leq y_i \\ -y_1 - s_1 \leq -y^l \\ \vdots \\ -y_N - s_N \leq -y^l \end{array} \right\} \text{And,} \quad \left. \begin{array}{l} y^u + S_i \geq y_i \\ -y_1 - s_1 \leq y^u \\ \vdots \\ -y_N - s_N \leq y^u \end{array} \right\} \quad (4-26)$$

Comparing and rearranging equation (4-26) to standard inequality constraint, the kronical value of the output and slack variable will be: $A_{i,14} = -I_{N.ny}$ and $A_{i,16} = -I_{N.ny}$, other terms in the first row of A_i are zeros.

Similarly, $A_{i,24} = I_{N.ny}$ and $A_{i,26} = -I_{N.ny}$, other terms in the second row of A_i are zeros.

The b_i vector: $b_{i,1} = (-1_{N.ny})y^l$ and $b_{i,2} = (1_{N.ny})y^u$

Moreover, we need to consider the minimum outflow from a lake. We have provided a combined outflow from the turbine and flood gate minimum of $4 \text{ m}^3/\text{s}$ i.e,

$$\dot{V}_t + \dot{V}_g \geq 4 \text{ m}^3/\text{s}$$

And we have, $\dot{V}_g = hgw\sqrt{2gh_2}$.

Solving these above two equations using Euler's seires expansion with necessary parameters and steady state values we find,

$$-h_g \leq \frac{(\dot{V}_t - 4)}{78.4}$$

Expanding up to N^{th} term we have,

$$\begin{array}{l} -h_{g_1} \leq \frac{(\dot{V}_{t_1} - 4)}{78.4} \\ \vdots \\ -h_{g_N} \leq \frac{(\dot{V}_{t_N} - 4)}{78.4} \end{array}$$

This yeilds $A_{i,31} = -I_{N.nu}$ and $b_{i,3} = 1_{nu.N} \times \frac{(\dot{V}_{t_N} - 4)}{78.4}$

Further, we have put slack variable, so we will consider a constraint to slack variable. The slack variable must be always positive,

$$Sv \geq 0$$

$$\begin{aligned} s_{v1} &\geq 0 \\ &\vdots \\ s_{vN} &\geq 0 \end{aligned}$$

This yields $A_{i,46} = -I_{N.ny}$ and $b_{i,4} = 0_{ny.N}$

Thus, the matrixes for inequality constraint are:

$$A_i = \begin{bmatrix} 0_{N.ny \times N.nu} & 0_{N.ny \times N.nu} & 0_{N.ny \times N.ny} & I_{N \times ny} & 0_{N.ny \times N.nx} & I_{N \times ny} & 0_{N.ny \times N.nu} \\ 0_{N.ny \times N.xy} & 0_{N.ny \times N.xy} & 0_{N.xy \times N.ny} & I_{N \times ny} & 0_{N.xy \times N.nx} & -I_{N \times xy} & 0_{N.xy \times N.nu} \\ -I_{N \times nu} & 0_{N.nu \times N.xu} & 0_{N.nu \times N.nu} & 0_{N.nu \times N.nu} & 0_{N.nu \times N.nx} & 0_{N.nu \times N.xu} & I_{N \times nu} \\ 0_{N.xy \times N.ny} & 0_{N.xy \times N.ny} & 0_{N.ny \times N.ny} & 0_{N.ny \times N.ny} & 0_{N.ny \times N.nx} & -I_{N \times ny} & 0_{N.ny \times N.nx} \end{bmatrix}$$

And

$$b_i = \left[(-1_{N.ny})y^l \quad (1_{N.ny})y^u \quad 1_{N \times nu,1} \times \frac{(\dot{V}_{tN} - 4)}{78.4} \quad 0_{n,xy \times 1} \right]$$

4.2.3 MPC Implementation

The MPC solution we have developed are based on the MATLAB codes developed by (Gøthesen et al., 2013) but with modifications. All the related MATLAB files (M-files) are attached in the appendix of the report under 'MPC' portion. The results are obtained after running MATLAB script files with all necessary parameters and MATLAB functions defined. We have developed a m-file in order to run either choosing linear model or nonlinear model of the lake Toke. The m-file named 'nonlinear_model.m' is used to describe the nonlinear model and 'linear_model.m' describes the linearized model.

The main MPC program is in the file named 'MPC.m' which takes its input variables from an excel file 'inputs.xlsx'. The excel file 'inputs.xlsx' contains predicted inflow \dot{V}_i , approximated turbine flow \dot{V}_i and HRV as well as LRV defined for lake Toke from NVE. The MPC.m file calculates every time a new control vector and the process is repeated until the controlled output is achieved. The prediction horizon N , number of samples n , parameter for selecting the reference level a , selection of linear or nonlinear model L or NL , weighting matrices P, Q, R, S and discretization time t were defined by the users.

We have another file 'Run_MPC.m' associated with the main MPC file which gives the result for MPC program. This file reads the data from an excel file 'inputs.xlsx' and calls the function MPC.m to execute and gives the output showing the controlled gate height h_g along with corresponding controlled level at Merkebekk h_1 , turbine flow and gate flow.

We have used the lower bound (z_l^T) and upper bound (z_u^T) for the unknown variables in equation (4-8). The lower and upper bounds are defined from $-\infty$ to ∞ for $\delta u^T, e^T, x^T, S_v^T$ and V_t^T and the user defined limit for u^T and y^T . The bounds we have used in MATLAB program is as follows:

$$z_l^T = (U_l^T, \delta U_l^T, E_l^T, Y_l^T, X_l^T, S_{vl}^T, V_{tl}^T)$$

$$z_u^T = (U_u^T, \delta U_u^T, E_u^T, Y_u^T, X_u^T, S_{v_u}^T, V_{t_u}^T)$$

And each elementary bounds for lower bound (z_l^T) and upper bound (z_u^T) are described as:

$$\left. \begin{array}{l} U_u^T = (u_{u1}, \dots, u_{un})^T \\ \delta U_u^T = (\infty, \dots, \infty)^T \\ E_u^T = (\infty, \dots, \infty)^T \\ Y_u^T = (y_{u1}, \dots, y_{un})^T \\ X_u^T = (\infty, \dots, \infty)^T \\ S_{vU}^T = (\infty, \dots, \infty)^T \\ V_{tu}^T = (\infty, \dots, \infty)^T \end{array} \right\} \text{upper bounds}$$

$$\left. \begin{array}{l} U_l^T = (u_{l1}, \dots, u_{lN})^T \\ \delta U_l^T = (-\infty, \dots, -\infty)^T \\ E_l^T = (-\infty, \dots, -\infty)^T \\ Y_l^T = (y_{l1}, \dots, y_{lN})^T \\ X_l^T = (-\infty, \dots, -\infty)^T \\ S_{v_l}^T = (-\infty, \dots, -\infty)^T \\ V_{t_l}^T = (-\infty, \dots, -\infty)^T \end{array} \right\} \text{lower bounds}$$

4.2.4 Result Intrepretation

The MATLAB simulation of the developed MPC application is analysed and concluded with appropriate remarks for our study. We have run the MPC application both in linear and nonlinear models and assuming different inflow for the lake Toke.

During the simulation process, we observed that simulation time depends on the parameters of weighting matrices, number of sample points, type of model used and length of prediction horizon. The initial heights of both compartment h_1 & h_2 were assumed 2.5 m.

We are concerned on the control of flooding situations, so here we have predicted inflow to the lake operated under normal and extreme flooding condition ($250 - 900 \text{ m}^3/\text{s}$). After number of simulations, we tuned the weighting matrices P, Q and R to 10, 8 and 10 respectively and S to 1, to obtain acceptable response in MPC. We have used discrete sampling time of 4 hours, 70 samples and prediction horizon 10. Though we are provided around 8,300 sample points, but we have only used 70 samples among them selecting the most relevant data for flooding conditions. If we choose all the data samples, then it requires very large computational time.

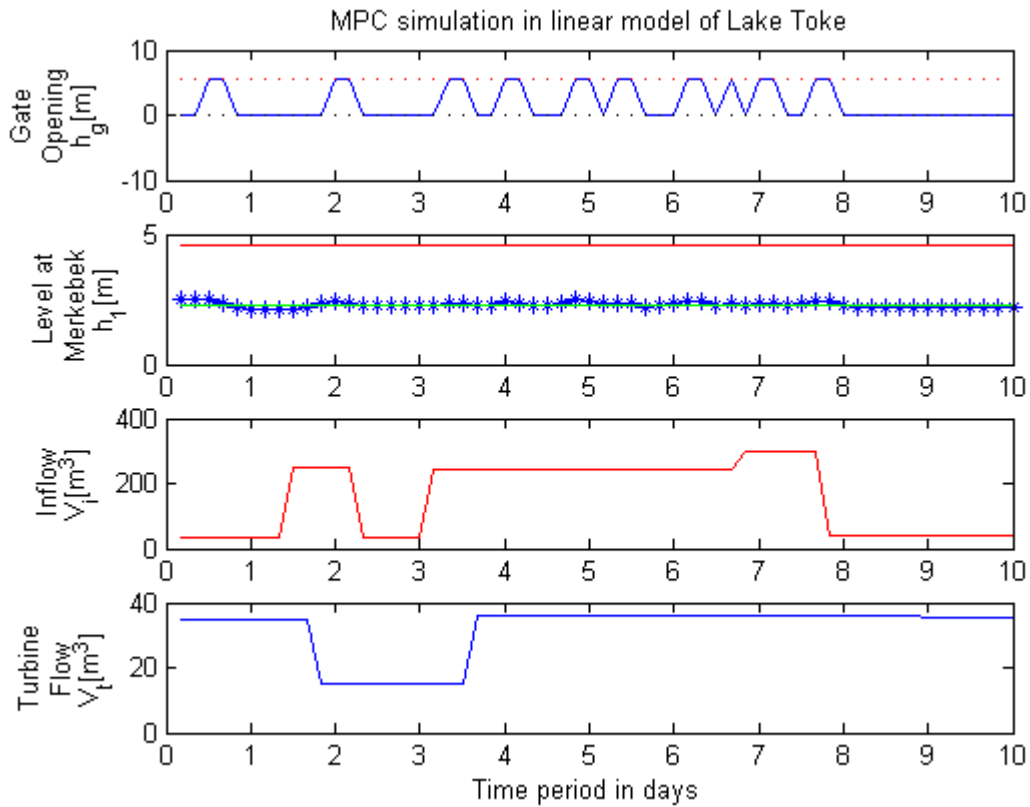


Figure 4-3: MPC simulation of linear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 5.5 days over a time stamp of 10 days.

Figure 4-3 illustrates the MPC simulation of linear model with predicted inflow $250 \text{ m}^3/\text{s}$ for around 5.5 days. In both Figure 4-3 and Figure 4-4, the gate opening h_g in meter, controlled height of Merkebekk h_1 , predicted inflow and turbine flow is shown. According to Figure 4-3 and Figure 4-4, it can be noticed that when the inflow is increased heavily at around $250 \text{ m}^3/\text{s}$ (maximum inflow), the gate opening increases immediately by passing the water and decreases after the inflow is dropped. MPC controller increases the gate opening in order to track the reference level so, the gate height depends the inflow to the lake. Mean while, when predicted inflow is decreased sharply, the level in the Merkebekk h_1 , tries to deviate from reference level but the MPC controller makes to increase gate opening in order to maintain the level at around reference level. Hence, due to the MPC controller action, it is observed that the height at Merkebekk seems controlled enough following the reference level at around 2.5 m, even if predicted inflow changes heavily. Moreover, the MPC controller is able to handle the constraints easily, non of the bounds are touched through entire simulation time even we had fed comparatively large inflow value continuously.

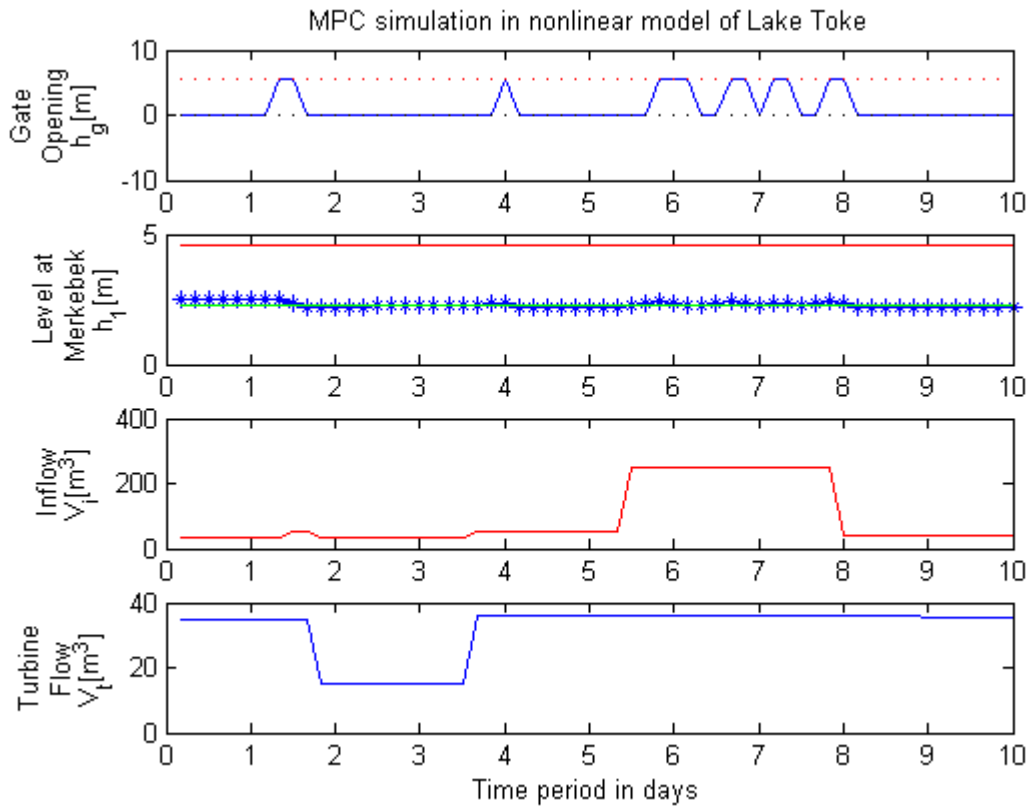


Figure 4-4: MPC simulation of nonlinear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 2.5 days over a time stamp of 10 days

Similarly, Figure 4-4 is the MPC simulation of nonlinear model with predicted inflow $250 \text{ m}^3/\text{s}$ for around 5.5 days. The difference in Figure 4-3 and Figure 4-4 is the type of model used during the simulation.

Figure 4-5 is the MPC simulation of linear model with predicted inflow $500 \text{ m}^3/\text{s}$ for around 2.5 days whereas Figure 4-6 is the MPC simulation of nonlinear model with predicted inflow $500 \text{ m}^3/\text{s}$ for around 5.5 days. The difference in Figure 4-3 and Figure 4-4 is the type of model used during the simulation.

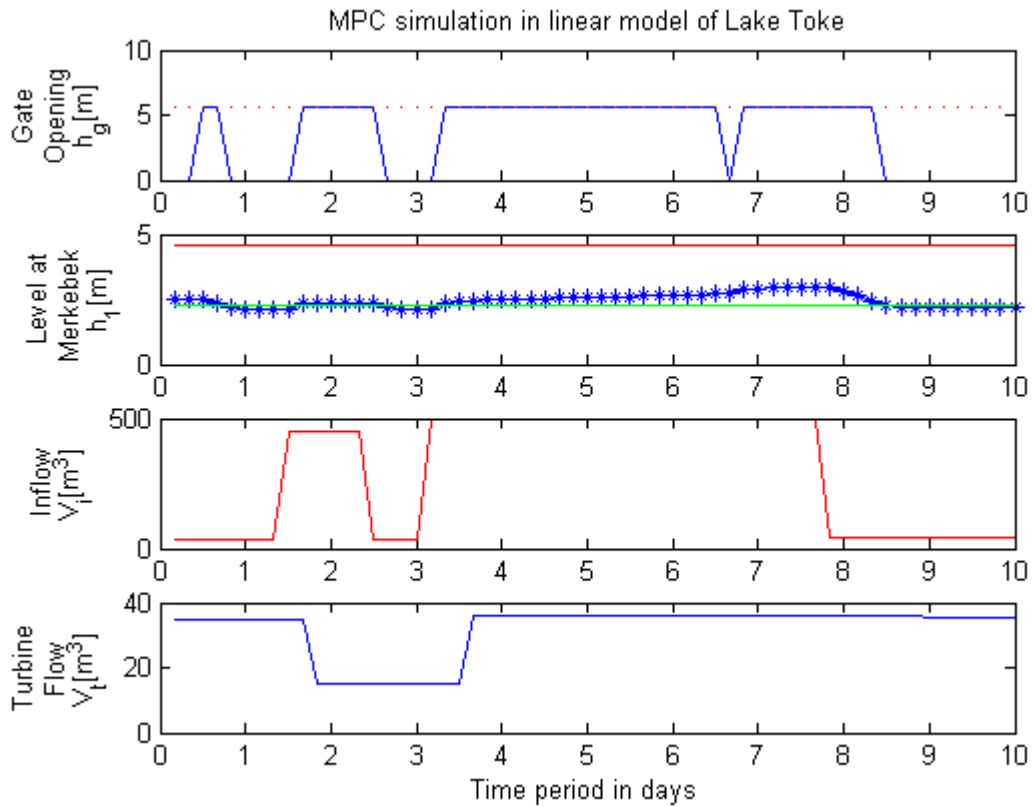


Figure 4-5: MPC simulation of linear model with predicted inflow $500 \text{ m}^3/\text{s}$ for approximately 5.5 days over a time stamp of 10 days

Due to the nonlinear nature of the model the response of nonlinear model is quiet slow than that of linear model. The response seems quiet same in both the models, but actually the the response time of output 'h₁' towards reference 'r' in a linearized model is faster than the nonlinear model comparing Figure 4-5 and Figure 4-6. The response of linear model is very close to reference level in linear model whereas nonlinear model is still following the reference.

During MPC simulation, the turbine flow is same in all the plots. The turbine flow is approximated and scaled using the generated power relation that we have developed in chapter 2. When turbine flow is less, the opening of gate is almost zero in all cases.

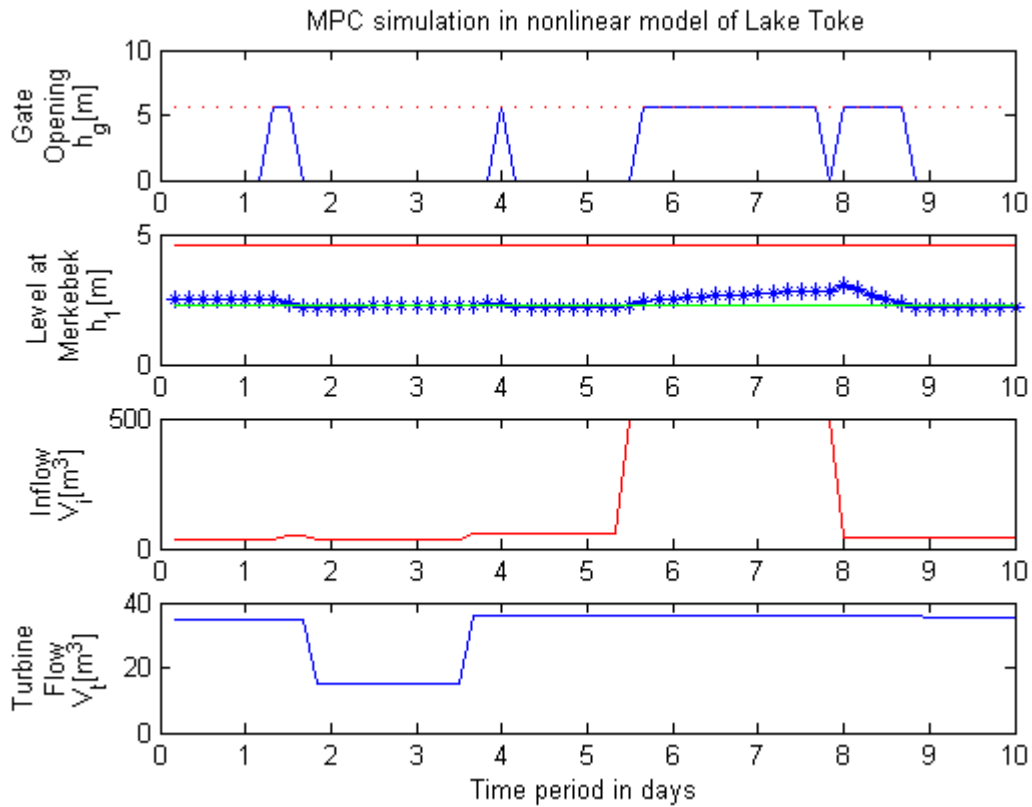


Figure 4-6: MPC simulation of nonlinear model with predicted inflow $250 \text{ m}^3/\text{s}$ for approximately 2 days over a time stamp of 10 days

Figure 4-7 represents the MPC simulation of nonlinear model with excess of inflow (predicted) about $900 \text{ m}^3/\text{s}$ for approximately 2 days. The upper boundary for the level is destroyed and the level jumps when the excess of inflow is treated to the system. The soft constraint that we have considered is violated when the very large inflow is feed. On the contrary, the water level is still above the lower bound that we had considered as a minimum outflow constrain during the entire simulation. This seems that our MPC application throughly worked on our model and assumptions that we have developed and a considered during the project.

Weighting matrices are very important while developing the MPC application. They play vital role on the system response, hence a careful consideration should be carried on during the simulation. We noticed that when the value of Q matrix is less, the gate opening is more and vice versa. this might happened beacuse the Q matrix is associated with control vector in performance index. Also, the value of $\frac{Q}{P}$ effects the system performance. If $\frac{Q}{P}$ is higher, then the system behaves faster and if $\frac{Q}{P}$ becomes less, the system becomes slower.

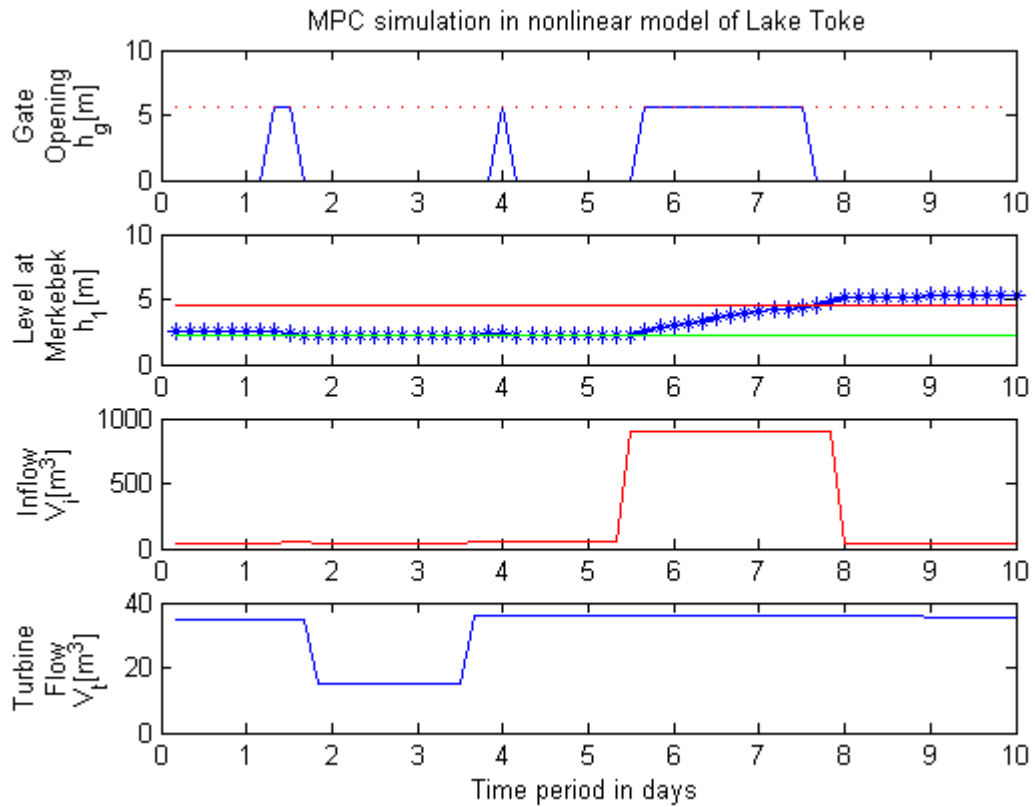


Figure 4-7: MPC simulation of nonlinear model with predicted inflow 900 m³/s for approximately 2 days showing the extreme flooding condition over a time stamp of 10 days

From the above MPC simulation, it can be concluded that introducing model error, where future inflow of water is unknown or big uncertainties are introduced, MPC can handle it.

5 Stochastic MPC

5.1 Background

A control problem is applied on a real process where there is a demand of controlling too many uncertain variables that are needed to track. We cannot guarantee the pattern of inflow to the system neither we can predict the ensemble that likely to happen in adverse situations, so basically we can say that in real control problems there is uncertainty. Our main task in this thesis work is to do stochastic analysis of the MPC application of Lake Toke. Though we have developed a simple MPC application for the Lake Toke, still our main task is to do further analysis on random MPC problem. The solutions and simplifications we had made in previous chapter may be inadequate for the stochastic phenomena and may lead misconception for the random variables that are to be controlled during stochastic process. The stochastic process contains sequence of random variables and time series even if the initial value is known (Wikipedia, 2014b). So, we can say that to describe stochastic solution in our case the probability and statistical considerations should be made.

5.2 Markowitz Portfolio Optimization

The Markowitz portfolio theory was developed by an economist Harry Markowitz in 1952 which mainly uses mathematical models to minimize risk and optimize return depending on risk (Marling and Emanuelsson, 2012) .

The Markowitz portfolio Optimization theory is based up on probability theory and concept of random variables. Let us consider the input vector x and output vector y of random variables as,

$$X = (x_1, \dots, x_n)^T$$

$$Y = (y_1, \dots, y_n)^T$$

Then, Markowitz portfolio Optimization theory describes,

$$Y = \min \sum_{i=1}^n p_i x_i$$

subjected to: $A_e \geq b_e$ (constraints)

Where

p_i is the probability of ensemble of future disturbances (input) to a system.

We can use this theory in our MPC application with modifications and necessary assumption for a stochastic process.

5.3 Stochastic analysis

We have done a simple stochastic analysis in order to observe how well the MPC works with different inflow scenario to Lake Toke model. The stochastic process comprises of random input variables. Figure 5-1 represents the simulation of real prediction inflow of $300 \text{ m}^3/\text{s}$ of around 8.5 day. The MPC controller seems worked well tracking the reference level in this condition. This implies that our MPC application is able to handle the continuous flooding situation to produce a controlled water level giving sufficient gate opening.

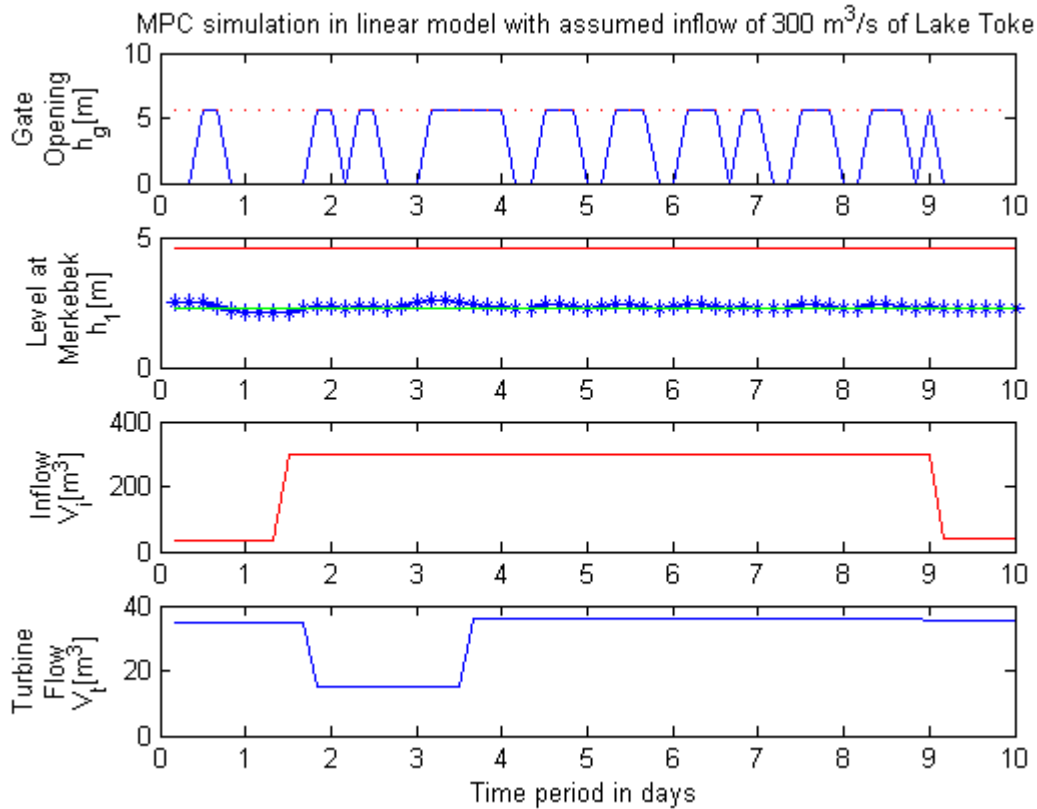


Figure 5-1: MPC simulation with assumed inflow of $300 \text{ m}^3/\text{s}$ in linear model

Similarly, Figure 5-2 represents the simulation of randomly assumed inflow throughout the entire simulation time. The MPC controller seems worked well tracking the reference level until the acceptable inflow of approximately $500 \text{ m}^3/\text{s}$, and then the controlled level started to raise towards upper bound as well as the maximum gate opening is achieved in this condition. This is the condition where the MPC cannot handle extremely very big inflow and may violate the soft constrain but still the lower bound is satisfied. This implies that our MPC application is able to handle the continuous randomly assumed flooding situation to produce a controlled water level giving sufficient gate opening.

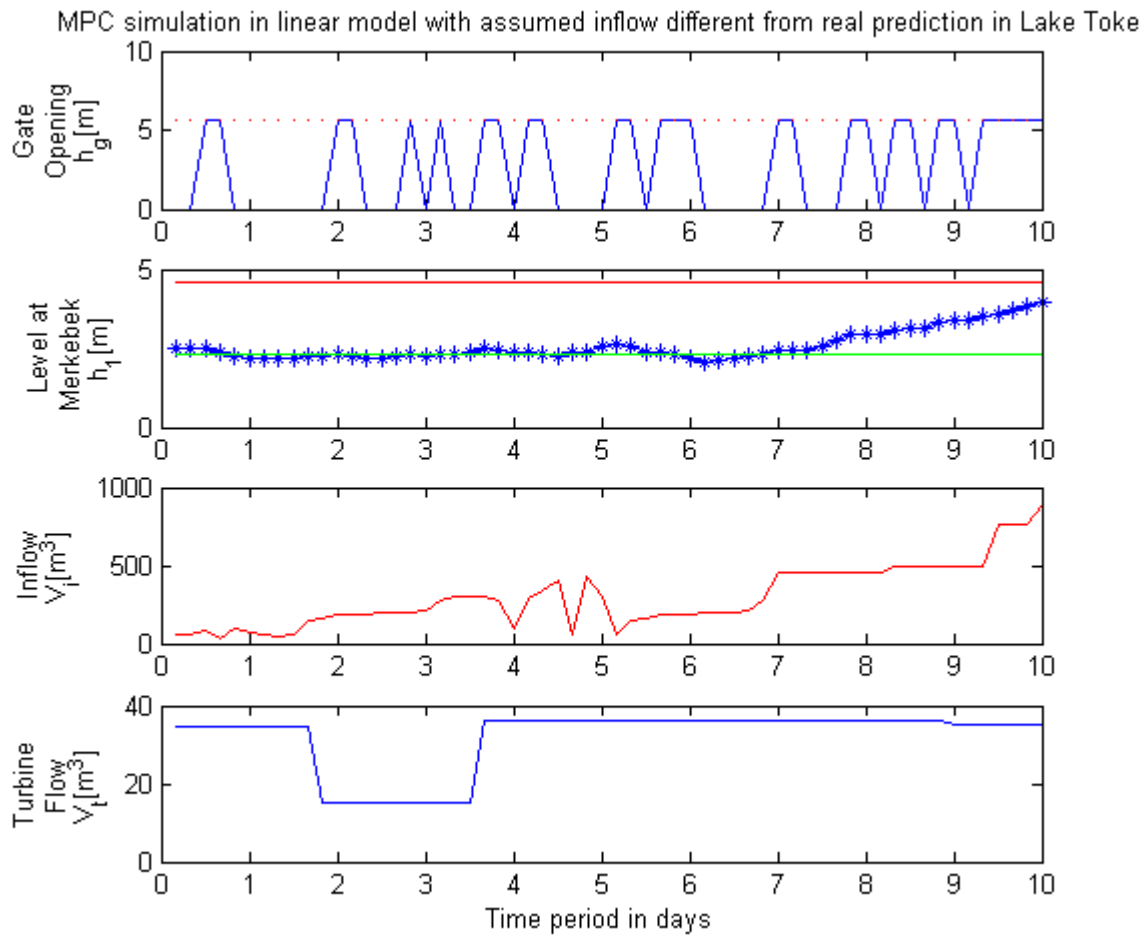


Figure 5-2: MPC simulation with random inflow in linear model

To conclude the stochastic MPC analysis, we can summed up that introducing model error where the future inflow of water was unknown in a MPC application yields almost desired reference level without violating the constraint condition, this indicates that our developed MPC application can handle the unknown inflow predictions.

6 Conclusion and Future work

The summary of the main ideas that we have implemented in this thesis work is highlighted in this section with recommendation for future work. The thesis is about the stochastic flood control using MPC. The main objective of the thesis was to analyze the necessary measurements/information and use them to develop the dynamic model of Lake Toke and to implement MPC along with stochastic MPC based on Markowitz Portfolio Optimization.

The relevant measurements/information were analyzed and described. We found that measurement error in the data provided by Skagerak Energi in height of Dalsfoss dam so we have applied the quay level to simulate and analyze the model during the entire thesis. The prediction model for turbine flow was developed.

The 2-state dynamic model was developed using general mass balance principle and validated through simulation to a steady state time response. The developed model were linearized at around nominal value and discretized using `c2d()` MATLAB function. The sensitivity of parameters that influence the system model was compared and

The linearized model is formulated as a quadratic programming problem. A performance index was defined to achieve optimum output. Both the constraints: hard and soft constraints were defined. Input flow and minimum outflow were assumed to be hard constraints. Output constraints are implemented as soft constraints in order to handle critical conditions (such as: excessive rain fall) as much as possible and violate them when it is not possible.

Both linear and nonlinear Model predictive control (MPC) application using MATLAB was developed and implemented. The results from the simulation seems well controlled fulfilling all the system requirements.

Finally, a simple stochastic MPC analysis was done using real values and randomly assumed inflow by adding random Gaussian noise. MPC application was simulated introducing model error where the future inflow of water was unknown seems to track the desired reference level without violating any system requirement. This indicates that our MPC application can handle unknown inflow predictions or big uncertain inputs can be handled by MPC.

The defined tasks for the thesis is almost fulfilled but the work could have been made more efficient with more assumptions and modifications that could not have been done in this work so far. Some of the recommendation for future work are illustrated below:

- The accurate data received from Skagerak Energi could really improve the system accuracy, so it is important that the correct information from is received to analyse more efficiently and to high accuracy.
- We have assumed only two state model, a third state model can be developed to get more efficient control.
- The Extended Kalman filter algorithm can be used to estimate the system state in quadratic programming and then can be implemented in MPC.

- A different method such as DSR method can be used to obtain a linear matrix model which is more accurate and can be find the good control in MPC.
- The Kalman filter algorithm can be used to estimate the model parameters α, β, K_{12} to achieve more accurate result.
- A sequence of highly uncertain random inflow can be assume to implement Stochastic based MPC solution.
- The optimization problem can be modified using Markowitz Portfolio Optimization theory to develop Stochastic based MPC solution.
- The MPC solution could be implemeted in LabVIEW and can be compared with MATLAB simulation results.

References

- Breckpot, M. 2013. "Flood control of river systems with Model Predictive Control", The river Demer as a case study. *Ph. D Thesis, Faculty of Engineering Science, Arenberg Doctoral School, Heverlee, Belgium.*
- Delgoda, D. K., Saleem, S. K., Halgamuge, M. N. & Malano, H. 2012. "Multiple Model Predictive Flood Control in Regulated River Systems with Uncertain Inflows". *Water Resour Manage, The University of Melbourne, Australia*
- Gøthesen, D.-K., Haile, U. K., Kharel, B. B., Kuznetsov, A., Njoku, I. O. & Rabchuk, K. V. 2013. "Flood Control using MPC of the Kragerø waterways". *MSc Project, Telemark University College, Porsgrunn.*
- Lie, B. 2013a. Group Project 2013: SCE 4016 Predictive Control with Implementation. *Telemark University College, Porsgrunn, Norway.*
- Lie, B. 2013b. Solution, Final test: FM1006 Modelling and Simulation of Dynamic Systems (Modelling Part). *Telemark University College, Porsgrunn, Norway.*
- Lie, B. 2014. Flood Control using MPC of Kragerø Waterways- Early Version. *Project work for Skagerak Energi.*
- Marling, H. & Emanuelsson, S. 2012. The Markowitz Portfolio Theory. *University of Gothenburg.*
- Thorsen, H. M. 2011. "Control and Optimization of Kragerø waterways". *Masters Thesis, Telemark University College, Porsgrunn.*
- Wikipedia 2014a. Optimal Control.
- Wikipedia 2014b. Stochastic Process.

Appendices

Appendix 1: Task Description



Telemark University College

Faculty of Technology

FMH606 Master's Thesis

Title: Stochastic Flood Control using MPC of the Kragerø Waterways

TUC supervisor: Bernt Lie, prof., Telemark University College

External partner: Skagerak Energi, contact: Ingvar Andreassen

Task description:

The following tasks should be carried out:

1. A functional description should be given of a planned Kragerø Waterways flood control system, and the quality of the information should be ascertained.
2. Necessary measurements/information for managing floods in Lake Toke should be described (level measurements, current and future inflow predictions, current and future turbine production flow, etc.).
3. A dynamic model of the relevant water levels at Lake Toke should be developed. The model should be validated against experimental/historic data, and an assessment of the accuracy of the model should be given.
4. Managing floods should be posed as an MPC problem, and an MPC solution should be developed and tested based on proposed deterministic inflow from a hydrological model and production flow through the turbine.
5. The method of Markowitz Portfolio Optimization should be considered for stochastic MPC based on an ensemble of future disturbances (inflow, production flow). Use of stochastic MPC should be compared to use of deterministic MPC.
6. The work that has been carried out should be documented in a master thesis.

Task background:

Five hydro power stations in the Kragerø Waterways, starting at the Dalsfoss hydro power station, receive their water from Lake Toke in Telemark. The catchment of Lake Toke covers ca. 1156 km²; the surface of the lake itself covers 32 km². The lake holds some 150 million m³ of water, and the annual average flow out of the lake is ca. 24 m³/s; the residence time of the lake is thus ca. 72 days, which is relatively little in a hydro power context. The Dalsfoss hydro power turbines can maximally utilize 36 m³/s; with a higher flow rate than this, the water must be allowed to bypass the turbine, which implies a lost opportunity from a hydro power point of view – it would be advantageous to use the buffer capacity of the lake to smooth out some variations in the flow. With a relatively heavy rainfall of 10 mm/h, this

Address: Kjølnes ring 56, NO-3918 Porsgrunn, Norway. **Phone:** 35 57 50 00. **Fax:** 35 55 75 47.



implies ca. 90 m³/s of water hitting the lake surface and ca. 3400 m³/s of water hitting the catchment. The Dalsfoss hydro power station will be built out to allow for a maximal capacity of 960 m³/s flood bypass. Normally, a flow of 300 m³/s is considered a dramatic flood. The main spring flood starts in April each year; floods are caused by snow melting and rain, and hydrological models are used to describe the complex flow through the catchment and into the lake.

The operation of the hydro power station at Dalsfoss is strictly constrained by maximally and minimally allowed levels – these constraints change during the year. The flow out of Lake Toke is also constrained. In addition, the operation is constrained by economic considerations.

At the moment, the floodgates are operated by a specialist operator who is approaching retirement. It is thus of interest to develop an automatic system for controlling the flood gates. In an initial attempt, it is reasonable to develop a dynamic model including a mass balance for Lake Toke in combination with a hydrological model. The hydrological model will describe the flow into Toke, and will be provided by Skagerak Energi. The model will be used in a Model based Predictive Control setting, and the initial goal is to compute a proposed flood gate opening off-line. This proposed flood gate opening will then be evaluated by a specialist operator; this is necessary in a trial period due to the consequences if anything goes wrong. This approach will lead to manual closed loop. In a later work, it is of interest to close the loop automatically, but this will not be done right away.

Thus, a solution should be based on receiving hydrological predictions from Skagerak Energi together with production plans. This should be used as input to a dynamic model of the lake level, and an MPC controller should be used for on-line computation of flood gate opening, which should then be passed back to Skagerak Energi. A short term plan is that this solution should be tested during the spring flood of 2014, following this MSc theses in the spring of 2014.

References:

- Gøthesen, D.-K., Haile, H.K., Khare, B.B., Kuznetsov, A., Njoku, I.O., Rabchuk, K.V. (2013). “Flood Control using MPC of the Kragerø Waterways”, MSc project, Telemark University College, Porsgrunn.
- Jørgensen, J.B., Capolei, A., Völcker, C. (2013). «Introduction to Economic MPC». Presentation at Nordic Process Control Workshop, Oulu, Finland, August 22-23.
- Thoresen, Hege Marie (2011). “Control and optimisation of “Kragerø-vassdraget””, MSc thesis, Telemark University College, Porsgrunn.

Student category:

The project requires background in simple modelling of dynamic systems, some understanding of model fitting, and optimization based control (MPC = Model based Predictive Control).



Practical arrangements:

The work will be carried out at the Kjølnes campus of Telemark University College. It will be useful with an excursion to see the physical system (Lake Toke and the Kragerø Waterways, including the Dalsfoss hydro power station), as well as some interaction with Skagerak Energi in Porsgrunn.

There will be weekly meetings with the supervisor, either face-to-face or by Skype, with hand-in of partial reports every 3 weeks.

The work on the thesis will start in the period January 13-24 2014, and the thesis is to be handed in by June 4 2014 at 14:00. It is not possible to delay the start-up of the thesis work: the supervisors have no possibility to carry out supervision after June 4.

Signatures:

Student (date and signature): Bhuwan Dhakal

Supervisor (date and signature): Bernt Lie

Appendix 2: Linearization and discretization of the model

The model is of the form

$$\frac{dh_1}{dt} = \frac{1}{(1-\alpha)A(h_1)} [(1-\beta)\dot{V}_i - \dot{V}_{12}]$$

$$\frac{dh_2}{dt} = \frac{1}{\alpha A(h_2)} [\beta\dot{V}_i - \dot{V}_t - \dot{V}_g + \dot{V}_{12}]$$

Considering \dot{V}_{in} , \dot{V}_t and \dot{V}_g as inputs and \dot{V}_{12} as state function.

Model matrix 'A' of continuous time model is,

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial h_1} & \frac{\partial f_1}{\partial h_2} \\ \frac{\partial f_2}{\partial h_1} & \frac{\partial f_2}{\partial h_2} \end{bmatrix}$$

Here,

$$\frac{dh_1}{dt} = \frac{1}{(1-\alpha) \times 28 \times 10^6} \times h_1^{-0.1} [(1-\beta)\dot{V}_{in} - 800(h_1 - h_2)(h_1 - h_2)^{1/2}]$$

$$\text{Or, } \frac{dh_1}{dt} = \frac{(1-\beta)\dot{V}_{in} \times h_1^{-0.1}}{(1-\alpha) \times 28 \times 10^6} - \frac{800}{(1-\alpha) \times 28 \times 10^6} \times h_1^{-0.1} \times (h_1 - h_2)^{3/2}$$

Now,

$$\frac{\partial}{\partial h_1} \left(\frac{dh_1}{dt} \right) = \frac{\partial f_1}{\partial h_1} = \frac{(1-\beta)\dot{V}_{in}}{(1-\alpha) \times 28 \times 10^6} \frac{\partial(h_1^{-0.1})}{\partial h_1} - \frac{800}{(1-\alpha) \times 28 \times 10^6} \left\{ (h_1 - h_2) \times \frac{\partial(h_1^{-0.1})}{\partial h_1} + h_1^{-0.1} \times \frac{\partial(h_1 - h_2)^{3/2}}{\partial(h_1 - h_2)} \right\} \frac{\partial(h_1 - h_2)}{\partial h_1}$$

Using the values $h_1=5.4$ and $h_2=4.75$ and computing the equation we get,

$$\frac{\partial f_1}{\partial h_1} = 0.0227$$

Similarly,

$$\frac{\partial f_1}{\partial h_2} = \frac{(1-\beta)\dot{V}_{in}}{(1-0.05) \times 28 \times 10^6} \frac{\partial(h_1^{-0.1})}{\partial h_2} - \frac{800}{(1-0.05) \times 28 \times 10^6} \left\{ (h_1 - h_2) \times \frac{\partial(h_1^{-0.1})}{\partial h_2} + h_1^{-0.1} \times \frac{\partial(h_1 - h_2)^{3/2}}{\partial(h_1 - h_2)} \right\} \times \frac{\partial(h_1 - h_2)}{\partial h_2}$$

$$\frac{\partial f_1}{\partial h_2} = -0.0279$$

Again,

$$\frac{dh_2}{dt} = \frac{1}{\alpha A(h_2)} (\dot{V}_{12} - \dot{V}_t - \dot{V}_g)$$

$$\frac{dh_2}{dt} = \frac{1}{\alpha \times 28 \times 10^6} h_2^{-0.1} \{ 800 \times (h_1 - h_2)^{3/2} - \dot{V}_t - h_g w \sqrt{2gh_2} \}$$

Or,

$$\frac{\partial f_2}{\partial h_1} = \frac{1}{0.05 \times 28 \times 10^6} \left\{ 800 \times h_2^{-0.1} \times \frac{\partial(h_1 - h_2)^{3/2}}{\partial(h_1 - h_2)} \times \frac{\partial(h_1 - h_2)}{\partial h_1} - 0 - 0 \right\}$$

Substituting $h_1=5.4$, $h_2=4.75$, values we found,

$$\frac{\partial f_2}{\partial h_1} = 0.5424$$

Similarly,

$$\frac{\partial f_2}{\partial h_2} = 0.5424$$

Therefore the A matrix becomes,

$$A = \begin{bmatrix} 0.0279 & -0.0279 \\ -0.5424 & 0.5424 \end{bmatrix}$$

For model matrix B,

As B, input is the function of \dot{V}_{in} , \dot{V}_t and h_g

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \text{ where,}$$

$$B_1 = \begin{bmatrix} \frac{\partial f_1}{\partial h_g} & \frac{\partial f_1}{\partial \dot{V}_{in}} & \frac{\partial f_1}{\partial \dot{V}_t} \end{bmatrix}$$

$$B_2 = \begin{bmatrix} \frac{\partial f_2}{\partial h_g} & \frac{\partial f_2}{\partial \dot{V}_{in}} & \frac{\partial f_2}{\partial \dot{V}_t} \end{bmatrix}$$

$$\frac{\partial f_1}{\partial \dot{V}_{in}} = \frac{\partial(1 - \beta) \times h_1^{-0.1} \dot{V}_{in}}{\partial \dot{V}_{in} [(1 - \alpha) \times 28 \times 10^6]}$$

Putting values we get $\frac{\partial f_1}{\partial \dot{V}_{in}} = 0.0003$

$$\frac{\partial f_1}{\partial h_g} = 0$$

$$\frac{\partial f_1}{\partial \dot{V}_{in}} = 0.0003$$

$$\frac{\partial f_1}{\partial \dot{V}_t} = 0$$

Similarly, for B_2

$$\frac{\partial f_2}{\partial h_g} = \frac{1}{\alpha A(h_2)} \frac{\partial}{\partial h_g} (h_g w \sqrt{2gh_2})$$

$$\frac{\partial f_2}{\partial h_g} = -0.6008$$

$$\frac{\partial f_2}{\partial \dot{V}_{in}} = 0.0001$$

$$\frac{\partial f_2}{\partial \dot{V}_t} = \frac{\partial}{\partial \dot{V}_t} \left(\frac{1}{\alpha A(h_2)} (-\dot{V}_t) \right)$$

$$\frac{\partial f_2}{\partial \dot{V}_t} = -0.0056$$

Which gives the B matrix as, $B = \begin{bmatrix} 0 & 0.003 & 0 \\ -0.6008 & 0.0001 & -0.0056 \end{bmatrix}$

For model matrix C,

Since $y = C\delta h$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

And for D matrix,

$$D=0$$

Hence, the linearized model matrices in continuous form are:

$$A = \begin{bmatrix} 0.0279 & -0.0279 \\ -0.5424 & 0.5424 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0.0003 & 0 \\ -0.6008 & 0.0001 & -0.0056 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore

$$\frac{d(\delta h)}{dt} = A\delta h + B\delta u$$

$$y = C\delta h$$

Appendix 3: MATLAB file for simulating the data

$(x_o, x_q, \dot{V}_t, \dot{V}_g \text{ and } \dot{W}_e)$

```
%Matlab file for simulating the data (xo,xq, (Vt,) ?(Vg ) ?and (We ) ?)
clc
data=xlsread('dalsfos.xlsx');
xo=data(:,1);
xu=data(:,2);
xq=data(:,3);
Vdo=data(:,4);
Wde=data(:,8);
%

find(xo>75);
xo(6968)=(xo(6967)+xo(6969))/2;
xo(15062)=(xo(15061)+xo(15063))/2;
xq(6968)=(xq(6967)+xq(6969))/2;
xq(15062)=(xq(15061)+xq(15063))/2;
find(xu>50);
xu(6968)=(xu(6967)+xu(6969))/2;
xu(15062)=(xu(15061)+xu(15063))/2;
tm = (1:length(xo))';
idx = find(~isnan(xo))';
xo1=xo(idx);
xq1=xq(idx);
Vdo1=Vdo(idx);
Wde1=Wde(idx);
polxq=polyfit(xq1,Vdo1,2);
xq1s=sort(xq1);
%for vdt....
ix=find(Vdo>36);
Vdt=Vdo;
Vdt(ix)=36;
Vdt1=Vdt(idx);
tml=(1:length(xo1))';
% correlating power
dx=xo1-xq1;
z=dx.*Vdt1;
zs=sort(z);
polW=polyfit(z,Wde1,1);
%figure(1)
%plot(z,Wde1,'bo',zs,polyval(polW,zs),'r-',zs,zs*9.81*1e-3,'k-',
'linewidth',1.5)
% calculating the roots for re-parameterizing
ht=xo1-xq1;
Vdg1=Vdo1-Vdt1;
%N=length(Vdg1);
%Wde1=(c1.*(xq1.^2).*ht)+c2.*(xq1.*ht)+c3.*ht+c4.*Vdg1.*ht;ht)
for i=1:length(ht)
    a(i)=1;
end

Y=[(xq1.^2).*ht xq1.*ht ht Vdg1.*ht a'];
%Wde1= Y.* C;
%[c1; c2; c3; c4; c5]=(inv(Y'.*Y)).*Y'.*Wde1
c=(inv(Y'*Y))*(Y'*Wde1);

% calculation of roots
n= length(Vdg1);
```

```
d=zeros(1,4);
for I=1:n
d(1)=c(1);
d(2)=c(2)-(c(1).*x01(I));
d(3)=(c(3)-c(2).*x01(I)+c(4).*Vdg1(I));
d(4)=(Wde1(I)-c(3).*x01(I)-c(4).*Vdg1(I).*x01(I)-c(5));
r=roots(d)
end
```

Appendix 4: MATLAB code for model validation and simulation

Defining model:

```
% Simulating h1, h2, hg_max, vd_in, vd_12 and vd_g;
function dy = model(t,y)
area=@(y) (28*10^6*1.1*y.^(1/10));
%parameters
alfa= 0.05; beta= 0.02;K12=800;
Vt=36; w=11.2; g=9.8;Cd=1;hg_max=5.6;
[m,n]=size(y);
dy= zeros(m,n);
if t<25000
    hg=0.5*hg_max;
else
    hg=hg_max;
end
Vdotg=Cd*w*min(hg,y(2))*sqrt(2*g*max(y(2),0));
Vdot12= K12*(y(1)-y(2))*sqrt(abs(y(1)-y(2)));
if t< 25000
Vdotin=36;
dy(1)= ((1-beta)*Vdotin)-Vdot12)/((1-alfa)*area(y(1)));
dy(2)= (beta*Vdotin+Vdot12-Vt-Vdotg)/(alfa*area(y(2)));
else
Vdotin = 24;
dy(1)= ((1-beta)*Vdotin-Vdot12)/((1-alfa)*area(y(1)));
dy(2)= (Vdot12-Vt-Vdotg+beta*Vdotin)/(alfa*area(y(2)));
end
end
```

Calling model file:

```
clc
clear all
timespan=(0:2000:40000);
[t,y]= ode15s(@model,timespan,[5.4;4.75]);hg_max=5.6;g=9.8;
k12=800; Cd=1; w=11.2;
a=hg_max*ones(length(y),1);
clf
%%%showing hg in fig 1
hg=zeros(size(t));
for i=1:length(t)
if t(i)<25000
hg(i)=0.5*hg_max;
else
hg(i)=hg_max;
end
end
%
figure (1)
plot(t,y(:,1),'g')
hold on
plot(t,y(:,2),'r')
hold on
plot(t,a,'b')
hold on
stairs(t,hg,'k');
grid on
legend('h1','h2','hgmax','hg')
xlabel('time[secs]')
```

```

ylabel('level h1, h2 , hgmax & h_g[meters]')
title('levels h1 and h2 in two compartments,hgmax & h_g ')
vdot12=k12.*(y(:,1)-y(:,2)).*sqrt(abs(y(:,1)-y(:,2)));
vdotg=Cd*w*min(hg,y(:,2)).*sqrt(2*g*max(y(:,2),0));
figure(2)
stairs(t,hg,'r','LineWidth',2);grid on;title('simulating
h_g');xlabel('time[secs]');ylabel('level[m]');
for j=1:length(t)
if t(j)<50000
vdotin(j)=400;
end
if t(j)>50000
vdotin(j)=24;
end
end
figure(3)
plot(t,vdot12,'r')
hold on
plot(t,vdotg,'b')
hold on
plot(t,vdotin,'g')
grid on
legend('Vdot12','Vdotg','Vdotin')
xlabel('Time [s]')
ylabel('Flow [m^3/s]')

```


Appendix 5: MATLAB file for Linearization and discretizationⁱ

```

function [A,B,C,D] = linear_model(h1,h2,hg,Vin,Vt,Ts)
% parameters
alpha = 0.05; % surface area fraction in compartment 2
beta = 0.02; % Inflow fractin to compartment 2
w = 11.2; % width of the gate
g = 9.81; % gavitational acceleration
area=28e6;
k12=800;

Vg =2^(1/2)*hg*w*(g*h2)^(1/2);
V12 =k12*abs(h1 - h2)^(1/2)*(h1 - h2);
Ah1 =1.1*h1^(1/10)*area;
Ah2 =1.1*h2^(1/10)*area;

%function
f1 = (1/((1 - alpha)* Ah1))* (((1 - beta)* Vin)- V12);
f2 = (1/(alpha* Ah2))* (beta* Vin + V12- Vt- Vg);

% a11 = df1/dx1;
a11 =(10*(k12*abs(h1 - h2)^(1/2) + (k12*sign(h1 - h2)*(h1 - h2)))/(2*abs(h1 - h2)^(1/2)))/(11*h1^(1/10)*area*(alpha - 1)) - (Vin*(beta - 1) + k12*abs(h1 - h2)^(1/2)*(h1 - h2))/(11*h1^(11/10)*area*(alpha - 1));
% a12 = df2/dx2;
a12 = -(10*(k12*abs(h1 - h2)^(1/2) + (k12*sign(h1 - h2)*(h1 - h2)))/(2*abs(h1 - h2)^(1/2)))/(11*h1^(1/10)*area*(alpha - 1));
% a21 = df2/dx1;
a21 = (10*(k12*abs(h1 - h2)^(1/2) + (k12*sign(h1 - h2)*(h1 - h2)))/(2*abs(h1 - h2)^(1/2)))/(11*alpha*h2^(1/10)*area);
% a22 = df2/dx2;
a22 = (Vt - Vin*beta - k12*abs(h1 - h2)^(1/2)*(h1 - h2) + 2^(1/2)*hg*w*(g*h2)^(1/2))/(11*alpha*h2^(11/10)*area) - (10*(k12*abs(h1 - h2)^(1/2) + (k12*sign(h1 - h2)*(h1 - h2)))/(2*abs(h1 - h2)^(1/2)) + (2^(1/2)*g*hg*w)/(2*(g*h2)^(1/2)))/(11*alpha*h2^(1/10)*area);
% b11 = df1/dhg;
b11 = 0;
% b12 = df1/dVin;
b12 = (10*(beta - 1))/(11*h1^(1/10)*area*(alpha - 1));
% b13 = df1/dVt;
b13 = 0;
% b21 = df2/dhg;
b21 = -(10*2^(1/2)*w*(g*h2)^(1/2))/(11*alpha*h2^(1/10)*area);
% b22 = df2/dVin;
b22 = (10*beta)/(11*alpha*h2^(1/10)*area);
% b23 = df3/dVt;
b23 = -10/(11*alpha*h2^(1/10)*area);
A = [a11 a12; a21 a22]
Bc = [b11; b21];
Bd = [b12 b13; b22 b23];
B=[Bc Bd]
C=[1 0];
D=[0 0 0];
%Discretization
[A,B,C,D] = c2dm(A,B,C,D,Ts);
end

```

Appendix 6: MATLAB file for MPCⁱⁱ

```
function [U] = MPC (Vin,Vt,LRV,HRV,Ts,P,Q,R,S,N,n,Select,a)
%% Parameters
g=9.81;hg=0.056;
h1=2.5;
h2=2.4;
w=11.2;
%%Calling Linearization file
[A,B,C,D] = linear_model (h1,h2,hg,Vin(1),Vt(1),Ts(end));
Bc=B(:,1);
dx1 = [h1,h2]';
nx = length(dx1);
u0=0;
nu = size(Bc,2);
ny = size(C,1);
%%Defining Constraints
ul = 0;
uu = 5.6;
dul = -inf;
duu = inf;
el= -inf;
eu = inf;
xl = -inf;
xu = inf;
yl=LRV;
yu=HRV;

%%
uLB = ones (N*nu,1) *ul;
uUB = ones (N*nu,1) *uu;
duLB = ones (N*nu,1) *dul;
duUB = ones (N*nu,1) *duu;
eLB = ones (N*ny,1) *el;
eUB = ones (N*ny,1) *eu;
yLB = -inf (N*ny,1);
yUB = inf (N*ny,1);
svLB = -inf (N*ny,1);
svUB = inf (N*ny,1);
vtLB = -inf (N*nu,1);
vtUB = inf (N*nu,1);
xLB = ones (N*nx,1) *xl;
xUB = ones (N*nx,1) *xu;

z1 = [uLB; duLB; eLB; yLB; xLB; svLB; vtLB]';
zu = [uUB; duUB; eUB; yUB; xUB; svUB; vtUB]';
%%
H11 = kron (eye (N),Q);
H22 = kron (eye (N),R);
H33 = kron (eye (N),P);
H44 = zeros (N*ny);
H55 = zeros (N*nx);
H66 = kron (eye (N),S);
H77 = zeros (N*nu);

H = blkdiag (H11, H22, H33, H44, H55, H66, H77);
%
c = zeros (1, (3*nu+3*ny+nx) *N);

%%
Y=zeros (ny);
```

```

U=zeros(nu);
V_g=zeros(nu);
% V_t=zeros(nu);
% V_in=zeros(ny);
Yl=zeros(ny);
YU=zeros(ny);
Re=zeros(ny);
%%
for I = 1:n-N

    %reference level
    r =LRV + a*(yu(I)-yl(I));
    % Present state and measurements
    dx = dx1;
    dy = C*dx1;

    %% Equality Constraints
    Ae1u = zeros(N*ny,N*nu);
    Ae1du = zeros(N*ny,N*nu);
    Ae1e = zeros(N*ny,N*ny);
    Ae1dy = eye(N*ny);
    Ae1dx = -kron(diag(ones(N-abs(-1),1),-1),C);
    Ae1sv = zeros(N*ny,N*ny);
    Ae1vt = zeros(N*ny,N*nu);

    Ae2u = zeros(N*nx,N*nu);
    Ae2du = -kron(eye(N),Bc);
    Ae2e = zeros(N*nx,N*ny);
    Ae2dy = zeros(N*nx,N*ny);
    Ae2dx = eye(N*nx)-kron(diag(ones(N-abs(-1),1),-1),A);
    Ae2sv = zeros(N*nx,N*ny);
    Ae2vt = zeros(N*nx,N*nu);

    Ae3u = zeros(N*ny,N*nu);
    Ae3du = zeros(N*ny,N*nu);
    Ae3e = eye(N*ny);
    Ae3dy = eye(N*ny);
    Ae3dx = zeros(N*ny,N*nx);
    Ae3sv = zeros(N*ny,N*ny);
    Ae3vt = zeros(N*ny,N*nu);

    Ae4u = kron(diag(ones(N-abs(-1),1),-1),eye(nu)) - eye(N*nu);
    Ae4du = eye(N*nu);
    Ae4e = zeros(N*nu,N*ny);
    Ae4dy = zeros(N*nu,N*ny);
    Ae4dx = zeros(N*nu,N*nx);
    Ae4sv = zeros(N*nu,N*ny);
    Ae4vt = zeros(N*nu,N*nu);

    Ae = [Ae1u Ae1du Ae1e Ae1dy Ae1dx Ae1sv Ae1vt;
          Ae2u Ae2du Ae2e Ae2dy Ae2dx Ae2sv Ae2vt;
          Ae3u Ae3du Ae3e Ae3dy Ae3dx Ae3sv Ae3vt;
          Ae4u Ae4du Ae4e Ae4dy Ae4dx Ae4sv Ae4vt];

    be1 = [C*dx1; zeros(ny*(N-1),1)];
    be2 = [A*dx1 ; zeros((nx*(N-1)),1)];
    be3 = ones(ny*N,1)*r(I);
    be4 = [-u0; zeros(nu*(N-1),1)];
    be = [be1; be2; be3; be4];

```

```

%% Inequality Constraints
Ailu = zeros(N*ny,N*nu);
Aildu = zeros(N*ny,N*nu);
Aile = zeros(N*ny,N*ny);
Aildy = -eye(N*ny);
Aildx = zeros(N*ny,N*nx);
Ailsv = -eye(N*ny);
Ailvt = zeros(N*ny,N*nu);

Ai2u = zeros(N*ny,N*nu);
Ai2du = zeros(N*ny,N*nu);
Ai2e = zeros(N*ny,N*ny);
Ai2dy = eye(N*ny);
Ai2dx = zeros(N*ny,N*nx);
Ai2sv = -eye(N*ny);
Ai2vt = zeros(N*ny,N*nu);

Ai3u = -eye(N*nu);
Ai3du = zeros(N*nu,N*nu);
Ai3e = zeros(N*nu,N*ny);
Ai3dy = zeros(N*nu,N*ny);
Ai3dx = zeros(N*nu,N*nx);
Ai3sv = zeros(N*nu,N*ny);
Ai3vt = zeros(N*ny,N*nu);

Ai4u = zeros(N*ny,N*nu);
Ai4du = zeros(N*ny,N*nu);
Ai4e = zeros(N*ny,N*ny);
Ai4dy = zeros(N*ny,N*ny);
Ai4dx = zeros(N*ny,N*nx);
Ai4sv = -eye(N*ny);
Ai4vt = zeros(N*ny,N*nu);

Ai = [Ailu Aildu Aile Aildy Aildx Ailsv Ailvt;
      Ai2u Ai2du Ai2e Ai2dy Ai2dx Ai2sv Ai2vt;
      Ai3u Ai3du Ai3e Ai3dy Ai3dx Ai3sv Ai3vt;
      Ai4u Ai4du Ai4e Ai4dy Ai4dx Ai4sv Ai4vt];

bi = [-ones(nu*N,1)*yl(I);
      ones(ny*N,1)*yu(I);
      ones(nu*N,1)*(Vt(I)-4)/78.4;
      zeros(N*ny,1)];

%% Use Quadratic Programming

options = optimset('Display', 'off', 'LargeScale', 'on', 'MaxIter', 200,
'Algorithm', 'active-set');
z = quadprog(H,c,Ai,bi,Ae,be,zl,zu,[],options);
% optimal value of u
u0 = z(1);
u = u0;
if Select == 'L'
dx1=A*dx + B*[u;Vin(I);Vt(I)];
else

%% Simulate non linear model with optimal u

mymodel = @(t,h) nonlinear_model(t,h,u,Vin(I),Vt(I));

```

```

[t,y1]=ode15s(mymodel, Ts, dx);
% Steady state values from non linear model
dx1=[y1(end,1);y1(end,2)];
end
% % flow through the gate
Vg = w*u*sqrt(2*g*h2);
%% Store Values for plotting

V_g(:,I)=Vg;
V_t(:,I)=Vt(I);
V_in(:,I)=Vin(I);
U(:,I)=u;
Y(:,I)=dy;
YL(:,I)=y1(I);
YU(:,I)=yu(I);
Re(:,I)=r(I);
end
%% Plotting
close all
figure(1)
index=1:n-N;
time=index*4/24;

subplot(411)
plot(time,U);
ylabel({'Gate';'Opening';'h_g[m]'})
hold on
plot(time,ul,'k');
hold on
plot(time,uu,'r-');
title('MPC simulation in linear model with assumed inflow different from
real prediction in Lake Toke')

subplot(412)
plot(time,Y,'*-','linewidth',1);
hold on
% ylabel('h1[m]')
ylabel({'Level at';'Merkebek ';'h_1[m]'})
plot(time,Re,'g-');
hold on
plot(time,YL,'r-');
hold on
plot(time,YU,'r-');

subplot(413)
plot(time,V_in,'r');
ylabel({'Inflow';'V_i[m^3]'})
subplot(414)
plot(time,V_t);
ylabel({'Turbine';'Flow';'V_t[m^3]'})
xlabel({'Time period in days'})

end

```

Appendix 7: MATLAB file to run MPC

```
%% Specify Discretization Time
Ts = 0:60*60*4; % Sampling Time
%% Prediction Horizon and number of samples
%Prediction Horizon
N=10;
% Number of sample points
n = 70; %% Import the data
[~, ~, raw] =
xlsread('C:\Users\Bhuwan\Desktop\Inputs.xlsx', 'Sheet1', 'A3:D72');
raw(cellfun(@x) ~isempty(x) && isnumeric(x) && isnan(x), raw) = {' '};
% Allocate imported array to column variable names
Vin = data(:,1); % Inflow
Vt = data(:,2); % flow through turbines
LRV = data(:,3); % LRV
HRV = data(:,4); %HRV

%Define Weighting Matrices
P = 10; % error weighting matrix
Q = 10; % input weighting matrix
R = 10; % change of input weighting matrix
S = 1; % weight slack variable
%%parameter for selecting reference signa
a=0.5;
%% Select the model
Select = 'L';
%% Run MPC
[U] = MPC(Vin, Vt, LRV, HRV , Ts, P, Q, R, S, N, n, Select, a);
```

Appendix 8: MATLAB file for Non-linear model

```
function dhdt = nonlinear_model(t,h,hg,v_dot_in,v_dot_t)
dhdt = zeros(2,1);
%%% parameters
alpha = 0.05; % Surface are fraction in compartment 2
beta = 0.02; % inflow fraction to compartment 2
omega = 11.2; % width of the gate
g = 9.81; % acceleration due to gravity
%%%
A_h1 = 2.8e7*1.1* h(1)^0.1;
A_h2 = 2.8e7*1.1* h(2)^0.1;
v_dot_12 = 800* (h(1) - h(2))* sqrt(abs(h(1)- h(2)));
v_dot_g = hg* omega* sqrt(2* g* h(2));
dhdt(1) = (1/((1 - alpha)* A_h1))* ((1 - beta)* v_dot_in)- v_dot_12);
dhdt(2) = (1/(alpha* A_h2))* (beta* v_dot_in + v_dot_12- v_dot_t- v_dot_g);

end
```

ⁱ Based on the MATLAB code developed by GØTHESEN, D.-K., HAILE, U. K., KHAREL, B. B., KUZNETSOV, A., NJOKU, I. O. & RABCHUK, K. V. 2013. "Flood Control using MPC of the Kragerø waterways". *MSc Project, Telemark University College, Porsgrunn*.

ⁱⁱ Based on the MATLAB code developed by *ibid*.

End of thesis

Bhuwan Dhakal (2014). Stochastic Flood Control using MPC of Kragerø Waterways. Master thesis submitted to faculty of technology, Telemark University College(TUC), Porsgrunn, Norway.