Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Changhun Jeong

Control Policies for Thermal Energy Storage Systems

Master's thesis in Chemical Engineering

Supervisor: Sigurd Skogestad | David Pérez Piñeiro

July 2020

NTNU
Norwegian University of
Science and Technology

Changhun Jeong

Control Policies for Thermal Energy Storage Systems

Master's thesis in Chemical Engineering
Supervisor: Sigurd Skogestad | David Pérez Piñeiro
July 2020

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

I dedicated this work to my parents
who have always been supportive of my decisions and challenges.

Summary

The thermal energy storage is the advanced technology which is gaining its popularity constantly in many sectors such including of residential sector due to increasing environmental awareness. It enables to exploit unconventional energy sources such as the waste heat energy or renewable energies. Although energy-saving and maximizing the use of the energy resources are major action to environmental footprint, it cannot be neglected that it is important to satisfy the thermal energy demand most of the time. To tackle the problem of the energy network system, the right control policy must be chosen and implemented. However, due to its stochastic nature on the fluctuating thermal energy demand and supply, and the unpredictable price of the energy, it is not easy to determine how to operate the energy storage system to exploit the supply of the energy.

This thesis aims to find an effective control policy for the operation of the system in the presence of uncertainty or random disturbances. This work considered various control policies such as a simple structured policy, model predictive control(MPC), parametric modified MPC, and scenario-based MPC. To evaluate these control policies, the mathematical model and uncertainty models are formulated. Uncertainty models generate sample paths of disturbances, considering its stochastic nature, based on probability distribution and Markov chain model. The policies are tuned for their best performances and evaluated based on their expected total cost which can be obtained by using stochastic optimization.

The expected total costs of the control policies are compared to the deterministic optimal cost under the assumption of a perfect forecast and the expected total cost of the system with thermal energy storage.

Preface

This project report is completed and delivered for the fulfillment of the requirement on the course *TKP4900 Chemical Process Technology, Master's Thesis* at the Norwegian University of Science and Technology (NTNU). The project is conducted during the Spring semester of 2020 at the Department of Chemical Engineering.

Although the global pandemic broke out and the university was closed for a while, this project could be completed thanks to many supports from the people around me. Firstly, I would like to appreciate my supervisor Professor for the guidance he has been constantly offering me. From the beginning of the project, he worked closely with me and constantly paid attention to my work by asking questions and offering tips. Secondly, I would like to send my earnest gratitude to my co-supervisor PhD candidate David Pérez Piñeiro who has been giving me assistance and support day and night whenever I faced big challenges during the semester. All of the helpful discussions, we had based on our shared interests on the optimal control and thermal energy storage, gave me the great insights and ideas to solve problems and complete this thesis project. Throughout this work, I gained an understanding of stochastic optimization. This project was a great opportunity to improve programming skills using MATLAB. Lastly, I would like to thank everyone in the Process Systems Engineering research group for their willingness to give help when needed. Thanks to all of them, NTNU has become home for me.

Declaration of Compliance

I declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology(NTNU)

Changhun Jeong
trondheim
July 18, 2020

Table of Contents

Summary i

Preface ii

Table of Contents v

List of Tables vii

List of Figures x

Abbreviations xi

1 Introduction 1
- 1.1 Project Background . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 2
- 1.4 Organization . . . 3

2 Background on Stochastic Optimization 5
- 2.1 Introduction . . . 5
- 2.2 A Universal Canonical Model . . . 5
- 2.3 Modeling Disturbance . . . 9
- 2.4 Designing Control Policies . . . 11
  - 2.4.1 Policy Function Approximations . . . 11
  - 2.4.2 Cost Function Approximations . . . 14
  - 2.4.3 Value Function Approximations . . . 14
  - 2.4.4 Direct Lookahead Approximations . . . 15

3 Background on Energy Storage 17
- 3.1 Introduction . . . 17
- 3.2 Thermal Energy Storage . . . 18
  - 3.2.1 Concept of Thermal Storage System . . . 18

3.2.2 Types of Thermal Energy Storage . . . 19
3.3 Control Policy of Energy Storage System . . . 20

4 Case Study: A District Heating Network **25**
4.1 Case Study . . . 25
4.2 Mathematical Model . . . 25
4.2.1 Static Parameters . . . 26
4.2.2 State Variables . . . 27
4.2.3 Decision Variables . . . 27
4.2.4 Constraints . . . 28
4.2.5 Exogenous Information . . . 28
4.2.6 System Model . . . 29
4.2.7 Cost Function . . . 29
4.3 Exogenous Information Processes . . . 30
4.3.1 Heat Supply . . . 30
4.3.2 Heat Demand . . . 30
4.3.3 Electricity Prices . . . 31

5 Designing Control Policies **33**
5.1 Case without Thermal Energy Storage . . . 33
5.2 Simple Policy . . . 34
5.2.1 Policy Tuning . . . 34
5.3 Certainty Equivalence MPC . . . 38
5.4 Parametric Modified MPC . . . 43
5.4.1 Modification of the Maximum Capacity of TES . . . 43
5.4.2 Modification of the Demand Constraint . . . 45
5.4.3 Modification of the Charging and Discharging Rate . . . 47
5.4.4 Overall of the Parametric Modified MPC . . . 49
5.5 Scenario-Based MPC . . . 51
5.6 Optimal Operation of the District Heating System . . . 58
5.7 Comparison of Policies . . . 60

6 Discussion **63**
6.1 Performance of MPC . . . 63
6.2 Issues with Assumptions . . . 63
6.3 Possible Future Work . . . 65

7 Conclusion **67**

Bibliography **69**

A MATLAB code **71**
A.1 Static parameters Initial states . . . 71
A.2 Disturbances . . . 71
A.3 Expected cost of the system without TES . . . 73
A.4 Simple policy . . . 73

A.5 Certainty equivalence MPC 75
A.6 Parametric modified MPC 78
A.7 Scenario-based MPC 87
A.8 The deterministic optimal solution(perfect forecast) 97

List of Tables

2.1 A set of sample realizations of prices(p_t) and the changes in prices ($\hat{p}_t$) . . 8

3.1 Basic MPC algorithm 21

4.1 Values for the static parameters. 27
4.2 Values for the initial state, x_0. 27

5.1 Comparison of the expected costs on all polices 60

List of Figures

2.1 A set of state variables 7
2.2 Four policies identified using two strategies 11

3.1 Scheme of classification of different storage systems according the storage concept 18
3.2 Illustration of MPC principle 21
3.3 Scenario tree representation of uncertainty evolution for multi-stage MPC 22
3.4 Scenario tree representation of the uncertainty evolution for multi-stage MPC with robust horizon 23

4.1 Heat flows in the system. Q_t^{IJ} is the amount of heat transferred from I to J at time t. 26
4.2 Demand model: Sinusoidal 31
4.3 Price model: First-order Markov chain + jump 32

5.1 The diagram of the district heat network without the thermal energy storage device. 33
5.2 Total expected cost as a function of the policy parameters. Each point is the average of 500 simulations corresponding to different uncertainty scenarios. 35
5.3 The storage level changes during the operation with the the simple policy 36
5.4 The heat flows during the operation with the simple policy 37
5.5 The scheme of certainty equivalence MPC 38
5.6 The history and the future prediction trajectories of the heat demand (a) and the spot electricity price (b) 39
5.7 The storage level changes during the heat network operation with MPC . 41
5.8 The heat flows during the heat network operation with MPC 42
5.9 Policy search of CFA on θ_R 44
5.10 The future trajectory of the demand depend on θ_D 45
5.11 Policy search of CFA on θ_D 46
5.12 The storage level changes depend on the parameter θ_D 47

5.13 Policy search of CFA on θ_R 48
5.14 The storage level changes during the heat network operation by CFA with the limited maximum charging and discharging rate 49
5.15 The heat flows during the heat network operation by CFA with the limited maximum charging and discharging rate 50
5.16 The scheme of the scenario tree applied on the case study 51
5.17 The evolution of the electricity price 53
5.18 The storage level changes during the heat network operation with scenario MPC on the electricity price 53
5.19 The heat flows during the heat network operation with scenario MPC on the electricity price 54
5.20 The evolution of the thermal energy demand 55
5.21 The storage level changes during the heat network operation with scenario MPC on demand 56
5.22 The heat flows during the heat network operation with scenario MPC on the demand 57
5.23 The storage level changes during the optimal heat network operation . . . 58
5.24 The heat flows during the optimal heat network operation 59
5.25 The converged costs of the implemented policies 61

6.1 The operation cost of certain simulations 64

Abbreviations

CFA	=	Cost Function Approximations
DLA	=	Direct Lookahead Approximations
LP	=	Linear program
MCMC	=	Markov Chain Monte-Carlo
MIMO	=	Multiple Inputs and multiple output
MPC	=	Model Predictive Control
PCM	=	Phase Change Material
PFA	=	Policy Function Approximations
TES	=	Thermal Energy Storage
VFA	=	Value Function Approximations

Chapter 1

Introduction

1.1 Project Background

Due to the increasing concerns on climate change and its environmental impact, the maximization of resource utilization has been the focus of process sustainability. One of the most energy-demanding sectors is the residential energy sector. The energy is consumed largely in domestic homes, for purposes such as heating and lighting, to make the quality of human life better. The total amount of domestic energy demand will surely increase in the future because studies show that the world's population is predicted to continuously grow. We now have the question of whether there will be adequate supplies of energy and if energy sources can be sustained, guaranteeing continuous supply for the future. It is unrealistic for mankind to give up technological advancements by cutting off the usage of resources. Therefore, to address the need for continuously improving quality of life through advances in technology and industry while maintaining sustainability, the challenge of energy and resource utilization must be tackled early on. Through optimization and control technologies, usage of energy and resources can be maximized and sustained, allowing the world to use energy without the danger of depletion.

The energy storage system is a critical component in optimizing energy utilization. It allows for flexible operations, making it possible for energy to be stored at one point in time and then utilized at another time in the future when needed. Therefore, the systems enable the utilization of hidden types of unconventional energy, such as waste energy, which is excessively supplied energy compared to the demand, from the industry(Aneke and Wang, 2016).

In common conventional energy systems without storage devices, it is not only impossible to exploit the energy supply, but it also costs more to operate. This is because the peak demand can only be satisfied only by purchasing another type of expensive energy from the market. Moreover, the excess supply of energy from the primary source is wasted instead of being used or stored. These problems are easily solved by using an energy storage device which stores energy for future use. It helps to reduce the mismatch between the energy generation and the energy demand for both short- and long-terms(Kalaiselvam and

Parameshwaran, 2014).

With the increasing attention on the utilization of the storage system, many control policies have been suggested and tested. However, it is difficult to design one control policy that fits all cases due to uncertainties such as the future demand, the energy price, and the energy supply.

1.2 Motivation

As discussed above, energy storage is essential to operate an energy system flexibly with low cost by capitalizing on the primary energy supply and minimizing the purchase of extra energy from the market. To reap the benefits of energy storage, it is mandatory to have a robust operating strategy or control system, also called control policy, that considers the various uncertainties the energy system could be subject to.

A strong control policy for the energy storage system guarantees that the energy demand is satisfied with the minimum operating costs at all times. It also provides the following advantages:

- **High peak capacity:** The demand usually fluctuates depends on many factors such as temperature and sometimes the peak demand happens. When it happens, the stored energy, when generated excessively, can be supplied to satisfied the demand.
- **Exploit energy prices from the market:** The buy-low and sell-high policy can be implemented to take advantage of fluctuating energy prices in the market. Energy is purchased when the price is low and it is then stored for future usage or sold for a higher price in the future.
- **Reject power supply surge:** Although energy tends to be supplied consistently, there may be fluctuations in the energy supply due to unexpected disturbances during operation. With energy storage, the reliability of the energy supply can be improved and a consistent flow of energy is guaranteed.

To create an effective policy in the presence of uncertainty, stochastic optimization can be employed as a tool. Various communities have conducted their research and invented tools on stochastic optimization for their purposes. Often, these communities would create their names and notations. Due to this, it is not easy to perform cross-work. A unified framework of the stochastic programming has then been suggested. The framework introduces how to model the stochastic problem, how to design different types of policies, and how to tune the policiesPowell (2019).

1.3 Objectives

The objectives of this thesis paper are to create and evaluate various types of control policies for the case study based on the unified framework of stochastic programming. Accordingly, among those policies, the most effective one will be chosen. The case study is the district heating system with a thermal storage tank at Heimdal. The case study consists of its uncertainties. Therefore, the following tasks will be implemented in this paper:

1. Building a mathematical model of the thermal energy storage system for simulations
2. Building uncertainty models/sample paths of the thermal energy demand, supply and the spot energy price which can be used for the evaluation of the expected cost of designed control policies.
3. Designing policies based on the unified framework of stochastic programming.
4. Tuning the parameters in control policies
5. Evaluating control policies and comparing them with the optimal operation cost.

1.4 Organization

This master thesis paper consists of six chapters. After chapter one, the introduction, the next following six chapters are briefly described:

- ***Chapter 2.*** **The Background on Stochastic Optimization**

  In this chapter, the background on stochastic optimization is introduced. It will discuss a universal canonical model from the unified framework of stochastic programming. It will also explain how to model uncertainties. Lastly, the types of policies and the policy search method for optimizing the policies are discussed briefly.

- ***Chapter 3.*** **Background on Energy Storage**

  Chapter 3 introduces the concept of energy storage and its benefits. More specifically, the thermal energy storage is explained in terms of its concept and types. Lastly, it discusses the control policies of energy storage system such as buy-low and sell-high, model predictive control, and scenario-based model predictive control.

- ***Chapter 4.*** **Case Study: A District Heating Network**

  This chapter begins by illustrating the case study, the waste solid incineration plant at Heimdal for a district heating network system. Thereafter, the system of the thermal energy storage is mathematically modelled according to the guidelines of the canonical model from the unified framework of stochastic programming. Lastly, the uncertainty models of the thermal energy supply, the thermal energy demand, and the spot electricity are introduced based on the sinusoidal and 1^{st} order Markov chain model with normal distribution function.

- ***Chapter 5.*** **Control Policies for the Heating Network**

  In this chapter, some policies are designed and tuned based on the guidelines of the unified framework of stochastic programming. Other control policies, such as certainty equivalence model predictive control and scenario-based model predictive control are implemented. All of the policies designed and introduced are evaluated and then compared with the *apriori* optimal operation policy.

- ***Chapter 6.*** **Discussion**

  The general discussion about the possible issues from the result and the problems from assumptions are presented. Also, possible future work is briefly discussed.

- ***Chapter 7.*** **Conclusions**

  Lastly, the conclusions regarding the result and the objectives of this thesis paper are presented.

Chapter 2

Background on Stochastic Optimization

2.1 Introduction

Stochastic optimization is defined as a group of methods to minimize or maximize an objective function under uncertainty. Decision-making under uncertainties is a common experience of both humans and organizations. There is a broad range of research communities working on the problem setting of sequential decisions under uncertainty. These communities include: decision analysis; stochastic search; ranking and selection; simulation optimization; online computation; optimal control; robust optimization; optimal stopping; Markov decision processes; approximate/adaptive dynamic programming, and much more.

So much research has been completed and is still continuously performed in these different fields that they have grown into neighbouring areas. Sometimes, new challenges, which are unable to be tackled by the original technique, are solved by tools from other communities. In light of these affairs, the unified framework for stochastic optimization was introduced to facilitate the cross-fertilization of ideas across communities. Therefore, techniques from different fields, such as MPC and dynamic programming, can be applied in solving a single problem by using the framework. Furthermore, the best control policy can be selected based on the specific characteristics of the data. Combining tools from various fields can result in a strong hybrid control policy.(Powell, 2019)

2.2 A Universal Canonical Model

Stochastic optimization is often used as a great tool to find an effective or optimal control policy under uncertainty for the future, instead of finding an optimal control action. However, sequential decision problems in the presence of uncertainty are much more complicated than deterministic optimization problems. Various types of modelling are required,

not only the sequencing of control variables and complex dynamics governing how the system evolves over time but also the flows of information. There are five elements to every sequential decision problem: State variables, control variables, disturbances, system model, and objective function.(Powell, 2020)

State Variables

Denoted x, the state variables are the most important quantity in every sequential decision process. It is the set of every captured variable to model the system. More precisely, state variables are defined in two ways:

- **Policy-dependent version:**
  1. A function of history which is necessary and sufficient with disturbances and a control policy to calculate the contribution/cost function, and the control policy (the decision function).
  2. Any information needed to model the information evolution for the contribution/cost function and control policies.
- **Optimization version:**
  1. A history function which is necessary and sufficient to calculate the contribution/cost function, and the constraints.
  2. Any information used to model the information evolution for the cost/contribution function and the constraints

In the definition of state variables, the term *"necessary and sufficient"* is used to keep state variables as compact as possible.

The state variables are also distinguished between initial state variables and dynamic state variables. Initial state variables, denoted x_0, are specified initial information such as deterministic parameters, initial values of any dynamically varying parameters, and probability distributions about any unknown parameters. Dynamic state variables, denoted x_t for $t > 0$, are changing variables over time. The deterministic parameters are excluded from dynamic state variables.

The state variables can be divided into three categories: The physical state, the information state, the belief/knowledge state. The physical sate, denoted R_t, is the status of the managed physical resources. It can be, for example, the water level in a reservoir. The information state, denoted I_t, includes any other information required to make a control variable and compute the system model or the objective function. Lastly, the belief/knowledge state, denoted B_t, is information which determines a probability distribution for unknown parameters. Therefore, the state variables can be described mathematically as:

$$x_t = (R_t, I_t, B_t)$$

These three states have a relationship as shown on **Figure 2.1**. The physical state is part of the information state. The information state will be all of the rest of the variables known

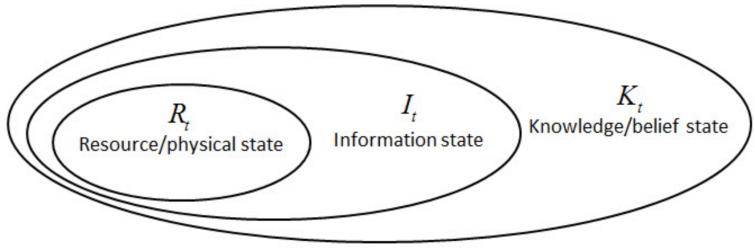


Figure 2.1: A set of state variables

perfectly after the physical states are chosen. The belief states include the probability distribution which determines unknown parameters.

Most problems use the state variables of the system as $x_t = R_t$. However, in case that other unspecified information is required, state variables can become a combination of the physical state, the information state, and belief state.

State variables can also be classified as the pre-decision state variables and post-decision state variables based on a history of information process. If the state variables are defined after new information such as disturbances arrives and before control variables are made as shown on equation (2.1), the state variables are defined as the pre-decision state variables.

$$h_t = (x_0, u_0, W_1, x_1, u_1, W_2, x_2, u_2, ..., u_{t-1}, W_t) \tag{2.1}$$

where u_t and W_t represent control variables and disturbances at time t respectively.

Equation (2.2) shows the sequence of the post-decision state variables which introduces state variables, x_t^u, after control variables are made and before disturbances are realized.

$$h_t = (x_0, u_0, x_0^u, W_1, x_1, u_1, x_1^u, W_2, x_2, u_2, x_2^u, ..., u_{t-1}, x_{t-1}^u, W_t, x_t) \tag{2.2}$$

The pre-decision state variables describe the system naturally, while the post-decision state variables are better suited for dynamic programming in some cases.

Control Variables

In the sequential dynamic programs, control variables u are made over time. A form of control variables can be either vector or scalar.

Control variables are determined by a control policy π. A control policy is a relationship of how state variables are processed in stochastic programming to determine control variables. The relationship of a control policy and control variables can be expressed as equation (2.3).

$$u_t = \mu^{\pi}(x_t) \tag{2.3}$$

Since the purpose of the stochastic optimization is to find an effective policy, it is important to define right control variables for designing a good control policy.

Disturbances

Disturbances are often called as exogenous information. The arrival of disturbances is an important feature in the sequential stochastic dynamic problem because it influences the state variables in the system.

disturbances, denoted W_t, are the unknown future information but first becomes known at time t. For example, the change of the price over time interval, $\hat{p}_t$, can the random disturbance as described on equation (2.4).

$$p_{t+1} = p_t + \hat{p}_t \tag{2.4}$$

The disturbance on equation (2.4) is updated based on 1^{st} order Markov chain model. $\hat{p}_t$ can come from probability distribution. Therefore, sample realization is often used. **Table 2.1** shows an example of 3 sample paths of the electricity price from sample realization based on probability distribution. The sequence of sample realization of the process is referred as a sample path, ω. The set of all of the possible sample realization is denoted Ω, therefore $\omega \in \Omega$.

Sample path	$t=0$	$t=1$		$t=2$		$t=3$	
ω	p_0	$\hat{p}_1$	p_1	$\hat{p}_2$	p_2	$\hat{p}_3$	p_3
1	29.80	2.44	32.24	1.71	33.95	-1.65	32.30
2	29.80	-1.96	27.84	0.47	28.30	1.88	30.18
3	29.80	-1.05	28.75	-0.77	27.98	1.64	29.61

Table 2.1: A set of sample realizations of prices(p_t) and the changes in prices ($\hat{p}_t$)

For disturbances, it is essential to know that $\hat{p}_{t'}(\omega)$ means a random variable at time t for $t < t'$ but $p_t(\omega)$ is a sample realization, not a random variable. It is because the information becomes first known at time t.

Disturbance information processes can be distinguished as either state-independent processes or state/action-dependent information processes based on the level of its complexity. The state-independent processes indicate the processes evolving independently regardless of the state variables. They are, for example, wind, stock price, and demand for services or products. State-independent information processes are very useful. They are often used to test policies simply after they are generated and stored. State/action-dependent information processes evolves depending on state variables x_t or control variables u_t. These processes are, unlike the state-independent information process, unable to create sample paths in advance for testing control policy.

System Model

In engineering problems, the system model is often considered the first thing to do when developing a dynamic problem. The system model shows how the system evolves from one state variables to another based on the control variables and disturbances. Equation (2.5) shows a general form of system model.

$$x_{t+1} = f(x_t, \mu_t^{\pi}(x_t), W_{t+1}) \tag{2.5}$$

Objective Function

Objective function plays an important role as performance metrics or stage cost for evaluating control variables and evaluating a control policy. It is most widely known as a cost function or a contribution function to be minimized or maximized.

In stochastic programming, performance metrics are often denoted as $g(x_t, u_t, W_{t+1})$. It means a cost function is computed based on the state, the control variables, and disturbances.

To find the best effective policy, the universal objective function can be formulated as equation (2.6).

$$\max_{\pi} \mathbb{E}_{x_0} \mathbb{E}_{W_1,\ldots,W_N|x_0} \left\{ \sum_{t=0}^{N} g_t(x_t, \mu_t^{\pi}(x_t), W_t)|x_0 \right\} \tag{2.6}$$

Equation (2.6) covers all range of problems, either state-dependent and state-independent problems, either final reward or cumulative reward.

2.3 Modeling Disturbance

It is not possible to find an effective control policy without modelling disturbance models. Although the importance of disturbance models are often underestimated, a practitioner often faces the challenge to solve a stochastic problem due to disturbance models. There are several types of disturbances. They are briefly explained below (Powell, 2020):

- Observational errors: It often occurs with the unknown state which can not be observed directly or accurately.
- Exogenous uncertainty: It means exogenous information which is realized at time t. It includes, for example, demand, prices and weather.
- Prognostic uncertainty: It is defined as the deviation between the forecast of the information and the realization of the information in the future.
- Inferential uncertainty: It arises when observations are used to estimate another set of parameters due to the lack of understanding of the system. It could be partially from noise in the observation and from the model.
- Experimental variability: It is the difference in the results from experiments run in similar conditions. It can occur in a computer simulation, a laboratory experiment or a field implementation.
- Model uncertainty: It may come from the transition function or the model of the stochastic process due to lack of understanding.
- Transitional uncertainty: It happens due to only exogenous shocks under the assumption of the perfect model of transition function.
- Control/implementation uncertainty: It occurs because of the unintended random perturbation on a controller.

- Communication errors and biases: It means accidental or intended miscommunication about states
- Algorithmic instability: Subtle changes of the input data or parameters can lead to the different path of an algorithm with variability.
- Goal uncertainty: It happens when there are multiple different expected solutions. For example, when a single model produces the results for different stakeholders.
- Political/regulatory uncertainty: Cost and constraints are influenced by the uncertainty about rules and requirements.

The sequential stochastic optimization problem is driven mainly by two information processes: control variables and disturbance information. The information sequence of the stochastic problem looks like:

$$x_0 \rightarrow u_0 = \mu_0^\pi(x_0) \rightarrow W_1 \rightarrow x_1 \rightarrow u_1 = \mu_1^\pi(x_1) \rightarrow W_2 \rightarrow x_3 \rightarrow$$

The system model, such as equation (2.5), can be computed by using a policy $\mu_t^\pi(x_t)$, the initial state, and disturbances known ahead. However, it is often a challenge to simulate the disturbance information sequence.

By introducing sample paths for disturbances, the sequential stochastic problem can be formulated as:

$$\begin{aligned} \underset{\pi}{\text{minimize}} \quad & J^\pi(\omega) = \sum_{t=0}^{N} g_t(x_t(\omega), \mu_t^\pi(x_t(\omega))) \\ \text{subject to} \quad & x_{t+1} = f(x_t(\omega), \mu_t^\pi(x_t(\omega)), W_{t+1}(\omega)) \end{aligned} \tag{2.7}$$

By modelling a sample path for disturbance information, stochastic programming can be easily solved. There are many tools available to model sample paths of disturbances. For example, Markov chain modelling, Monte Carlo sampling, probability distribution, numerical simulation, and observational sampling can be performed. Among these tools, Markov Chain Monte-Carlo sampling and probability distribution are powerful tools and often applied(Powell, 2020). Especially, Markov Chain Monte–Carlo (MCMC) is a very useful method for making sample paths of disturbances. It is a computer-driven sampling method that generates samples randomly based on the distribution information. MCMC is a combination of two properties: Monte–Carlo and Markov chain. Monte-Carlo exploits the properties of a distribution and Markov chain gives an idea that random sample is dependent on the previous one(van Ravenzwaaij et al., 2016).

Also, to apply the tools for modelling distributions it is important to realize the behaviours and the distribution shapes of the random variables. The major classes of distributions includes(Powell, 2020):

- Exponential families of random variables: Most of the well-known distributions fall into this category such as normal distribution.
- Heavy-tailed distributions: As time goes, it does not shows much variance in the distribution. The price is a good example.

- Spikes: The distribution displays infrequent but extreme observation. The peak of electricity price is regarded as a good example.
- Rare events: Rare events such as spikes, however, these are characterized as events instead of values. A good example of the rare events might be a jet engine failure.
- Burst: A sequence of observations over a short period can be characterized as a burst. It can be the behaviour of the process due to the extreme condition such as power outages.
- Regime shifting: A data series shift from one region to another. Data does not always stay around the mean.

2.4 Designing Control Policies

A control policy is defined as a function or a rule which determines a control variables based on the state variables. There are two fundamental strategies which create control policies. One is policy search, and the other is policies based on lookahead approximations. In policy search, equation (2.8) is directly used to search over policies and parameters.

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi} \left\{ \sum_{t=0}^{T} g_t(x_t, \mu_t^{\pi}(x_t), W_{t+1}) | x_0 \right\} \tag{2.8}$$

Policies based on lookahead approximations approximates the states in the future from the impact of a decision made at present.

Both of the strategies can create optimal policies under specific situations. Each strategy can be classified into two meta policies as **Figure 2.2** shows.

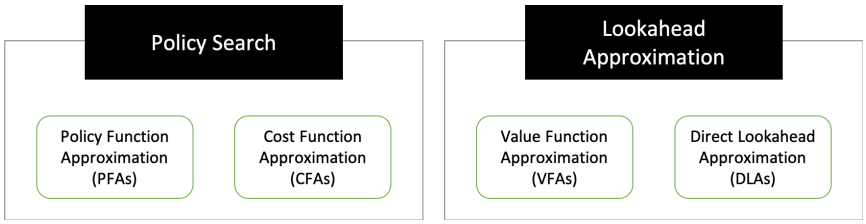


Figure 2.2: Four policies identified using two strategies

These four classes of policies can cover most of the communities introduced in chapter 2.1 and can offer a foundation for designing effective policies. Additionally, tuning allows for the creation of very high-quality results from simple policies(Powell, 2020).

2.4.1 Policy Function Approximations

Policy Function Approximations (PFA) are analytic functions which make feasible control variables by using the states without optimization. It is often applied to low-dimensional

vectors or discrete actions. If there is a very well structured idea of how to create a control variable, effective policy can be designed by PFA. Example cases where policy function approximations can be applied are control policies for the inventory, the water reservoir and the dam control.

Policy function approximations can be written as a form of parametric or locally parametric function. Depend on the form, classes of PFA can be classified. A few of them are explained briefly below:

- **Lookup table policies:**

  A lookup table policy can be simply expressed:

  $$u = \mu^{\pi}(x)$$

  It returns to discrete control action, u, from a particular discrete state variables, x. Therefore, there is one control action parameter stored for each state variable. Lookup tables are widely used in business because it is easy to understand and to define. However, it may be very difficult to optimize in practice.

- **Affine policies**

  Affine policies are written in the linear form with the unknown parameters as following:

  $$\mu^{\pi}(x_t|\theta) = \theta_0 + \theta_1\phi_1(x_t) + \theta_2\phi_2(x_t)$$

  Affine problem is often used in the quadratic control problem. By applying considerable algebra with weighting factors of the quadratic control problem, the optimal policy is simply expressed as:

  $$u_t^*(x_t) = K_t \cdot x_t$$

  However, the problem must be unconstrained to use this approach, well known as the Ricatti equation.

- **Monotone policies**

  If the state variables are multidimensional, the control variables change on each dimension of the state variables in monotone policies. The most common example of monotone policies is the buy-low and high-sell policy as:

  $$\mu^{\pi}(x_t|\theta) = \begin{cases} -1 & If \quad p_t \leq \theta_{min} \\ 0 & If \quad \theta_{min} < p_t < \theta_{max} \\ 1 & If \quad p_t \geq \theta_{max} \end{cases}$$

  where p_t is the energy price at time t. In the example policy, it buys energy when the price goes down below θ_{min} and sells it when the price goes up above θ_{max}. The action increases monotonically.

Given a control policy structures, as described above, with parameters θ, there are multiple ways to optimize the policy by finding the best values of θ for maximizing or minimizing the objective function as:

$$\max_{\theta} J^{\pi}(\theta) = \mathbb{E}^{\pi}\left\{\sum_{t=0}^{T} g(x_t, \mu^{\pi}(x_t|\theta))|x_0\right\} \tag{2.9}$$

where the dynamics evolves according to:

$$x_{t+1} = f^{M}(x_t, u_t, W_{t+1})$$

This process is called policy search. There are two approaches to perform the policy search as following:

- **Batch learning**

  Equation (2.9) is replaced with an average over N samples as:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \bar{J}^{\pi}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T} g(x_t(\omega^n), \mu^{\pi}(x_t(\omega^n)|\theta)) \\ \text{subject to} \quad & x_{t+1}(\omega^n) = f^{M}(x_t(\omega^n), \mu^{\pi}(x_t(\omega^n)|\theta)), W_{t+1}(\omega^n)) \end{aligned} \tag{2.10}$$

  where the constraint is a system model which generates the sequence of the state variables generated following sample path ω^n. This is a classical statistical estimation problem.

- **Adaptive learning**

  Instead of batch learning, also the logic of updating standard stochastic gradient can be used as:

$$\theta^{n+1} = \theta^n + \alpha_n \nabla_{\theta} F^{\pi}(\theta^n, W^{n+1})$$

  The stochastic gradient $F^{\pi}(\theta^n, W^{n+1})$ can be obtained by differentiating equation (2.9) with respect to θ. By applying the chain rule, the stochastic gradient can be computed by:

$$\begin{aligned} \nabla_{\theta} F^{\pi}(\theta, \omega) = \left(\frac{\partial C_0(x_0, u_0}{\partial u_0}\right)\left(\frac{\partial \mu_0^{\pi}(x_0|\theta)}{\partial \theta}\right) + \sum_{t'=1}^{T}\Bigg[\left(\frac{\partial C_{t'}(x_{t'}, \mu_{t'}^{\pi}(x_{t'}))}{\partial x_{t'}}\frac{\partial x_{t'}}{\partial \theta}\right) \\ + \frac{\partial C_{t'}(x_{t'}, u_{t'})}{\partial u_{t'}}\left(\frac{\partial \mu_{t'}^{\pi}(x_{t'}|\theta)}{\partial x_{t'}}\frac{\partial x_{t'}}{\partial \theta} + \frac{\partial \mu_{t'}^{\pi}(x_{t'}|\theta)}{\partial \theta}\right) \end{aligned} \tag{2.11}$$

  where

$$\frac{\partial x_{t'}}{\partial \theta} = \frac{\partial x_{t'}}{\partial x_{t'-1}}\frac{\partial x_{t'-1}}{\partial \theta} + \frac{\partial x_{t'}}{\partial u_{t'-1}}\left[\frac{\partial \mu_{t'-1}^{\pi}(x_{t'-1}|\theta)}{\partial x_{t'-1}}\frac{\partial x_{t'-1}}{\partial \theta} + \frac{\partial \mu_{t'-1}^{\pi}(x_{t'-1})}{\partial \theta}\right] \tag{2.12}$$

The derivative $\partial x_{t'}/\partial\theta$ at time $t = 0$ are computed using equation (2.12) where $\partial x_0/\partial\theta = 0$ and stepping forward in time.

The complexity of the computation of these derivatives is highly problem dependent. The computation of the derivation, using equations (2.11) and (2.12), requires assumptions that the cost function, the policy and the transition functions can be differentiated(Powell, 2020).

2.4.2 Cost Function Approximations

Cost Function Approximations (CFA) are widely used in industry, although it is not getting much attention in academia. In CFA, the approximation of objective function is maximized or minimized, subject to the approximation of constraints. Equation (2.13) shows the general form of CFA.

$$\mu^{CFA}(x_t|\theta) = argmax_(u)\bar{g}_t(x_t, u|\theta) \tag{2.13}$$

where $\bar{g}_t(x_t, u|\theta)$ represents a the parametrically modified cost function, subject to a modified set of constraints.

CFA pulls off the pure exploitation policy by adding an uncertainty bonus as a form of parameters in a pure problem. CFA is a heuristic approach to adjusting a tradeoff between exploration and exploitation.

The objective function is modified to achieve desired behaviour by heuristic approaches such as bonuses and penalties. It helps to get cost-based optimization models which produce robust behaviour in the presence of uncertainty.

The constraints can be modified by introducing parameters as well to produce more robust solutions. It is commonly applied practically in real-world problems. Constraint-modified CFA is given by:

$$U^{Con-CFA}(x_t|\theta) = arg\max_{u_t} C(x_t, u_t)$$

subject to

$$A_t x_t = b(\theta)$$

$$x_t > 0$$

The policy search of CFA can be easily implemented by generating a set of parameters $\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, then using the equation (2.10) which is known as the batch learning method. Also, the stochastic gradient equations (2.11) and (2.12) can be applied for the policy search of CFA(Powell, 2020).

2.4.3 Value Function Approximations

The idea of Value Function Approximations (VFA) started from Bellman's equation. VFA uses an approximation of the value of being in a state resulting from control variables taken now. Equation (2.14) shows the general form of VFA(Powell, 2020).

$$\mu^{VFA}(x_t|\theta) = argmax_(u)(g(x_t, u) + \bar{V}_t^u(x_t, u|\theta)) \tag{2.14}$$

where $\bar{V}_t^u(x_t, u|\theta)$ is the approximation of the future impact.

2.4.4 Direct Lookahead Approximations

Direct Lookahead Approximations (DLA) maximize or minimize based on an approximate model of current and future cost. It is the most brute-force method. When the future approximation can not be computed by VFA, DLA is often used and equation (2.14) is converted into equation (2.15), the decision tree problem, to find optimal control variables.

$$\mu_t^*(x_t) = arg\max_{u_t}\left(g(x_t,u_t) + \mathbb{E}\left\{\max_{\pi}\mathbb{E}\left\{\sum_{t'=t+1}^{T} g(x_{t'},\mu_{t'}^{\pi}(x_{t'}))|x_{t+1}\right\}|x_t,u_t\right\}\right) \quad (2.15)$$

In equation (2.14), the inner term, after $\max_\pi$, represents the process of finding the best action among a class of policies. However, it is not possible to solve. The easiest solution to compute equation (2.15) is to make an approximation. The deterministic approximation of the future is the most popular methods for modelling the solvable lookahead problem. It can be done easily by replacing the end of the horizon from T to $t+H$, which limits the size of the matrix(Powell, 2020).

Chapter 3

Background on Energy Storage

3.1 Introduction

The history of the energy storage system started from the ancient civilization. The oldest form of energy storage is the storage of natural ice or snow from mountains and lakes during the wintertime and they often were used for food preservation, cold drinks and space cooling. However, energy storage is not considered as primitive technology despite its long history. These days, the energy storage system is considered as an advanced energy technology with enormous potential economically and environmentally for now and the future.

Since the development of modern energy storage systems, they have been playing an essential role in energy management, especially in the matter of satisfying energy demand by using the minimum amount of energy resources. The benefit of the energy storage system has become even broader because it also enables the utilization of unpredictable and intermittent renewable energy resources such as wind and solar power. For example, the energy generated from the wind power is stored when the wind is strong, then the energy is used at the peak of energy demand or when the wind does not blow. Not only because of better utilization of the renewable natural resources but also because the energy storage can be employed to capture energy waste from other units, it is undeniable that the energy storage system is an environmentally friendly technology(Kalaiselvam and Parameshwaran, 2014).

There are plenty of other benefits from energy storage systems, aside from being environmentally friendly. The installation of the energy storage system will result in the following advantages(Aneke and Wang, 2016):

- Economic operation from reducing energy costs and consumption.
- Enhanced operation flexibility and reliability.
- Smaller equipment sizes.
- Better efficiency of process equipment.

- Less usage of fossil fuel and reduced greenhouse gas emission accordingly.

Energy storage consists of substances which can hold energy. The type of energy storage is identified based on the form of the stored energy such as mechanical, chemical, biological, magnetic and thermal energy storage(Kalaiselvam and Parameshwaran, 2014).

3.2 Thermal Energy Storage

A thermal energy storage stores heat as sensible heat energy or latent heat energy for later uses. The operation of thermal energy storage helps to stitch the gap between energy supply and energy demand. How to operate the thermal energy storage system, in terms of making decisions about charging, storing, and discharging the thermal energy, is dependent on factors such as temperature, place, demand and supply at present or in future(Kalaiselvam and Parameshwaran, 2014).

3.2.1 Concept of Thermal Storage System

The concept of the thermal energy storage system is distinguished as active or passive ones as shown on **Figure 3.1**.

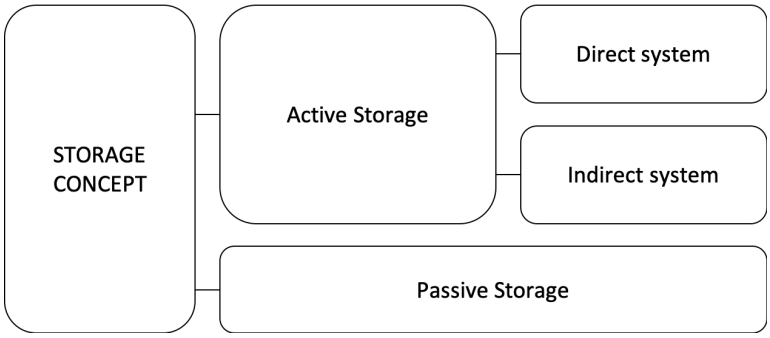


Figure 3.1: Scheme of classification of different storage systems according the storage concept

The forced convection heat transfer into the storage substance is the feature of active systems. In active storage, the storage medium, where the heat is stored, flows through a heat exchanger. Active systems are classified as direct or indirect systems. The difference between direct and indirect system is whether the heat transfer fluid is used as the storage medium as well or not. If the heat transfer fluid is also used as the storage medium, the thermal energy storage system is the direct active system. If the different medium for storing the heat, then the system is defined as an indirect active system.

In passive storage systems, the heat transfer fluid flows around only storage for charging and discharging the system. Therefore, the storage medium does not circulate itself to transfer the heat. The storage medium of a passive storage system is often solid such as concrete and PCM(Gil et al., 2010).

3.2.2 Types of Thermal Energy Storage

Thermal energy storage is categorized into three types: sensible heat storage, latent heat storage, and thermochemical storage.

Sensible Heat Storage

When heat is stored in sensible heat storage, the temperature of the storage medium changes, either increasing or decreasing. There are plenty of available storage mediums with its own advantages and disadvantages for sensible heat storage: water, air, oil, brick, brick, concrete, etc. The material of the storage medium is chosen depending on the heat capacity and the available space for the storage. (Cabeza et al., 2015)

Generally, the material used for sensible heat storage tends to have a high thermal capacity and is cheap and easy to obtain. When choosing the material, its properties must be considered such as density, specific heat, thermal conductivity and diffusivity, vapour pressure, compressibility, and chemical stability.(Fernandez et al., 2010)

Latent Heat Storage

Latent heat storage utilizes the huge energy transfer from the phase transition of a material, latent heat. Solid-liquid phase transition, by melting and solidifying a material, is commonly applied for latent heat storage technology. The benefit of this storage is that it can store and release a large amount of thermal energy at a constant temperature. There are also many materials available for latent heat storage. They are called phase change materials (PCM). However, only a few of PCMs are available in the industrial area due to problems such as phase separation, subcooling, corrosion, long-term stability, and low heat conductivity. When choosing a proper PCM for the system, melting enthalpy and temperature, availability and cost must be considered(Cabeza et al., 2015).

Thermochemical Energy Storage

Many chemical reactions are either exothermic and endothermic. Thermochemical energy storage utilizes these reactions, but only reversible reactions. Although chemical energy conversion has better performance efficiency than sensible and latent heat storage, it is difficult to find the appropriate reversible chemical reaction.

Thermochemical energy storage consists of chemical reactions and the sorption system. There are several chemical reactions studied for thermochemical energy storage such as hydration reaction, carbonation reaction, ammonia decomposition, metal oxidation reactions and sulfur cycles. As sorption system, there are adsorption on solid materials or liquids(Orosz and Dickes, 2017).

Thermal Storage Using Water

It is ideal to use a material with a large heat storage density per volume as the storage medium of sensible heat storage. The material should have high specific heat value and high density. Among all the available materials for sensible storage, water shows a very high heat storage density. Not only that, but water also has many other advantages: cheaper

price, better availability, and less reactivity. Therefore, water is often considered as the most suitable material as storage medium if the operation temperature is between $0°C$ and $100°C$. The material used for the storage tank must be watertight and highly insulating to minimize the heat losses from the storage. It becomes often more critical when operating at higher temperatures because the higher temperature is, the larger the heat loss that occurs. Moreover, it is better to have the charging and discharging rate as high as possible for better operation(Furbo, 2015).

3.3 Control Policy of Energy Storage System

To choose a good control policy for the operation is as important as to select the right energy storage device for the purpose and the unique situation of the system. This section briefly discusses the control policies applied in the energy management system.

Buy-Low and Sell-High

Buy-low and sell-high is one of the simple investment strategies used by many investors or business area(Zervos et al., 2013). This strategy is designed to obtain benefits from energy storage. As the energy price does not stay constant and fluctuates, energy can be purchased from the market then stored into the energy storage system when the price is low. The stored energy can then be used or discharged when the energy price is high. By doing so, the operation cost of energy storage will be reduced, which leads to maximum profit(Berrada and Loudiyi, 2019).

Model Predictive Control

The idea of Model Predictive Control (MPC) was raised by (Richalet et al., 1978) and (Cutler and Ramaker, 1979). MPC has a strong advantage in that it applies the optimization theory in control. It uses the mathematical models of the system to predict the future states and control variables, at each discrete time step, based on the current states. It then applies the first control variables into the system. It enables the system to achieve the control objective. The optimization problem is updated at each time step because of the feedback. Therefore, it is fixed automatically when the system is driven away from the model prediction. This procedure of MPC is implemented repeatedly on each time stepFoss (2013).

The MPC strategy is well explained by Mayne as(Mayne et al., 2000):

> Model predictive control is a form of control in which the current control action is obtained by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.

Table 3.1 shows the algorithm of MPC. A prediction horizon is how far the model predicts the future through optimization. The prediction horizon moves forward accordingly as the optimal solution is computed for each step. This technique is called the moving horizon approach and shown on **Figure 3.2**.

Algortihm: State feedback MPC procedure
for
t = 0,1,2,. . . **do**
Get the current state x_t
Solve a dynamic optimization problem on the prediction horizon from t to $t + N$ with x_t as initial condition.
Apply the first control move u_t from the solution above.
end for

Table 3.1: Basic MPC algorithm

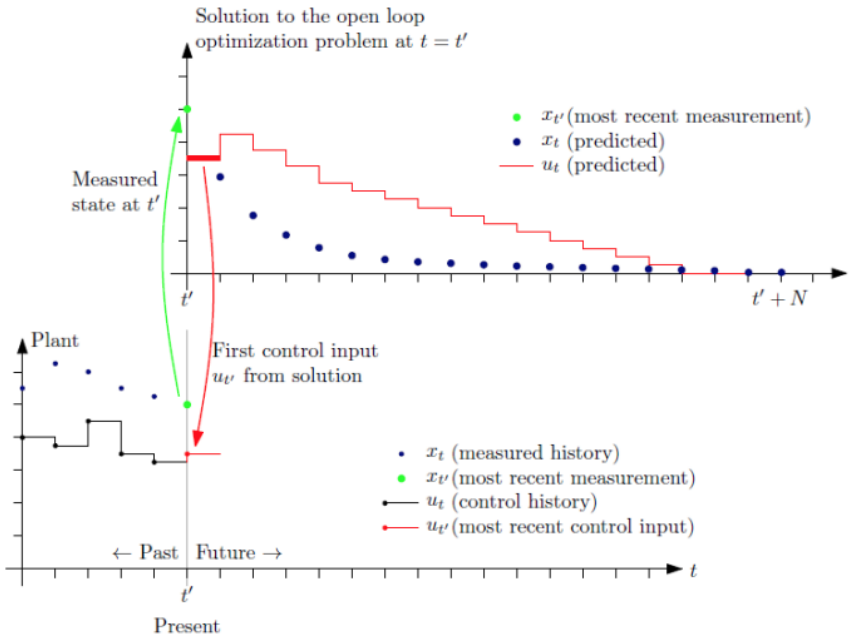


Figure 3.2: Illustration of MPC principle

A sequence of optimal control variables is obtained by computing the optimization problem (3.2), over and over again. The first variable of the sequence is then applied to the system(Foss, 2013).

$$\begin{aligned} \underset{x,\,u}{\text{minimize}} \quad & \phi(x,u) \\ \text{subject to} \quad & F(\dot{x},x,u)=0, \\ & x^{low} \le x \le x^{high}, \\ & u^{low} \le u \le u^{high} \end{aligned} \tag{3.1}$$

Multistage Model Predictive Control

Model predictive control (MPC) handles multi inputs and multi outputs (MIMO) system which is highly coupled. However, standard MPC can not deal properly with uncertainty because the optimisation technique does not consider uncertainty (Birge, 1997)(Shapiro et al., 2009).

After many years of research, multi-stage MPC is presented by (Lucia et al., 2013). The main idea of this approach is to apply a scenario tree which describes the growth of uncertainty with each discrete time step. Therefore, multi-stage MPC is often represented as scenario-based MPC.

The principal assumption is that the uncertainty can be properly modelled by a scenario tree. The scenario tree that is employed in multi-stage MPC is shown on **Figure 3.3**.

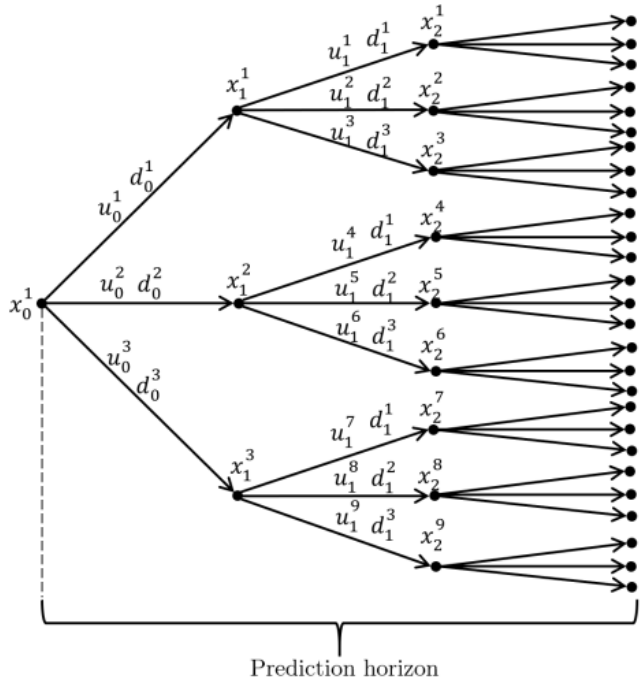


Figure 3.3: Scenario tree representation of uncertainty evolution for multi-stage MPC

It describes the uncertainty evolution by branching out at each node. Every node represents expected states by uncertain events and the control inputs according to the events. The tree describes the future control actions according to the previous uncertainty realisation obtained from measurement information. Therefore, future control input sequences are used to cancel out the effect of future uncertainty. It guarantees that multi-stage MPC has a lower degree of conservativeness compared to other approaches under uncertainty.

However, the problem size increases exponentially as the number of uncertainty realisation and prediction horizon length increases. Therefore, it is important to build a proper

scenario tree for the desired system. To keep the problem size compact, the uncertainty must remain constant after a certain point in time called the robust horizon as described on **Figure 3.4**.

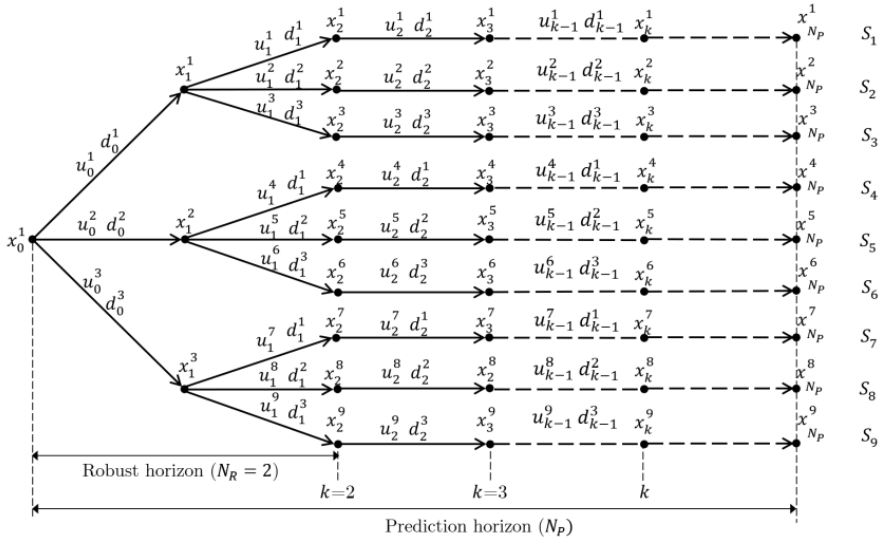


Figure 3.4: Scenario tree representation of the uncertainty evolution for multi-stage MPC with robust horizon

To formulate the scenario-based MPC mathematically, the discrete-time transition function as:

$$x_{t+1,j} = f(x_{t,j}, u_{t,j}, d_{t,j})$$

where, the subscription $(\cdot)_{t,j}$ means the j^{th} scenario at time t.

Firstly, the scenario tree such as **Figure 3.4** is built by the uncertainty realisation. Each scenario is formulated as the path from the root node to the leaf node.

Then, optimization problem for the scenario-based MPC is:

$$\begin{aligned} \underset{x_{t,j},\, u_{t,j}}{\text{minimize}} \quad & \sum_{j=1}^{S} \left[\omega_j \sum_{t=1}^{N} J(x_{t,j}, u_{t,j}) \right] \\ \text{subject to} \quad & f(x_{t,j}, u_{t,j}, d_{t,j}) = x_{t+1,j}, \\ & g(x_{t,j}, u_{t,j}) \leq 0, \\ & \sum_{j=1}^{S} \bar{E}_j u_j = 0 \quad \forall j \in 1, ..., S \end{aligned} \tag{3.2}$$

where ω represents the probability or weight for each scenario, $J(x_{k,j}, u_{k,j})$ is the cost function, $g(x_{k,j}, u_{k,j})$ is the constraints. The last constraint in problem (3.2) is known as the non-anticipativity constraints which make sure that the future control variables cannot anticipate the realisation of the uncertainty. It also guarantees that the states from the same parent node must have the same control input.

$$p = n_u \sum_{j=1}^{S-1} n_{c,(j,j+1)} \tag{3.3}$$

where $n_{c,(j,j+1)}$ is the number of common nodes for two consecutive scenarios j and $j+1$ in the scenario tree. The matrices $\bar{E}_{j,j}$ is given as:

$$\bar{E} = \begin{bmatrix} E_{1,2} & -E_{1,2} & & & \\ & E_{2,3} & -E_{2,3} & & \\ & & \ddots & \ddots & \\ & & & E_{S-1,S} & -E_{S-1,S} \end{bmatrix} \tag{3.4}$$

Chapter 4

Case Study: A District Heating Network

4.1 Case Study

As a case study, we will analyze a district heating network with a thermal storage device. An incineration plant at Heimdal burns municipal solid waste to heat up water that runs in pipes to large parts of the city of Trondheim. Most residential areas in the city are connected to the district heating network. In addition, institutions such as St. Olav's Hospital, the Norwegian University of Science and Technology, Lerkendal Stadium and Nidaros Cathedral also receive district heating.

The waste incineration plant is designed to provide the base-load of the heating demand. Without storage, peak heating would have to be supplied by backup electric boilers. This peaking/backup system is expensive and its operating cost depends on the spot electricity prices, which are highly stochastic. As an alternative, a thermal storage device has been installed. It consists of a hot water tank that can be charged/discharged on-demand. The main idea is to store the excess heat from the waste incineration plant when demand is low and use it at a later time when demand is high. However, we can also benefit from fluctuations in spot electricity prices. For example, we can use electric boilers to charge the hot water tank when electricity prices are low and use the hot water tank to satisfy demand when electricity prices are high. This gives great flexibility, but we need to design a control policy to operate the system.

4.2 Mathematical Model

We consider the problem of controlling the heat flows in the district heating network with a thermal storage device over a finite-time horizon $t = 0, 1, \ldots, T$, where t is the time index and T is the horizon or number of times control is applied. The goal is to find a policy, or control law, that minimizes the total expected cost.

The possible heat flows within the system are illustrated in **Figure 4.1**. The heat production plant may send heat to satisfy the demand or to charge the storage device. Similarly, heat can be purchased from the grid on-demand at the spot electricity price by using the electric boilers to either satisfy the demand or to charge the storage device. The energy stored in the device may be used to satisfy the demand at any time. All of these decisions have to be made in the face of uncertain demand and highly stochastic spot electricity prices.

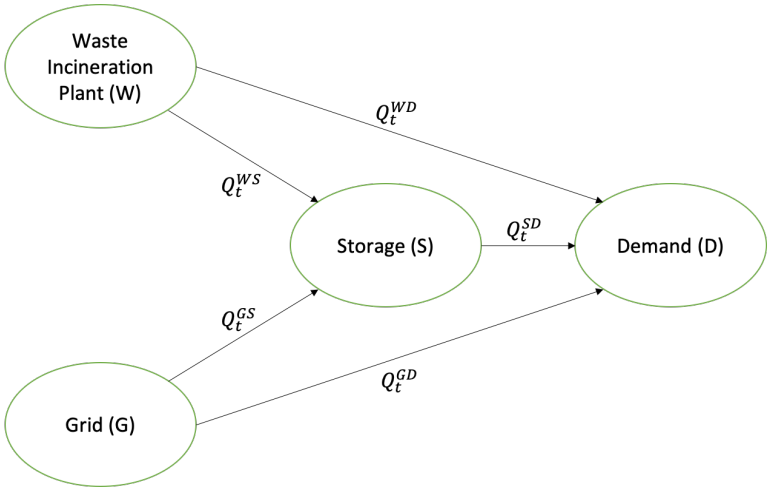


Figure 4.1: Heat flows in the system. Q_t^{IJ} is the amount of heat transferred from I to J at time t.

We make the following assumptions:

- We do not model the heat loss in the storage device. However, we account for losses indirectly by introducing the round-trip efficiency as a model parameter.
- We consider that we are a small player in the electricity market and we can purchase unlimited energy from the grid. Furthermore, our actions do not affect the spot electricity prices.

4.2.1 Static Parameters

The thermal energy storage device can be characterized by the following static parameters:

- S_{max}: Maximum energy capacity of the device in MWh.
- η_c, η_d: Charging and discharging efficiency of the storage device, respectively.
- γ_c, γ_d: Maximum charging and discharging rate of the storage device, respectively, given as MWh per time period.

For this case study, we will consider the following numerical values for the static parameters:

S_{max}	500 [MWh]
η_c	0.9
η_d	0.9
γ_c	50 [MWh per time]
γ_d	50 [MWh per time]

Table 4.1: Values for the static parameters.

4.2.2 State Variables

The state of the system at time t is given by the vector

$$x_t = (S_t, E_t, D_t, p_t), \tag{4.1}$$

which includes all the information that is necessary and sufficient to compute control actions, calculate costs and simulate the process over time:

- S_t: Amount of heat in the storage device at time t in MWh.
- E_t: Amount of heat produced by the waste incineration plant at time t in MWh.
- D_t: Aggregate heat demand at time t in MWh.
- p_t: Spot electricity price at time t in NOK/MWh.

For this case study, we will consider the following values for the initial state:

S_0	0 [MWh]
D_0	200 [MWh per time]
E_0	200 [MWh per time]
p_0	210 [NOK/MWh]

Table 4.2: Values for the initial state, x_0.

4.2.3 Decision Variables

The decision or control variables at time t are given by the vector

$$u_t = (Q_t^{WD}, Q_t^{GD}, Q_t^{SD}, Q_t^{WS}, Q_t^{GS}), \tag{4.2}$$

where Q_t^{IJ} represents the amount of thermal energy transferred from I to J at time t. The superscript W stands for waste incineration plant, D for demand, S for storage and G for grid. Arguably, the main decision we must make is whether to use the energy from storage now or hold it for a later time when we anticipate the demand will exceed the supply or the electricity prices may go up.

4.2.4 Constraints

In our model, the control actions are subject to several constraints. We require that all components of u_t be nonnegative for all t:

$$Q_t^{WD}, Q_t^{GD}, Q_t^{SD}, Q_t^{WS}, Q_t^{GS} \geq 0. \tag{4.3}$$

Furthermore, the total amount of energy stored in the device at time t must not exceed the energy capacity available:

$$Q_t^{WS} + Q_t^{GS} \leq S_{max} - S_t. \tag{4.4}$$

We also make the assumption that all demand at time t must be satisfied:

$$Q_t^{WD} + \eta_d Q_t^{SD} + Q_t^{GD} = D_t. \tag{4.5}$$

Additionally, the amount of energy drawn from the storage device to satisfy the demand must not exceed the amount of energy that is available in the device at time t:

$$Q_t^{SD} \leq S_t. \tag{4.6}$$

The total amount of energy charged to or withdrawn from the device is also constrained by the maximum charging and discharging rates:

$$Q_t^{WS} + Q_t^{GS} \leq \gamma_c, \tag{4.7}$$

$$Q_t^{SD} \leq \gamma_d. \tag{4.8}$$

Finally, flow conservation requires that:

$$Q_t^{WS} + Q_t^{WD} \leq E_t. \tag{4.9}$$

The control u_t is constrained to take values in a given subset $U_t(x_t)$ defined by (4.3)-(4.9), which depends on the current state x_t. We assume that decisions are made with a policy (also called control law) that consists of a sequence of functions $\pi = \{\mu_0, \ldots, \mu_{T-1}\}$, where μ_t maps states x_t into controls $u_t = \mu_t(x_t)$, and satisfies the control constraints, i.e., is such that $\mu_t(x_t) \in U_t(x_t)$.

4.2.5 Exogenous Information

We denote by ω_t the vector of exogenous information (also known as random variables or disturbances), which denote all the variables that arrive enxogenously to the system between t and $t+1$. When modeling specific variables, we use "hats" to indicate exogenous information. In our model,

$$\omega_t = (\hat{E}_t, \hat{D}_t, \hat{p}_t), \tag{4.10}$$

where:

- $\hat{E}_t$: Amount of energy supply by the waste incineration plant between times t and $t+1$.

- $\hat{D}_t$: Amount of aggregated heat demand between times t and $t+1$.
- $\hat{p}_t$: The spot electricity price between times t and $t+1$.

To complete the model, we would have to either provide the probability distribution for the exogenous variables or to specify the source for actual observations. In Section 4.3 we describe the stochastic models we use for the exogenous information processes.

4.2.6 System Model

The mapping from a state x_t to the next state x_{t+1}, given our control action u_t and a random disturbance ω_t is given by the following transition function:

$$x_{t+1} = f_t(x_t, u_t, \omega_t). \tag{4.11}$$

The dynamics in our energy system are described by the following equations:

$$S_{t+1} = S_t + \eta_c(Q_t^{WS} + Q_t^{GS}) - Q_t^{SD}, \tag{4.12}$$

$$E_{t+1} = \hat{E}_{t+1}, \tag{4.13}$$

$$D_{t+1} = \hat{D}_{t+1}, \tag{4.14}$$

$$p_{t+1} = \hat{p}_{t+1}. \tag{4.15}$$

Equation (4.12) is an energy balance in the storage device, while equations (4.13)-(4.15) update the heat supply, the aggregate heat demand and the spot electricity price based on the realization of the random disturbance.

4.2.7 Cost Function

Every time we purchase electricity from the grid at a price p_t (using the electric boilers) to either satisfy the demand or charge the storage device, we incur in the following stage cost:

$$g_t(x_t, u_t, \omega_t) = p_t(Q_t^{GS} + Q_t^{GD}). \tag{4.16}$$

Given an initial state x_0, the total expected cost of a policy $\pi = \{\mu_0, \dots, \mu_{T-1}\}$ over a finite-time horizon $t = 0, 1, \dots, T$ is given by

$$J_\pi(x_0) = \mathbb{E}\left\{ g_T(x_T) + \sum_{t=0}^{T-1} g_t(x_t, \mu_t(x_t), \omega_t) \right\}, \tag{4.17}$$

where the expected value operator $\mathbb{E}\{\cdot\}$ is over all the random variables ω_t and x_t. Therefore, our goal is to find a policy π that minimizes this cost. Mathematically, we can write our problem in a compact form as

$$\min_{\pi \in \Pi} J_\pi(x_0), \tag{4.18}$$

where Π is the set of all admissible policies (i.e., policies that satisfy the constraints). Note that this problem requires optimizing a functional over functions, which is a mathematical problem with roots on calculus of variations. This is the main point of departure from deterministic optimal control problems, where we optimize a cost function over a set of real values (mathematical programming).

If the problem above was deterministic and the exogenous information was known *a priori*, we could solve the control problem using a standard batch linear program (LP):

$$\min_{u_0,\ldots,u_T} \sum_{t=0}^{T} g_t(x_t, u_t), \tag{4.19}$$

such that $u_t \in U_t(x_t)$ for each time t and subject to transition dynamics expressed as a set of constraints linking all time points. This formulation is only valid if we can make exact predictions about the energy produced in the waste incineration plant, demand and spot electricity prices. This is hardly ever the case with physical processes that are intrinsically stochastic. However, this formulation is useful as a performance bound to benchmark policies.

4.3 Exogenous Information Processes

In this section, we present the models of the exogenous information processes. These include the heat supply by the waste incineration plant, the aggregated heat demand, and the spot electricity prices.

4.3.1 Heat Supply

The heat is generated by burning solid waste in the incinerator. The amount of heat supply depends on the amount of waste available and the maximum capacity of the incinerator. Assuming that the waste arrives every morning, the incinerator is most likely to operate at steady-state. Therefore, the heat supply is assumed deterministic:

$$\hat{E}_t = 210 \quad [\text{MWh}] \tag{4.20}$$

The heat energy supply is shown as a red line on **Figure 4.2**.

4.3.2 Heat Demand

The heat demand follows a daily pattern characterized by peaks in the morning (6-7am) and in the evening (7-8pm). We model this predictable variability as a sinusoidal function. To account for stochastic uncertainty, we introduce Gaussian noise in this function. Therefore, the stochastic model is as follows:

$$\hat{D}_t = \min\left\{\max\left\{\mu_0 + A_0 \sin\left(\frac{4\pi t}{T} - \frac{\pi}{2}\right) + \epsilon_D, D_{min}\right\}, D_{max}\right\} \quad [\text{MWh}] \tag{4.21}$$

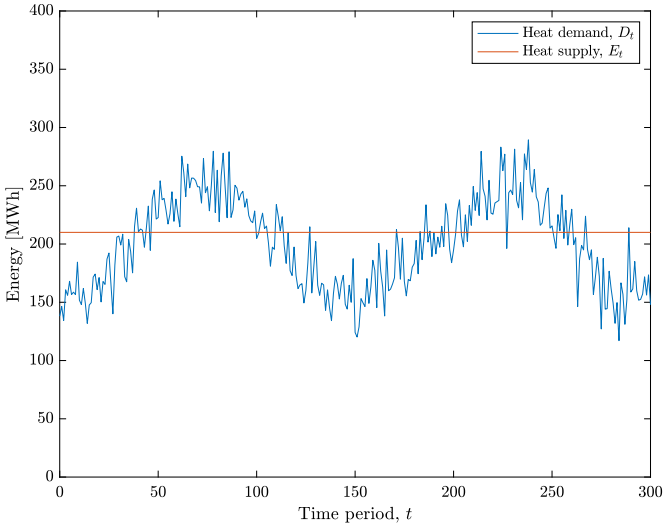


Figure 4.2: Demand model: Sinusoidal

Where D_{min} is the minimum heat demand, D_{max} is the maximum heat demand, and ϵ_D is random noise. The parameters used in (4.21) for the case study are:

$$D_{min} = 100, \quad D_{max} = 300, \quad \mu_0 = 200 \quad A_0 = 50, \quad \epsilon_D \sim N(0, 20)$$

A sample realization of this stochastic model is shown (blue line) in **Figure 4.2**.

4.3.3 Electricity Prices

Real-time electricity prices are spiky and highly stochastic. We model them using a first order Markov chain in combination with a jump diffusion model:

$$\hat{p}_t = \min\left\{\max\left\{p_0 + p_1 + 1_{\{u_t \leq p\}} p_2, p_{min}\right\}, p_{max}\right\} \quad \text{[NOK/MWh]} \tag{4.22}$$

where p_0 is the average price, p_1 captures Gaussian noise and p_2 models sudden jumps in prices, which occur according to:

$$1_{\{u_t \leq p\}} = \begin{cases} 0 & \text{if} \quad u_t \leq p, \\ 1 & \text{if} \quad u_t \geq p. \end{cases}$$

Here, u_t is a continuous uniform random variable between 0 and 1. In our case study, the parameters for (4.22) are set as:

$$p_{min} = 0, \quad p_{max} = 2500, \quad p_0 = 200$$

$$p_1 \sim N(0, 50), \quad p_2 \sim N(0, 500), \quad u_t \sim U(0, 1) \quad p = 0.0031$$

A sample realization of this stochastic model is shown in **Figure 4.3**.

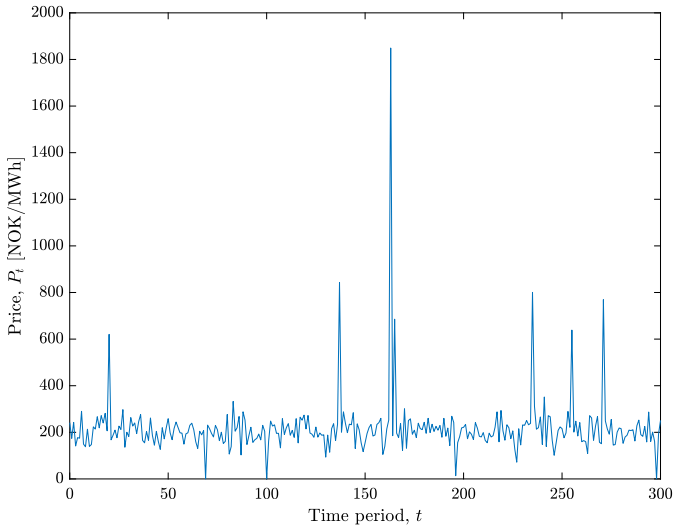


Figure 4.3: Price model: First-order Markov chain + jump

Chapter 5

Designing Control Policies

This chapter is devoted to the design of control policies. We consider four different policies: a simple policy given by a closed-form expression, a certainty equivalence model predictive control, a parametric modified model predictive control, and a scenario-based stochastic model predictive control. Each policy will be presented in detail in the following sections.

Our goal in this chapter is to test these policies on the case study presented in the previous chapter. We will benchmark their relative performance against the posterior optimal solution, i.e., deterministic bound when we assume perfect knowledge of the future. We also examine the total expected cost we incur to when storage is not considered.

5.1 Case without Thermal Energy Storage

We first consider the case without the storage device. In this case, electric boilers have to be used as a backup to supply the peak heating, purchasing electricity at the spot electricity price. A diagram of the system in this case is shown in **Figure 5.1**. We compute the total expected cost for this case by simulating 500 uncertainty scenarios. The result is **7.7016e+05 NOK**.

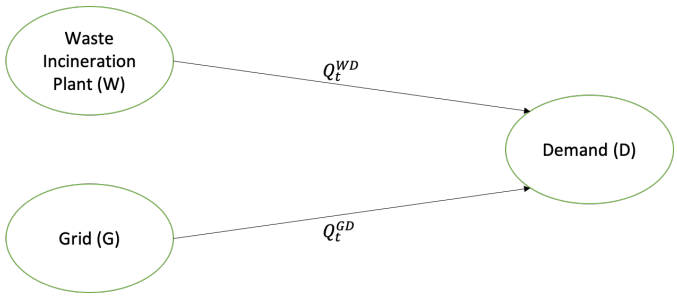


Figure 5.1: The diagram of the district heat network without the thermal energy storage device.

5.2 Simple Policy

In this section, we derive a simple policy given by a closed-form expression based on PFA. It follows the philosophy of "buy low, sell high", meaning that it searches for opportunities to charge the storage device when electricity prices are low and use the storage device when prices are high. The structure of the policy is given below:

$$\mu_t(x_t|\theta) = \begin{cases} Q_t^{WD} = \min\{D_t, E_t\} \\ Q_t^{SD} = \begin{cases} \min\{D_t - Q_t^{WD}, \min\{S_t, \gamma_d\}\} & \text{If } p_t > \theta^{high} \\ 0 & \text{If } p_t \leq \theta^{high} \end{cases} \\ Q_t^{GD} = D_t - Q_t^{WD} - Q_t^{SD} \\ Q_t^{WS} = \min\{E_t - Q_t^{WD}, \min\{S_{max} - S_t, \gamma_c\}\} \\ Q_t^{GS} = \begin{cases} \min\{\gamma_c, S_{max} - S_t - Q_t^{WS}\} & \text{If } p_t < \theta^{low} \\ 0 & \text{If } p_t \geq \theta^{low} \end{cases} \end{cases} \tag{5.1}$$

The policy is a parametric function which depends on the parameters $\theta = (\theta^{high}, \theta^{low})$, which are related to the spot electricity prices. The policy is guided by the following principles:

- Use the heat from the waste incineration plant first to satisfy the demand.
- If supply exceeds demand, store the surplus heat in the storage device, respecting the capacity constraint and the maximum charging rate.
- Only use the electric boilers (grid) to satisfy the demand as the last resource, when the demand cannot be met in any other way.
- Charge the storage device with energy from the grid when electricity prices are low (below some value θ^{low}), respecting the capacity constraint and the maximum charging rate.
- Use the storage device to satisfy the demand when electricity prices are high (above θ^{high}), respecting the capacity constraint and the maximum discharging rate.

The performance of the policy under uncertainty depends on the parameters $\theta = (\theta^{high}, \theta^{low})$, which must be properly tuned.

5.2.1 Policy Tuning

To tune the policy, we perform an exhaustive search over all possible combinations of parameters. Exhaustive search is feasible in this problem because we only have two parameters. For a larger number of parameters, stochastic search algorithms can be used. The total expected cost for each set of parameters is computed by averaging the result of 500 simulations corresponding to different uncertainty scenarios.

We start by discretizing the parameter space defined by $\theta = (\theta^{low}, \theta^{high})$ as follows:

$$\theta^{low} = \{0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300\},$$
$$\theta^{plus} = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}.$$

Then, the parameter θ^{high} is determined by

$$\theta^{high} = \theta^{low} + \theta^{plus}.$$

The result of the exhaustive search is shown in **Figure 5.2**.

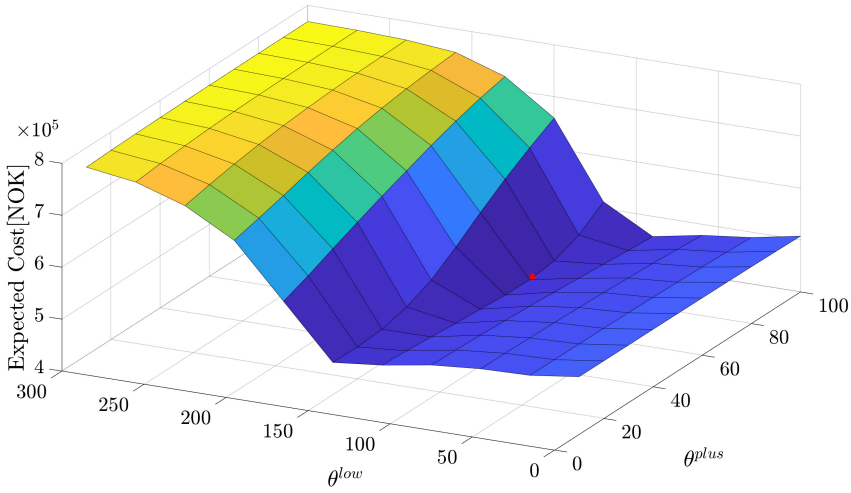


Figure 5.2: Total expected cost as a function of the policy parameters. Each point is the average of 500 simulations corresponding to different uncertainty scenarios.

The set of parameters that minimize the total expected cost is

$$\theta = (\theta^{low}, \theta^{plus}) = (120, 70),$$

which is marked with a red point in **Figure 5.2**. This corresponds with the following set:

$$\theta = (\theta^{low}, \theta^{high}) = (120, 190).$$

The total expected cost of this simple policy for the optimal tuning parameters is **4.5397e+05 NOK**.

During the operation of the simple policy with the optimal parameters, the energy level of the storage device is shown in **Figure 5.3** and the corresponding heat flows in the system are shown in **Figure 5.4**. As the policy is intended, it is observed that the rest of the thermal energy generated from the plan after meeting the demand is stored and used it when the electricity price is higher than θ^{high}. The electricity is purchased rarely only when its price is lower than θ^{low}.

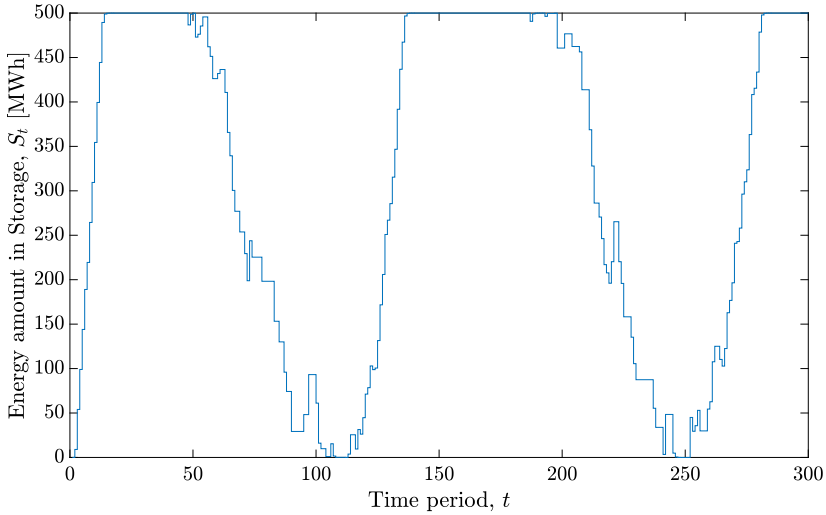


Figure 5.3: The storage level changes during the operation with the the simple policy

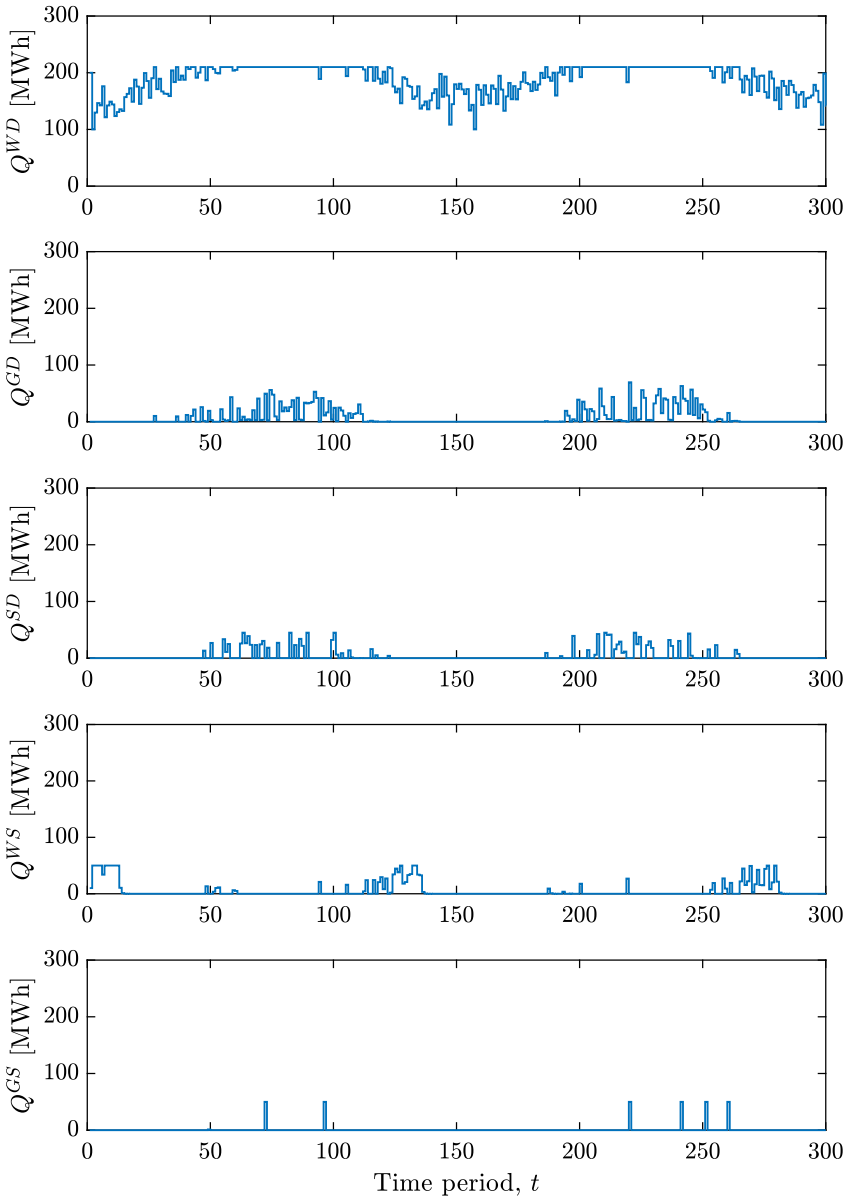


Figure 5.4: The heat flows during the operation with the simple policy

5.3 Certainty Equivalence MPC

Certainty equivalence MPC is applied in the same manner as the described algorithm in **Table 3.1**. However, the realization of disturbances and their forecast processes must be included additionally in the algorithm. The structure of the certainty equivalence MPC follows **Figure 5.5**.

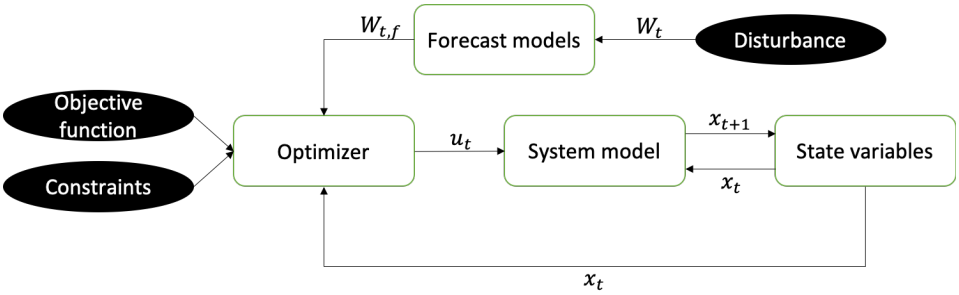


Figure 5.5: The scheme of certainty equivalence MPC

Each time step t, the realization of disturbances W_t occurs. The disturbance W_t is converted into the sequence of disturbance forecast $W_{w,f}$ by forecast models. Combined with the objective function and the constraints, the state variable x_t and the sequence of disturbance forecast $W_{t,f}$ are delivered to the optimizer to determine the control variables u_t. The state variables are updated by the system model with control variables u_t and state variables x_t. This process repeats over and over on every time step.

The forecast models, which generates the sequence of disturbance forecast $W_{t,f}$ from the relaization og disturbances W_t, are formulated as follows:

$$D_{t,f} = \min\left\{\max\left\{\mu_0 + A_0 \sin\left(\frac{4\pi t}{T} - \frac{\pi}{2}\right), D_{min}\right\}, D_{max}\right\} \quad [MWh] \tag{5.2}$$

Where D_{min} and D_{max} are the minimum and maximum of the possible thermal energy demand and are set as 100 and 300, respectively. Model coefficients μ_0 and A_0 are set as 200 and 50 respectively.

$$p_{t+1,f} = \min\left\{\max\left\{p_t + \hat{P}_1, p_{min}\right\}, p_{max}\right\} \tag{5.3}$$

Where p_{min} and p_{max} represent the minimum and maximum of the possible spot electricity prices and are set as 0 and 2500 respectively. $\hat{P}_1$ is the probability distribution which can be expressed as $\hat{P}_1 \sim \nu(0, 50)$

Equation (5.2) is a forecast model of thermal energy demand. It generates the sequence of the demand forecast, $D_{t,f}$. The demand forecast model has the same sinusoidal model as the thermal demand model, equation (4.21), which generates a sample path of the thermal energy demand. Only difference between equations (5.2) and (4.21) is that equation (5.2) does not have stochastic noise ϵ_D. Thus, the demand forecast model, equation (4.21), reflects the daily trend of thermal energy demand only.

Equation (5.3) is a model which forecasts the spot electricity price from the previous one. It generates the sequence of the spot electricity price forecast, $p_{t,f}$. This model is similar to the model of the spot electricity price, equation (4.22). However, equation (5.3) does not include the price jump and spike term. So, equation (5.3) consider a casual fluctuation of the spot electricity price.

The disturbance forecast models, equations (5.2) and (5.3), are projected on **Figure 5.6**. The red dots represents the the future forecast of disturbances and the the blue line before $t = 0$ describes the historical disturbances. **Figure 5.6 (a)** is the history and forecast of demand. **Figure 5.6 (b)** is the history and forecast of the spot electricity price.

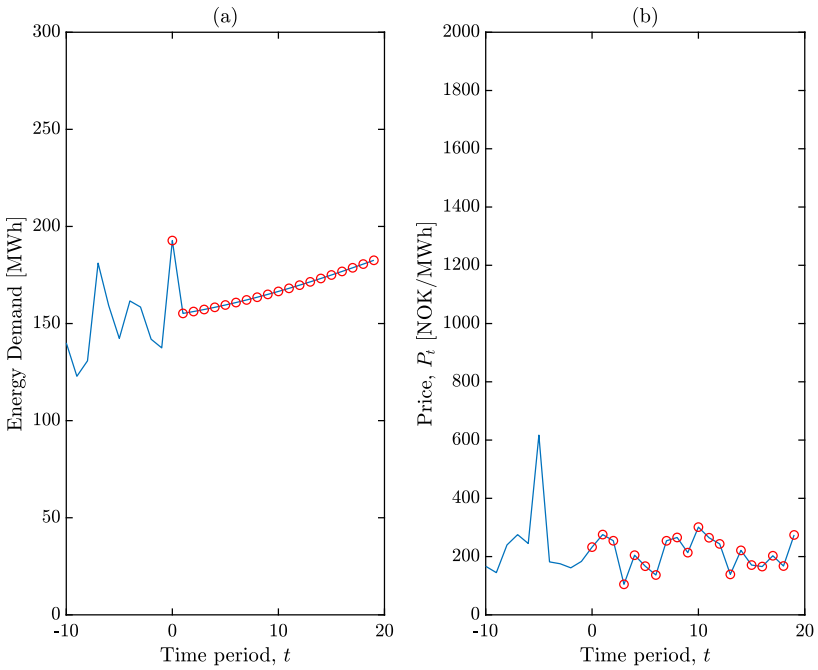


Figure 5.6: The history and the future prediction trajectories of the heat demand (a) and the spot electricity price (b)

With the sequence of the disturbance forecasts $W_{t,f} = \{D_{t,f}, p_{t,f}\}$ and the state variables, the open loop optimization problem is solved in the optimizer as:

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{t=t'}^{t'+H} p_{t,f}(Q_t^{GD} + Q_t^{GS}) \\
\text{subject to} \quad & S_t + \eta_c(Q_t^{WS} + Q_t^{GS}) - Q_t^{SD} = S_{t+1}, \quad t = t', \ldots, t'+H, \\
& Q_t^{WD} + \eta_d Q_t^{SD} + Q_t^{GD} = D_{t,f}, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} + S_t \leq S_{max}, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq S_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{WD} \leq E_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} \leq \gamma_c, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq \gamma_d, \quad t = t', \ldots, t'+H, \\
& x_t, u_t \geq 0, \quad t = t', \ldots, t'+H
\end{aligned}
\tag{5.4}
$$

where H indicates the length of the prediction horizon for the open loop optimization. In the case study, the length of the prediction horizon H is set as 20. As a result of the open loop optimization, the sequence of the control variables $u = \{u_1, , u_H\}$ is obtained. The first control variable of the control sequence u_1 is applied to the system.

The procedure described above is repeated on each time step during the operation time T. As the result, The energy level change of TES during the operation is described on **Figure 5.7** and the heat flows during the operation is shown **Figure 5.8**. The energy level change during the operation by the certainty equivalence MPC seems more unstable and fluctuating compared to the operation by the simple policy. The heat flows from the certainty equivalence MPC shows ironical behaviour that Q^{GD} is used instead of Q^{WD} to satisfy the demand a few times during the operation. The expected cost of the operation by certainty equivalence MPC is **6.8444e+05[NOK]**.

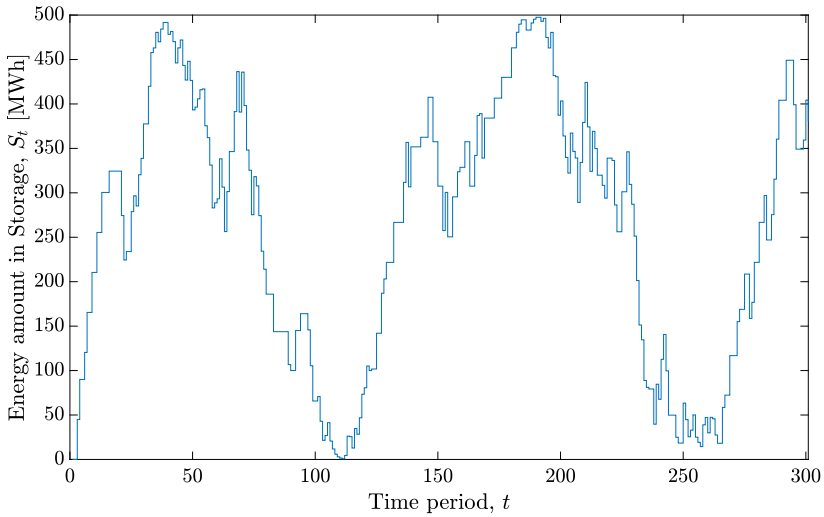


Figure 5.7: The storage level changes during the heat network operation with MPC

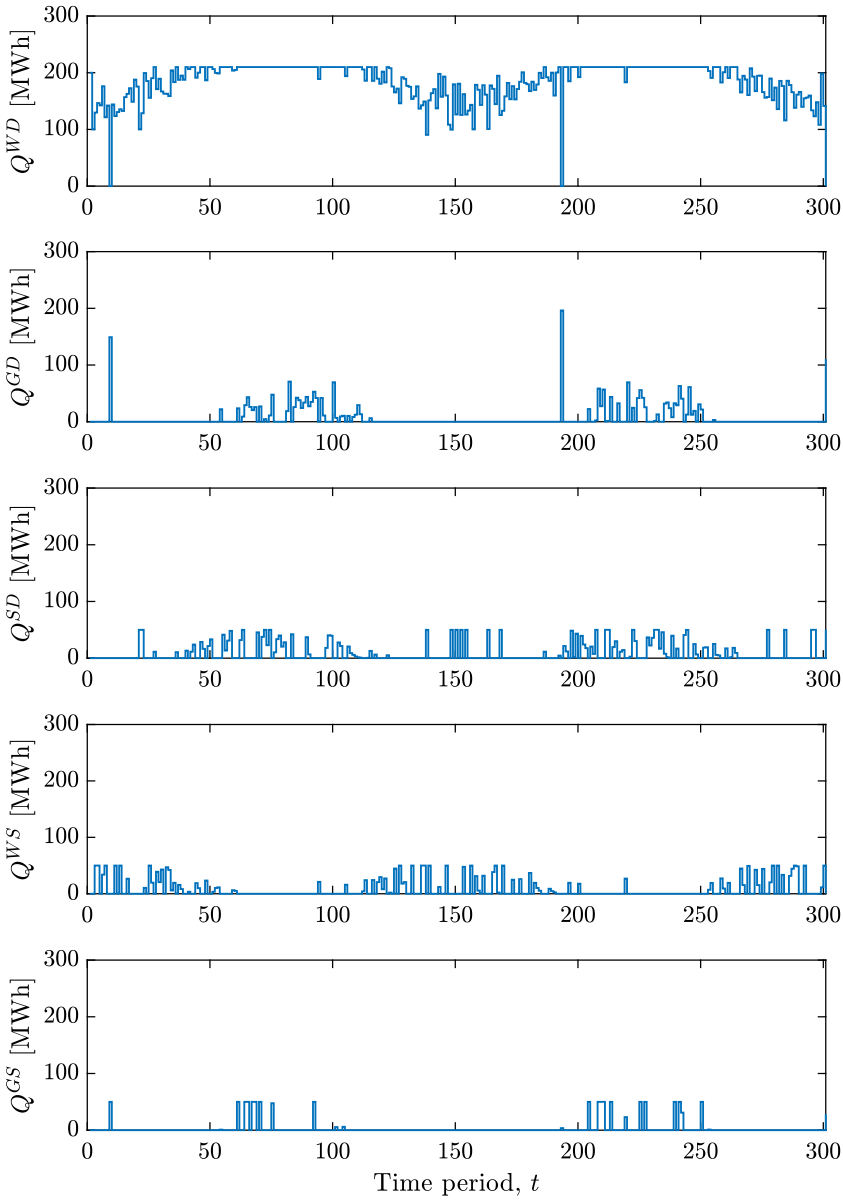


Figure 5.8: The heat flows during the heat network operation with MPC

5.4 Parametric Modified MPC

Certainty equivalence MPC can be converted to Parametric modified MPC by using cost function approximations (CFA). It introduces parameters on either objective function or constraints to get desired behaviour on the system. In this thesis work, three cases of parametric modified MPC are considered such as:

1. modification of the maximum capacity of TES S_{max} by a parameter θ_S
2. modification of the demand forecast $D_{t,f}$ by a parameter θ_D
3. modification of the maximum charging and discharging rates, γ_c and γ_d, by a parameter θ_R

Parametric modified MPC is tuned, by adjusting an introduced parameter, to achieve the minimum total expected cost of control policy. The array of a parameter must be first defined depend on what value of the parameter is desired to be searched for tuning. The total expected cost of control policy, parametric modified MPC, can be computed with each parameter by the batch learning method, equation (2.10). A parameter which has the minimum total expected cost is chosen as the best tuning parameter for the parametric modified MPC.

5.4.1 Modification of the Maximum Capacity of TES

There are two main reasons to modify the maximum capacity of TES. The first reason is due to safety. In reality, it is often prohibited to charge the storage to the maximum level due to the safety reasons. If the surge of the energy supply occurred and there was not enough space in TES, the hot water could overflow. This accident can be dangerous. It is important to know how a total expected cost changes when the maximum capacity of TES is limited.

The second reason is because of potential projects for increasing capacity. It is recommended to know how a total expected cost changes as the capacity of TES increases for a potential project of increasing the capacity. If the information is known, it is easier to decide on the investment for the projects.

The two reasons make it worth to adjust and manipulate the maximum capacity of TES by introducing a parameter θ_S.

Policy Structure

The problem of parametric modified MPC is easily formulated on this case by introducing a parameter θ_S on the constraint of the maximum capacity of TES on the optimization problem of the certainty equivalence MPC, the optimization problem (5.4). The optimization problem of the parametric modified MPC can be written as the optimization problem (5.9). The parameter θ_S is marked in red colour. The algorithms and other variables are as same as the certainty equivalence MPC.

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{t=t'}^{t'+H} p_{t,f}(Q_t^{GD} + Q_t^{GS}) \\
\text{subject to} \quad & S_t + \eta_c(Q_t^{WS} + Q_t^{GS}) - Q_t^{SD} = S_{t+1}, && t = t', \ldots, t'+H, \\
& Q_t^{WD} + \eta_d Q_t^{SD} + Q_t^{GD} = D_{t,f}, && t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} + S_t \leq \theta_S S_{max}, && t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq S_t, && t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{WD} \leq E_t, && t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} \leq \gamma_c, && t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq \gamma_d, && t = t', \ldots, t'+H, \\
& x_t, u_t \geq 0, && t = t', \ldots, t'+H
\end{aligned}
\tag{5.5}
$$

Policy Search

An array of the parameter, which is applied for parametric modified MPC on the maximum capcity of TES, is specified as:

$$\theta_S = \{0.5, 0.6, \ldots, 1.4, 1.5\}$$

Using each value of the parameters, each total expected cost is calculated. The result is described on **Figure 5.9**.

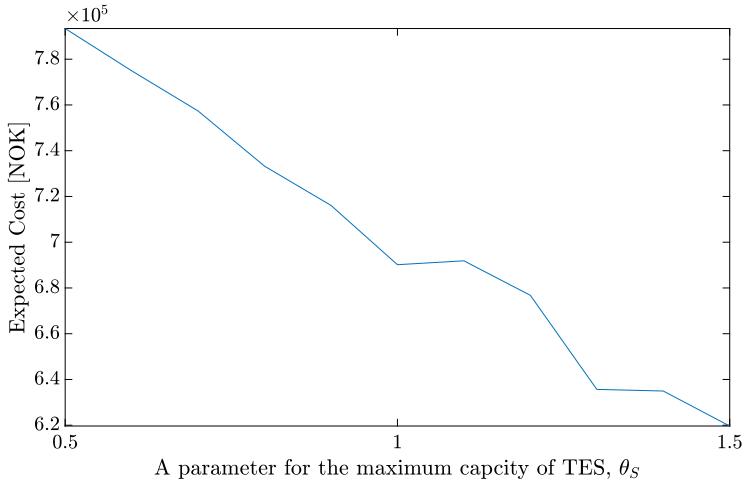


Figure 5.9: Policy search of CFA on θ_R

The total expected cost decreases in general as the parameter θ_S increases. It shows a trend such that the larger TES is, the less total expected cost is obtained. However, when

the parameter θ_S is more than 1, the gradient of the expected cost over the parameter θ_S does not seem constant. When considering to increase the size of TES, more investigation and consideration must be required.

5.4.2 Modification of the Demand Constraint

The demand forecast is computed by equation (5.2) which reflect the hourly trend of thermal energy demand. However, the demand forecast cannot include the actual variation. To operate more conservatively and to satisfy the demand easily, the demand forecast can be manipulated by introducing a parameter θ_D on it. For example, when the parameter θ_D is set as 1.0 and 1.5, the demand forecasts are described on **Figure 5.10**. It shows the history of the demand before time $t = 0$ and the demand forecasts afterwards. The blue colour line with circles is the demand forecast when the parameter θ_D is set as 1.5. The red colour line with crosses is the demand forecast the parameter θ_D is set as 1.0.

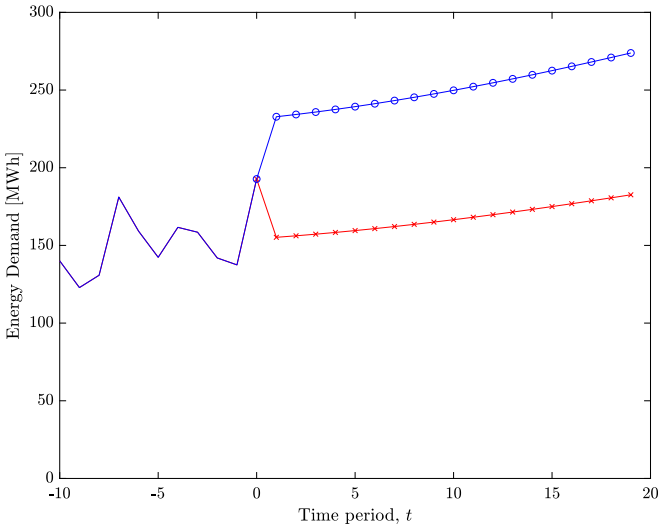


Figure 5.10: The future trajectory of the demand depend on θ_D

As the parameter θ_D increase, the demand forecast is overestimated. It leads to store more thermal energy in TES for future use. It can be done by either buying more electricity when the electricity is relatively cheaper or storing the thermal energy from the plant when the energy is excessively supplied compared to the demand.

Policy Structure

The certainty equivalence MPC, equation (5.4), is converted to the parametric modified MPC for this case by introducing a parameter θ_D on the energy demand satisfaction constraint. The optimization problem of the parametric modified MPC can be written as (5.6).

The parameter θ_D is marked in red color. The other information is treated in the manner with the certainty equivalence MPC.

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{t=t'}^{t'+H} p_{t,f}(Q_t^{GD} + Q_t^{GS}) \\
\text{subject to} \quad & S_t + \eta_c(Q_t^{WS} + Q_t^{GS}) - Q_t^{SD} = S_{t+1}, \quad t = t', \ldots, t'+H, \\
& Q_t^{WD} + \eta_d Q_t^{SD} + Q_t^{GD} = \theta_D D_{t,f}, t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} + S_t \leq S_{max}, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq S_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{WD} \leq E_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} \leq \gamma_c, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq \gamma_d, \quad t = t', \ldots, t'+H, \\
& x_t, u_t \geq 0, \quad t = t', \ldots, t'+H
\end{aligned}
\tag{5.6}
$$

Policy Search

In order to search through the policy, the array of the parameter θ_D is specified as:

$$\theta_D = \{0.5, 0.6, \ldots, 1.4, 1.5\}$$

The total expected costs are calculated by the batch learning method, equation (2.10) with every parameter. The result is plotted on **Figure. 5.11**.

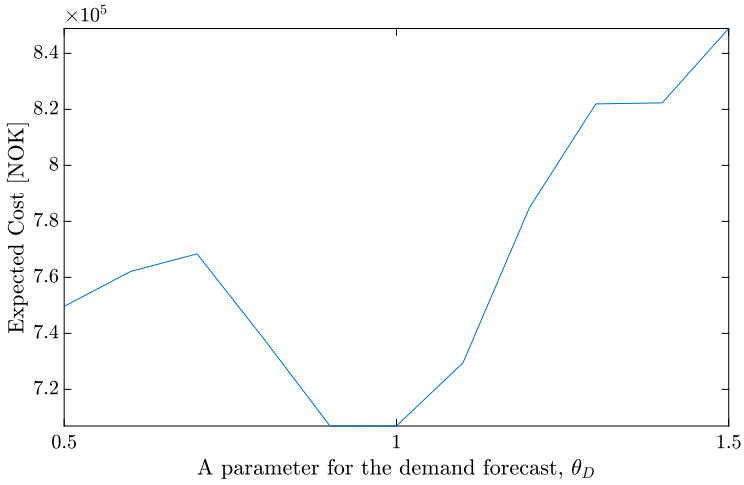


Figure 5.11: Policy search of CFA on θ_D

The minimum total expected cost is obtained when the parameter θ_D is set as 1.0. Although the total expected cost increases due to more consumption of electricity, the

energy level of TES can be controlled as desired by setting the parameter θ_D as described on **Figure. 5.12**. The blue line represents the energy change of TES during the operation for $\theta_D = 0.7$, The red line is for $\theta_D = 1.0$, and the yellow line is for $\theta_D = 1.3$ The higher the parameter θ_D is, the more contingency level of energy is kept in TES.

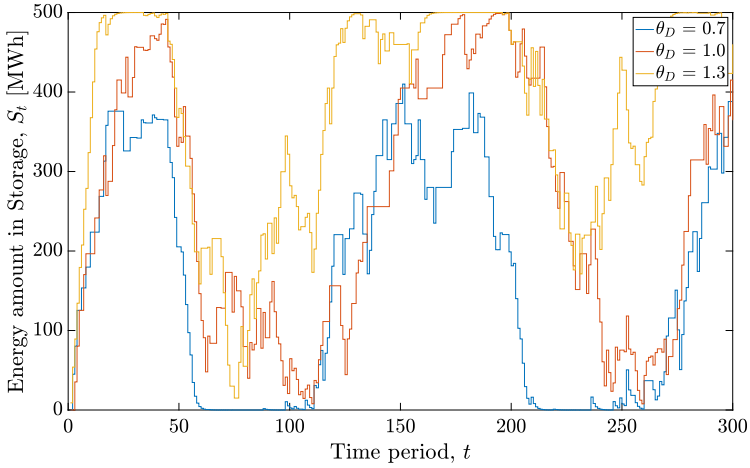


Figure 5.12: The storage level changes depend on the parameter θ_D

5.4.3 Modification of the Charging and Discharging Rate

Initially, the charging rate and discharging rate on the thermal energy storage device is limited by their maximum charging and discharging rates, γ_C and γ_d, respectively. Tuning of these constraints can give more flexibility in the utilization of TES.

Policy Structure

The parametric modified MPC can be generated for this case by putting a parameter θ_R on the charging and discharging rate constraints. The optimization problem is written as problem (5.7). The introduced parameters for the rates θ_R are marked in red colour.

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{t=t'}^{t'+H} p_{t,f}(Q_t^{GD} + Q_t^{GS}) \\
\text{subject to} \quad & S_t + \eta_c(Q_t^{WS} + Q_t^{GS}) - Q_t^{SD} = S_{t+1}, \quad t = t', \ldots, t'+H, \\
& Q_t^{WD} + \eta_d Q_t^{SD} + Q_t^{GD} = D_{t,f}, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} + S_t \leq S_{max}, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq S_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{WD} \leq E_t, \quad t = t', \ldots, t'+H, \\
& Q_t^{WS} + Q_t^{GS} \leq \theta_R \gamma_c, \quad t = t', \ldots, t'+H, \\
& Q_t^{SD} \leq \theta_R \gamma_d, \quad t = t', \ldots, t'+H, \\
& x_t, u_t \geq 0, \quad t = t', \ldots, t'+H
\end{aligned}
\tag{5.7}
$$

Policy Search

In order to perform the policy search on this case, the array of the parameter θ_R is set as:

$$\theta_{rate} = \{0.1, 0.2, \ldots, 1.9, 2.0\}$$

The parameter θ_R shows what percentage of the heat transfer rates is utilized on the thermal energy storage device. For example, if the parameter θ_R is set as 0.5, the maximum charging and discharging rates, γ_c and γ_d, is limited by their 50%.

The expected costs are computed throughout all elements of the parameter array θ_R by using the batch learning method, equation (2.10). The result is shown on **Figure 5.13**.

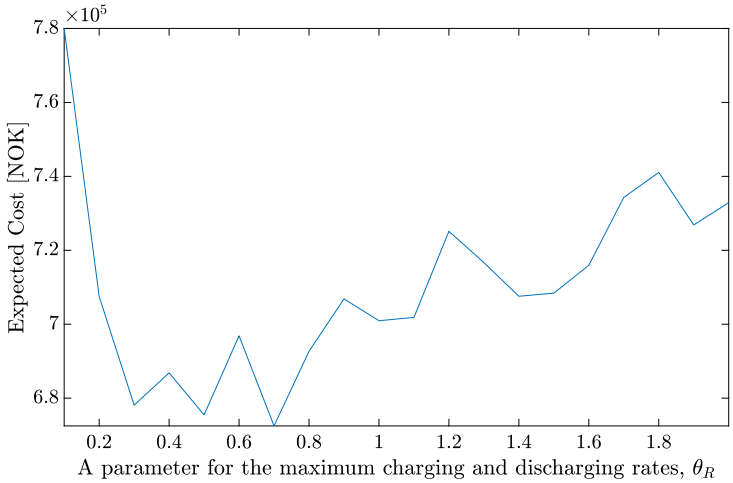


Figure 5.13: Policy search of CFA on θ_R

It describes that the total expected cost of the operation with this policy is the minimum when the parameter θ_R is set as 0.7.

5.4.4 Overall of the Parametric Modified MPC

In this work, three types of parametric modified MPC are performed. The total expected cost did not get lowered by adjusting parameters to manipulate the maximum capacity of TES and the demand forecast. However, it is meaningful to know how the total expected cost changes along with tuning the parameter because the information can play an essential role when making certain types of decisions such as the size of TES when installing or increasing the size, safety bounds for operation, and how to control the energy level of TES.

When performing the policy search over the maximum charging and discharging rates, the minimum total expected cost is found when the parameter θ_R is set to 0.7. The total expected cost of the operation with the control policy is computed as **6.6977e+05[NOK]**. During this operation, the energy level of TES changes as described on **Figure 5.14** and the heat flows are controlled as shown on **Figure 5.15**. Although the energy level change is less stable than the operation by the simple policy, it is less fluctuating than the operation by the certainty equivalence MPC. The heat flows are operated no longer ironically as the operated by the certainty equivalence MPC. Unlike the operation controlled by the certainty equivalence MPC, Only Q^{GD} is not used completely to satisfy the demand during the operation.

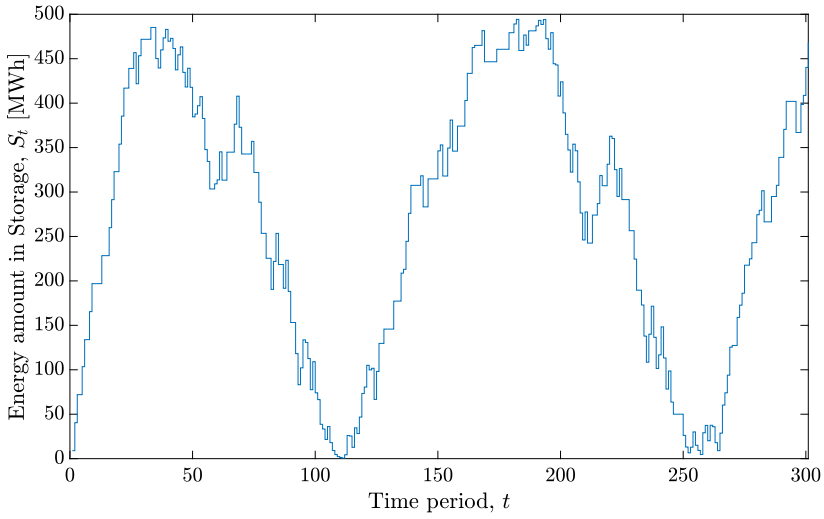


Figure 5.14: The storage level changes during the heat network operation by CFA with the limited maximum charging and discharging rate

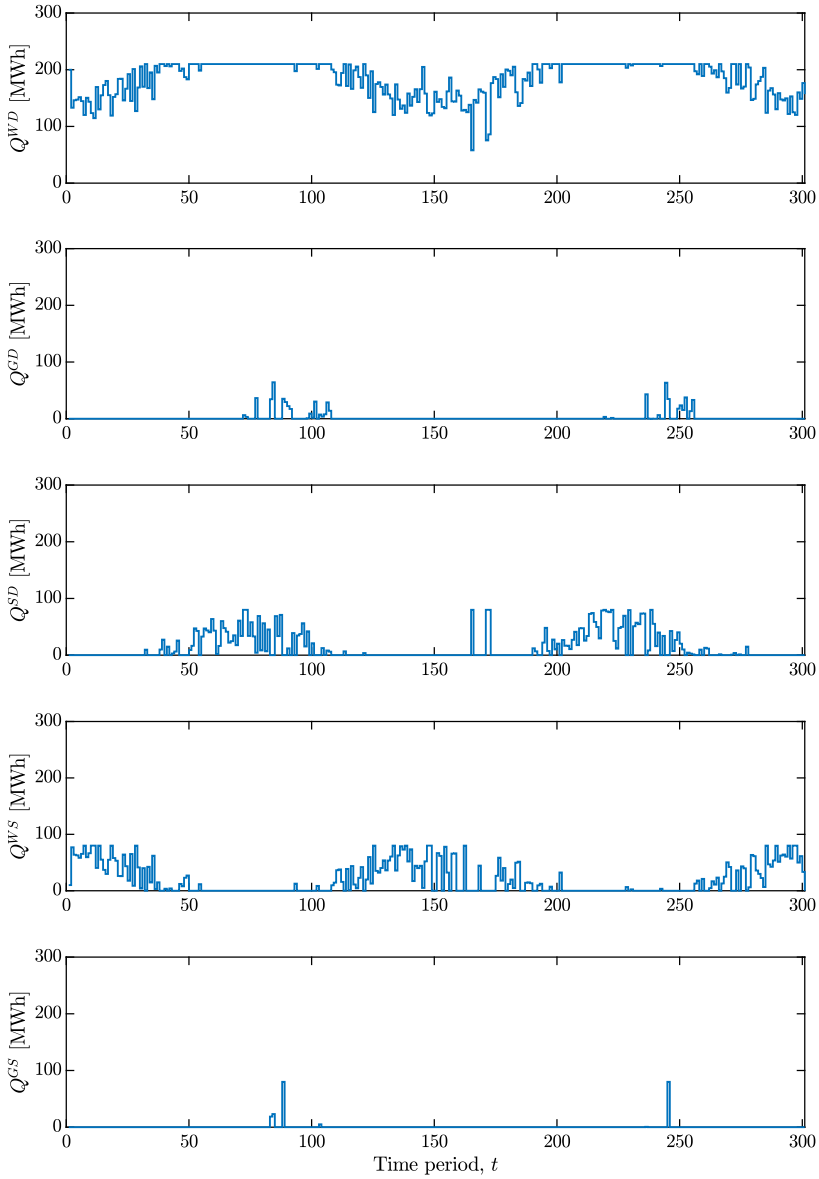


Figure 5.15: The heat flows during the heat network operation by CFA with the limited maximum charging and discharging rate

5.5 Scenario-Based MPC

In this section, two scenario-based MPCs are considered and implemented to counteract the presence of uncertainty such as the future demand or the spot electricity price on the district heating network system. The scenario-based MPC offers conservative operation. Both of the scenario-based MPCs employ a scenario tree described on **Figure 5.16**.

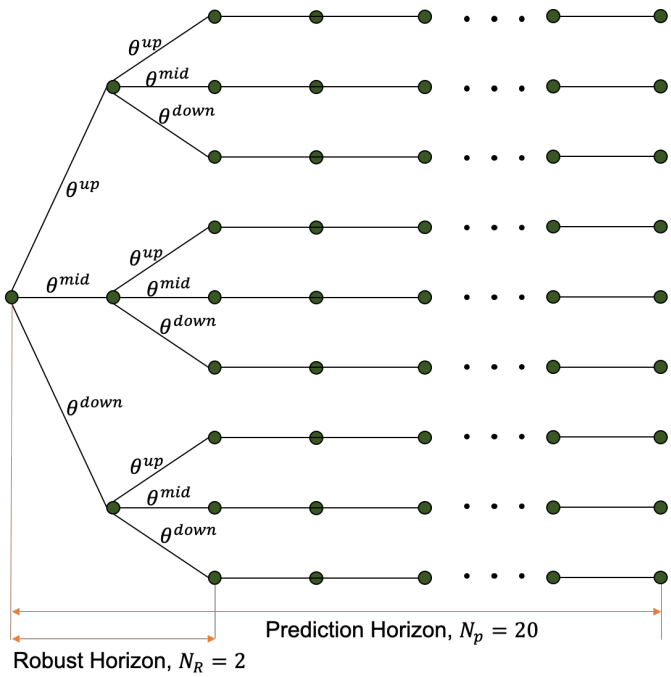


Figure 5.16: The scheme of the scenario tree applied on the case study

The scenario tree is set to evolve based on the parameters $\theta = \{\theta^{up}, \theta^{mid}, \theta^{down}\}$. Each node makes three branches to capture the uncertainty according to the parameters. The robust horizon N_R is set as 2. After the robust horizon, the predictions remain constant until the end of the prediction horizon N_p which is set as 20. Consequently, the nine scenarios are created and evaluated to make a decision for the control action through the following optimization process:

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{j=1}^{N_S} p_j^{\omega} \sum_{t=t'}^{t'+N_p} \hat{p}_{t,j,f}(Q_{t,j}^{GD} + Q_{t,j}^{GS}) \\
\text{subject to} \quad & S_{t,j} + \eta_c(Q_{t,j}^{WS} + Q_{t,j}^{GS}) - Q_{t,j}^{SD} = S_{t+1,j}, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{WD} + \eta_d Q_{t,j}^{SD} + Q_{t,j}^{GD} = \hat{D}_{t,j,f}, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{WS} + Q_{t,j}^{GS} + S_{t,j} \leq S_{max}, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{SD} \leq S_{t,j}, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{WS} + Q_{t,j}^{WD} \leq E_{t,j}, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{WS} + Q_{t,j}^{GS} \leq \gamma_c, \quad t = t', \ldots, t'+H, \\
& Q_{t,j}^{SD} \leq \gamma_d, \quad t = t', \ldots, t'+H, \\
& x_{t,j}, t, j_t \geq 0, \quad t = t', \ldots, t'+H
\end{aligned}
\tag{5.8}
$$

where j represent a sample path such as $j = \{1, 2, \ldots, N_S\}$, p_j^{ω} means the probability or weight factor of a scenario j. In the case, p_j^{ω} is equally set as $1/N_S$ for all j. When the optimization is done one more constraint called the non-anticipativity constraint must be considered in order to make the unique control input. The non-anticipativity constraints must be written in form of equation (3.4) including of the following terms:

$$
\begin{aligned}
{}_{1,1} = u_{1,2} = u_{1,3} = u_{1,4} = u_{1,5} = u_{1,6} = u_{1,7} = u_{1,8} = u_{1,9} \\
u_{2,1} = u_{2,2} = u_{2,3}, \quad u_{2,4} = u_{2,5} = u_{2,6}, \quad u_{2,7} = u_{2,8} = u_{2,9}
\end{aligned}
\tag{5.9}
$$

Scenario-Based MPC of the Electricity Price

In the price scenario MPC, a scenario tree evolves from the electricity price prediction dependent on the parameters $\theta = \{\theta^{up}, \theta^{mid}, \theta^{down}\}$. In this case study, θ^{up} is considered as 30% increase from the prediction, θ^{mid} allows the prediction to stay constant, and θ^{down} means 30% decrease from the prediction. The prediction of the price is obtained from the price model (4.22). The example of the scenario evolution of the spot electricity price is described on **Figure 5.17**. The electricity is realized on time $t = 1$ and the future scenarios of the disturbance are projected.

The possible decisions of the heat flows are diversified according to the evolution of the electricity price. The first input is applied to the system. As a result, the heat flow for the whole operation is described on **Figure 5.19** and the level change of the storage device is shown on **Figure 5.18**. The energy level is tended to be kept higher for the conservative operation than certainty equivalence MPC. The expected cost of the price scenario MPC is **6.9666e+05[NOK]**.

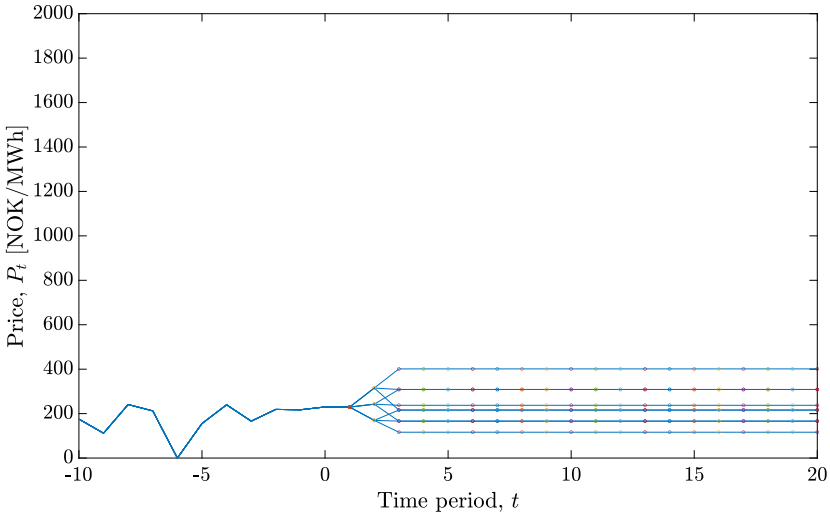


Figure 5.17: The evolution of the electricity price

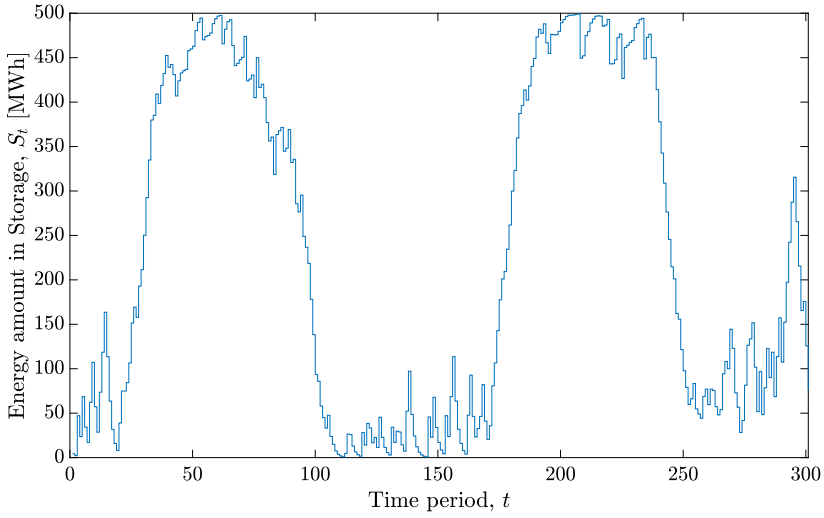


Figure 5.18: The storage level changes during the heat network operation with scenario MPC on the electricity price

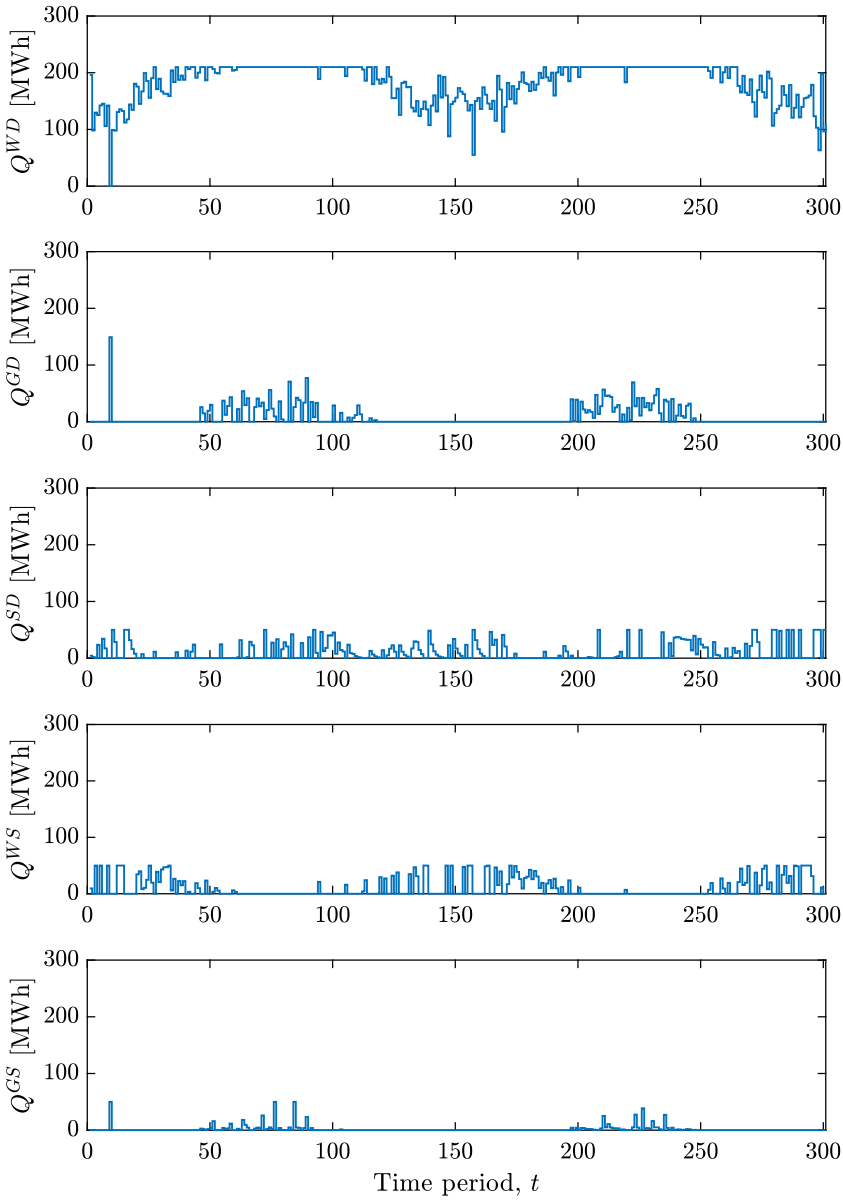


Figure 5.19: The heat flows during the heat network operation with scenario MPC on the electricity price

Scenario-Based MPC of the Thermal Energy Demand

A scenario tree considers the variation of the thermal energy demand in the scenario MPC of thermal energy demand. It evolves from the current thermal demand and the future prediction based on the parameters $\theta = \{\theta^{up}, \theta^{mid}, \theta^{down}\}$. In the case study, θ^{up} is set as $+20MWh$, θ^{mid} is no change and θ^{down} means $-20MWh$ from the prediction of demand. The prediction of the thermal energy demand is computed from the demand model (4.21). The thermal energy demand is evolved into nine scenarios as described on **Figure 5.20**.

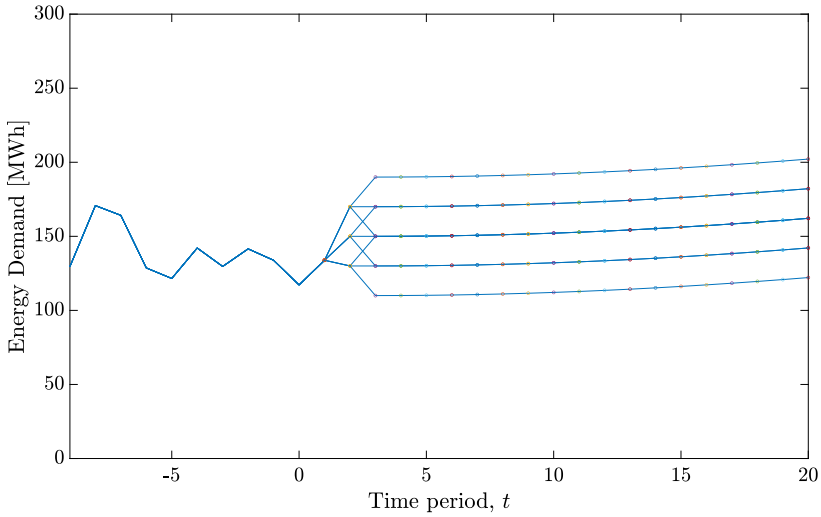


Figure 5.20: The evolution of the thermal energy demand

The possible decisions of the heat flows are diversified according to the evolution of the electricity price. The first input is applied to the system. As a result, the heat flow for the whole operation is described on **Figure 5.22** and the level change of the storage device is shown on **Figure 5.21**. The thermal energy storage is operated more conservatively by keeping some energy always high in the storage for more conservative operation. The heat flow Q^{GS} is more often used than other policies to keep the storage level. The expected cost of the price scenario MPC is **7.3122e+05[NOK]**.

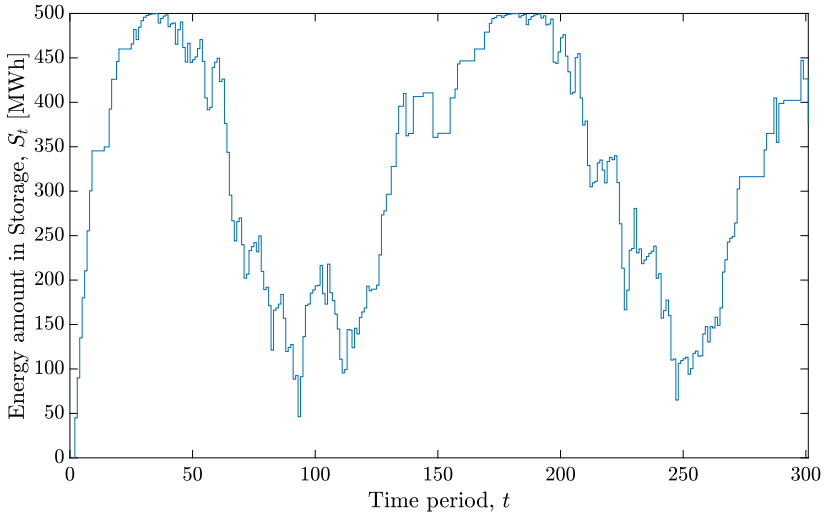


Figure 5.21: The storage level changes during the heat network operation with scenario MPC on demand

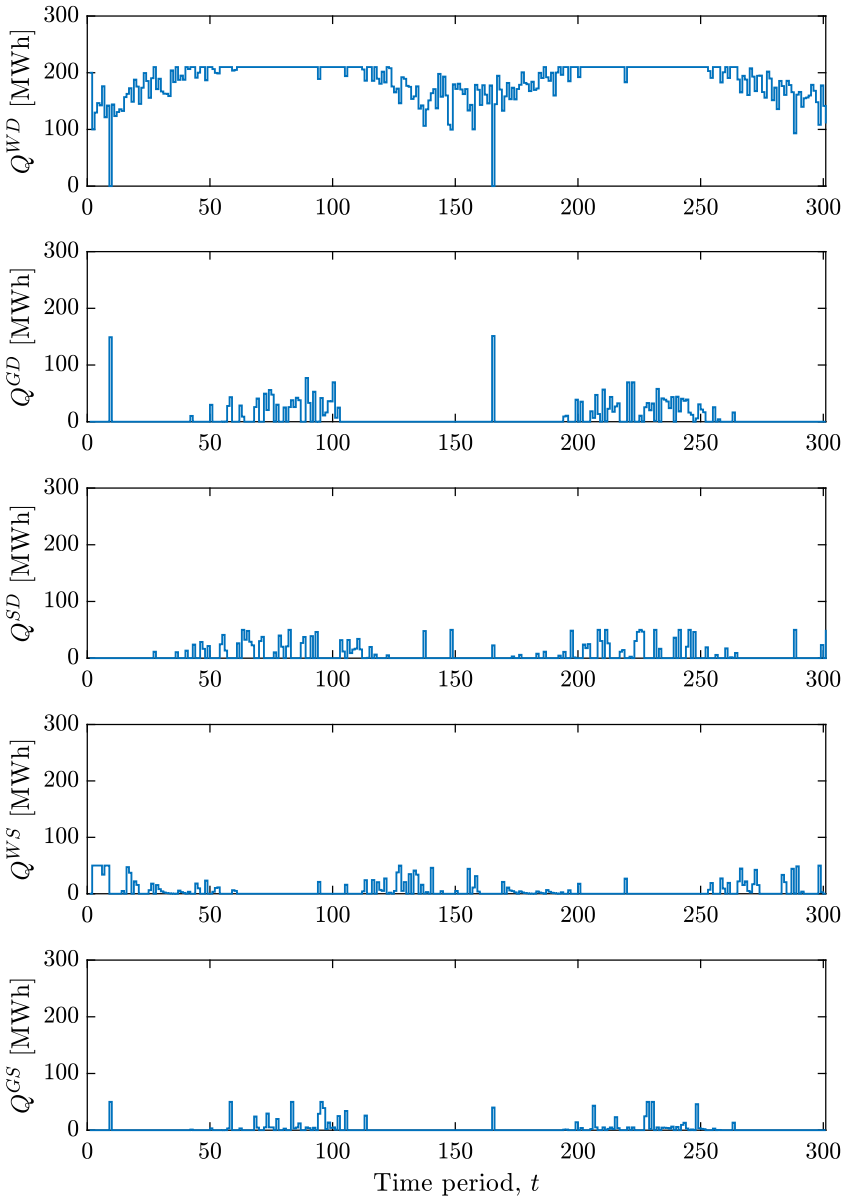


Figure 5.22: The heat flows during the heat network operation with scenario MPC on the demand

5.6 Optimal Operation of the District Heating System

To compare the policies implemented above, the deterministic optimal solution must be calculated. It is assumed, for the calculation, that all of the current and future disturbances are perfectly and precisely realized in advance. It can be computed by a standard batch linear program (LP), equation (**??**), over each simulation. the energy level change of TES in the optimal control policy can be described on **Figure 5.23** and the heat flows during the optimal operation is shown on **Figure 5.24**.

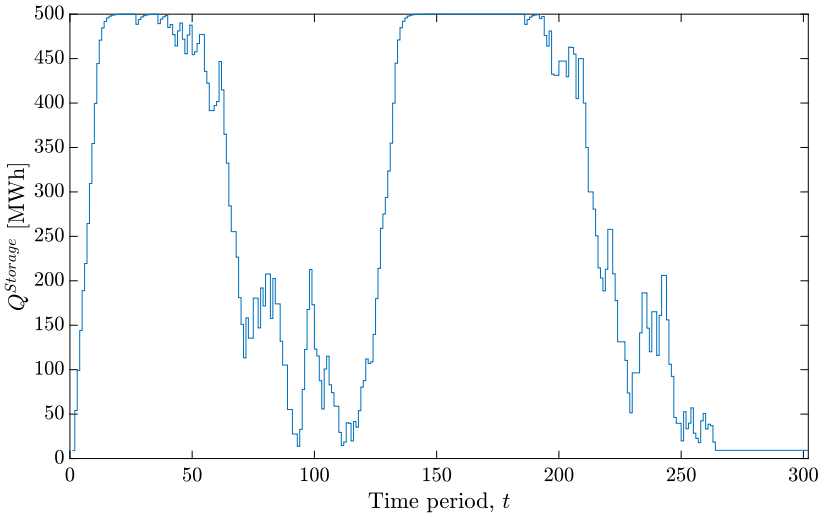


Figure 5.23: The storage level changes during the optimal heat network operation

The optimal total expected cost is computed by averaging out all of the costs on every simulation. It is **4.2467e+05[NOK]**

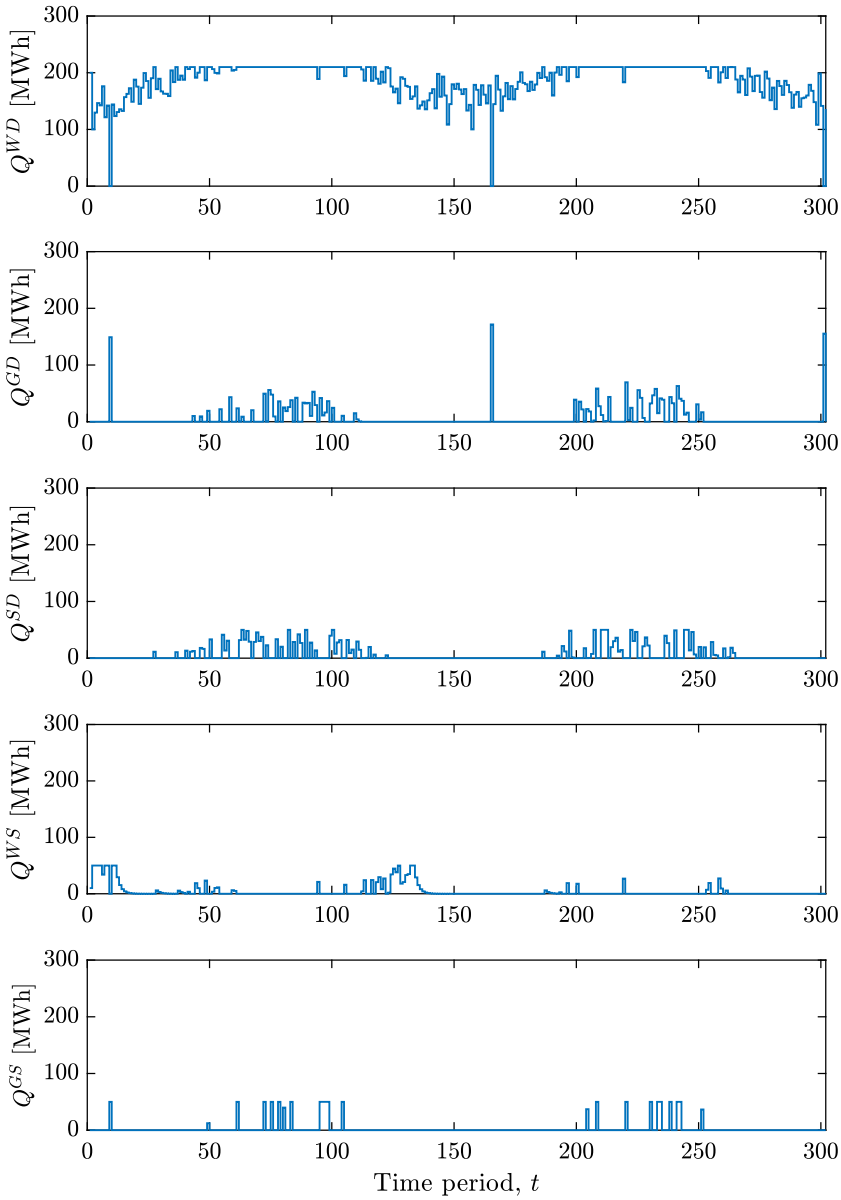


Figure 5.24: The heat flows during the optimal heat network operation

5.7 Comparison of Policies

The previous sections show the evaluation of four policies such as the simple policy, the certainty equivalence MPC, the parametric modified MPC, and scenario-based MPC by the batch learning method with 500 simulations. Each simulation adopts disturbance from a sample path generated from equations (4.20) - (4.22). As the number of simulation increases, the total expected costs of the policies are converged to certain numbers as described on **Figure 5.25**. It shows that it is beneficial to have a TES in the system.

The expected cost on each policy is shown on **Table 5.1**. The deterministic optimum with a perfect forecast has the lowest expected cost. The second-lowest expected cost comes from PFA. The expected cost increases in the following order: CFA on MPC, standard MPC, scenario MPC on price, and scenario MPC on demand. Lastly, the biggest total expected cost is required for the operation on the system without TES. It is shown the comparison value on each policy with the optimum. This result shows that the simple policy can provide near-optimal control policy than others in the presence of uncertainty.

	Expected Cost [NOK]$(\cdot 10^5)$	**Relative value to optimal solution** $[J_\pi / J_{\pi^*}]$
System without TES	7.7016	1.8135
Simple Policy	4.5397	1.0690
Certainty equivalence MPC	6.8444	1.6117
Parametric modified MPC $[\theta_R = 0.7]$	6.6977	1.5771
Scenario-based MPC [Price]	6.9666	1.6405
Scenario-based MPC [Demand]	7.3122	1.7219
Determistic optimal solution (Perfect forecast)	4.2467	1

Table 5.1: Comparison of the expected costs on all polices

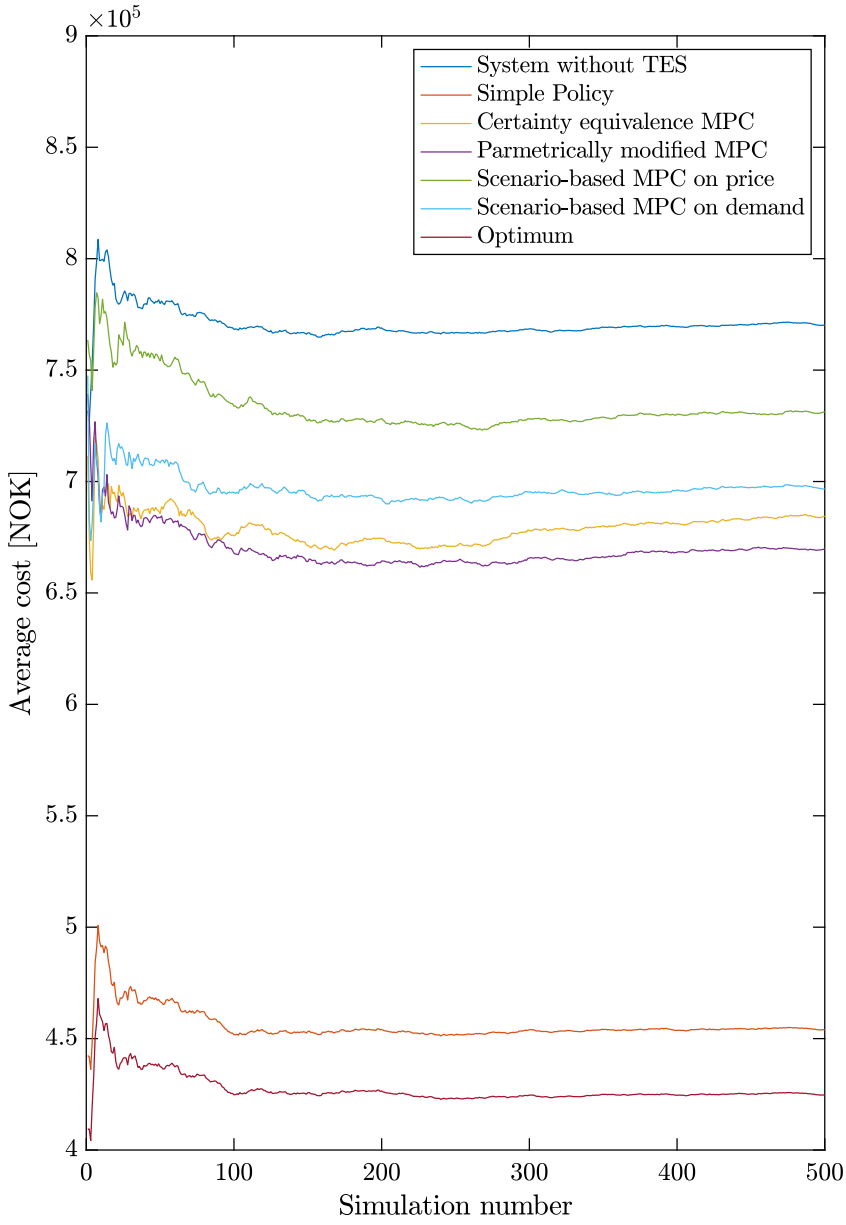


Figure 5.25: The converged costs of the implemented policies

Chapter 6

Discussion

This chapter includes the general discussion about issues from the result of this work and possible future work for improvement.

6.1 Performance of MPC

Although it is profitable to employ thermal energy storage (TES) by all of the control policies suggested in this work, the MPC control strategies such as certainty equivalence, parametric modified MPC and scenario-based MPC have quite higher operation cost compared to the simple policy. It is because the forecast model is not reflecting reality enough. If the spot electricity was described more precisely for the future by updating its forecast model or making a certain contract with the supplier, the expected cost is lower than the MPC control policies implemented in this work. For example, the expected total cost of MPC, with the existing demand forecast, equation (5.2), and the perfect price forecast assumed, is computed **4.3833e+05[NOK]**. It is approximately 35% deduction of the operating cost of the certainty equivalence MPC implemented in this work. It even shows a better result than the simple policy. This example emphasizes the importance of correct forecast models.

6.2 Issues with Assumptions

There are a few assumptions made for modelling the district heat network system. One of them is the capability to generate infinite amount of thermal energy by the electric boiler. This assumption simplifies the case study overall. Without the presumption, it becomes very difficult that the buy-low use-high policy satisfies the fluctuating demand for the entire operation. However, due to this assumption, sometimes it becomes disadvantageous to utilize the heat flow Q^{GS}. It is because the charging and discharging efficiencies are set to 0.9, which means the electricity purchased to store the thermal energy in TES is reduced by approximately 20% in use, meanwhile, the shortage of the demand can be fulfilled by

the heat flow $Q^G D$ instantly. This causes a tiny problem on simulation which is the system without TES can be cheaper sometimes than the operation with TES by some policies. It is described on **Figure 6.1**.

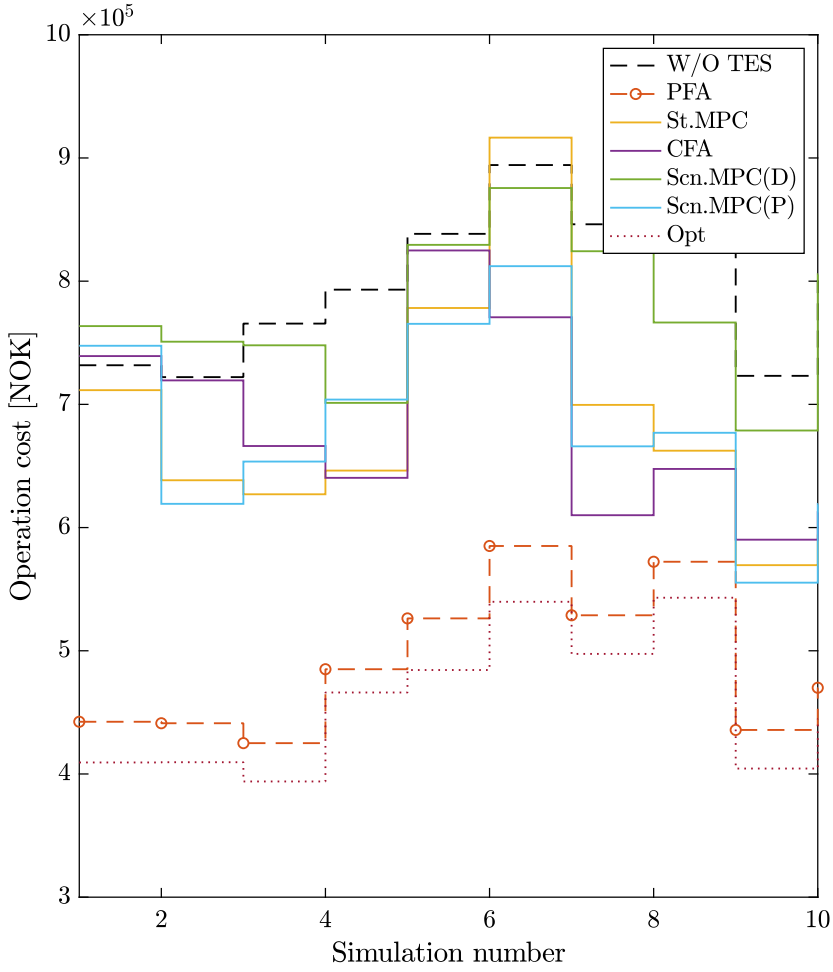


Figure 6.1: The operation cost of certain simulations

If the assumption was annulled so that the amount of thermal energy which the electric boiler can generate is limited and there was a large penalty cost for not satisfying demand, the heat flow Q^{GS} would become more valuable economically. The operation of the system without TES would become more expensive due to the failure to satisfy demand and the usage of TES will be even more advantageous by reducing the mismatch between the thermal energy supply and demand.

6.3 Possible Future Work

In this work, the simple policy such as buy-low use-high policy is chosen as the nearest policy to optimal operation. However, there are many potentials to change these evaluations for improvement after performing:

- Reformulate the TES model. In this work, the model is designed under the assumption that there is no heat loss over time on the temperature gradient, no cost to hold up the energy in storage, and the infinity capacity of electric boiler. For future work, a more rigorous model can be formulated.
- Apply the adaptive learning method for policy search online instead of keeping the parameters fixed on certain values. With varying parameters for better performance at each time, an interesting result may be observed.
- Upgrade the uncertainty models. For example, there are many forms of electricity subscriptions such as cheaper electricity price during the night or before a certain amount of consumption. It will give insight that which policy is more effective on which electricity subscription.
- Upgrade the disturbance forecast model. It would make MPC more competitive.
- To consider more various types of control policies, control policy based on Lookahead can be employed to formulate more types of control policies.

In general, future work on finding the most effective policy for the operation of a district heating system can be implemented from the upgrade of the TES modelling and the uncertainty models, and from formulating the policies.

Chapter 7

Conclusion

The conclusions and discussion regarding the objectives of this thesis are presented in this chapter.

The mathematical model for the district heating network system with one thermal energy storage device has been formulated with a few assumptions which simplify the case study. Static parameters, state variables, control variables, disturbances, constraints, system model, and objective function are defined for modelling. The formulated mathematical model is a linear programming problem because all constraints, system model, and the objective function are linear function, which made the case study easy to solve. The mathematical model is well explained out through **Chapter 4.2**

The uncertainty models, such as the thermal energy generated from the waste incineration plant, the thermal energy demand in the city, and the spot electricity price, are formulated to reflect the stochastic nature. The energy generation model, equation (4.20), is made based on assumption that the plant operates constantly. The energy demand model, equation (4.21), is forged based on sinusoidal equation and the noise is added in the form of normal distribution function with a fixed value of standard deviation and constraints on possible maximum and minimum demand amount. Lastly, the spot electricity price model, equation (4.22), is designed by Markov 1st order chain. The initial price is given from the previous data. The noise is added by normal distribution function and the jumps and spikes, which rarely happens, in the price is considered by the combination of normal distribution and condition with random number generation. As a result, 500 sample paths are created by using the three model mentioned above.

Various types of control policies are considered and implemented in this work. Firstly, the simple policy, which is formulated based on policy function approximation, is generated. It has the principal of the buy-low and high-use policy as described on equation (5.1). Secondly, the certainty equivalence MPC is formulated and implemented for the district heating network system. To run the optimization, the disturbance forecast models are created. The demand forecast model, equation (5.2), is made in form sinusoidal which reflects the hourly trend of the thermal energy demand without stochastic noise. The forecast model of the spot electricity price, equation (5.3), is formulated by using 1^{st} order Markov

chain model. It considers the random variation from the previous price but does not involves the occurrences of jump and spike in the price. Equation (5.4) shows the optimization problem of the certainty equivalence MPC to determine the control variables. Thirdly, the parametric modified MPC is structured based on the certainty equivalence MPC by using the cost function approximation (CFA). The parameters are introduced into constraints of the certainty equivalence MPC. In this work, three following parametric modified MPCs are implemented by manipulating: the maximum capacity of TES; estimation of the demand forecast; the maximum charging and discharging rates. The optimization problems of three parametric modified MPCs are written as equations (5.6), (5.5), and (5.7). Lastly, two scenario-based MPCs are created with the evolution of the scenario tree on either the electricity price or the energy demand. The optimization problem of the scenario-based MPC is presented on equation (5.8) with its non-anticipative constraint, equation (5.9).

The parameters in policies, such as the simple policy and parametric modified MPC, are tuned for the lower expected total costs during the operation of the system. For the simple policy, possible sets of the parameter $\theta = \{\theta^{low}, \theta^{high}\}$ are specified. The expected total costs are estimated on the possible sets of the parameter by the batch learning method, equation (2.10). The best parameters are determined as a set of parameters which exhibits the minimum expected total cost after comparing all of the calculated expected total costs. Similarly, the best parameters are found on the parametric modified MPC after computing expected total costs on specified parameters by the batch learning method, equation (2.10). Although two of all the parametric modified MPCs, which manipulate the maximum capacity of TES with a parameter θ_S and the demand forecast with a parameter θ_D, do not display the benefits in the expected total costs, the parametric modified MPC, adjusting the maximum charging and discharging rates with a parameter θ_R, showed the maximum reduction on the expected total cost when the parameter θ_R is tuned as 0.7.

After all of the policies are implemented with their best tuning, the expected total costs of every control policy are obtained by the batch learning method, equation (2.10). These expected total costs are compared with the cost of deterministic optimal solution and the heating network system without TES as shown on **Table 5.1**.

As a result of this work, the expected total cost of the simple policy is the lowest as the near-optimal operation. It shows that the simple policy can perform well in the presence of uncertainty if the policy is well structured. The parametric modified MPC policy shows the second-lowest. The certainty equivalence MPC policy has the third-lowest expected total cost. The scenario-based MPC policies on the spot electricity price and the demand come the next. As, lastly, the operation of the system without TES is the most expensive, it elucidates that it is advantageous to have TES as well.

Bibliography

Aneke, M., Wang, M., 2016. Energy storage technologies and real life applications – a state of the art review. Applied energy 179, 350–377.

Berrada, A., Loudiyi, K., 2019. Chapter 4 - gravity energy storage applications, in: Berrada, A., Loudiyi, K. (Eds.), Gravity Energy Storage. Elsevier, pp. 75 – 103. URL: http://www.sciencedirect.com/science/article/pii/B9780128167175000049, doi:https://doi.org/10.1016/B978-0-12-816717-5.00004-9.

Birge, J.R., 1997. State-of-the-art-survey–stochastic programming: Computation and applications. INFORMS Journal on Computing 9, 111–133.

Cabeza, L., Martorell, I., Miró, L., Fernández, A., Barreneche, C., 2015. 1 - introduction to thermal energy storage (tes) systems, in: Cabeza, L.F. (Ed.), Advances in Thermal Energy Storage Systems. Woodhead Publishing. Woodhead Publishing Series in Energy, pp. 1 – 28. URL: http://www.sciencedirect.com/science/article/pii/B9781782420880500018, doi:https://doi.org/10.1533/9781782420965.1.

Cutler, C.R., Ramaker, B.L., 1979. Dynamic matrix control??a computer control algorithm. IEEE Transactions on Automatic Control 17, 72.

Fernandez, A., Martínez, M., Segarra, M., Martorell, I., Cabeza, L., 2010. Selection of materials with potential in sensible thermal energy storage. Solar Energy Materials and Solar Cells 94, 1723 – 1729. URL: http://www.sciencedirect.com/science/article/pii/S0927024810003296, doi:https://doi.org/10.1016/j.solmat.2010.05.035.

Foss, B., H., 2013. Merging optimization and control. Lecture Notes.

Furbo, S., 2015. 2 - using water for heat storage in thermal energy storage (tes) systems, in: Cabeza, L.F. (Ed.), Advances in Thermal Energy Storage Systems. Woodhead Publishing. Woodhead Publishing Series in Energy, pp. 31 – 47. URL: http://www.sciencedirect.com/science/article/

pii/B978178242088050002X, doi:https://doi.org/10.1533/9781782420965.1.31.

Gil, A., Medrano, M., Martorell, I., Lázaro, A., Dolado, P., Zalba, B., Cabeza, L.F., 2010. State of the art on high temperature thermal energy storage for power generation. part 1—concepts, materials and modellization. Renewable and Sustainable Energy Reviews 14, 31 – 55. URL: http://www.sciencedirect.com/science/article/pii/S1364032109001774, doi:https://doi.org/10.1016/j.rser.2009.07.035.

Kalaiselvam, S., Parameshwaran, R., 2014. Chapter 2 - energy storage, in: Kalaiselvam, S., Parameshwaran, R. (Eds.), Thermal Energy Storage Technologies for Sustainability. Academic Press, Boston, pp. 21 – 56. URL: http://www.sciencedirect.com/science/article/pii/B9780124172913000025, doi:https://doi.org/10.1016/B978-0-12-417291-3.00002-5.

Lucia, S., Finkler, T., Engell, S., 2013. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. Journal of Process Control 23, 1306–1319.

Mayne, D., Rawlings, J., Rao, C., Scokaert, P., 2000. Constrained model predictive control: Stability and optimality. Automatica (Oxford) 36, 789–814.

Orosz, M., Dickes, R., 2017. 16 - solar thermal powered organic rankine cycles, in: Macchi, E., Astolfi, M. (Eds.), Organic Rankine Cycle (ORC) Power Systems. Woodhead Publishing, pp. 569 – 612. URL: http://www.sciencedirect.com/science/article/pii/B9780081005101000168, doi:https://doi.org/10.1016/B978-0-08-100510-1.00016-8.

Powell, W., 2020. REINFORCEMENT LEARNING AND STOCHASTIC OPTIMIZATION A unified framework for sequential decisions. John Wiley Sons, Inc.

Powell, W.B., 2019. A unified framework for stochastic optimization. European journal of operational research 275, 795–821.

van Ravenzwaaij, D., Cassey, P., Brown, S.D., 2016. A simple introduction to markov chain monte–carlo sampling. Psychonomic bulletin review 25, 143–154.

Richalet, J., Rault, A., Testud, J., Papon, J., 1978. Model predictive heuristic control: Applications to industrial processes. Automatica 14, 413 – 428. URL: http://www.sciencedirect.com/science/article/pii/0005109878900018, doi:https://doi.org/10.1016/0005-1098(78)90001-8.

Shapiro, A., Dentcheva, D., Ruszczyński, A., 2009. Lectures on stochastic programming: modeling and theory. MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics.

Zervos, M., Johnson, T.C., Alazemi, F., 2013. Buy-low and sell-high investment strategies. Mathematical Finance 23, 560–578.

Appendix A

MATLAB code

This section contains some important pieces of source code developed in this project. The code is written in MATLAB.

A.1 Static parameters Initial states

The function `s0.m` gives inputs of static parameters and initial state variables for the system.

```
function S0 = S0()
% Static parameters
S0.R_max = 500;      % The energy capacity of TES in MWh
S0.eff_c = 0.9;      % The charging efficiency of TES
S0.eff_d = 0.9;      % The discharging efficiency of TES
S0.rate_c = 50;      % The maximum charging rate of TES, given as MWh per
    time period
S0.rate_d = 50;      % The maximum discharging rate of TES, given as MWh
    per time period

% Initial states
S0.S.R = 0;      % Energy level of TES
S0.S.E = 210;    % Energy supply
S0.S.D = 200;    % Energy demand
S0.S.P = 200;    % Spot electricity price
end
```

A.2 Disturbances

The file `Exogenous_gen.m` creates sample paths of disturbance from the demand and the spot electricity price model, the functions `Demand.m` and `Electricity_price` respectively.

Exogenous_gen.m

```
clear
clc
close all
tic


T = 300; % Time
n_scenario = 500; %Number of desired sample paths

%Demand parameter
mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;

%Electricity price parameter
Pmin = 0;
Pmax = 2500;
P0 = 200;
P1std = 50;
P2std = 500;
%Exogenous information storage.
sample_path.D = [];
sample_path.P = [];
sample_path.E = ones(1,T+1)*210;

for i = 1:n_scenario
%Disturbance functions
D_one = Demand(mean,amplitude,Dmin,Dmax,T);
P_one = Electricity_price(Pmin,Pmax,P0,P1std,P2std,T);

sample_path.D = [sample_path.D; D_one];
sample_path.P = [sample_path.P; P_one];
end

save('sample_path','sample_path')
```

Demand.m

```
function D = Demand(mean,amplitude,Dmin,Dmax,T)
t = 0:T;
D = mean + amplitude * sin(2*2*pi*t/T -pi/2) + normrnd(0,20,[1,T+1]);
D = min(max(D,Dmin),Dmax);
end
```

Electricity_price.m

```
function P = Electricity_price(Pmin,Pmax,P0,P1std,P2std,T)
P1 = normrnd(0,P1std,1,T+1);%P1std=50
P2 = normrnd(0,P2std,1,T+1);%P2std=500
a =  unifrnd(0,1,1,T+1);
p = 0.031;
for i = 1:length(a)
    if a(i)>p
        a(i) = 0;
```

```
    else
        a(i) = 1;
    end
end
P = P0 + P1 + a.*P2;
P = min(max(P,Pmin),Pmax);
end
```

A.3 Expected cost of the system without TES

The file `P00_woTES_main.m` computes the expected total cost of the system without TES.

```
Clear
clc
close all
%% Load
addpath('../Exogenous_information')
load('sample_path.mat')
plot_settings

%%
Elec_use = sample_path.D - 210; % Demand - Supply
Unsat_Demand = Elec_use > 0; % If Demand - Supply > 0, then 1, or 0.
Elec_use = Elec_use.*Unsat_Demand; % Required heat to satisfy the demand
spot_cost = Elec_use.*sample_path.P; %The cost to satisfy the demand on
    each time

operation_cost = sum(spot_cost,2); %The cost to satisfy the demand on each
     operation
Expected_cost = mean(operation_cost);
```

A.4 Simple policy

The file `P01_SP_xyz.m` performs the policy search and shows the plots of the result. The expected total cost is computed by the function `PFA_a_cost.m`.

`P01_SP_xyz`

```
clear all
close all
clc

%% Load
addpath('../Exogenous_information')
load('sample_path.mat')
plot_settings

%% Simple policy

%parameter array
low_price = linspace(0,300,11)
plus_price = linspace(10,100,10)
```

```

%Computation of expected total cost
for i = 1:size(low_price,2)
for k = 1:size(plus_price,2)
P_low = low_price(i);
P_plus = plus_price(k);
P_high = P_low + P_plus;
Avg_Cost(i,k) = PFA_a_cost(S0,sample_path,P_high,P_low);
end
end
%% Policy search
mini = min(min(Avg_Cost)) %minimum expected total cost
[x,y]=find(Avg_Cost==mini);
plus = plus_price(y);
low = low_price(x); % theta low
high = low + plus;  % theta high

%% Ploting
Font = 30;

plot_settings
figure()
surf(plus_price,low_price,Avg_Cost);
hold on
plot3(plus_price(y),low_price(x),min(min(Avg_Cost)),'-o','Color','r','
    MarkerSize',20,'MarkerFaceColor','r')

xlabel('$\theta^{plus}$','FontSize',Font)
ylabel('$\theta^{low}$','FontSize',Font)
zlabel('Expected Cost[NOK]','FontSize',Font)
set(gca,'FontSize',Font)
```

PFA_a_cost.m

```
function Avg_Cost = PFA_a_cost(S0,sample_path,P_high,P_low)

% number of scenario & time horizon
n_scenario = size(sample_path.D,1);
time = size(sample_path.D,2);

% Initial state
S = S0.S;
rate_d = S0.rate_d*ones(n_scenario,1);
rate_c = S0.rate_c*ones(n_scenario,1);
R_max = S0.R_max;
    %initial states
    R = S.R*ones(n_scenario,1);
    D = S.D*ones(n_scenario,1);
    E = S.E*ones(n_scenario,1);
    P = S.P*ones(n_scenario,1);

%% The simple policy based on specific parameters for low and high prices.

u.WD = [];
u.GD = [];
u.RD = [];
u.WR = [];
```

```
u.GR = [];
x.R = [];
x.D = [];
x.E = [];
x.P = [];
x.C = [];
for t = 1:time

    %policy => control variable
    WD = min(D,E);
    comp_h = P>P_high;
    RD = comp_h.*min(min(D-WD,rate_d),R)*S0.eff_d;
    GD = D-WD-RD;
    WR = min(min(E-WD,rate_c),R_max-R);
    comp_l = P<P_low;
    GR = comp_l.*min(rate_c-WR,R_max-R-WR);
    C = P.*(GD+GR);

    %Data storage
    u.WD = [u.WD, WD];
    u.GD = [u.GD, GD];
    u.RD = [u.RD, RD];
    u.WR = [u.WR, WR];
    u.GR = [u.GR, GR];
    x.R = [x.R, R];
    x.D = [x.D, D];
    x.E = [x.E, E];
    x.P = [x.P, P];
    x.C = [x.C, C];

    %update states
    R = R+S0.eff_c*(WR+GR)-RD;
    D = sample_path.D(1:n_scenario,t);
    E = sample_path.E(1,t)*ones(n_scenario,1);
    P = sample_path.P(1:n_scenario,t);
end
% Cost
Cost = sum(x.P.*(u.GD+u.GR),2);
for k = 1:n_scenario
    Avg_Cost_cum(k) = mean(Cost(1:k));
end
Avg_Cost = mean(Cost);

end
```

A.5 Certainty equivalence MPC

The file `P02_CEMPC_main.m` runs the simulation of the certainty equivalence MPC.

```
close all
clear all
clc
tic

%% Load
addpath('../P05_Apriori_optimal')
addpath('../Exogenous_information')
```

```
addpath('../P01_PFA')
load('sample_path.mat')

%% Settings

S0 = S0();
S = S0.S;

%Scenario number
n_scenario = size(sample_path.D,1);
%total time horizon
N = size(sample_path.D,2);

%Open loop optimization time horizon
t = 20;

%Extention of sample path
sample_path.E = [S.E sample_path.E sample_path.E(1,1:t+1)];
sample_path.D = [sample_path.D]; %sample_path.D(1:n_scenario,1:t+1)];
sample_path.P = [sample_path.P]; %sample_path.P(1:n_scenario,1:t+1)];

%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];

%% simulation

Cost2 =[];
for w = 1:n_scenario
ss
%Initial setup
S0 = S0();
S = S0.S;

%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];


for k = 1:N

    %exogenous information model
        Price_horizon =  normrnd(0,50,1,t) + S.P;
        %Price_horizon =  S.P*ones(1,t);
        Demand_horizon = [S.D D(1,k+1:k+t-1)];
        Energy_horizon = sample_path.E(1,k:k+t-1);

    %optimize the system
        %%% EQUALITY CONSTRAINT
            %Demand constraint
```

```
        Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        Beq_D = Demand_horizon';
            %Transition function
        Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(eye
    (t),[0 0 0 0 0 1]);
        Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        Aeq = [Aeq_D;Aeq_T];
        Beq = [Beq_D;Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        A_1 = kron(eye(t), [0 0 0 1 1 1]);
        B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
        A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        B_2 = zeros(t,1);
            %Charing rate limitation
        A_3 = kron(eye(t),[0 0 0 1 1 0]);
        B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        A_4 = kron(eye(t),[0 0 1 0 0 0]);
        B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
        A_5 = kron(eye(t),[1 0 0 1 0 0]);
        B_5 = Energy_horizon';
            % A&B integration
        A = [A_1;A_2;A_3;A_4;A_5];
        B = [B_1;B_2;B_3;B_4;B_5];

        %%% Objective function
        f = kron(Price_horizon,[-0.02 1 0 -0.01 1 0]);

        %%% Boundaries - lower and upper
        lb = zeros(6*t,1);
        ub = ones(6*t,1)*inf;

        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.WD = w_opt(1:t:6*t);
        u.GD = w_opt(2:t:6*t);
        u.SD = w_opt(3:t:6*t);
        u.WS = w_opt(4:t:6*t);
        u.GS = w_opt(5:t:6*t);

        % Take an action
        uk = [u.WD(1);u.GD(1);u.SD(1);u.WS(1);u.GS(1)];

        %Simulating the plant behavior during dt
        S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;

        %realization
        S.P = sample_path.P(w,k);
        S.D = sample_path.D(w,k);
        %Energy_horizon = sample_path.E(1,k);
```

```
122
123         %For plotting
124         xSim = [xSim, S.R];
125         uSim = [uSim, uk];
126
127 end
128 Cost1 = sum(sample_path.P(w,1:N).*(uSim(2,1:N) + uSim(5,1:N)));
129 Cost2 = [Cost2; Cost1];
130 end
131 Cost = mean(Cost2);
132
133 for i = 1:size(Cost2,1)
134 Avg_cost_cum(i) = mean(Cost2(1:i));
135 end
136
137 % save
138 name = strcat(date,'_st_MPC_',num2str(w),'itr_',num2str(t),'_hz');
139 save(name)
```

A.6 Parametric modified MPC

The files `P03_theta_D_MPC_main.m`, `P03_theta_S_MPC_main.m`, and `P03_theta_R_MPC_main.m` runs the simulations for policy search on the parametic modified MPC.

`P03_theta_D_MPC_main.m`

```
1  close all
2  clear all
3  clc
4
5  %% Load
6  addpath('../P05_Apriori_optimal')
7  addpath('../Exogenous_information')
8  addpath('../P01_PFA')
9  load('sample_path.mat')
10
11 %% Optimum calculation
12
13 S0 = S0();
14 S = S0.S;
15
16 %Scenario number
17 n_scenario = size(sample_path.D,1);
18 %total time horizon
19 N = size(sample_path.D,2);
20
21 %Open loop optimization time horizon
22 t = 20;
23
24 %Extention of sample path
25 sample_path.E = [S.E sample_path.E sample_path.E(1,1:t+1)];
26 sample_path.D = [sample_path.D];
```

```
sample_path.P = [sample_path.P];

%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];

% parameter tuning for D
theta_array = linspace(0.5, 1.5, 11)

%% simulation

Modified_cost = [];
for kk = 1:size(theta_array,2)
    theta = theta_array(kk);
Cost2 =[];
for w = 1:n_scenario
ss
%Initial setup
S0 = S0();
S = S0.S;

%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];

for k = 1:N

    %exogenous information model
        Price_horizon =  normrnd(0,50,1,t) + S.P;
        Demand_horizon = [S.D D(1,k+1:k+t-1)*theta];
        Energy_horizon = sample_path.E(1,k:k+t-1);

    %optimize the system
        %%% EQUALITY CONSTRAINT
            %Demand constraint
        Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        Beq_D = Demand_horizon';
            %Transition function
        Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(eye
    (t),[0 0 0 0 0 1]);
        Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        Aeq = [Aeq_D;Aeq_T];
        Beq = [Beq_D;Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        A_1 = kron(eye(t), [0 0 0 1 1 1]);
        B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
```

```
        A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        B_2 = zeros(t,1);
            %Charing rate limitation
        A_3 = kron(eye(t),[0 0 0 1 1 0]);
        B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        A_4 = kron(eye(t),[0 0 1 0 0 0]);
        B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
        A_5 = kron(eye(t),[1 0 0 1 0 0]);
        B_5 = Energy_horizon';
            % A&B integration
        A = [A_1;A_2;A_3;A_4;A_5];
        B = [B_1;B_2;B_3;B_4;B_5];

        %%% Objective function
        f = kron(Price_horizon,[-0.02 1 0 -0.01 1 0]);

        %%% Boundaries - lower and upper
        lb = zeros(6*t,1);
        ub = ones(6*t,1)*inf;

        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.WD = w_opt(1:t:6*t);
        u.GD = w_opt(2:t:6*t);
        u.SD = w_opt(3:t:6*t);
        u.WS = w_opt(4:t:6*t);
        u.GS = w_opt(5:t:6*t);

        %realization
        S.P = sample_path.P(w,k);
        S.D = sample_path.D(w,k);

        %Energy_horizon = sample_path.E(1,k);

        % Take an action
        uk = [u.WD(1);u.GD(1);u.SD(1);u.WS(1);u.GS(1)];

        %Simulating the plant behavior during dt
        S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;

        %For plotting
        xSim = [xSim, S.R];
        uSim = [uSim, uk];

end

Cost1 = sum(sample_path.P(w,1:N).*(uSim(2,1:N) + uSim(5,1:N)));
Cost2 = [Cost2; Cost1];
end
Cost = mean(Cost2);
Modified_cost = [Modified_cost, Cost];
end

```

```
for i = 1:size(Cost2,1)
Avg_cost_cum(i) = mean(Cost2(1:i));
end

name = strcat(date,'_D_MPC_',num2str(w),'itr_',num2str(t),'_hz');
save(name)

%% plot
plot_settings
close all

figure()
plot(theta_array,Modified_cost,'LineWidth',1.5)
axis([min(theta_array) max(theta_array) min(Modified_cost) max(
    Modified_cost)])
xlabel('A parameter for the demand forecast, $\theta_{D}$','FontSize',33)
ylabel('Expected Cost [NOK]','FontSize',33)
set(gca,'FontSize',33)
```

P03_theta_S_MPC_main.m

```
close all
clear all
clc

%% Load
addpath('../P05_Apriori_optimal')
addpath('../Exogenous_information')
addpath('../P01_PFA')
load('sample_path.mat')

%% Optimum calculation

S0 = S0();
S = S0.S;

%Scenario number
n_scenario = size(sample_path.D,1);
%total time horizon
N = size(sample_path.D,2);

%Open loop optimization time horizon
t = 20;

%Extention of sample path
sample_path.E = [S.E sample_path.E sample_path.E(1,1:t+1)];
sample_path.D = [sample_path.D];
sample_path.P = [sample_path.P];


%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
```

```
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];

% parameter tuning for D
theta_array = linspace(0.5, 1.5, 11)

%% simulation

Modified_cost = [];
for kk = 1:size(theta_array,2)
    theta = theta_array(kk);
Cost2 =[];
for w = 1:n_scenario

%Initial setup
S0 = S0();
S = S0.S;

%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];

for k = 1:N

    %exogenous information model
        Price_horizon =  normrnd(0,50,1,t) + S.P;
        Demand_horizon = [S.D D(1,k+1:k+t-1)];
        Energy_horizon = sample_path.E(1,k:k+t-1);

    %optimize the system
        %%% EQUALITY CONSTRAINT
            %Demand constraint
        Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        Beq_D = Demand_horizon';
            %Transition function
        Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(eye
    (t),[0 0 0 0 0 1]);
        Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        Aeq = [Aeq_D;Aeq_T];
        Beq = [Beq_D;Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        A_1 = kron(eye(t), [0 0 0 1 1 1]);
        B_1 = S0.R_max*ones(t,1)*theta;
            %Discharging limitation : Q,SD < amount in Storage
        A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        B_2 = zeros(t,1);
            %Charing rate limitation
        A_3 = kron(eye(t),[0 0 0 1 1 0]);
        B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        A_4 = kron(eye(t),[0 0 1 0 0 0]);
        B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
```

```
        A_5 = kron(eye(t),[1 0 0 1 0 0]);
        B_5 = Energy_horizon';
            % A&B integration
        A = [A_1;A_2;A_3;A_4;A_5];
        B = [B_1;B_2;B_3;B_4;B_5];

        %%% Objective function
        f = kron(Price_horizon,[-0.02 1 0 -0.01 1 0]);

        %%% Boundaries - lower and upper
        lb = zeros(6*t,1);
        ub = ones(6*t,1)*inf;

        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.WD = w_opt(1:t:6*t);
        u.GD = w_opt(2:t:6*t);
        u.SD = w_opt(3:t:6*t);
        u.WS = w_opt(4:t:6*t);
        u.GS = w_opt(5:t:6*t);

        %realization
        S.P = sample_path.P(w,k);
        S.D = sample_path.D(w,k);
        %Energy_horizon = sample_path.E(1,k);

        % Take an action
        uk = [u.WD(1);u.GD(1);u.SD(1);u.WS(1);u.GS(1)];

        %Simulating the plant behavior during dt
        S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;

        %For plotting
        xSim = [xSim, S.R];
        uSim = [uSim, uk];

end

Cost1 = sum(sample_path.P(w,1:N).*(uSim(2,1:N) + uSim(5,1:N)));
Cost2 = [Cost2; Cost1];
end
Cost = mean(Cost2);
Modified_cost = [Modified_cost, Cost];
end

for i = 1:size(Cost2,1)
Avg_cost_cum(i) = mean(Cost2(1:i));
end

%% Plot
plot_settings
close all

figure()
plot(theta_array,Modified_cost,'LineWidth',1.5)
```

```
axis([min(theta_array) max(theta_array) min(Modified_cost) max(
    Modified_cost)])
xlabel('A parameter for the maximum capcity of TES, $\theta_{S}$','
    FontSize',33)
ylabel('Expected Cost [NOK]','FontSize',33)
set(gca,'FontSize',33)
```

P03_theta_R_MPC_main.m

```
close all
clear all
clc


%% Load
addpath('../P05_Apriori_optimal')
addpath('../Exogenous_information')
addpath('../P01_PFA')
load('sample_path.mat')

%% Optimum calculation

S0 = S0();
S = S0.S;

%Scenario number
n_scenario = size(sample_path.D,1);
%total time horizon
N = size(sample_path.D,2);

%Open loop optimization time horizon
t = 20;

%Extention of sample path
sample_path.E = [S.E sample_path.E sample_path.E(1,1:t+1)];
sample_path.D = [sample_path.D];
sample_path.P = [sample_path.P];

%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];

% parameter tuning for D
theta_array = linspace(0.1, 2.0, 20)

%% simulation

Modified_cost = [];
for kk = 1:size(theta_array,2)
    theta = theta_array(kk);
Cost2 =[];
```

```
for w = 1:n_scenario
ss
%Initial setup
S0 = S0();
S = S0.S;

%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];

for k = 1:N

    %exogenous information model
        Price_horizon =  normrnd(0,50,1,t) + S.P;
        Demand_horizon = [S.D D(1,k+1:k+t-1)];
        Energy_horizon = sample_path.E(1,k:k+t-1);

    %optimize the system
        %%% EQUALITY CONSTRAINT
            %Demand constraint
        Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        Beq_D = Demand_horizon';
            %Transition function
        Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(eye
    (t),[0 0 0 0 0 1]);
        Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        Aeq = [Aeq_D;Aeq_T];
        Beq = [Beq_D;Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        A_1 = kron(eye(t), [0 0 0 1 1 1]);
        B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
        A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        B_2 = zeros(t,1);
            %Charing rate limitation
        A_3 = kron(eye(t),[0 0 0 1 1 0]);
        B_3 = S0.rate_c*ones(t,1)*theta;
            %Discharing rate limitation
        A_4 = kron(eye(t),[0 0 1 0 0 0]);
        B_4 = S0.rate_d*ones(t,1)*theta;
            %Waste incinerator : Heat supply amount
        A_5 = kron(eye(t),[1 0 0 1 0 0]);
        B_5 = Energy_horizon';
            % A&B integration
        A = [A_1;A_2;A_3;A_4;A_5];
        B = [B_1;B_2;B_3;B_4;B_5];

        %%% Objective function
        f = kron(Price_horizon,[-0.02 1 0 -0.01 1 0]);

        %%% Boundaries - lower and upper
        lb = zeros(6*t,1);
        ub = ones(6*t,1)*inf;
```

```
105
106         %%% Optimization
107         w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);
108
109         % Data store from openloop optimization
110         u.WD = w_opt(1:t:6*t);
111         u.GD = w_opt(2:t:6*t);
112         u.SD = w_opt(3:t:6*t);
113         u.WS = w_opt(4:t:6*t);
114         u.GS = w_opt(5:t:6*t);
115
116         %realization
117         S.P = sample_path.P(w,k);
118         S.D = sample_path.D(w,k);
119         %Energy_horizon = sample_path.E(1,k);
120
121         % Take an action
122         uk = [u.WD(1);u.GD(1);u.SD(1);u.WS(1);u.GS(1)];
123
124         %Simulating the plant behavior during dt
125         S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;
126
127         %For plotting
128         xSim = [xSim, S.R];
129         uSim = [uSim, uk];
130
131 end
132
133 Cost1 = sum(sample_path.P(w,1:N).*(uSim(2,1:N) + uSim(5,1:N)));
134 Cost2 = [Cost2; Cost1];
135 end
136 Cost = mean(Cost2);
137 Modified_cost = [Modified_cost, Cost];
138 end
139
140 for i = 1:size(Cost2,1)
141 Avg_cost_cum(i) = mean(Cost2(1:i));
142 end
143
144
145 %% Plot
146 plot_settings
147 close all
148
149 figure()
150 plot(theta_array,Modified_cost,'LineWidth',1.5)
151 axis([min(theta_array) max(theta_array) min(Modified_cost) max(
        Modified_cost)])
152 xlabel('A parameter for the maximum charging and discharging rates, $\
        theta_{R}$','FontSize',33)
153 ylabel('Expected Cost [NOK]','FontSize',33)
154 set(gca,'FontSize',33)
```

A.7 Scenario-based MPC

The files `P04_P_scn_MPC_main.m` and `P04_D_scn_MPC_main.m` runs the scenario-based MPC on the spot electricty price and the demand respectively.

`P04_P_scn_MPC_main.m`

```
close all
clear all
clc

%% Load
addpath('../Exogenous_information')
addpath('../P01_PFA')
load('sample_path.mat')

%% Optimum calculation

%initial setup
S0 = S0();
S = S0.S;

%Scenario number
n_scenario = size(sample_path.D,1);

%total time horizon
N = size(sample_path.D,2);

%Open loop optimization time horizon
t = 20;

%Extention of sample path
sample_path.E = [S.E sample_path.E sample_path.E(1,1:t+1)];

%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];

%% setting

%Price_scneario parameters.
Theta_up = 1.3;
Theta_mid = 1.0;
Theta_down = 0.7;

% %% simulation
Cost2 =[];

for w = 1:1%n_scenario
% initial states
```

```
S = S0.S;

%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];

for k = 1:N

        %exogenous information model
        Price_R =  normrnd(0,50,1,t) + S.P;
        Demand_horizon = [S.D D(1,k+1:k+t-1)];
        Energy_horizon = sample_path.E(1,k:k+t-1);

%% Process constraint : Same for all senarios.
        %%% EQUALITY CONSTRAINT
            %Demand constraint
        S_Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        S_Beq_D = Demand_horizon';
            %Transition function
        S_Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(
    eye(t),[0 0 0 0 0 1]);
        S_Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        S_Aeq = [S_Aeq_D; S_Aeq_T];
        S_Beq = [S_Beq_D; S_Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        S_A_1 = kron(eye(t), [0 0 0 1 1 1]);
        S_B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
        S_A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        S_B_2 = zeros(t,1);
            %Charing rate limitation
        S_A_3 = kron(eye(t),[0 0 0 1 1 0]);
        S_B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        S_A_4 = kron(eye(t),[0 0 1 0 0 0]);
        S_B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
        S_A_5 = kron(eye(t),[1 0 0 1 0 0]);
        S_B_5 = Energy_horizon';
        % A&B integration
        S_A = [S_A_1; S_A_2; S_A_3; S_A_4; S_A_5];
        S_B = [S_B_1; S_B_2; S_B_3; S_B_4; S_B_5];

%% PRICE SCENARIOS : 9 trees.
f_form = [-0.02 1 0 -0.01 1 0]
% Scenario. 01 : Present-UP-UP-Costant
        pt.S1_Price_horizon = [Price_R(1) (Theta_up*Price_R(2)) Theta_up*(
    Theta_up*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S1_f = kron(pt.S1_Price_horizon,f_form);

% Scenario. 02 : Present-UP-MID-Costant
```

```
        pt.S2_Price_horizon = [Price_R(1) (Theta_up*Price_R(2)) Theta_mid*(Theta_up*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S2_f = kron(pt.S2_Price_horizon,f_form);

% Scenario. 03 : Present-UP-DOWN-Costant
        pt.S3_Price_horizon = [Price_R(1) (Theta_up*Price_R(2)) Theta_down*(Theta_up*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S3_f = kron(pt.S3_Price_horizon,f_form);

% Scenario. 04 : Present-MID-UP-Costant
        pt.S4_Price_horizon = [Price_R(1) (Theta_mid*Price_R(2)) Theta_up*(Theta_mid*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S4_f = kron(pt.S4_Price_horizon,f_form);

% Scenario. 05 : Present-MID-MID-Costant
        pt.S5_Price_horizon = [Price_R(1) (Theta_mid*Price_R(2)) Theta_mid*(Theta_mid*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S5_f = kron(pt.S5_Price_horizon,f_form);

% Scenario. 06 : Present-MID-DOWN-Costant
        pt.S6_Price_horizon = [Price_R(1) (Theta_mid*Price_R(2)) Theta_down*(Theta_mid*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S6_f = kron(pt.S6_Price_horizon,f_form);

% Scenario. 07 : Present-DOWN-UP-Costant
        pt.S7_Price_horizon = [Price_R(1) (Theta_down*Price_R(2)) Theta_up*(Theta_down*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S7_f = kron(pt.S7_Price_horizon,f_form);

% Scenario. 08 : Present-DOWN-MID-Costant
        pt.S8_Price_horizon = [Price_R(1) (Theta_down*Price_R(2)) Theta_mid*(Theta_down*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S8_f = kron(pt.S8_Price_horizon,f_form);

% Scenario. 09 : Present-DOWN-DOWN-Costant
        pt.S9_Price_horizon = [Price_R(1) (Theta_down*Price_R(2)) Theta_down*(Theta_down*Price_R(3))*ones(1,t-2)];
        %%% Objective function
        S9_f = kron(pt.S9_Price_horizon,f_form);

%% Non-anticipativity constraints
nas = [1 1 1 1 1 0];
naz = zeros(1,6);

na_A1 = [nas kron(ones(1,t-1),naz) -nas kron(ones(1,t-1),naz) kron(ones(1,t*7), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*1), naz) -nas kron(ones(1,t-1),naz) kron(ones(1,t*6), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*2), naz) -nas kron(ones(1,t-1),naz) kron(ones(1,t*5), naz)
```

```
    nas kron(ones(1,t-1),naz) kron(ones(1,t*3), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*4), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*4), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*3), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*5), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*2), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*6), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*1), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*7), naz) -nas kron(ones(1,t-1)
    ,naz) ];

na_A2 = [naz nas kron(ones(1,t-2),naz) naz -nas kron(ones(1,t-2),naz) kron
    (ones(1,t*7),naz)
        naz nas kron(ones(1,t-2),naz) kron(ones(1,t),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t*6),naz)
        kron(ones(1,t*3),naz) naz nas kron(ones(1,t-2),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t*4),naz)
        kron(ones(1,t*3),naz) naz nas kron(ones(1,t-2),naz) kron(ones(1,t)
    ,naz) naz -nas kron(ones(1,t-2),naz) kron(ones(1,t*3),naz)
        kron(ones(1,t*6),naz) naz nas kron(ones(1,t-2),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t),naz)
        kron(ones(1,t*6),naz) naz nas kron(ones(1,t-2),naz) kron(ones(1,t)
    ,naz) naz -nas kron(ones(1,t-2),naz)];

na_Aeq = [na_A1; na_A2];
na_Beq = zeros(size(na_Aeq,1),1);

%% Integration of linprog factors

        Aeq = kron(eye(9),S_Aeq);
        Aeq = [Aeq; na_Aeq];
        Beq = kron(ones(9,1),S_Beq);
        Beq = [Beq; na_Beq];
        A = kron(eye(9),S_A);
        B = kron(ones(9,1),S_B);
        f = [S1_f S2_f S3_f S4_f S5_f S6_f S7_f S8_f S9_f];
        lb = zeros(6*9*t,1);
        ub = ones(6*9*t,1)*inf;

%%        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.S1.WD = w_opt(1:6:6*t);
        u.S1.GD = w_opt(2:6:6*t);
        u.S1.SD = w_opt(3:6:6*t);
        u.S1.WS = w_opt(4:6:6*t);
        u.S1.GS = w_opt(5:6:6*t);
        u.S1.R = w_opt(6:6:6*t);

        u.S2.WD = w_opt(6*t+1:6:6*2*t);
        u.S2.GD = w_opt(6*t+2:6:6*2*t);
        u.S2.SD = w_opt(6*t+3:6:6*2*t);
        u.S2.WS = w_opt(6*t+4:6:6*2*t);
        u.S2.GS = w_opt(6*t+5:6:6*2*t);
        u.S2.R = w_opt(6*t+6:6:6*2*t);

```

```
196         u.S3.WD = w_opt(6*2*t+1:6:6*3*t);
197         u.S3.GD = w_opt(6*2*t+2:6:6*3*t);
198         u.S3.SD = w_opt(6*2*t+3:6:6*3*t);
199         u.S3.WS = w_opt(6*2*t+4:6:6*3*t);
200         u.S3.GS = w_opt(6*2*t+5:6:6*3*t);
201         u.S3.R = w_opt(6*2*t+6:6:6*3*t);
202 
203         u.S4.WD = w_opt(6*3*t+1:6:6*4*t);
204         u.S4.GD = w_opt(6*3*t+2:6:6*4*t);
205         u.S4.SD = w_opt(6*3*t+3:6:6*4*t);
206         u.S4.WS = w_opt(6*3*t+4:6:6*4*t);
207         u.S4.GS = w_opt(6*3*t+5:6:6*4*t);
208         u.S4.R = w_opt(6*3*t+6:6:6*4*t);
209 
210         u.S5.WD = w_opt(6*4*t+1:6:6*5*t);
211         u.S5.GD = w_opt(6*4*t+2:6:6*5*t);
212         u.S5.SD = w_opt(6*4*t+3:6:6*5*t);
213         u.S5.WS = w_opt(6*4*t+4:6:6*5*t);
214         u.S5.GS = w_opt(6*4*t+5:6:6*5*t);
215         u.S5.R = w_opt(6*4*t+6:6:6*5*t);
216 
217         u.S6.WD = w_opt(6*5*t+1:6:6*6*t);
218         u.S6.GD = w_opt(6*5*t+2:6:6*6*t);
219         u.S6.SD = w_opt(6*5*t+3:6:6*6*t);
220         u.S6.WS = w_opt(6*5*t+4:6:6*6*t);
221         u.S6.GS = w_opt(6*5*t+5:6:6*6*t);
222         u.S6.R = w_opt(6*5*t+6:6:6*6*t);
223 
224         u.S7.WD = w_opt(6*6*t+1:6:6*7*t);
225         u.S7.GD = w_opt(6*6*t+2:6:6*7*t);
226         u.S7.SD = w_opt(6*6*t+3:6:6*7*t);
227         u.S7.WS = w_opt(6*6*t+4:6:6*7*t);
228         u.S7.GS = w_opt(6*6*t+5:6:6*7*t);
229         u.S7.R = w_opt(6*6*t+6:6:6*7*t);
230 
231         u.S8.WD = w_opt(6*7*t+1:6:6*8*t);
232         u.S8.GD = w_opt(6*7*t+2:6:6*8*t);
233         u.S8.SD = w_opt(6*7*t+3:6:6*8*t);
234         u.S8.WS = w_opt(6*7*t+4:6:6*8*t);
235         u.S8.GS = w_opt(6*7*t+5:6:6*8*t);
236         u.S8.R = w_opt(6*7*t+6:6:6*8*t);
237 
238         u.S9.WD = w_opt(6*8*t+1:6:6*9*t);
239         u.S9.GD = w_opt(6*8*t+2:6:6*9*t);
240         u.S9.SD = w_opt(6*8*t+3:6:6*9*t);
241         u.S9.WS = w_opt(6*8*t+4:6:6*9*t);
242         u.S9.GS = w_opt(6*8*t+5:6:6*9*t);
243         u.S9.R = w_opt(6*8*t+6:6:6*9*t);
244 
245         % Take an action
246         uk = [u.S1.WD(1);u.S1.GD(1);u.S1.SD(1);u.S1.WS(1);u.S1.GS(1)];
247 
248         % Exogenous information realization
249         S.P = sample_path.P(w,k);
250         S.D = sample_path.D(w,k);
251 
252         %Simulating the plant behavior during dt
```

```
        S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;

        %For plotting
        xSim = [xSim, S.R];
        uSim = [uSim, uk];

end

Cost1 = sum(sample_path.P(w,:).*(uSim(2,:) + uSim(5,:)));
Cost2 = [Cost2; Cost1];
end
Cost = mean(Cost2);

for i = 1:size(Cost2,1)
Avg_cost_cum(i) = mean(Cost2(1:i));
end

name = strcat(date,'_Pscn_MPC_',num2str(w),'itr_',num2str(t),'_hz');
save(name)
```

P04_D_scn_MPC_main.m

```
close all
clear all
clc

%% Load
addpath('../Exogenous_information')
addpath('../P01_PFA')
load('sample_path.mat')

%% Optimum calculation

%initial setup
S0 = S0();
S = S0.S;

%Scenario number
n_scenario = size(sample_path.D,1);
%total time horizon
N = size(sample_path.D,2);

%Open loop optimization time horizon
t = 20;

%Extention of sample path
sample_path.E = [sample_path.E sample_path.E(1,1:t+1)];

%Demand model without stochastic value used for prediction
z = 0:N-1;
D_mean = 200;
amplitude = 50;
Dmin = 100;
Dmax = 300;
D = D_mean + amplitude * sin(2*2*pi*z/(N-1) -pi/2);
D = min(max(D,Dmin),Dmax);
D = [S.D D D(1,1:t+1)];
```

```
36
37 %% setting
38
39 %Price_scneario parameters.
40 Theta_up = 20;
41 Theta_mid = 0;
42 Theta_down = -20;
43
44 % %% simulation
45 Cost2 =[];
46 for w = 1:1%n_scenario
47 % initial states
48 S = S0.S;
49
50 %Plotting Arrays
51 xSim = [];
52 uSim = [];
53 timeSim = [];
54
55 for k = 1:N
56
57         %exogenous information model
58         Price_horizon =  normrnd(0,50,1,t) + S.P;
59         Demand_horizon = [S.D D(1,k+1:k+t-1)];
60         Energy_horizon = sample_path.E(1,k:k+t-1);
61
62 %% Demand SCENARIOS : 9 trees.
63         dt.S1_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_up) (Demand_horizon(3)+2*Theta_up) (Demand_horizon(4:t)+2*
    Theta_up)];
64         dt.S2_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_up) (Demand_horizon(3)+Theta_up+Theta_mid) (Demand_horizon(4:t)+
    Theta_up+Theta_mid)];
65         dt.S3_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_up) (Demand_horizon(3)+Theta_up+Theta_down) (Demand_horizon(4:t)
    +Theta_up+Theta_down)];
66         dt.S4_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_mid) (Demand_horizon(3)+Theta_mid+Theta_up) (Demand_horizon(4:t)
    +Theta_mid+Theta_up)];
67         dt.S5_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_mid) (Demand_horizon(3)+Theta_mid+Theta_mid) (Demand_horizon(4:t
    )+Theta_mid+Theta_mid)];
68         dt.S6_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_mid) (Demand_horizon(3)+Theta_mid+Theta_down) (Demand_horizon(4:
    t)+Theta_mid+Theta_down)];
69         dt.S7_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_down) (Demand_horizon(3)+Theta_down+Theta_up) (Demand_horizon(4:
    t)+Theta_down+Theta_up)];
70         dt.S8_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_down) (Demand_horizon(3)+Theta_down+Theta_mid) (Demand_horizon
    (4:t)+Theta_down+Theta_mid)];
71         dt.S9_Demand_horizon = [Demand_horizon(1) (Demand_horizon(2)+
    Theta_down) (Demand_horizon(3)+Theta_down+Theta_down) (Demand_horizon
    (4:t)+Theta_down+Theta_down)];
72
73 %% Process constraint : Same for all senarios.
74         %%% EQUALITY CONSTRAINT
```

```
            %Demand constraint => when eqaulity : infeasible => inequality
    .
        S_Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        S_Beq_D = [dt.S1_Demand_horizon'
            dt.S2_Demand_horizon'
            dt.S3_Demand_horizon'
            dt.S4_Demand_horizon'
            dt.S5_Demand_horizon'
            dt.S6_Demand_horizon'
            dt.S7_Demand_horizon'
            dt.S8_Demand_horizon'
            dt.S9_Demand_horizon'];

            %Transition function
        S_Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(
    eye(t),[0 0 0 0 0 1]);
        S_Beq_T = S.R*ones(t,1);

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        S_A_1 = kron(eye(t), [0 0 0 1 1 1]);
        S_B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
        S_A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        S_B_2 = zeros(t,1);
            %Charing rate limitation
        S_A_3 = kron(eye(t),[0 0 0 1 1 0]);
        S_B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        S_A_4 = kron(eye(t),[0 0 1 0 0 0]);
        S_B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
        S_A_5 = kron(eye(t),[1 0 0 1 0 0]);
        S_B_5 = Energy_horizon';
        % A&B integration
        S_A = [S_A_1; S_A_2; S_A_3; S_A_4; S_A_5];
        S_B = [S_B_1; S_B_2; S_B_3; S_B_4; S_B_5];

        S_f = kron(Price_horizon,[-0.02 1 0 -0.01 1 0]);

%% Non-anticipativity constraints
nas = [1 1 1 1 1 0];
naz = zeros(1,6);

na_A1 = [nas kron(ones(1,t-1),naz) -nas kron(ones(1,t-1),naz) kron(ones(1,
    t*7), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*1), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*6), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*2), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*5), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*3), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*4), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*4), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*3), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*5), naz) -nas kron(ones(1,t-1)
    ,naz) kron(ones(1,t*2), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*6), naz) -nas kron(ones(1,t-1)
```

```
    ,naz) kron(ones(1,t*1), naz)
    nas kron(ones(1,t-1),naz) kron(ones(1,t*7), naz) -nas kron(ones(1,t-1)
    ,naz) ];

na_A2 = [naz nas kron(ones(1,t-2),naz) naz -nas kron(ones(1,t-2),naz) kron
    (ones(1,t*7),naz)
        naz nas kron(ones(1,t-2),naz) kron(ones(1,t),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t*6),naz)
        kron(ones(1,t*3),naz) naz nas kron(ones(1,t-2),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t*4),naz)
        kron(ones(1,t*3),naz) naz nas kron(ones(1,t-2),naz) kron(ones(1,t)
    ,naz) naz -nas kron(ones(1,t-2),naz) kron(ones(1,t*3),naz)
        kron(ones(1,t*6),naz) naz nas kron(ones(1,t-2),naz) naz -nas kron(
    ones(1,t-2),naz) kron(ones(1,t),naz)
        kron(ones(1,t*6),naz) naz nas kron(ones(1,t-2),naz) kron(ones(1,t)
    ,naz) naz -nas kron(ones(1,t-2),naz)];


na_Aeq = [na_A1; na_A2];
na_Beq = zeros(size(na_Aeq,1),1);

%% Integration of linprog factors
        Aeq_T = kron(eye(9),S_Aeq_T);
        Aeq_D = kron(eye(9),S_Aeq_D);
        Aeq = [Aeq_D; Aeq_T; na_Aeq];

        Beq = [S_Beq_D; kron(ones(9,1),S_Beq_T)];
        Beq = [Beq; na_Beq];

        A = kron(eye(9),S_A);
        B = kron(ones(9,1),S_B);
        f = kron(ones(1,9),S_f);
        lb = zeros(6*9*t,1);
        ub = ones(6*9*t,1)*inf;

%%        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.S1.WD = w_opt(1:6:6*t);
        u.S1.GD = w_opt(2:6:6*t);
        u.S1.SD = w_opt(3:6:6*t);
        u.S1.WS = w_opt(4:6:6*t);
        u.S1.GS = w_opt(5:6:6*t);
        u.S1.R = w_opt(6:6:6*t);

        u.S2.WD = w_opt(6*t+1:6:6*2*t);
        u.S2.GD = w_opt(6*t+2:6:6*2*t);
        u.S2.SD = w_opt(6*t+3:6:6*2*t);
        u.S2.WS = w_opt(6*t+4:6:6*2*t);
        u.S2.GS = w_opt(6*t+5:6:6*2*t);
        u.S2.R = w_opt(6*t+6:6:6*2*t);

        u.S3.WD = w_opt(6*2*t+1:6:6*3*t);
        u.S3.GD = w_opt(6*2*t+2:6:6*3*t);
        u.S3.SD = w_opt(6*2*t+3:6:6*3*t);
        u.S3.WS = w_opt(6*2*t+4:6:6*3*t);
```

```
        u.S3.GS = w_opt(6*2*t+5:6:6*3*t);
        u.S3.R = w_opt(6*2*t+6:6:6*3*t);

        u.S4.WD = w_opt(6*3*t+1:6:6*4*t);
        u.S4.GD = w_opt(6*3*t+2:6:6*4*t);
        u.S4.SD = w_opt(6*3*t+3:6:6*4*t);
        u.S4.WS = w_opt(6*3*t+4:6:6*4*t);
        u.S4.GS = w_opt(6*3*t+5:6:6*4*t);
        u.S4.R = w_opt(6*3*t+6:6:6*4*t);

        u.S5.WD = w_opt(6*4*t+1:6:6*5*t);
        u.S5.GD = w_opt(6*4*t+2:6:6*5*t);
        u.S5.SD = w_opt(6*4*t+3:6:6*5*t);
        u.S5.WS = w_opt(6*4*t+4:6:6*5*t);
        u.S5.GS = w_opt(6*4*t+5:6:6*5*t);
        u.S5.R = w_opt(6*4*t+6:6:6*5*t);

        u.S6.WD = w_opt(6*5*t+1:6:6*6*t);
        u.S6.GD = w_opt(6*5*t+2:6:6*6*t);
        u.S6.SD = w_opt(6*5*t+3:6:6*6*t);
        u.S6.WS = w_opt(6*5*t+4:6:6*6*t);
        u.S6.GS = w_opt(6*5*t+5:6:6*6*t);
        u.S6.R = w_opt(6*5*t+6:6:6*6*t);

        u.S7.WD = w_opt(6*6*t+1:6:6*7*t);
        u.S7.GD = w_opt(6*6*t+2:6:6*7*t);
        u.S7.SD = w_opt(6*6*t+3:6:6*7*t);
        u.S7.WS = w_opt(6*6*t+4:6:6*7*t);
        u.S7.GS = w_opt(6*6*t+5:6:6*7*t);
        u.S7.R = w_opt(6*6*t+6:6:6*7*t);

        u.S8.WD = w_opt(6*7*t+1:6:6*8*t);
        u.S8.GD = w_opt(6*7*t+2:6:6*8*t);
        u.S8.SD = w_opt(6*7*t+3:6:6*8*t);
        u.S8.WS = w_opt(6*7*t+4:6:6*8*t);
        u.S8.GS = w_opt(6*7*t+5:6:6*8*t);
        u.S8.R = w_opt(6*7*t+6:6:6*8*t);

        u.S9.WD = w_opt(6*8*t+1:6:6*9*t);
        u.S9.GD = w_opt(6*8*t+2:6:6*9*t);
        u.S9.SD = w_opt(6*8*t+3:6:6*9*t);
        u.S9.WS = w_opt(6*8*t+4:6:6*9*t);
        u.S9.GS = w_opt(6*8*t+5:6:6*9*t);
        u.S9.R = w_opt(6*8*t+6:6:6*9*t);

        % Take an action
        uk = [u.S1.WD(1);u.S1.GD(1);u.S1.SD(1);u.S1.WS(1);u.S1.GS(1)];

        S.D-uk(1)-uk(2)-uk(3)

        % Exogenous information realization
        S.P = sample_path.P(w,k);
        S.D = sample_path.D(w,k);

        %Simulating the plant behavior during dt
        S.R = S.R + [0 0 -1 S0.eff_c S0.eff_c]*uk;

```

```
        %Data storing
        xSim = [xSim, S.R];
        uSim = [uSim, uk];

end

Cost1 = sum(sample_path.P(w,:).*(uSim(2,:) + uSim(5,:)));
Cost2 = [Cost2; Cost1];
end
Cost = mean(Cost2);

for i = 1:size(Cost2,1)
Avg_cost_cum(i) = mean(Cost2(1:i));
end

name = strcat(date,'_Dscn_MPC_',num2str(w),'itr_',num2str(t),'_hz');
save(name)
```

A.8 The deterministic optimal solution(perfect forecast)

The file `P05_Opt_main.m` runs the simulation of the deterministic optimal policy under the assumption of the perfect forecast.

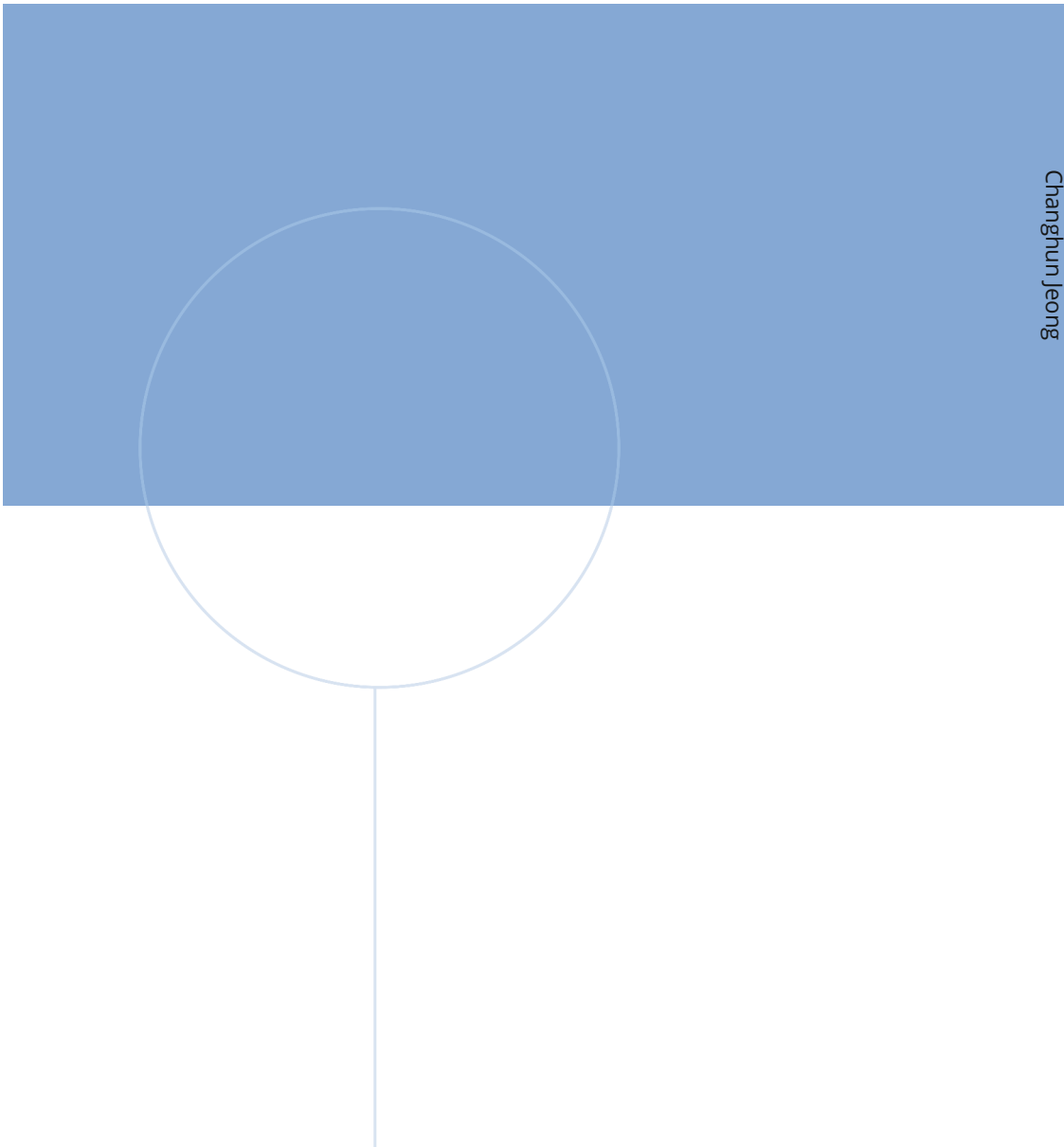
`P05_Opt_main.m`

```
close all
clear all
clc


%% Load
addpath('../Exogenous_information')
addpath('../P01_PFA')
load('sample_path.mat')
plot_settings

%% Optimum calculation

%initial setup
S0 = S0();
S=S0.S;
%scenario number
n_scenario = size(sample_path.D,1);

%Extention of sample path
sample_path.E = [S.E sample_path.E];
sample_path.D = [S.D*ones(n_scenario,1) sample_path.D];
sample_path.P = [S.P*ones(n_scenario,1) sample_path.P];

%time horizon
t = size(sample_path.D,2);

tic
T_Cost = [];
for w = 1:n_scenario
```

```
    w
%Choosing the price policy
Price_horizon = sample_path.P(w,:);
Demand_horizon = sample_path.D(w,:);
Energy_horizon = sample_path.E(1,:);

%optimize the system
        %%% EQUALITY CONSTRAINT
            %Demand constraint
        Aeq_D = kron(eye(t),[1 1 S0.eff_d 0 0 0]);
        Beq_D = Demand_horizon';
            %Transition function
        Aeq_T = kron(tril(ones(t)),[0 0 1 -S0.eff_c -S0.eff_c 0])+kron(eye
    (t),[0 0 0 0 0 1]);
        Beq_T = S.R*ones(t,1);
        % Aeq & Beq integration
        Aeq = [Aeq_D;Aeq_T];
        Beq = [Beq_D;Beq_T];

        %%% INEQUALITY CONSTRAINTS : Ax<=b
            %Charging limitation : WS+GS+S <=R.max
        A_1 = kron(eye(t), [0 0 0 1 1 1]);
        B_1 = S0.R_max*ones(t,1);
            %Discharging limitation : Q,SD < amount in Storage
        A_2 = kron(eye(t),[0 0 1 0 0 -1]);
        B_2 = zeros(t,1);
            %Charing rate limitation
        A_3 = kron(eye(t),[0 0 0 1 1 0]);
        B_3 = S0.rate_c*ones(t,1);
            %Discharing rate limitation
        A_4 = kron(eye(t),[0 0 1 0 0 0]);
        B_4 = S0.rate_d*ones(t,1);
            %Waste incinerator : Heat supply amount
        A_5 = kron(eye(t),[1 0 0 1 0 0]);
        B_5 = Energy_horizon';
        % A&B integration
        A = [A_1;A_2;A_3;A_4;A_5];
        B = [B_1;B_2;B_3;B_4;B_5];

        %%% Objective function
        f = kron(Price_horizon,[0 1 0 0 1 0]);

        %%% Boundaries - lower and upper
        lb = zeros(6*t,1);
        ub = ones(6*t,1)*inf;

        %%% Optimization
        w_opt = linprog(f,A,B,Aeq,Beq,lb,ub);

        % Data store from openloop optimization
        u.WD = w_opt(1:6:6*t);
        u.GD = w_opt(2:6:6*t);
        u.SD = w_opt(3:6:6*t);
        u.WS = w_opt(4:6:6*t);
        u.GS = w_opt(5:6:6*t);
        x.R = w_opt(6:6:6*t);

```

```
        M_Cost = sum((u.GD+u.GS).*Price_horizon');

T_Cost = [T_Cost; M_Cost];
end
Avg_Cost = mean(T_Cost);

for i = 1:size(T_Cost,1)
Avg_cost_cum(i) = mean(T_Cost(1:i));
end

name = strcat(date,'_OPT_',num2str(w),'itr_',num2str(t),'_hz');
save(name)

%% Plot
Line_thick = 1.1
Font = 14;

figure()
subplot(5,1,1)
stairs(1:t,u.WD,'LineWidth',Line_thick)
%legend('$Q_{WD}$')
axis([0 t 0 300])
%xlabel('Time period, $t$')
ylabel('$Q^{WD}$ [MWh]','FontSize',Font)
set(gca,'FontSize',Font)


subplot(5,1,2)
stairs(1:t,u.GD,'LineWidth',Line_thick)
%legend('$Q_{GD}$')
axis([0 t 0 300])
%xlabel('Time period, $t$')
ylabel('$Q^{GD}$ [MWh]','FontSize',Font)
set(gca,'FontSize',Font)


subplot(5,1,3)
stairs(1:t,u.SD,'LineWidth',Line_thick)
%legend('$Q_{GD}$')
axis([0 t 0 300])
%xlabel('Time period, $t$')
ylabel('$Q^{SD}$ [MWh]','FontSize',Font)
set(gca,'FontSize',Font)

subplot(5,1,4)
stairs(1:t,u.WS,'LineWidth',Line_thick)
%legend('$Q_{WS}$')
axis([0 t 0 300])
%xlabel('Time period, $t$')
ylabel('$Q^{WS}$ [MWh]','FontSize',Font)
set(gca,'FontSize',Font)

subplot(5,1,5)
stairs(1:t,u.GS,'LineWidth',Line_thick)
%legend('$Q_{GS}$')
axis([0 t 0 300])
xlabel('Time period, $t$','FontSize',Font)
```

```
144 set(gca,'FontSize',Font)
145
146 ylabel('$Q^{GS}$ [MWh]','FontSize',Font)
147
148 Font = 30;
149 figure()
150 stairs(1:t,x.R,'LineWidth',1.5)
151 %legend('$Q_{GS}$')
152 axis([0 t 0 S0.R_max])
153 xlabel('Time period, $t$','FontSize',Font)
154 ylabel('$Q^{Storage}$ [MWh]','FontSize',Font)
155 set(gca,'FontSize',Font)
```



Changhun Jeong

NTNU
Norwegian University of
Science and Technology