NTNU
Kunnskap for en bedre verden

Department of Chemical Engineering

TKP4580 - Chemical Engineering Specialization Project

Optimal control of energy storage systems

Author:
Alireza Mirzaei

Supervisors:
Sigurd Skogestad — David Pérez Piñeiro

December, 2020

Abstract

Fossil fuel consumption and global warming are among the most pressing problems in present times. These issues have become a focal point for scientific engineering research. In light of this, various studies are being undertaken to develop solutions to reduce conventional energy consumption and establish renewable sources as the key to sustainable energy production. This, in turn, has necessitated the development of advanced technologies to enhance the thermal energy performance of the buildings. Thermal energy storage, also known as TES, is an example of such technology.

Energy storage problems are highly stochastic in nature. Not only the generation of the energy from the resources is uncertain, but also the energy prices are not constant. Therefore finding and implementing true control and optimization policies which can decrease the cost and meet the demand is essential. In this project, we studied the optimal control of borehole energy storage system. The idea comes from the fact that due to lower variations of the underground temperature compared to air, we can use it as a suitable energy storage system. Coupling it with a heat pump, make it suitable to be used in parallel with a conventional heater and cooler to provide energy for a typical building.

With the objective of minimizing the annual energy costs, we developed a Certainty Equivalence Model Predictive Control (CEMPC) algorithm which is suitable for minimization of the total cost for thermal energy consumption. This algorithm helps us to know that for example how much heat energy is better to be taken from borehole, heater or both in summer days, or in an inverse route how much cooling might be provided from borehole and cooler on summer days. We designed some prediction set variations for CEMPC and calculated the performance of these variations by applying a large number of simulation sample paths to identify how much we are far from the real optimal value. The performance of this algorithm can be compared with other control policies in further stages of the project.

Acknowledgements

This specialization project did not reached end with kind supports and help of many individual. I would like to extend my thanks to all of them.

I highly appreciate chemical process system engineering group and my supervisor, Professor Sigurd Skogestad, to provided me with enough knowledge and relevant prerequisites needed to complete this project.

After that, I would like to express my special gratitude to my co-supervisor David Pérez Piñeiro who accompanied me till end of this project and spent a lot of time and endeavor for different part of the project. Without his coordination, it was not possible to finish this project on time.

I am highly indebted to Jose Otavio Assumpcao Matias for his guidance on coding and troubleshooting of the program. My thanks and appreciations also go to classmates who have willingly helped me out with their abilities.

Contents

Nomenclature

Constants Parameters

λ	heat transfer coefficient between borehole and underground	$KWh/^{\circ}C$
C_p	Water heat capacity	$KWh/Kg^{\circ}C$
COP	Heat pump Coefficient of Performance	$-$
D	Diameter of pipes in borehole	$meter$
L	Length of the pipes in borehole	$meter$
m	Water mass in borehole	Kg
n	Number of pipes in borehole	$-$
Q_{max}	Maximum allowable energy flow between demand and borehole	KWh

Model Variables

A_d	Domain of demand change	KWh
A^{inf}	Domain of underground temperature change	$^{\circ}C$
C	Stage cost at time t	NOK
D_t	Demand at time t	KWh
$\hat{D}_t$	change of demand between time t and t+1	KWh
D^{std}_{avg}	Standard deviation of demand noise	NOK/KWh
P^c_t	Cooler operating price at time t	NOK/KWh
P^h_t	Heater operating price at time t	NOK/KWh
P^{hh}_t	Heat pump operating price at time t	NOK/KWh
$\hat{P}^c_t$	change of cooler operating price between time t and t+1	NOK/KWh
$\hat{P}^h_t$	change of heater operating price between time t and t+1	NOK/KWh
$\hat{P}^{hh}_t$	change of heat pump operating price between time t and t+1	NOK/KWh
P^c_{avg}	average cooler operating price	NOK/KWh
P^h_{avg}	average heater operating price	NOK/KWh
P^{hh}_{avg}	average heat pump operating price	NOK/KWh
P^{std}_{avg}	Standard deviation from average price	NOK/KWh
Q^c_t	Energy flow between cooler and building at time t	KWh
Q^h_t	Energy flow between heater and building at time t	KWh
Q^b_t	Energy flow between borehole and building at time t	KWh
S_t	Set of state variables at time t	$-$
T^b_t	Borehole temperature at time t	$^{\circ}C$
T^{inf}_t	Underground temperature at time t	$^{\circ}C$
$\hat{T}^{inf}_t$	Change of underground temperature between time t and t+1	$^{\circ}C$
T^{inf}_{max}	Maximum underground temperature	$^{\circ}C$
T^{std}_{avg}	Standard deviation of underground temperature noise	NOK/KWh
X_t	Set of decision variables at time t	$-$

List of Figures

List of Tables

1 Introduction

1.1 Importance of Renewable Energy and the Storage

In these years of global market uncertainty, an important matter is that the world needs energy and the global trend shows that demand of energy continuously increasing in order to support economic and social improvement and build a better life standards, but providing this energy around the world should be taken with responsibility and commitment to developing and using our resources more efficiently. Today, all of us are committed to protecting both people and the environment and making positive economic contributions.

Here the problem is not only that the conventional energy resource such as fossil fuels are depleting as fast as possible, but also there are increasing concerns about the environmental negative effects of hydrocarbons as the primary source of energy. It is said that electricity production is our first source of greenhouse gases, more than all of our driving and flying combined. Electricity generation has been reported as the second leading cause of industrial air pollution in the U.S. Most of our electricity comes from coal, nuclear, and other non-renewable power plants. Producing energy from these resources takes a severe toll on our environment, polluting our air, land, and water. Using renewable and clean sources of energy is one of the most important measures we can take to reduce these adverse impacts on our planet. Renewable energy is the energy that is derived from natural resources that replenish themselves over a period of time without depleting the Earth's resources. These resources also have the benefit of being abundant, available in some capacity nearly everywhere, and they cause little, if any, environmental damage. Energy from the sun, wind, and thermal energy stored in the Earth's crust are examples. Clean energy can reduce harmful smog, toxic buildups in our air and water, and the impacts caused by coal mining and gas extraction. Renewable energy sources can also be used to produce electricity with fewer environmental impacts. It is possible to make electricity from renewable energy sources without producing carbon dioxide (CO_2), the leading cause of global climate change.

Alongside with gradual technical improvement in equipment and installation of renewable energy, the public intention to replacing it with fossil fuels increased within recent years. The strongest motivation over renewable sources is that renewable energy is nearly free of charge compared to conventional one. In many cases such as solar cells it is also possible to sell extra amount of produced electricity to the grid.

On the other hand, replacing fossil-fuel infrastructure or conventional electricity with renewable sources will take time because applying these energy has always been associated with technical problems. One of the most important issues with these kinds of energy is unreliability. This comes from the point that the most renewable resources are out of our control. In case of wind towers or solar cells, we are uncertain that how we have to manage to use the produced energy, because we do not know how much energy we are able to produce over the next days. An effective approach for solving this issue is to use a battery enabling us to store the energy and using it in more appropriate time later. In energy storage domain, the essence of some optimal control policies that help us to reduce energy consumption cost is highly needed from economical point of view because most of the time the prices are fluctuating. Moreover due to uncertain nature of renewable energy availability at different times, such policies should ensure us that we can always provide required energy from both conventional energy installation and these renewable resources.

1.2 Contributions of the Project

In this project, we develop a stochastic optimization approach for a specific energy storage problem. We demonstrate that how it is possible to economically feed a building with required energy by applying a borehole in parallel with conventional heater and cooler when the energy price experiences random variation and the storage device is subject to stochastic noise.

We evaluate the performance of Certain Equivalence of Model Predictive Control (CEMPC) by defining some problem variation and demonstrate under which condition this policy can potentially keep the annual cost of a random time-varying energy storage problem near to its real optimized cost.

1.3 Organization of the Report

We start by defining the base model, and then describe how we can formulate the problem by manipulating the nature of the underlying data. We then discuss the policy which is used to minimize annual energy cost for the problem and compare it with the real optimum point with the perfect forecast.

We start by reviewing the literature in chapter 2 to present other researches about borehole energy storage systems, uncertainty modeling and Model Predictive Control algorithm. In chapter 3 we present a base model of the energy storage problem and demonstrate different variables of the system. Then we provide the formulation of the cost and show how the stochastic variables could be formulated. The algorithms used in this report and the benchmark policy they are compared against are presented in Chapter 4. We implement CEMPC policy in the context of problem and provide performance of these algorithm in terms of numerical results and profiles of energy flow in Chapter 5. Our findings are analyzed in Chapter 6, and conclusion are presented in Chapter 7.

2 Literature Review

Optimal Control Energy storage under stochastic condition has been interested over recent years and different case studies investigated and appropriate control policies developed; however, according to author knowledge, few number of research have been carried out on optimal control of borehole systems. In this chapter, we provide a literature review of borehole energy storage systems. Afterwards, we continue to review the different studies of Model Predictive Control techniques on optimization of different energy storage problems, and then we explore strategies applied to model uncertainty which have been implemented in previous works.

2.1 Borehole Energy Storage System

Borehole as an efficient tool for recovery of geothermal energy storage purposes have been increasingly trended over recent years. Welsch et al. 2016 shows that there is an optimal borehole pipes arrangement where the highest storage efficiencies and the highest heat extraction rates are achieved. He also demonstrates that under the very simplified operating procedure and subsurface conditions, storage efficiencies of up to 83 percent are reached and it is also possible to increase this efficiency even further by adjusting supply temperatures for the heating system or increasing the loading temperature of the storage. Schulte et al. 2015 designs an approach for the simulation and optimization of borehole thermal energy storage systems using software tools. His methodology successfully determines the ideal size of the thermal energy storage and estimates the optimal number and length of borehole heat exchangers with regard to a specific annual heat demand. Furthermore, they indicates that borehole thermal energy storage systems only operate efficiently in large-scale applications. Bär et al. 2015 suggests that coupling of different renewable energy sources, i.e. solar thermal and geothermal – with already existing district heating systems – e.g. combined (biofuel-driven) heat and power stations (CHP) is a very promising approach to cover the heating demand of renovated or old buildings. He justified that how he design, completion and performance of medium sized borehole systems are strongly depending on the knowledge about the subsurface and the energy flows between the heat source, the storage system and the building.

Spitler 2005 expresses the importance of a control algorithm that can control the borehole energy storage system based on a long term,. Cui et al. 2008 and Z. Deng 2005 provides some methods in modeling for ground-source heat pump, but no model based control algorithms have been presented in these papers. Rink has provided with some papers about heat pumps control extended with a battery subjecting to variable energy prices, but his works does not cover long term energy storage or the coupling of an underground heat storage system to an heat pump system(Rink et al. 1988 and Rink 1994). Zaheeruddin et al. 1988 describes sub-optimal control algorithms of heat-pump and heat storage systems based on short time term. In his article, energy prices with random variations are taken into account, too. A control algorithm for hybrid thermal energy storage combined with a short term weather forecast is proposed in LeBreux et al. 2006. This control algorithm optimizes the short term policy of the system.

2.2 Model Predictive Control

Model Predictive Control (CEMPC) is one of the solutions that have been studied in in energy storage systems. In particular, Model Predictive Control has been applied to maximize the efficiency of heating and cooling systems for large buildings as in Ma et al. 2012. Arnold and Andersson 2011 shows that a Model Predictive Control can be used to manage a storage hub with both battery and hot water storage devices to minimize the cost of satisfying loads from a collection of households in the presence of unreliable renewable sources, stochastic electricity and natural gas prices.

2.3 Modelling Uncertainty

Powell 2019 classifies the strategies for developing the policy functions to four different groups to be applied in sequential decision making problems:

a. Policy function approximations (PFA)

b. Cost function approximations (CFA)

c. Value function approximations (VFA)

d. Direct lookahead policies (DLA)

The first two methods are policies based on policy search, and the 3rd and fourth methods are policies based on look ahead approximations. In the policy search method strategies, an objective function is used to search within a family of functions to find the function that works best. Lookahead approximations are based on approximating the impact of a control action on the future. Various formulations of the energy storage problem have been previously studied, and policies from each of the four classes or some hybrid policies have been applied.

PFA and CFA policies have been investigated in several articles in the energy storage domain. In a short time horizon Warrington et al. 2012 effectively controls storage systems based on predefined function and wind forecast errors. Han and E 2016 applies the same technique to conduct energy flows from wind turbines, battery, and the power grid to meet a time-varying demand. H.P.Simão et al. 2017 uses a deterministic look ahead to control an energy system where battery is managed to handle the interruptions of renewable energy.

For lookahead approximation policies, it is very important to perform a good uncertainty modelling to handle stochastic optimization. There are several articles that addressed uncertainty in their energy storage case studies. Zhou et al. 2018 performed Backward Dynamic Programming (BDP) to find an optimal policy for managing wind energy storage system in presence of price variation and wind unreliability. Backward DP was also tested to find near-optimal solution in Durante et al. 2017, However, with regards to the problems that exact Backward DP have with the large space state energy storage problems, they has not been widely applied in the energy storage domain. Another challenge in direct lookahead policies is model simplification which should be taken carefully. This is essential and have been performed in most of the literature to allow the controller to solve the problem, but this have to keep the simplified model near the reality. In most of the literature, simplifications performed in the modelling of the renewable energies and price processes. For example, Ridder et al. 2011 assumes simple model to describe the dynamics of the underground storage system to be able to implement dynamic programming. He obtains a low root-mean-square value of the prediction error showing that his model is sufficiently good in predicting the temperature calculated with the complex model and have to be used in the controller. Price changes and production of renewable have also been subject to some simplifications in the previous works. For example Jiang et al. (2014), Cheng et al. 2018 and Mokrian et al. (2006) model it as a first order Markov chain in their works.

Deterministic prediction of random exogenous processes have been applied in the energy domain. Sioshansi et al. 2014 studies the benefits of integrating wind turbines and storage devices by utilizing a deterministic lookahead which makes decision based on a mixed integer program with prediction for two-week horizon. Wallace and Fleten 2003 applies deterministic forecasts of wind, solar and loads to manage energy storage. Most of these models are subject to argument in the research community that not take uncertainty into account. Model predictive control policies use rolling forecasts which update at each time step in the base model, but forecast error cannot be controlled in such models.

In this thesis, we design different predictions on stochastic changes in our borehole energy storage model and evaluate their performance by Certain Equivalence Model Predictive Control. We also benchmark these with real optimal value (perfect forecast). Description of our policy is presented in chapter 4.

3 Modeling a Borehole Energy Storage Problem

In this chapter, we provide the basic model of the energy storage problem by defining the state variables, the decision variables, exogenous information, transition function and objective function. As a case study, we will analyze a combined heating and cooling system with a thermal storage device. we also describe the strategies applied to our model in order to simulate the variations entered to the system e.g. random prices.

3.1 Basic Model

In most buildings around 40 percent of the energy consumed to heat and cool the building. One of the ways to lower the electricity energy consumption, operating cost, and in consequence carbon dioxide emission is to heat and cool buildings with an underground thermal energy storage system by applying boreholes. Briefly, such systems consist of a large underground volume, which can be loaded with heat or cold quantities from the building and can hold this energy for some months. An underground storage field typically includes several horizontal and vertical pipes, drilled on the nodes of a square grid and is filled with a water circuit. This construction looks like a big underground container. Based on the system type, the pipes are allowed to exchange thermal energy with the underground or have to exchange as minimum heat energy as possible. The idea of borehole comes from the fact that it is possible to dispatch and store the heat inside the building to the borehole over summer and use this heat in the winter. In inverse route, consuming this energy over winter will cause a lower borehole temperature which is suitable for summer cooling.

In our model, we assumed that a primary borehole network as our storage battery device in parallel with a conventional heater and cooler used to provide thermal energy for the demand. Heater and cooler function is to supply the shortage of the energy when it is not possible to take all the heat or cold from borehole. They are working with electricity and their operating cost is subject to stochastic variations.

The variation in the underground temperature is lower than the air and this make underground more reliable energy source than outside air in heat pumps, however these variations depend to so many factors such as geographical location, weather condition, depth and soil type and it is not possible to derive or predict it directly. Therefore we consider that the underground temperature variation can act like as a stochastic energy loss in the storage system. We will develop it further in the formulation of transition function in 3.1.6. We assume that there is no other loss in our system.

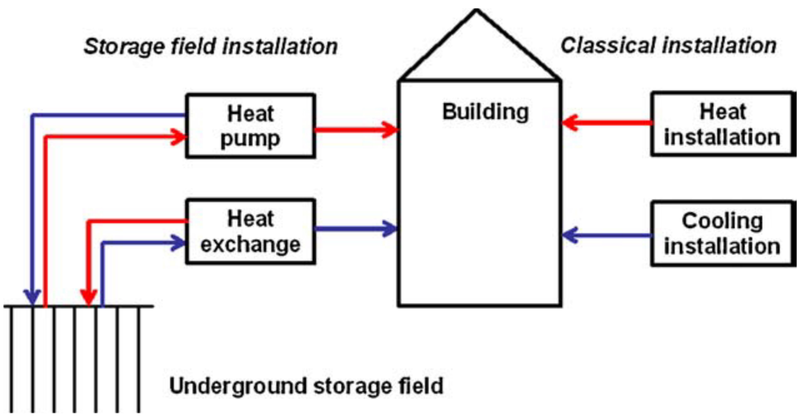

Figure 1: Heating and cooling of a building

We are interested to control the borehole temperature between 0 °C and 12 °C. Within this range we need a heat pump to be coupled with borehole storage system to provide the energy over winter, nevertheless, no building can be heated with such low temperatures. If the temperature goes below 0 °C, not only there is the risk of water freezing, but also heat pump do no work efficiently due to high temperature difference. According to literature setting 12 °C as higher bound helps us to use the cold water directly for cooling purpose, therefore there is no need to heat pump and it will shut down over summer. A typical sketch of our model is presented in Fig. 1.

The objective is to control the flow of energy between borehole, heater and cooler in such a way that we obtain the minimized energy cost. There are four nodes, three decision variables, and five exogenous processes in this model. In Fig. 2. We provide a more conceptual sketch of the model. Demand should be met at any time by the energy flows.

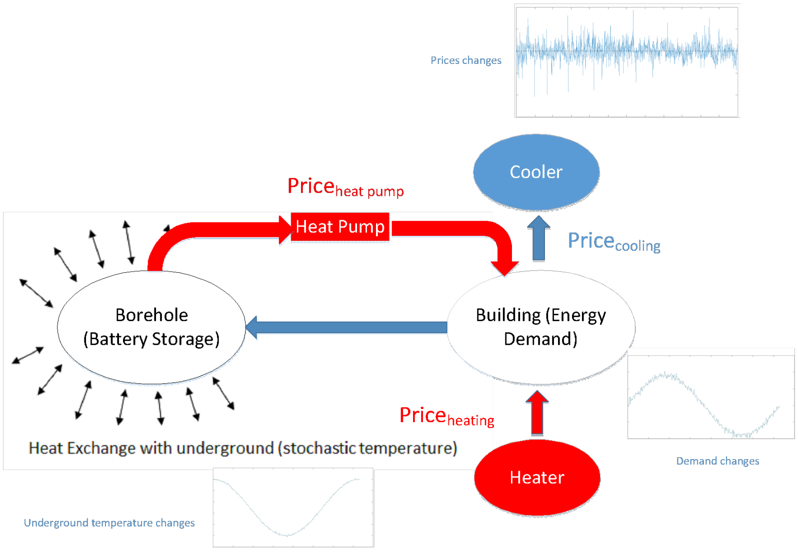

Figure 2: Basic model

3.1.1 Static Parameters

In our model it was assumed that some parameters are constant to allow us better investigate on this model, therefore some quantities have to be assigned to these parameters. First the geometry of the borehole is fixed, therefore the number of holes, their diameter and length have to be specified. Second we supposed that overall heat transfer coefficient from the borehole to the surrounding underground is not affected by dynamic of the system and are estimated to be constant. Third due to physical limitation of the system, the maximum flow of the energy from borehole to the system is limited to a certain value. In addition, our heat pump has a fixed COP. List of the static parameters is as following:

Q_{max}: Maximum allowable energy flow between he borehole and the demand in KWh.

n: The number of the pipes in the borehole system

D: Diameter of the pipes in inches

L: Length of the pipes in meter

m: Mass of the water in the borehole in kg

C_p: Water heat capacity in KWh/Deg. C

For our case study, we consider the following numerical values for the above parameters in Table 1.

Parameter	Value	Unit
R_{max}	30	KWh
n	30	-
D	40	Inches
L	10	meters
m	235619	Kg
C_p	0.001167	KWh/Kg°C

Table 1: Values of the static parameters

3.1.2 State Variables

The state variables include all information we need to know to model the process of the system over time and is sufficient to decide and take actions and calculate final cost. Initial state variables (S_0) specify initial values of these state and should be specified to obtain system performance.

Our thermal energy storage system has six dynamic states (S_t), including the demand, prices, underground temperature and borehole temperature. We assume three prices as Heater Price, cooler price and heat pump price enabling us to analyze the contribution of each of these sources. We also assumed that that there is no need to pay for the direct cooling by borehole and it is free of charge. Therefore the states of the system is given by:

$$S_t = (D_t, {T_t}^b, {T_t}^{\inf}, {P_t}^h, {P_t}^{hh}, {P_t}^c) \tag{1}$$

D_t: Heat demand at day t in KWh.

${T_t}^b$: Borehole temperature at day t in °C.

${T_t}^{inf}$: Underground temperature at day t in °C.

${P_t}^h$: Heating price by heater at day t in NOK/KWh

${P_t}^{hh}$: Heating price by heat pump at day t in NOK/KWh

${P_t}^c$: Cooling price by cooler at day t in NOK/KWh

We present the values of the initial states in Table 2:

Variable	Value	Unit
D_0	0	KWh
${T_0}^b$	11	°C
${T_0}^{inf}$	15	°C
${P_0}^h$	0.9	NOK/KWh
${P_0}^{hh}$	0.5	NOK/KWh
${P_0}^c$	0.6	NOK/KWh

Table 2: Values of the initial states,S_0

3.1.3 Decision Variables

Our decision variables ($Q_{tb,h,c}$) include all energy flow allocations made every day by controller. Heating of the building can be taken from both heater and borehole in our model. Also, cold could be supplied by borehole and installer cooler installed inside the building. Cold is normally considered as inverse of the heat. So direction of cooling flow would be from building to both borehole and cooler. Taking above points to the account, set of decision variables are as following:

$$X_t = ({Q_t}^b, {Q_t}^h, {Q_t}^c) \tag{2}$$

$Q_t{}^b$: The amount of heat and cold which is provided by borehole everyday in KWh/Day

$Q_t{}^h$: The amount of heat which is provided by heater everyday in KWh/Day.

$Q_t{}^c$: The amount of cold which is provided by cooler everyday in KWh/Day.

3.1.4 Constraints

In our model, defined variables and inputs which were specified in previous parts are subject to some constraints as presented here.

Due to maximum amount of water that that can be circulated by a household pump, and assuming 4 °C as approach temperature of inlet and outlet flow of the water to the evaporator of the heat pump, maximum flow of energy from borehole to building, and inversely from building to borehole, is limited to a certain value. Due to this limitation, there is a maximum value for the energy which can be taken from or given to the borehole. Here we assumed that this maximum is 30 KWh. Therefore:

$$0 \leq {Q_t}^b \leq 30KWh \tag{3}$$

As it was mentioned in basic model, we are interested in controlling the borehole temperature between 0 12 °C. Therefore the second constraint is:

$$0°C \leq {T_t}^b \leq 12°C \tag{4}$$

As it is clear, we cannot have the negative values for our inputs, therefore next constraint is that all inputs are positive:

$${Q_t}^b, {Q_t}^h, {Q_t}^c \geq 0 \tag{5}$$

Above energy allocations should be controlled in such a way that can meet the demand every day. Here we assumed that we do not heat and cool the building at same time. We can write this constraint as following formula:

$$D_t = (\frac{COP}{COP-1}{Q_t}^b + {Q_t}^h) * 1\_\{D_t \geq 0\} + ({Q_t}^b + {Q_t}^c) * 1\_\{D_t \prec 0\} \tag{6}$$

Which means that our daily heating demand shall be equal to sum of heating flows from both borehole and heater and our daily cooling demand shall be equal to sum of cooling flows by both borehole and cooler.

3.1.5 Exogenous Information

We let the ω_t be the vector of our exogenous information. These are random variables or disturbances that we do not know anything about them at the decision making time and enter to our model next day. We consider them as the exogenous change of states and we use "hats" to indicate this matter. In our model they are:

$\hat{D}_t$: The amount of demand change between t and t+1 i.e. two consequence days in KWh

$\hat{T}_{\mathrm{t}}{}^{\mathrm{inf}}$: The amount of underground temperature change between t and t+1 i.e. two consequence days in °C.

$\hat{P}_{\mathrm{t}}{}^{\mathrm{h}}$: The amount of heating price change by heater between t and t+1 i.e. two consequence days in NOK/KWh.

$\hat{P}_{\mathrm{t}}{}^{\mathrm{hh}}$: The amount of heaing price change by borehole between t and t+1 i.e. two consequence days in NOK/KWh.

$\hat{P}_{\mathrm{t}}{}^{\mathrm{c}}$: The amount of cooling price change by cooler between t and t+1 i.e. two consequence days in NOK/KWh.

It is very crucial to obtain a good model for probability distribution of the exogenous variables which help us to stay near optimal point in Lookahead Approximations or Value Function Approximation. We will discuss it in chapter 6.

3.1.6 State Transition Function

The transition function $\mathrm{S^M(S_t,\ X_t},\ \omega_{t+1})$ specifies the transition route of the states from t to t+1 and represents the dynamic of the model. In our model, transition function of the states that are subject to direct exogenous information are obtained by following equations:

$$D_{t+1} = D_t + \hat{D}_t \tag{7}$$

$$P^h{}_{t+1} = P^h{}_t + \hat{P}^h_t \tag{8}$$

$$P^{hh}{}_{t+1} = P^{hh}{}_t + \hat{P}^{hh} \tag{9}$$

$$P^c{}_{t+1} = P^c{}_t + \hat{P}^c_t \tag{10}$$

$$T^{\mathrm{inf}}{}_{t+1} = T_t^{\mathrm{inf}} + \hat{T}_t^{\mathrm{inf}} \tag{11}$$

In order to obtain an exogenous information for the borehole temperature, we have to write an energy balance over the borehole:

$$mC_p\frac{dT^b}{dt} = \lambda(\mathrm{T^{inf}\text{-}T}^b) \pm Q^b \tag{12}$$

The sign $\pm$ show the direction of the flow as it can be from borehole to demand in case of heating or inverse in case of cooling. λ represents the heat transfer coefficient of all borehole system with underground. Based on our model assumption, we need a borehole system with low λ to minimize the loss, because we are interested that borehole can store the thermal energy in itself. After discretizing and rearranging above equation we will have the last transition function as:

$$T_{t+1}{}^b = (1 - \frac{\lambda}{mC_p})T_t{}^b + \frac{\lambda}{mc}\mathrm{T}_t{}^{\mathrm{inf}} \pm Q^b \tag{13}$$

3.1.7 Objective Function

Our cost function characterizes our system and helps us to know that how much we are far from the optimal operation. In our model, we have to pay every time we use heater, cooler and heat pump due to their energy consumption.

Therefore our stage can be written as below equation:

$$C(S_t, X_t) = {Q_t}^b {P_t}^h + {Q_t}^{hh} {P_t}^{hh} + {Q_t}^c {P_t}^c \tag{14}$$

And the total cost over a year is equal to:

$$Total\,\mathrm{Cos}\,t = \sum_{t=1}^{365} C(S_t, X_t) \tag{15}$$

We are interested to minimize the annual paid energy cost. Therefore, our objective function is as following formula:

$$\min_{\pi} E[\sum_{t=1}^{365} C(S_t, X_t)|S_0] \tag{16}$$

Our objective is to apply a policy π that minimizes this cost and we have to choose it from $\pi \in \Pi$ which Π is the set of all possible policies that can meet the constraints.

3.2 Exogenous Iinformation

In this section, we present the models of the exogenous information processes. By design, there are four sources of randomness in our case study: the underground temperature (${\mathrm{T_t}}^{\mathrm{inf}}$) which can impact on our borehole temperature by heat exchange and the heating, cooling and heat pump prices. In this section, we will provide with the methods used to model the variations of underground temperature and prices.

3.2.1 Price Model

Electricity prices are stochastic and updated daily, hourly and in some cases even every 5 minutes. Here, we assume that heater, cooler and heat pump prices follow first order Markov chain model. we use following formula and represent the randomness by Guassian noise as:

$${P_t}^{h,hh,c} = \max\left\{ {P_{avg}}^{h,hh,c} + normrnd(0, {P_{avg}}^{Std}) \right\} \tag{17}$$

In the above model, prices fluctuates around an average value every day. These average values are are presented in Table 3.

Variable	Value	Unit
${\mathrm{P_{avg}}}^{\mathrm{h}}$	0.9	NOK/KWh
${\mathrm{P_{avg}}}^{\mathrm{hh}}$	0.7	NOK/KWh
${\mathrm{P_{avg}}}^{\mathrm{c}}$	0.6	NOK/KWh
${\mathrm{P_{avg}}}^{\mathrm{std}}$	0.5	NOK/KWh

Table 3: Average operating prices of heat pump, heater and cooler

Fig. 3 provides a typical profile of prices. It can be seen that how the model of stochastic prices over a year looks like.

3.2.2 Demand Model

We assume that control starts from first day of the winter. Based on experience, we can expect that demand load peaked to a maximum and then decline over the 6 colder months. Alternatively when summer starts, the demand pass zero and becomes negative which means it is needed to cool the building. As we go ahead in summer days, the negative demand increases, reaches a minimum and then returns to its initial value. We characterize this model with a sinusoidal function over the year, and to inject randomness to daily demand we add a Gaussian noise to demand function as following:

$$D_t = -A_d \sin(\frac{2\pi t}{365} + \pi) + normrnd(0, D_{avg}^{Std}) \tag{18}$$

We take 90 KWh for maximum heat and cold which is demanded by the building. Therefore we set following values to the parameters:

$\mathrm{A_d}$: 90 KWh which is typical for a household demand

$\mathrm{D_{avg}}^{\mathrm{std}}$: 5 KWh

Fig. 3 provides that how the stochastic price model over a year looks like.

3.2.3 Underground Temperature Model

Underground temperature and in consequence heat exchange between borehole and underground depends to different factors such as Geographical location, weather variation, depth of installation, soil. A number of quantitative models from geothermal low activity areas (i.e. on stable platforms outside tectonic and volcanically active areas) show that at shallow depths down to a few hundred meters, mean annual surface temperature is the main factor controlling subsurface temperature. But most of the resource agree that it fluctuates between 10 °C and 25 °C. In this model we assumed that this temperature varies in a range between 10-15 °C and in first days of the winter it is equal to its maximum value. As we go ahead through a year it reaches the minimum and returns again to this value after 365 days. These variation can be simulated by a sinusoidal function as following:

$$T_t{}^{\mathrm{inf}} = T_{\mathrm{max}}{}^{\mathrm{inf}} - A^{\mathrm{inf}} \sin(\frac{2\pi t}{365} - \frac{\pi}{2}) + normrnd(0, T_{\mathrm{inf}}{}^{Std}) \tag{19}$$

We set following values to the parameters of above equation:

$\mathrm{T_{max}}^{\mathrm{inf}}$: 285.5 °K (12.5 °C)

$\mathrm{A_{inf}}$: 2.5 °K

$\mathrm{T_{std}}^{\mathrm{inf}}$: 0.05 °K

Fig. 3 shows that how the variation of underground temperature looks like over a year.

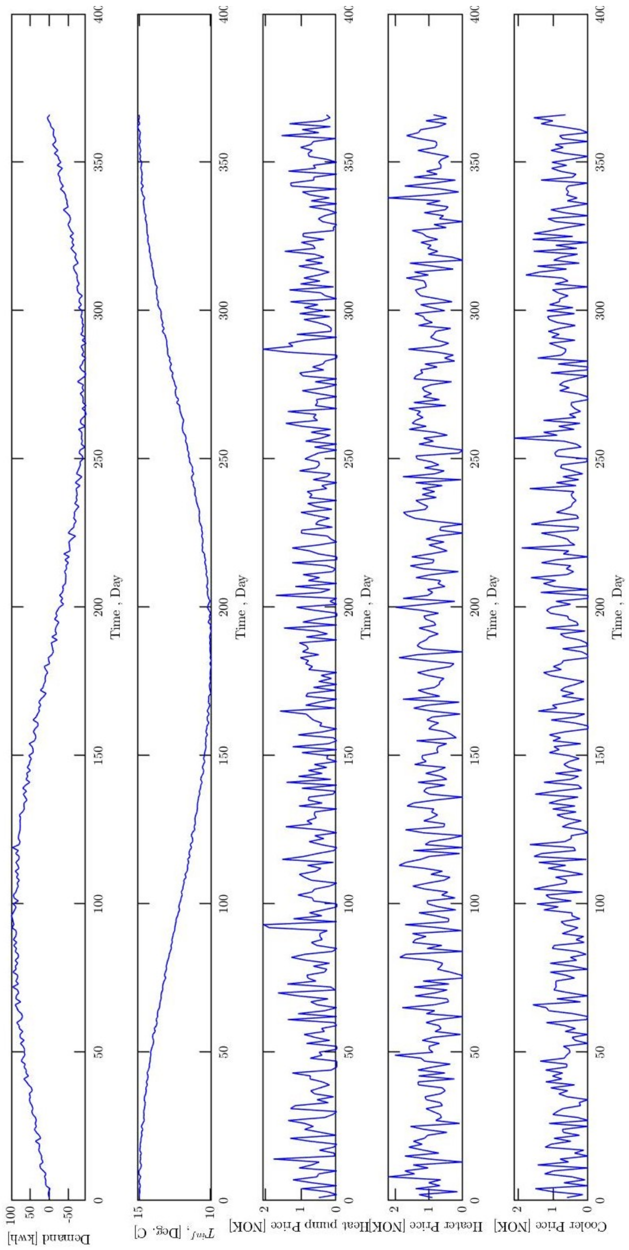

Figure 3: A sample path of exogenous information

3.3 Prediction Set Variations

We modeled our exogenous information as sinusoidal load, sinusoidal underground temperature function and first order Markov chain prices in the previous sections. We want to set a Certainty Equivalence Model Predictive Control algorithm to solve this problem. In order to verify the performance of our policy, we need a prediction set on our exogenous information. Here we build 3 different sets of prediction on exogenous information enabling us to evaluate the strengths of our policy with different predictions. These prediction sets are as following:

• Problem 1: We predict the load and underground temperature with sinusoidal functions but without randomness. Prediction of prices were first order Markov chain which leads to different values from the real prices but with same average and standard deviation.

• Problem 2: We predict the load and underground temperature with sinusoidal functions but without randomness. Prices were predicted to be constant same as mean value of the real prices.

- Problem 3: We predict the load with sinusoidal function but without randomness and underground temperature with constant value of 15 °C. Prediction of prices were first order Markov chain which leads to different values from the real prices but with same average and standard deviation.

4 Designing Policy

In this chapter we present the policies that are designed to allocate energy flows as manipulated variables in our case study. As mentioned before, Powell 2019 classified four different approaches for solving the stochastic optimization problem with sequential decision making as: policy function approximations (PFA), cost function approximations (CFA), value function approximations (VFA), and direct lookahead policies (DLA). Here we will take the Certain Equivalence Model Predictive Control policy which is a kind of DLA Policy and describe how we have to design it for our case study.

In order to test this policy and see how much the obtained cost via this policy is far from our real optimal value, we have to benchmark it against the perfect forecast. In the perfect forecast, we assume that we know stochastic changes in the future. Therefore the problem can be addressed as a deterministic model and obtained cost is the real optimal cost of the problem.

4.1 Direct Lookahead Models (DLA)

Sometimes the approximation of the cost function is very difficult due to large range of possible values of exogenous information in the model. In such cases we can use a direct lookahead model (DLA) policy. The DLA optimizes decisions over some pre-defined control horizon using a future approximation of the cost function. DLAs can be either deterministic or stochastic. Here we use Deterministic MPC as the Deterministic Direct Lookahead method and Certain Equivalence MPC as stochastic DLA strategy.

4.1.1 Deterministic MPC

A brief history of industrial MPC can be found at Qin and Badgwell 2002. Fig. 4 shows a diagram of MPC control and demonstrate how it works.

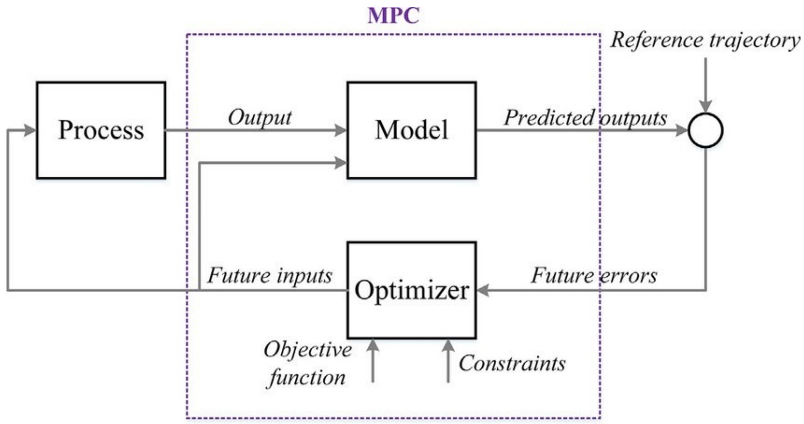

Figure 4: Model Predictive Control (MPC) Diagram

The principles of a deterministic MPC is as following:

a. At each time step, optimizer computes control actions by solving an open loop optimization problem for a certain prediction horizon which is 10 days in our case study. The optimizer uses current states as initial states and the cost function is subjected to main constraints for prediction horizon as it can be seen in above figure.

b. The first value of the computed control sequence in the horizon, i.e. first day in our model, is applied to the process.

c. At the next time step, the model update the states and above cycle is repeated.

The trajectory tracking by above algorithm can be seen in Fig. 5.

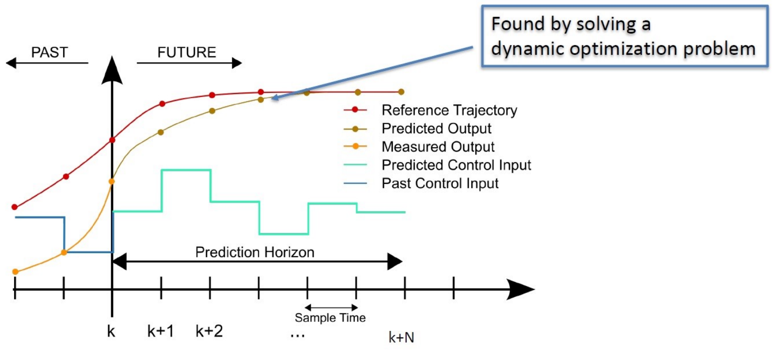

Figure 5: MPC trajectory tracking

For more detail explanation, we assume that we are at time k and states S_t. Knowing the difference between the reference state and measured state, i.e the state we want to be and the state we actually are, and considering a control horizon for removing this difference, we can solve an open loop optimization of cost function subject to defined constraints. Now a set of control actions over prediction horizon are obtained and the first one is taken and injected to the model. By implementing this action, we go forward one step time and again we measure the current state and this cycle continues.

In our case study, we can use the MPC model for the real optimal solution. In this case we assume that all uncertainty are clear to us when we are on first day of of control horizon. We are interested in minimizing the cost function subject to predefined constraints which was described in the base model section, whereas all prices, borehole temperature, demand and underground temperature over whole next days are clear to us. We will obtain this cost by a MPC algorithm with control horizon extended from the current day to last day of the year. The cost is the real minimum cost of the energy storage problem.

4.1.2 Certain Equivalence Model Predictive Control (CEMPC)

In a stochastic model, random values in the form of the exogenous information enter to the model, causes that the future states be uncertain and different from their deterministic value. It is clear that MPC solely cannot handle the uncertainty in the model. One of the proposed methods to handle uncertainty in the energy storage domain is the Uncertainty Equivalence MPC (CEMPC). This is same as MPC with the difference that uncertainty realized by the controller in form of the forecast functions. In other words, we estimate the uncertainty by some logical functions. By this technique, uncertainty is transformed to the values that can be calculated by approximation functions similar to other certain variables. Fig 6. Shows the flow of data in a CEMPC algorithm.

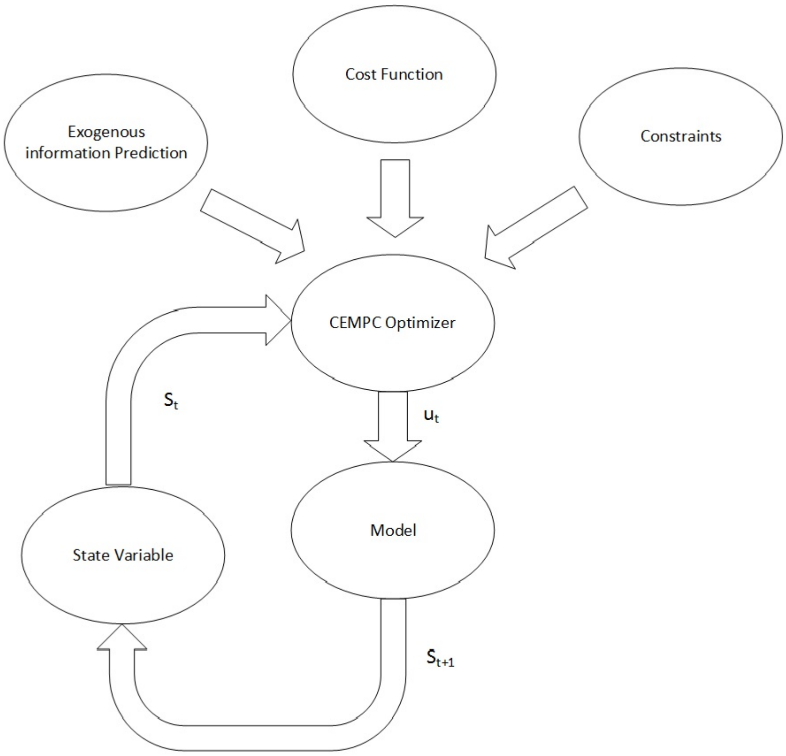

Figure 6: Certainty Equivalence Model Predictive Control(CEMPC) bloack diagram

At time step t, the predicted exogenous information, current states, cost function and constraint are imported to MPC and a series of actions in the prediction horizon is taken. The first action injected to the model and next set of states is obtained. These values are return to the MPC as current state alongside with prediction of exogenous information at time t+1 and the cycle continues till end of the horizon.

We have five exogenous information presented in our case study as it was stated in basic model section and we predict these variables as described in section 3.4. Sinusoidal prediction function of the demand is similar to its real function but without stochastic change. Therefor the prediction formula can be written as below:

$$D_t = -A_d \sin(\frac{2\pi t}{365} + \pi) \tag{20}$$

Again, for underground temperature we write the prediction equation as sinusoidal with same parameters of of its real function but without Gaussian noise:

$$T_t{}^{\text{inf}} = T_{\max}{}^{\text{inf}} - A^{\text{inf}} \sin(\frac{2\pi t}{365} - \frac{\pi}{2}) \tag{21}$$

It is clear that both demand and underground temperature prediction equations can estimate the value of these variables and trend of change is same as main equations. For prices prediction we consider the same first Markov Chain model with below formulas:

$$P_t{}^{h,hh,c} = \max\left\{P_{avg}{}^{h,hh,c} + normrnd(0, P_{avg}{}^{Std})\right\} \tag{22}$$

All parameters in above equation is same as real price equation. In case of constant value prediction, i.e.third prediction variation set in section 3.4, we assume that they are equal to average values $\mathrm{P_{avg}}^{\mathrm{h,hh,c}}$. These prediction equations generate a sample path of the prices, demand and underground temperature over prediction horizon to be applied in CEMPC method.

Now we can optimize the cost function with above predictions by CEMPC policy over a prediction horizon of 10 days. The optimization problem between the t_k and t_{k+10} is written as:

$$\min_{Q^i} \sum_{t_k}^{t_k+10} {Q_t}^b {P_t}^h + {Q_t}^{hh} {P_t}^{hh} + {Q_t}^c {P_t}^c \tag{23}$$

s.t.

$$D_t = \left(\frac{COP}{COP-1} {Q_t}^b + {Q_t}^h\right) * 1\_\{D_t \geq 0\} + ({Q_t}^b + {Q_t}^c) * 1\_\{D_t \prec 0\} \tag{24}$$

$${Q_t}^b, {Q_t}^{hh}, {Q_t}^c \geq 0 \tag{25}$$

$$0°C \leq {T_t}^b \leq 12°C \tag{26}$$

$$0 \leq {Q_t}^b \leq 30KWh \tag{27}$$

Every time this problem is solved, a set of control actions (${Q_{t_k}}^{b,h,c}$, ${Q_{t_{k+1}}}^{b,h,c}$,. . . , ${Q_{t_{k+10}}}^{b,h,c}$) are obtained over the control horizon. The first obtained action injected to the model and obtained states used as initial state of next time step and this procedure continues with prediction horizon shifted one step time forward.

5 Evaluating Policy Performance

For each prediction variations described in section 3.4.3, CEMPC policy tested over 30 sample paths of simulated demand, heat pump price, heater price, cooler price and underground temperature. We present the performance of the CEMPC with mentioned prediction sets in this chapter. Thermal Energy flows are controlled over 365 days, i.e. one year. We start the control from first day of the winter and update energy allocation decisions every day. We present profiles of energy allocation for each set of prediction and calculate the average cost showing that how much we have to pay as expected annual cost of the energy consumption over 30 tested trial.

5.1 CEMPC prediction set 1: underground temperature: sinusoidal, prices: constant value

We assume that prices are constant and equal to average prices in Table 3. Profiles of energy allocation decision are presented in Fig. 7. From the profiles it can be seen that the the priority of the controller for heating is always heat pump which has lower price. Whenever a constraint saturates, the controller decides to switch the heating from the borehole to the conventional heater.

Total cost of the energy is calculated 13,285 NOK with this set of prediction.

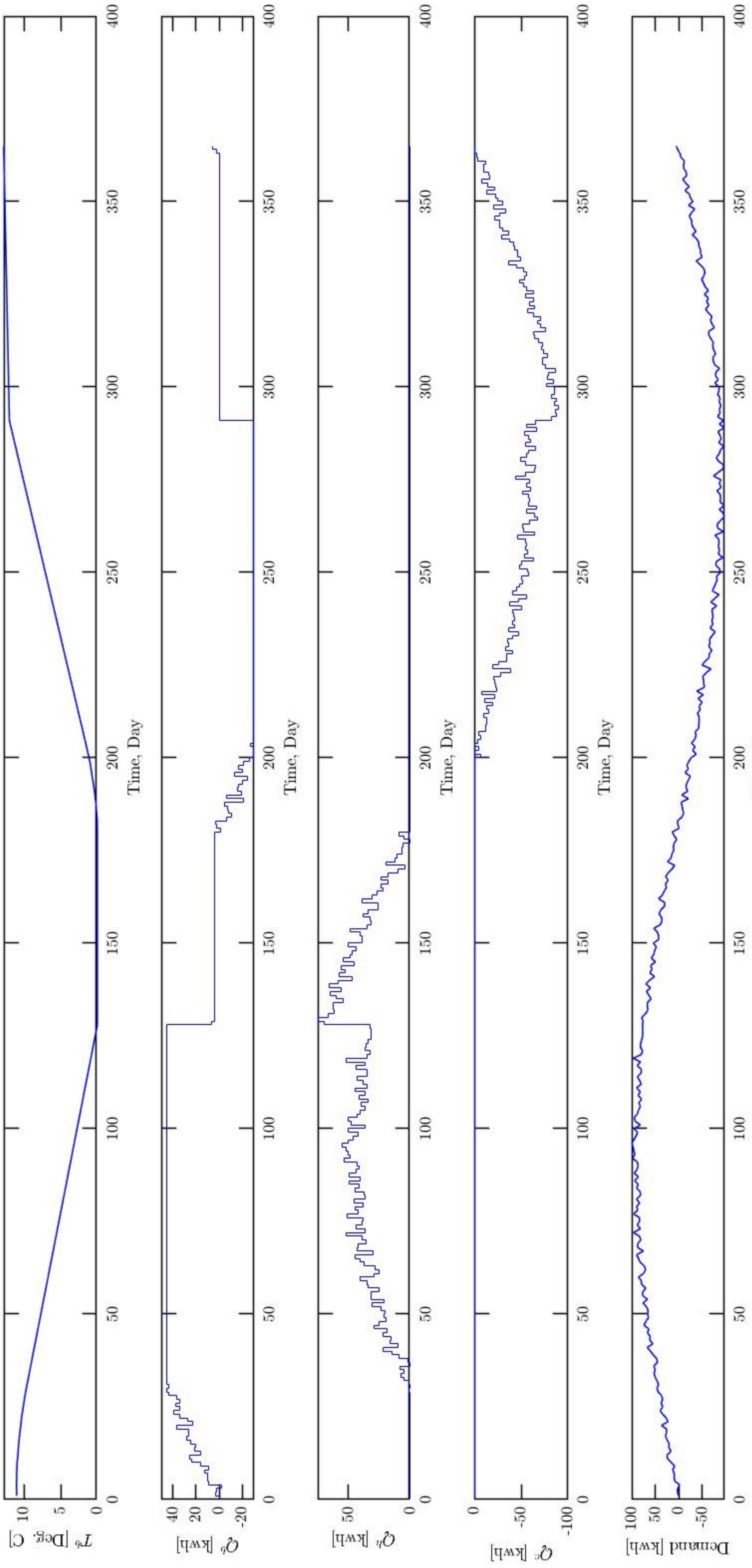

Figure 7: Solution profiles with prediction set 1, CEMPC method, demand and underground temperature prediction is sinusoidal and price predictions is constant average price

5.2 CEMPC [prediction set 2: underground temperature: sinusoidal, Prices: first order Markov chain

Fig. 8 shows the profiles of energy allocation for this set of prediction over one year. AS it is seen there are sudden switches in the profiles demonstrating that controller try to take the optimal decision based on the prediction of the price over prediction horizon and will select the cheapest route based on the prediction.

Total cost of the energy is calculated 13,749 NOK with these prediction.

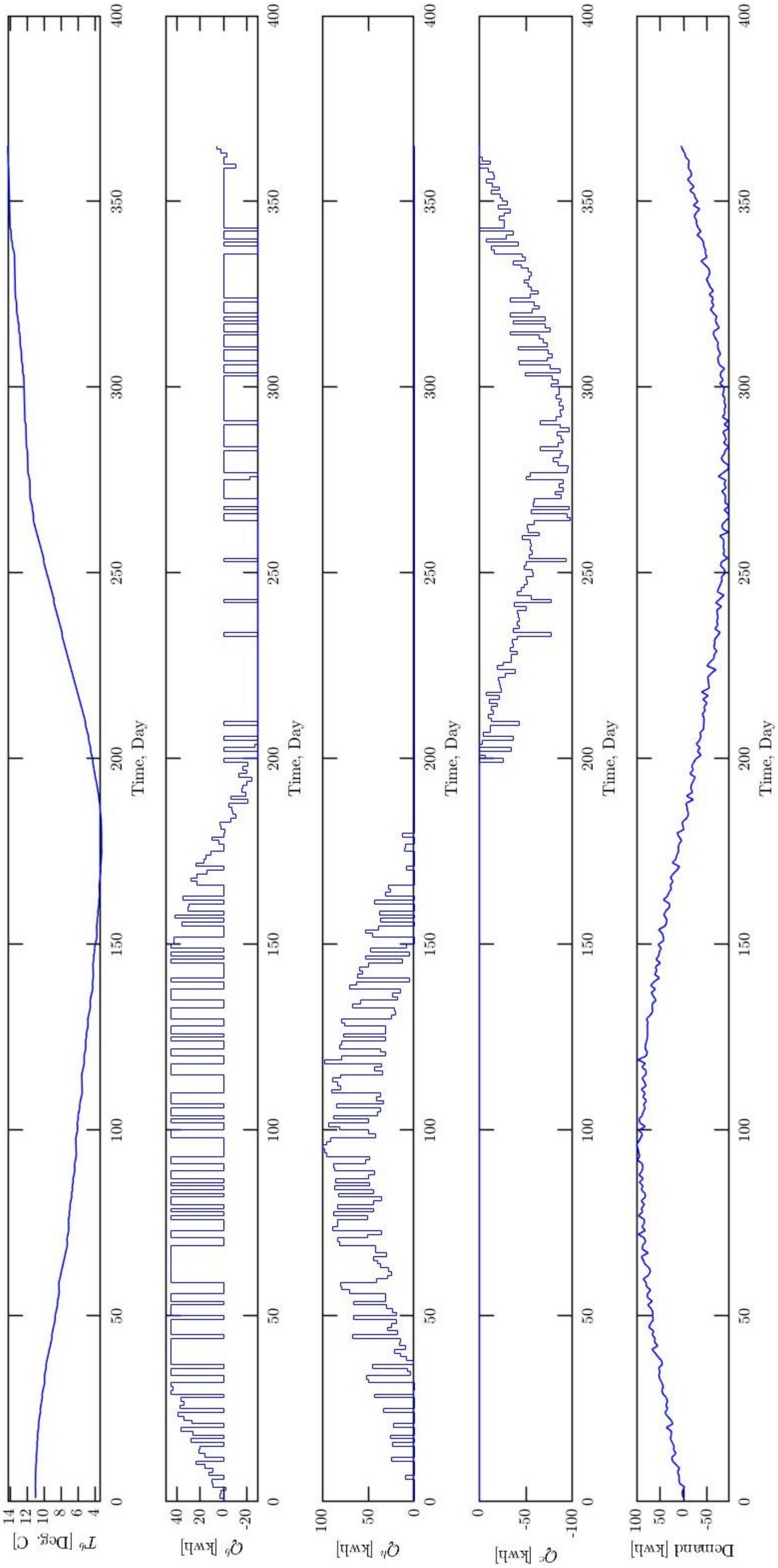

Figure 8: Solution profiles with prediction set 2, CEMPC method, demand and underground temperature prediction is sinusoidal and price prediction is first order Markov chain

5.3 CEMPC prediction set 3: underground temperature: constant, Prices: first order Markov chain

Here the prediction is like as previous set with the difference that underground temperature is predicted constant and equal to 15 °C over a year. The results of this problem can be seen in Fig. 9

Total cost of the energy is calculated 13,770 NOK with this set of prediction.

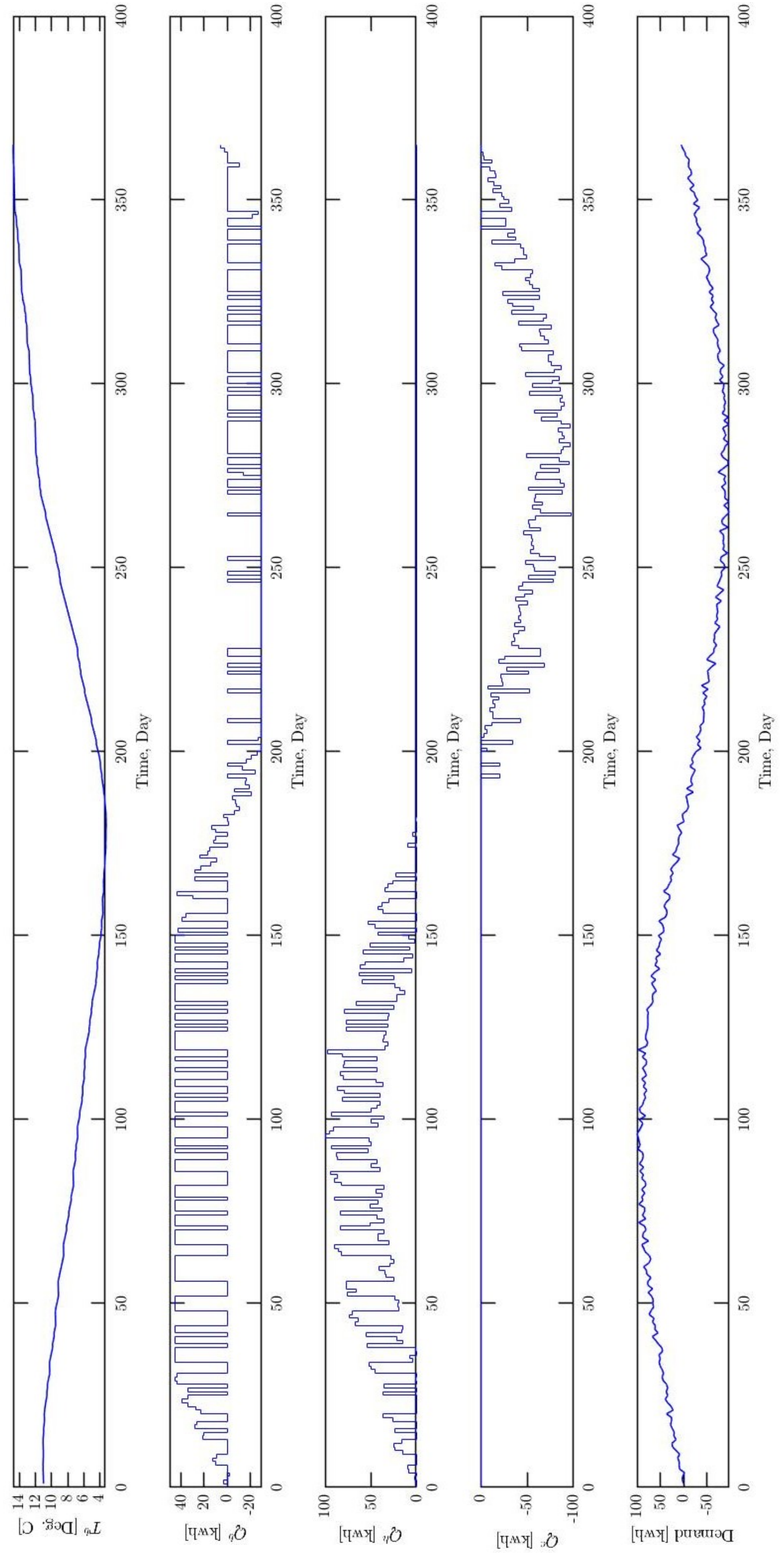

Figure 9: Solution profiles with prediction set 3, CEMPC method, demand prediction is sinusoidal, underground temperature prediction is constant and price prediction is first order Markov chain

5.4 Real Optimal Policy

In order to evaluate the performance of our CEMPC policy in previous section and to know how much we are far from real optimal point, we have to benchmark our policies against real optimal one. In this part we assume that we know the values of exogenous information of all future days over a year. It means that we are aware of all prices, demand values and underground temperature from tomorrow to last day of the year. In this case our problem converts from a stochastic problem to a deterministic problem and we can solve it by a deterministic MPC policy adjusting the prediction horizon from the current day to the lat day of the year.

The profiles of the real optimal solution are presented in Fig. 10 and the optimal price is calculated as 10,710 NOK.

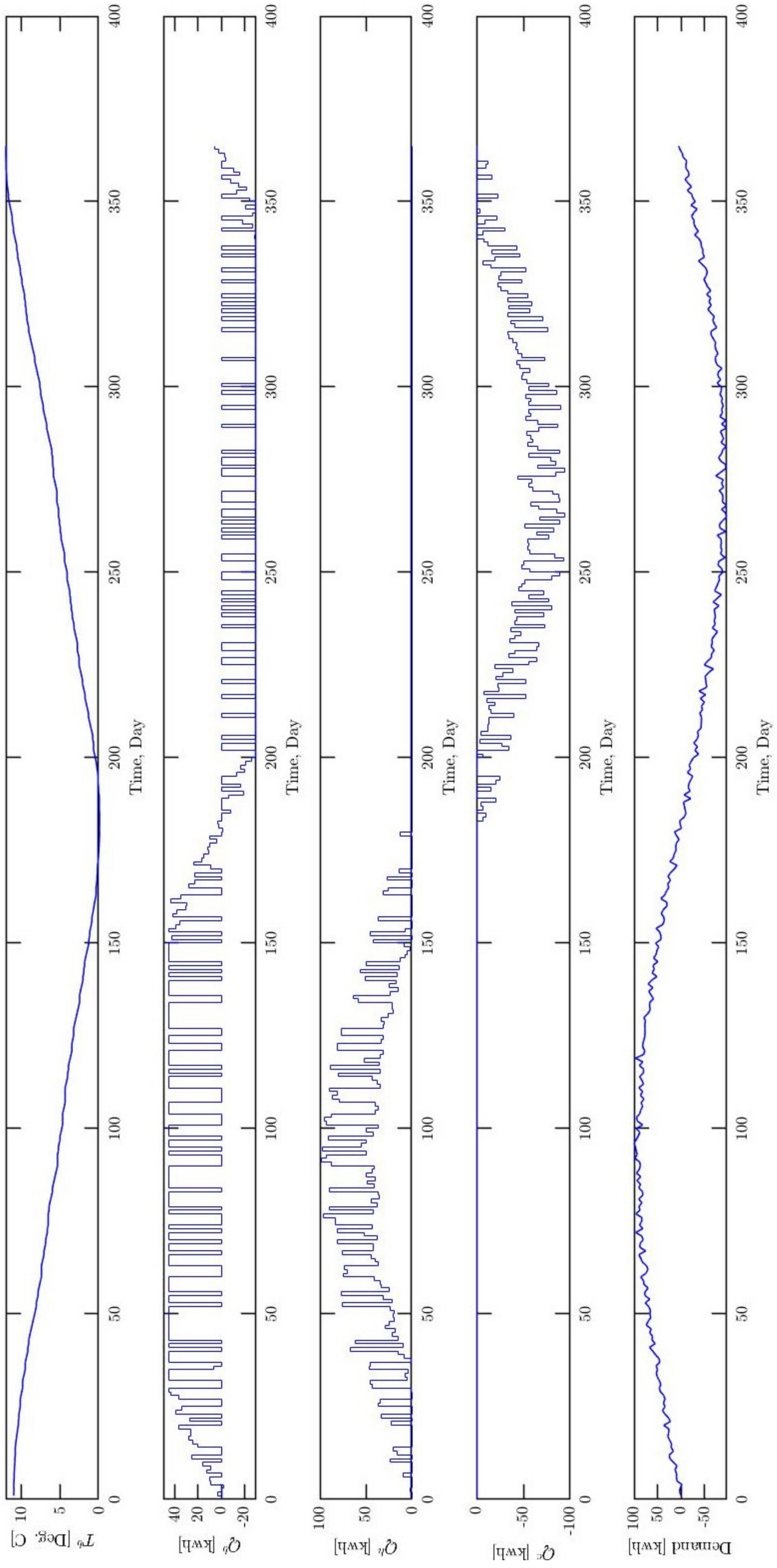

Figure 10: Solution profiles of real optimal solution with the perfect forecast

5.5 Comparison of the Results

In the previous sections we presented a sample path profile of underground temperature and energy flow allocations over a year for the different CEMPC prediction set and real optimal solution. We also calculated expected cost over 30 sample paths for each prediction set. In Table 4 we summarize the calculated costs of obtained for difwith real optimal cost.

Problem	Average cost [NOK]	Cost index value
CEMPC Prediction set 1	13,285	1.240
CEMPC Prediction set 2	13,749	1.283
CEMPC Prediction set 3	13,770	1.285
Real optimal cost	10710	1

Table 4: Comparison of the cost for CEMPC policy with different prediction sets

It can be seen that the lowest price after the optimal solution is for the prediction set 1 with constant average price prediction and is 24 percent higher than the real optimal solution. The cost for prediction sets 2 and 3 are higher than prediction set 1. In the next chapter, we analyze these results and provide explanation for the performance of the CEMPC with our defined prediction sets.

6 Discussion

This chapter includes discussion on results of the CEMPC policy. We also provide the difficulties with the simplification and reality of this model and suggest possible future expansion on this case study.

6.1 Performance of CEMPC

Comparing the price index of the variation in previous section, we see that the expected cost with the CEMPC control approach is 24 28 percent higher than real optimal point. It is natural as CEMPC works based on the forecast of the future changes of the exogenous information. Therefore, in order to keep the cost of the CEMPC acceptably close to real optimal point, it is very important to provide good forecast for the model. In our model we assumed three prices experience stochastic changes and there is one uncertainty present in the problem. In prediction sets 2 and 3 the prices fluctuate around an average point with first order Markov chain pattern. Results show that the performance of forecasting an average price is better works than prediction with first order Markov chain. The justification is that the number of the forecast error are higher in the case that we predict prices with random values, i.e. prediction sets 2 and 3. The forecast error happens when for examples heater price are lower than heat pump price and and the prediction shows that it is higher at discrete sample times. This is also true for inverse case. Although the difference between the cost with these prediction sets 2 and 3 are not so much higher than set 1 but they show the importance of the good prediction in a CEMPC policy. The cost of problem 3 do not differ from problem 2 and it is comply with our model, because in order to provide direct cooling for the building, we need a borehole system with low amount of thermal energy loss and high energy storage capacity. Moreover, temperature variation is not so much high that can affect final price in real systems and in many cases it is assumed that the underground temperature is constant. Briefly, the results show us that if we can make a detail forecast of the prices by for example taking peak period into account, we can decrease our costs even more.

6.2 Simplifications and Challenges of the Model

Nearly all stochastic control of energy storage system case studies need assumption to simplify the model and we did same in the project. Nevertheless, it is not possible to solve these kind of problems because the energy storage domain is a kind of large state space problem. Here we assumed that the COP of the system is constant, but in the reality, it might undergo variations based on the temperature of the hot side and cold side and flow rate of circulation medium. If the COP variation enter to the model, our linear problem will convert to a non linear one, However our assumption is not far from a real case as these variations are not huge. We did not entered more details to our borehole system, for example flowrate variation. These parameters increase the non-linearity of the problem and make it much more complex and time consuming to solve and the problem might not be solved with current methods over long periods.

Another assumption which is not true in real problems is zero price for direct cooling. In this case, a circulation cost of the water can be added to the system. The most important challenge in our work is to obtaining the true operating price for cooling, heating and heat pump. The resources just provides the electricity prices and we could not find a certain range for heating, cooling and heat pump prices. The applied price values were based on our assumption and close to electricity prices. In case of having real values and patterns of these prices we can set our model like as a real case and better design the optimization policies.

It is important to note that MPC model cannot check all the constraint till end of horizon when it the controller is in initial or middle days. We solved this problem by applying constraint handling method and used this technique for violation of borehole temperature as it is almost always violated at end of the horizon. That is because in these days, i.e. last days of the summer, it always receive heat from the underground even if the demands received the cold from the cooler and heat from the demand not added to the borehole. In our case study we handled it by changing this constraint to a soft one and penalize the controller for violating it, but in the case studies that constraints cannot physically violated, applying the MPC method should be taken carefully.

6.3 Suggestion for Further Studies

In this project, we checked the performance of CEMPC as a subclass of direct lookahead approximation with comparing it to real optimal point and noticed how different prediction can change the expected cost. Some improvements can be made in terms of both modeling and policies in this case study:

• Some fixed parameters can be reformulated. For example the heat exchange might not be constant over a year or COP can be written as formulation.

• The CEMPC performance can be compared and analyzed according to methods such as PFA or VFA which are outlined in Powell book.

• Different price and demand variation patterns can be applied to this case study or other prediction can be made in order to study the CEMPC method more deeply.

7 Conclusion

We present the conclusion of our project in this chapter. We modeled a borehole energy storage system in parallel to a heater and cooler to provide thermal energy requirement of a building. We applied some simplification to our model and formulated it and identified constant parameters, states, transition function and cost function. We also specified the constraint of the optimization problem in chapter 3. We considered the uncertainty in our problem as heater price, cooler price, heat pump price, underground temperature and demand. We assumed that demand and underground temperature change with sinusoidal pattern with some noise over a year as formulated with Eq. 18 and 19. We also used Eq. 17 to show the price variations in our case study.

We designed and tested Certainty Equivalence Model Predictive Control (CEMPC) with three different prediction sets of exogenous information. These set are seen in section 4. Comparing the results with the real optimal point, we found out that cost of CEMPC method is 24 to 28 percent higher than the real optimal cost. Also, constant average price yields a lower cost compared to first order Markov chain prediction due to less forecast error. We concluded that making more precise prediction is very important to obtain a cost closer to real optimal value. In order to check performance of the CEMPC against other approaches, it is suggested that different policies such as modified parametric MPC or PFA could be designed and tested in this case study.

Bibliography

Arnold, M. and G. Andersson (2011). 'Model predictive control of energy storage including uncertain forecasts'. In: *Power Systems Computation Conference (PSCC)* 23, pp. 24–29.

Bär, Kristian et al. (2015). 'Seasonal High Temperature Heat Storage with Medium Deep Borehole Heat Exchangers'. In: *Energy Procedia* 76, pp. 351–360.

Cheng, B., T. samov and W. B. Powell (2018). 'Low-Rank Value Function Approximation for Co-Optimization of Battery Storage'. In: *IEEE Transactions on Smart Grid* 9.6, pp. 6590–6598.

Cui, P. et al. (2008). 'Simulation of hybrid ground-coupled heat pump with domestic hot water heating systems using HVACSIM'. In: *Energy Buildings* 40, pp. 1731–1736.

Durante, Joseph, Juliana Nascimento and Warren Powell (2017). 'Backward Approximate Dynamic Programming with Hidden Semi-Markov Stochastic Models in Energy Storage Optimization'. In:

H.P.Simão et al. (2017). 'The challenge of integrating offshore wind power in the U.S. electric grid. Part II: Simulation of electricity market operations'. In: *Renewable Energy* 103, pp. 418–431.

Han, Jiequn and Weinan E (2016). 'Deep Learning Approximation for Stochastic Control Problems'. In: *CoRR* abs/1611.07422.

LeBreux, M., M. Lacroix and G. Lachiver (2006). 'Fuzzy and feedforward control of an hybrid thermal energy storage system'. In: *Journal of Optimization Theory and Applications* 38, pp. 1149–1155.

Ma, Y. et al. (2012). 'Predictive control for energy efficient buildings with thermal storage: Modelling, stimulation, and experiments'. In: *IEEE Control Systems* 32, pp. 44–64.

Powell, Warren B. (2019). *Stochastic Optimization and Learning.* John Wiley and Sons, Inc, Publication.

Qin, S. Joe and Thomas A. Badgwell (2002). 'A survey of industrial model predictive control technology'. In: *Control Engineering Practice* 11, pp. 733–764.

Ridder, Fjo De et al. (2011). 'An optimal control algorithm for borehole thermal energy storage systems'. In: *Energy and buildings* 43, pp. 2918–2925.

Rink, R.E. (1994). 'Optimal operation of solar heat-storage with off-peak energy price incentive'. In: *Journal of Optimization Theory and Applications* 15, pp. 251–266.

Rink, R.E., V. Gourishankar and M. Zaheeruddin (1988). 'Optimal-control of heat-pump heat-storage systems with time-of-day energy price incentive'. In: *Journal of Optimization Theory and Applications* 58, pp. 93–108.

Schulte, Daniel Otto et al. (2015). 'Optimization of Medium-Deep Borehole Thermal Energy Storage Systems'. In: *Energy Technology* 4, pp. 104–113.

Sioshansi, R., S. H. Madaeni and P. Denholm (2014). 'A Dynamic Programming Approach to Estimate the Capacity Value of Energy Storage'. In: *IEEE Transactions on Power Systems* 29.1, pp. 395–403.

Spitler, J.D. (2005). 'Ground-source heat pump system research – past, present, and future'. In: *HVAC and R Research* 11, pp. 165–167.

Wallace, S. W. and S. E. Fleten (2003). 'Stochastic Programming Models in Energy'. In: *Handbooks in Operations Research and Management Science* 10.1, pp. 637–677.

Warrington, J. et al. (2012). 'Robust reserve operation in power systems using affine policies. Decision and Control (CDC)'. In: *IEEE 51st Annual Conference on*, pp. 1111–1117.

Welsch, Bastian et al. (2016). 'Characteristics of medium deep borehole thermal energy storage'. In: *International Journal of Energy Research* 40, pp. 1855–1868.

Z. Deng S.J. Rees, J.D. Spitler (2005). 'A model for annual simulation of standing column well ground heat exchangers'. In: *HVAC and R Research* 11, pp. 355–367.

Zaheeruddin, M., V.G. Gourishankar and R.E. Rink (1988). 'Dynamic suboptimal control of a heat-pump heat-storage system'. In: *Journal of Optimization Theory and Applications* 9, pp. 341–355.

Zhou, Y. et al. (2018). 'Managing wind-based electricity generation in the presence of storage and transmission capacity'. In: *Production and Operations Management* 28, pp. 970–989.

Appendix

A Static Parameters and Initial Conditions

```
function S0 = S0()
% Model parameters
S0.Tbmax = 285;     % Maximum allowable ground temp   [Deg. K]
S0.Tbmin = 273;     % Minimum allowable ground temp   [Deg. K]
S0.Rmax  = 30;      % Maximum heating / cooling rate [kwh]
S0.m     = 235619; % Water mass in the bore based on:
                    % 30 pipes
                    % Dia.    = 1 meters
                    % Length = 10 meters
S0.Cpw   = 0.001167% Water Heat Capacity              [kwh/kG Deg. K]
S0.lambda= 0.01;     % Based on OHTC X Area           [kwh/Deg. K]
S0.COP   = 3;                                          %COP of heat pump
S0.Pmin  = 0        % We do not have negative price
S0.P1std = 0.5      % Price standard deviation         [NOK/kwh]

% Initial states
S0.S.Tb   = 284;    % Borehole temp                    [Deg. K]
S0.S.Tinf= 288;     % Ground temp                      [Deg. K]
S0.S.D    = 0;      % Demand energy                    [kwh]
S0.S.Ph   = 0.9;    % heater price                     [NOK/kwh]
S0.S.Phh  = 0.7;    % Heaatpump Heating price          [NOK/kwh]
S0.S.Pc   = 0.6;    % Electricity-based cooling price[NOK/kwh]
S0.S.s    = 30;     % Constraint Handling Factor
end
```

B Generation of Exogenous Information

```
close all
clear all
clc
addpath('C:\Users\win\Desktop\Specialization Project')
plot_settings
S0 = S0()

%% Exogenouse information generation
T        = 365;                        % Time horizon              [Day]
scenario = 30;                         % Number of sample paths

%Exogenous information storage.
sample_path.D   = [];
sample_path.Ph  = [];
sample_path.Phh = [];
sample_path.Pc  = [];
sample_path.Tinf= [];

for i = 1:scenario
    W = exogenous_information(S0,T);
    sample_path.D    = [sample_path.D; W.D];
    sample_path.Tinf = [sample_path.Tinf; W.Tinf];
    sample_path.Ph   = [sample_path.Ph; W.Ph];
    sample_path.Phh  = [sample_path.Phh; W.Phh];
    sample_path.Pc   = [sample_path.Pc; W.Pc];
end

%% Exogenous Information for a Sample Trial
Trial_plot=20

subplot (5,1,1)
plot(sample_path.D(Trial_plot,:),'b')
xlabel('Time period, Day')
ylabel('Demand [kwh]')

subplot (5,1,2)
plot(sample_path.Tinf(Trial_plot,:),'b')
xlabel('Time period, Day')
ylabel('$T_{underground}$, [Deg. C]')

subplot (5,1,3)
plot(sample_path.Phh(Trial_plot,:),'b')
xlabel('Time period, Day')
ylabel('HP Price [NOK]')

subplot (5,1,4)
plot(sample_path.Ph(Trial_plot,:),'b')
xlabel('Time period, Day')
ylabel('Heating Price [NOK]')

subplot (5,1,5)
plot(sample_path.Pc(Trial_plot,:),'b')
xlabel('Time period, Day')
ylabel('Cooling Price [NOK]')

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

```
save('C:\Users\win\Desktop\Specialization Project\sample_path.mat');
```

B.1 Exogenous Information Function

```
function W = exoinfo(S0,T)
%% Ground Temp Model
for N=1:T
muinf=285.5;
Ainf =2.5;
Tinf(N) =muinf-Ainf*sin ((2*pi*N/T)-(pi/2))+normrnd(0,0.05);         % [Deg. C]
end

%% Demand model
for N=1:T
    Ad = 90;                                    % Domain of demand energy [kwh]
    D(N)  = -Ad*sin((2*pi*N/T)+pi)+normrnd(0,5);
end

%% Price model
for N=1:T
    Pn = normrnd(0,S0.P1std);
    Ph(N) = S0.S.Ph + Pn;
    Pn = normrnd(0,S0.P1std);
    Phh(N)= S0.S.Phh + Pn;
    Pn = normrnd(0,S0.P1std);
    Pc(N) = S0.S.Pc + Pn;
end
Ph=max(S0.Pmin,Ph);
Phh=max(S0.Pmin,Phh);
Pc=max(S0.Pmin,Pc);

W.D   = [S0.S.D];          % Resetting demand at first of each scenario
W.Phh = [S0.S.Phh];        % Resetting heatpump price at first of each scenario
W.Ph  = [S0.S.Ph];         % Resetting heating price at first of each scenario
W.Pc  = [S0.S.Pc];         % Resetting cooling at first of each scenario
W.Tinf= [S0.S.Tinf];       % Resetting underground temperature at first of each
↪   scenario

W.D   = [W.D D];
W.Ph  = [W.Ph Ph];
W.Phh = [W.Phh Phh];
W.Pc  = [W.Pc Pc];
W.Tinf= [W.Tinf Tinf];
end
```

C Code of problem with prediction set 1: sinusoidal prediction of underground temperature and constant prediction of prices

```
close all
clear all
clc
tic
addpath('C:\Users\win\Desktop\Specialization Project')
load('sample_path.mat')

%% Settings
S0 = S0();
S = S0.S;
plot_settings

scenario = size(sample_path.D,1);     % Scenario number
N = size(sample_path.D,2)-1;          % Total time horizon              [Day]
nd = 10;                              % Open loop optimization time horizon

%Extention of sample path
sample_path.D    = [sample_path.D];
sample_path.Ph   = [sample_path.Ph];
sample_path.Phh = [sample_path.Phh];
sample_path.Pc   = [sample_path.Pc];
sample_path.Tinf= [sample_path.Tinf];

%% Prediction of demand and ground temp. model without exogenous information
Ad  = 90;                           % Overall domain of Demand energy  [kwh]
for i=1:N-1
    D(i)=Ad*sin(2*pi*i/N);
end
D = [S.D D D(1,1:nd)];

for j=1:N-1
    muinf=285.5;
    Ainf=2.5;
    Tinf(j)=muinf-Ainf*sin ((2*pi*j/N)-(pi/2));
end
 Tinf=[S.Tinf Tinf Tinf(1,1:nd)];

%% Simulation
PathsCost =[];
for w = 1:scenario
ss;
%Initial setup
S0 = S0();
S = S0.S;
%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];
    for k = 1:N
    % Exogenous information model
    % Ground temp horizon
    Tinf_horizon = [S.Tinf Tinf(1,k+1:k+nd-1)];

    % Demand horizon
```

```
Demand_horizon = [S.D D(1,k+1:k+nd-1)];
nd = length(Demand_horizon);

% Price horizon
%Price_noise1 =  normrnd(0,S0.P1std,1,nd);
Ph_horizon=S0.S.Ph*ones(1,nd);
%Ph_horizon=max(S0.Pmin,Price_noise1 + S0.S.Ph);
Phh_horizon=S0.S.Phh*ones(1,nd);
%Price_noise2 =  normrnd(0,S0.P1std,1,nd);
%Phh_horizon=max(S0.Pmin,Price_noise2 + S0.S.Phh);
%Price_noise3 =  normrnd(0,S0.P1std,1,nd);
Pc_horizon=S0.S.Pc*ones(1,nd);
%Pc_horizon=max(S0.Pmin,Price_noise3 + S0.S.Pc);


% Finding indexes of the negative demands
[~, col] = find(Demand_horizon<0);
% Creating "boolean" array
DProductArray = ones(nd,1);
% Indicating negative demands
DProductArray(col,:) = 0;

%% Equality constraint : Ax=b
% Demand constraint
Aeq_D = zeros(nd,6*nd);
%D_k = DPredictArray.*(1.5*U_k + Qheat_k) + (1 { DpredictArray).*(-U_k -
↪  Qcool_k)
for ii = 1:nd
    Aeq_D(ii,1 + 6*(ii - 1)) = (S0.COP/(S0.COP-1))*DProductArray(ii) +
    ↪  (-1)*(1 - DProductArray(ii)); % for U_k
    Aeq_D(ii,2 + 6*(ii - 1)) = (1)*DProductArray(ii) + (0)*(1 -
    ↪  DProductArray(ii));                    % for Qheat_k
    Aeq_D(ii,3 + 6*(ii - 1)) = (0)*DProductArray(ii) + (-1)*(1 -
    ↪  DProductArray(ii));                   % for Qcool_k
end
Beq_D = Demand_horizon';

%Transition function
Aeq_T = zeros(nd,6*nd);
a=1-(S0.lambda/(S0.m*S0.Cpw));
b=1/(S0.m*S0.Cpw);
for iii=1:nd
   Aeq_T(iii:end,1+6*(iii-1))= b*((+1)*DProductArray(iii) + (-1)*(1 -
   ↪  DProductArray(iii)));
   Aeq_T(iii:end,5+6*(iii-1))= (a-1);
   for jjj=iii:nd
       Aeq_T(jjj,1+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,1+6*(iii-1));
       Aeq_T(jjj,5+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,5+6*(iii-1));
   end
end
Aeq_T=Aeq_T + kron(eye(nd),[0 0 0 1 0 0]);
Beq_T=zeros(nd,1);
for kkk=1:nd
   Beq_T(kkk,1) = [a^kkk*S.Tb];
end

% Aeq & Beq integration
Aeq = [Aeq_D;Aeq_T];
```

```matlab
Beq = [Beq_D;Beq_T];

%% Inequality constraints : Ax<=b
%Constraint handling : Tb-s <=Tbmax
A = kron(eye(nd), [0 0 0 1 0 -1]);
B = S0.Tbmax*ones(nd,1);

%% Objective function
f = zeros(1,6*nd);
for kk=1:nd
    f(1,1 + 6*(kk - 1)) =
    ↪  (S0.COP/(S0.COP-1))*DProductArray(kk)*Phh_horizon(kk) + (1)*(1 -
    ↪  DProductArray(kk))*0 ;
    f(1,2 + 6*(kk - 1)) = (Ph_horizon(kk));
    f(1,3 + 6*(kk - 1)) = (Pc_horizon(kk));
    f(1,6 + 6*(kk - 1)) = 30;                        % Penalizing cost
end

%% Boundaries - lower and upper
lb = zeros(6*nd,1);
lb(4:6:6*nd,1) = S0.Tbmin;
for i=1:nd
    lb(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

ub = repmat([S0.Rmax inf inf inf 0 inf]',nd,1);
for i=1:nd
    ub(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

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

%  Data store from openloop optimization
u.u     = w_opt(1:6:6*nd);
u.qheat = w_opt(2:6:6*nd);
u.qcool = w_opt(3:6:6*nd);
u.Tb    = w_opt(4:6:6*nd);
u.Tinf  = w_opt(5:6:6*nd);
u.s     = w_opt(6:6:6*nd);

%% Take an action
uk = [u.u(1);u.qheat(1);u.qcool(1)];
sample_path.uk(1+5*(w-1):3+5*(w-1),k)=uk;            % For visual check
sample_path.uk(5+5*(w-1),k)=w_opt(6);                % For visual check

%% Transition during day
S.Tb = S.Tb + [((-1)*DProductArray(1) + (+1)*(1 - DProductArray(1)))*b 0 0]*uk
↪  + (1-a)*Tinf_horizon(1);
sample_path.uk(4+5*(w-1),k)=S.Tb-273;                % For visual check

%% Realization
S.Ph = sample_path.Ph(w,k+1);
S.Phh = sample_path.Phh(w,k+1);
S.Pc = sample_path.Pc(w,k+1);
S.D = sample_path.D(w,k+1);
S.Tinf = sample_path.Tinf(w,k+1);
```

```
    %% For plotting
    xSim = [xSim, S.Tb];
    uSim = [uSim, uk];
     end

     % Finding indexes of the negative demands
     [~, col] = find(sample_path.D(w,:)<0);
     % Creating "boolean" array
     DemandProductArray = S0.COP/(S0.COP-1)*ones(N+1,1);
     % Indicating negative demands
     DemandProductArray(col,:) = 0;

PathCost_i = sum((DemandProductArray(1:N)'.*sample_path.Phh(w,1:N).*uSim(1,1:N))
↪   + sample_path.Ph(w,1:N).*uSim(2,1:N) + sample_path.Pc(w,1:N).*uSim(3,1:N));
PathsCost = [PathsCost; PathCost_i];
end
Cost = mean(PathsCost);

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

% save
save = ('P02_MPC')

% Modifying values for Plotting the states and inputs for a given scenario
trial = 20;
sample_path.ukplot(1:4,:) = sample_path.uk(1+5*(trial-1):4+5*(trial-1),:);
sample_path.ukplot(5,:)= sample_path.D(trial,1:N);

for ind= 1:N
if sample_path.ukplot(5,ind)>=0
   sample_path.ukplot(1,ind)=(S0.COP/(S0.COP-1))*sample_path.ukplot(1,ind)
else
   sample_path.ukplot(1,ind)=-1*sample_path.ukplot(1,ind)
   sample_path.ukplot(3,ind)=-1*sample_path.ukplot(3,ind)
end
end

figure
subplot(5,1,1)
plot(sample_path.ukplot(4,:),'b')
xlabel('Time, Day' )
ylabel('$T^{b}$ [Deg. C]')

subplot(5,1,2)
stairs(0:N,[sample_path.ukplot(1,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{b}$ [kwh]')

subplot(5,1,3)
stairs(0:N,[sample_path.ukplot(2,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{h}$ [kwh]')

subplot(5,1,4)
stairs(0:N,[sample_path.ukplot(3,:), nan],'b-')
xlabel('Time, Day')
```

```
ylabel('$Q^{c}$ [kwh]')

subplot(5,1,5)
plot(sample_path.ukplot(5,:),'b')
xlabel('Time, Day')
ylabel('Demand [kwh]')
```

D Code of problem with prediction set 2: sinusoidal prediction of underground temperature and first order Markov chain prediction of prices

```
close all
clear all
clc
tic
addpath('C:\Users\win\Desktop\Specialization Project')
load('sample_path.mat')

%% Settings
S0 = S0();
S = S0.S;
plot_settings

scenario = size(sample_path.D,1);     % Scenario number
N = size(sample_path.D,2)-1;          % Total time horizon              [Day]
nd = 10;                              % Open loop optimization time horizon

%Extention of sample path
sample_path.D   = [sample_path.D];
sample_path.Ph  = [sample_path.Ph];
sample_path.Phh = [sample_path.Phh];
sample_path.Pc  = [sample_path.Pc];
sample_path.Tinf= [sample_path.Tinf];

%% Prediction of demand and ground temp. model without exogenous information
Ad  = 90;                            % Overall domain of Demand energy  [kwh]
for i=1:N-1
    D(i)=Ad*sin(2*pi*i/N);
end
D = [S.D D D(1,1:nd)];

for j=1:N-1
    muinf=285.5;
    Ainf=2.5;
    Tinf(j)=muinf-Ainf*sin ((2*pi*j/N)-(pi/2));
end
 Tinf=[S.Tinf Tinf Tinf(1,1:nd)];

%% Simulation
PathsCost =[];
for w = 1:scenario
ss;
%Initial setup
S0 = S0();
S = S0.S;
%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];
    for k = 1:N
    % Exogenous information model
    % Ground temp horizon
    Tinf_horizon = [S.Tinf Tinf(1,k+1:k+nd-1)];
```

```
% Demand horizon
Demand_horizon = [S.D D(1,k+1:k+nd-1)];
nd = length(Demand_horizon);

% Price horizon
Price_noise1 =  normrnd(0,S0.P1std,1,nd);
Ph_horizon=max(S0.Pmin,Price_noise1 + S0.S.Ph);
Price_noise2 =  normrnd(0,S0.P1std,1,nd);
Phh_horizon=max(S0.Pmin,Price_noise2 + S0.S.Phh);
Price_noise3 =  normrnd(0,S0.P1std,1,nd);
Pc_horizon=max(S0.Pmin,Price_noise3 + S0.S.Pc);


% Finding indexes of the negative demands
[~, col] = find(Demand_horizon<0);
% Creating "boolean" array
DProductArray = ones(nd,1);
% Indicating negative demands
DProductArray(col,:) = 0;

%% Equality constraint : Ax=b
% Demand constraint
Aeq_D = zeros(nd,6*nd);
%D_k = DPredictArray.*(1.5*U_k + Qheat_k) + (1 { DpredictArray).*(-U_k -
↪  Qcool_k)
for ii = 1:nd
    Aeq_D(ii,1 + 6*(ii - 1)) = (S0.COP/(S0.COP-1))*DProductArray(ii) +
    ↪  (-1)*(1 - DProductArray(ii)); % for U_k
    Aeq_D(ii,2 + 6*(ii - 1)) = (1)*DProductArray(ii) + (0)*(1 -
    ↪  DProductArray(ii));                   % for Qheat_k
    Aeq_D(ii,3 + 6*(ii - 1)) = (0)*DProductArray(ii) + (-1)*(1 -
    ↪  DProductArray(ii));                  % for Qcool_k
end
Beq_D = Demand_horizon';

%Transition function
Aeq_T = zeros(nd,6*nd);
a=1-(S0.lambda/(S0.m*S0.Cpw));
b=1/(S0.m*S0.Cpw);
for iii=1:nd
   Aeq_T(iii:end,1+6*(iii-1))= b*((+1)*DProductArray(iii) + (-1)*(1 -
   ↪  DProductArray(iii)));
   Aeq_T(iii:end,5+6*(iii-1))= (a-1);
   for jjj=iii:nd
       Aeq_T(jjj,1+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,1+6*(iii-1));
       Aeq_T(jjj,5+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,5+6*(iii-1));
   end
end
Aeq_T=Aeq_T + kron(eye(nd),[0 0 0 1 0 0]);
Beq_T=zeros(nd,1);
for kkk=1:nd
    Beq_T(kkk,1) = [a^kkk*S.Tb];
end

% Aeq & Beq integration
Aeq = [Aeq_D;Aeq_T];
Beq = [Beq_D;Beq_T];
```

```
%% Inequality constraints : Ax<=b
%Constraint handling : Tb-s <=Tbmax
A = kron(eye(nd), [0 0 0 1 0 -1]);
B = S0.Tbmax*ones(nd,1);

%% Objective function
f = zeros(1,6*nd);
for kk=1:nd
    f(1,1 + 6*(kk - 1)) =
    ↪  (S0.COP/(S0.COP-1))*DProductArray(kk)*Phh_horizon(kk) + (1)*(1 -
    ↪  DProductArray(kk))*0 ;
    f(1,2 + 6*(kk - 1)) = (Ph_horizon(kk));
    f(1,3 + 6*(kk - 1)) = (Pc_horizon(kk));
    f(1,6 + 6*(kk - 1)) = 30;                      % Penalizing cost
end

%% Boundaries - lower and upper
lb = zeros(6*nd,1);
lb(4:6:6*nd,1) = S0.Tbmin;
for i=1:nd
    lb(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

ub = repmat([S0.Rmax inf inf inf 0 inf]',nd,1);
for i=1:nd
    ub(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

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

%  Data store from openloop optimization
u.u     = w_opt(1:6:6*nd);
u.qheat = w_opt(2:6:6*nd);
u.qcool = w_opt(3:6:6*nd);
u.Tb    = w_opt(4:6:6*nd);
u.Tinf  = w_opt(5:6:6*nd);
u.s     = w_opt(6:6:6*nd);

%% Take an action
uk = [u.u(1);u.qheat(1);u.qcool(1)];
sample_path.uk(1+5*(w-1):3+5*(w-1),k)=uk;          % For visual check
sample_path.uk(5+5*(w-1),k)=w_opt(6);              % For visual check

%% Transition during day
S.Tb = S.Tb + [((-1)*DProductArray(1) + (+1)*(1 - DProductArray(1)))*b 0 0]*uk
↪  + (1-a)*Tinf_horizon(1);
sample_path.uk(4+5*(w-1),k)=S.Tb-273;              % For visual check

%% Realization
S.Ph = sample_path.Ph(w,k+1);
S.Phh = sample_path.Phh(w,k+1);
S.Pc = sample_path.Pc(w,k+1);
S.D = sample_path.D(w,k+1);
S.Tinf = sample_path.Tinf(w,k+1);

%% For plotting
xSim = [xSim, S.Tb];
```

```
   uSim = [uSim, uk];
    end

    % Finding indexes of the negative demands
    [~, col] = find(sample_path.D(w,:)<0);
    % Creating "boolean" array
    DemandProductArray = S0.COP/(S0.COP-1)*ones(N+1,1);
    % Indicating negative demands
    DemandProductArray(col,:) = 0;

PathCost_i = sum(DemandProductArray(1:N)' .* sample_path.Phh(w,1:N).*uSim(1,1:N)
↪  + sample_path.Ph(w,1:N).*uSim(2,1:N) + sample_path.Pc(w,1:N).*uSim(3,1:N));
PathsCost = [PathsCost; PathCost_i];
end
Cost = mean(PathsCost);

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

% save
save = ('P02_MPC')

% Modifying values for Plotting the states and inputs for a given scenario
trial = 20;
sample_path.ukplot(1:4,:) = sample_path.uk(1+5*(trial-1):4+5*(trial-1),:);
sample_path.ukplot(5,:)= sample_path.D(trial,1:N);

for ind= 1:N
if sample_path.ukplot(5,ind)>=0
   sample_path.ukplot(1,ind)=(S0.COP/(S0.COP-1))*sample_path.ukplot(1,ind)
else
   sample_path.ukplot(1,ind)=-1*sample_path.ukplot(1,ind)
   sample_path.ukplot(3,ind)=-1*sample_path.ukplot(3,ind)
end
end

figure
subplot(5,1,1)
plot(sample_path.ukplot(4,:),'b-')
xlabel('Time, Day')
ylabel('$T^{b}$ [Deg. C]')

subplot(5,1,2)
stairs(0:N,[sample_path.ukplot(1,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{b}$ [kwh]')

subplot(5,1,3)
stairs(0:N,[sample_path.ukplot(2,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{h}$ [kwh]')

subplot(5,1,4)
stairs(0:N,[sample_path.ukplot(3,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{c}$ [kwh]')
```

```
subplot(5,1,5)
plot(sample_path.ukplot(5,:), 'b')
xlabel('Time, Day')
ylabel('Demand [kwh]')
```

E Code of problem with prediction set 3: constant prediction of underground temperature and first order Markov chain prediction of prices

```
close all
clear all
clc
tic
addpath('C:\Users\win\Desktop\Specialization Project')
load('sample_path.mat')

%% Settings
S0 = S0();
S = S0.S;
plot_settings

scenario = size(sample_path.D,1);     % Scenario number
N = size(sample_path.D,2)-1;          % Total time horizon              [Day]
nd = 10;                              % Open loop optimization time horizon

%Extention of sample path
sample_path.D   = [sample_path.D];
sample_path.Ph  = [sample_path.Ph];
sample_path.Phh = [sample_path.Phh];
sample_path.Pc  = [sample_path.Pc];
sample_path.Tinf= [sample_path.Tinf];

%% Prediction of demand and ground temp. model without exogenous information
Ad  = 90;                            % Overall domain of Demand energy  [kwh]
for i=1:N-1
    D(i)=Ad*sin(2*pi*i/N);
end
D = [S.D D D(1,1:nd)];

%% Simulation
PathsCost =[];
for w = 1:scenario
ss;
%Initial setup
S0 = S0();
S = S0.S;
%Plotting Arrays
xSim = [];
uSim = [];
timeSim = [];
    for k = 1:N
    % Exogenous information model
    % Ground temp horizon
    Tinf_horizon = S0.S.Tinf*ones(1,nd);
    %Tinf_horizon = [S.Tinf Tinf(1,k+1:k+nd-1)];

    % Demand horizon
    Demand_horizon = [S.D D(1,k+1:k+nd-1)];
    nd = length(Demand_horizon);

    % Price horizon
    Price_noise1 =  normrnd(0,S0.P1std,1,nd);
```

```
Ph_horizon=max(S0.Pmin,Price_noise1 + S0.S.Ph);
Price_noise2 =  normrnd(0,S0.P1std,1,nd);
Phh_horizon=max(S0.Pmin,Price_noise2 + S0.S.Phh);
Price_noise3 =  normrnd(0,S0.P1std,1,nd);
Pc_horizon=max(S0.Pmin,Price_noise3 + S0.S.Pc);


% Finding indexes of the negative demands
[~, col] = find(Demand_horizon<0);
% Creating "boolean" array
DProductArray = ones(nd,1);
% Indicating negative demands
DProductArray(col,:) = 0;

%% Equality constraint : Ax=b
% Demand constraint
Aeq_D = zeros(nd,6*nd);
%D_k = DPredictArray.*(1.5*U_k + Qheat_k) + (1 { DpredictArray).*(-U_k - Qcool_k)
for ii = 1:nd
    Aeq_D(ii,1 + 6*(ii - 1)) = (S0.COP/(S0.COP-1))*DProductArray(ii) + (-1)*(1 - DProductArray(ii)); % for U_k
    Aeq_D(ii,2 + 6*(ii - 1)) = (1)*DProductArray(ii) + (0)*(1 - DProductArray(ii));                   % for Qheat_k
    Aeq_D(ii,3 + 6*(ii - 1)) = (0)*DProductArray(ii) + (-1)*(1 - DProductArray(ii));                 % for Qcool_k
end
Beq_D = Demand_horizon';

%Transition function
Aeq_T = zeros(nd,6*nd);
a=1-(S0.lambda/(S0.m*S0.Cpw));
b=1/(S0.m*S0.Cpw);
for iii=1:nd
   Aeq_T(iii:end,1+6*(iii-1))= b*((+1)*DProductArray(iii) + (-1)*(1 - DProductArray(iii)));
   Aeq_T(iii:end,5+6*(iii-1))= (a-1);
   for jjj=iii:nd
       Aeq_T(jjj,1+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,1+6*(iii-1));
       Aeq_T(jjj,5+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,5+6*(iii-1));
   end
end
Aeq_T=Aeq_T + kron(eye(nd),[0 0 0 1 0 0]);
Beq_T=zeros(nd,1);
for kkk=1:nd
    Beq_T(kkk,1) = [a^kkk*S.Tb];
end

% Aeq & Beq integration
Aeq = [Aeq_D;Aeq_T];
Beq = [Beq_D;Beq_T];

%% Inequality constraints : Ax<=b
%Constraint handling : Tb-s <=Tbmax
A = kron(eye(nd), [0 0 0 1 0 -1]);
B = S0.Tbmax*ones(nd,1);

%% Objective function
```

```
f = zeros(1,6*nd);
for kk=1:nd
    f(1,1 + 6*(kk - 1)) =
    ↪  (S0.COP/(S0.COP-1))*DProductArray(kk)*Phh_horizon(kk) + (1)*(1 -
    ↪  DProductArray(kk))*0 ;
    f(1,2 + 6*(kk - 1)) = (Ph_horizon(kk));
    f(1,3 + 6*(kk - 1)) = (Pc_horizon(kk));
    f(1,6 + 6*(kk - 1)) = 30;                    % Penalizing cost
end

%% Boundaries - lower and upper
lb = zeros(6*nd,1);
lb(4:6:6*nd,1) = S0.Tbmin;
for i=1:nd
    lb(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

ub = repmat([S0.Rmax inf inf inf 0 inf]',nd,1);
for i=1:nd
    ub(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

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

%  Data store from openloop optimization
u.u     = w_opt(1:6:6*nd);
u.qheat = w_opt(2:6:6*nd);
u.qcool = w_opt(3:6:6*nd);
u.Tb    = w_opt(4:6:6*nd);
u.Tinf  = w_opt(5:6:6*nd);
u.s     = w_opt(6:6:6*nd);

%% Take an action
uk = [u.u(1);u.qheat(1);u.qcool(1)];
sample_path.uk(1+5*(w-1):3+5*(w-1),k)=uk;           % For visual check
sample_path.uk(5+5*(w-1),k)=w_opt(6);               % For visual check

%% Transition during day
S.Tb = S.Tb + [((-1)*DProductArray(1) + (+1)*(1 - DProductArray(1)))*b 0 0]*uk
↪   + (1-a)*Tinf_horizon(1);
sample_path.uk(4+5*(w-1),k)=S.Tb-273;               % For visual check

%% Realization
S.Ph = sample_path.Ph(w,k+1);
S.Phh = sample_path.Phh(w,k+1);
S.Pc = sample_path.Pc(w,k+1);
S.D = sample_path.D(w,k+1);
S.Tinf = sample_path.Tinf(w,k+1);

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

 % Finding indexes of the negative demands
 [~, col] = find(sample_path.D(w,:)<0);
 % Creating "boolean" array
```

```
    DemandProductArray = S0.COP/(S0.COP-1)*ones(N+1,1);
    % Indicating negative demands
    DemandProductArray(col,:) = 0;

PathCost_i = sum(DemandProductArray(1:N)'.* sample_path.Phh(w,1:N).*uSim(1,1:N) +
↪   sample_path.Ph(w,1:N).*uSim(2,1:N) + sample_path.Pc(w,1:N).*uSim(3,1:N));
PathsCost = [PathsCost; PathCost_i];
end
Cost = mean(PathsCost);

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

% save
save = ('P02_MPC')

% Modifying values for Plotting the states and inputs for a given scenario
trial = 20;
sample_path.ukplot(1:4,:) = sample_path.uk(1+5*(trial-1):4+5*(trial-1),:);
sample_path.ukplot(5,:)= sample_path.D(trial,1:N);

for ind= 1:N
if sample_path.ukplot(5,ind)>=0
   sample_path.ukplot(1,ind)=(S0.COP/(S0.COP-1))*sample_path.ukplot(1,ind)
else
   sample_path.ukplot(1,ind)=-1*sample_path.ukplot(1,ind)
   sample_path.ukplot(3,ind)=-1*sample_path.ukplot(3,ind)
end
end

figure
subplot(5,1,1)
plot(sample_path.ukplot(4,:),'b')
xlabel('Time, Day')
ylabel('$T^{b}$ [Deg. C]')

subplot(5,1,2)
stairs(0:N,[sample_path.ukplot(1,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{b}$ [kwh]')

subplot(5,1,3)
stairs(0:N,[sample_path.ukplot(2,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{h}$ [kwh]')

subplot(5,1,4)
stairs(0:N,[sample_path.ukplot(3,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{c}$ [kwh]')

subplot(5,1,5)
plot(sample_path.ukplot(5,:), 'b')
xlabel('Time, Day')
ylabel('Demand [kwh]')
```

F Code of optimal solution with perfect forecast

```
close all
clear all
clc
tic
addpath('C:\Users\win\Desktop\Specialization Project')
load('sample_path.mat')

%% Settings
S0 = S0();
S = S0.S;
plot_settings

n_scenario = size(sample_path.D,1);  % Scenario number
N = size(sample_path.D,2)-1;         % Total time horizon= Control Horizon [Day]


%Extention of sample path
sample_path.D   = [sample_path.D];
sample_path.Ph  = [sample_path.Ph];
sample_path.Phh = [sample_path.Phh];
sample_path.Pc  = [sample_path.Pc];
sample_path.Tinf= [sample_path.Tinf];

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

%Choosing the price policy
Ph_horizon = sample_path.Ph(w,1:N);
Phh_horizon = sample_path.Phh(w,1:N);
Pc_horizon = sample_path.Pc(w,1:N);
Demand_horizon = sample_path.D(w,1:N);
Tinf_horizon = sample_path.Tinf(w,1:N);

% Finding indexes of the negative demands
[~, col] = find(Demand_horizon<0);
% Creating "boolean" array
DProductArray = ones(N,1);
% Indicating negative demands
DProductArray(col,:) = 0;

%% Equality constraint : Ax=b
% Demand constraint
Aeq_D = zeros(N,6*N);
%D_k = DPredictArray.*(1.5*U_k + Qheat_k) + (1 { DpredictArray).*(-U_k -
↪ Qcool_k)
for ii = 1:N
    Aeq_D(ii,1 + 6*(ii - 1)) = (S0.COP/(S0.COP-1))*DProductArray(ii) + (-1)*(1 -
    ↪ DProductArray(ii)); % for U_k
    Aeq_D(ii,2 + 6*(ii - 1)) = (1)*DProductArray(ii) + (0)*(1 -
    ↪ DProductArray(ii));                  % for Qheat_k
```

```
    Aeq_D(ii,3 + 6*(ii - 1)) = (0)*DProductArray(ii) + (-1)*(1 -
    ↪  DProductArray(ii));                    % for Qcool_k
end
Beq_D = Demand_horizon';

%Transition function
Aeq_T = zeros(N,6*N);
a=1-(S0.lambda/(S0.m*S0.Cpw));
b=1/(S0.m*S0.Cpw);
    for iii=1:N
       Aeq_T(iii:end,1+6*(iii-1))= b*((+1)*DProductArray(iii) + (-1)*(1 -
       ↪  DProductArray(iii)));
       Aeq_T(iii:end,5+6*(iii-1))= (a-1);
       for jjj=iii:N
           Aeq_T(jjj,1+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,1+6*(iii-1));
           Aeq_T(jjj,5+6*(iii-1))=a^(jjj-iii)* Aeq_T(jjj,5+6*(iii-1));
       end
    end
Aeq_T=Aeq_T + kron(eye(N),[0 0 0 1 0 0]);
Beq_T=zeros(N,1);
for kkk=1:N
    Beq_T(kkk,1) = [a^kkk*S.Tb];
end

% Aeq & Beq integration
Aeq = [Aeq_D;Aeq_T];
Beq = [Beq_D;Beq_T];

%% Inequality constraints : Ax<=b
% Constraint handling : Tb-s <=Tbmax
A = kron(eye(N), [0 0 0 1 0 -1]);
B = S0.Tbmax*ones(N,1);

%% Objective function
f = zeros(1,6*N);
for kk=1:N
    f(1,1 + 6*(kk - 1)) =  (S0.COP/(S0.COP-1))*DProductArray(kk)*Phh_horizon(kk)
    ↪  + (1)*(1 - DProductArray(kk))*0 ;
    f(1,2 + 6*(kk - 1)) = (Ph_horizon(kk));
    f(1,3 + 6*(kk - 1)) = (Pc_horizon(kk));
    f(1,6 + 6*(kk - 1)) = 30;                        % Penalizing cost
end

%% Boundaries - lower and upper
lb = zeros(6*N,1);
lb(4:6:6*N,1) = S0.Tbmin;
lb(6:6:6*N,1) = 0;
for i=1:N
    lb(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

ub = repmat([S0.Rmax inf inf inf 0 inf]',N,1);
for i=1:N
    ub(5 + 6*(i - 1),1)=Tinf_horizon (i);
end

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

```
%  Data store from openloop optimization
u.u     = w_opt(1:6:6*N)';
u.qheat = w_opt(2:6:6*N)';
u.qcool = w_opt(3:6:6*N)';
u.Tb    = w_opt(4:6:6*N)';
u.Tinf  = w_opt(5:6:6*N)';
u.s     = w_opt(6:6:6*N)';

%% Take all action
uk = [u.u;u.qheat;u.qcool];
sample_path.uk(1+5*(w-1):3+5*(w-1),:)=uk;          % For visual check
sample_path.uk(5+5*(w-1),:)=w_opt(6);              % For visual check

%% Transition over a year
S.Tb = u.Tb  %S.Tb + [((-1)*DProductArray(1) + (+1)*(1 - DProductArray(1)))*b 0
↪  0]*uk + (1-a)*Tinf_horizon(1);
sample_path.uk(4+5*(w-1),:)=S.Tb-273;              % For visual check

  % Finding indexes of the negative demands
    [~, col] = find(sample_path.D(w,:)<0);
    % Creating "boolean" array
    DemandProductArray = S0.COP/(S0.COP-1)*ones(N+1,1);
    % Indicating negative demands
    DemandProductArray(col,:) = 0;

PCost_i = sum(DemandProductArray(1:N)' .* Phh_horizon .* u.u + Ph_horizon .*
↪  u.qheat + Pc_horizon .* u.qcool);
P_Cost = [P_Cost; PCost_i];
end
Avg_Cost = mean(P_Cost);

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

% save
save = ('POPMPC')

% Modifying values for Plotting the states and inputs for a given scenario
trial = 20;
sample_path.ukplot(1:4,:) = sample_path.uk(1+5*(trial-1):4+5*(trial-1),:);
sample_path.ukplot(5,:)= sample_path.D(trial,1:N);

for ind= 1:N
if sample_path.ukplot(5,ind)>=0
   sample_path.ukplot(1,ind)=(S0.COP/(S0.COP-1))*sample_path.ukplot(1,ind)
else
   sample_path.ukplot(1,ind)=-1*sample_path.ukplot(1,ind)
   sample_path.ukplot(3,ind)=-1*sample_path.ukplot(3,ind)
end
end

figure
subplot(5,1,1)
plot(sample_path.ukplot(4,:),'b-')
xlabel('Time, Day')
ylabel('$T^{b}$ [Deg. C]')
```

```
subplot(5,1,2)
stairs(0:N,[sample_path.ukplot(1,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{b}$ [kwh]')

subplot(5,1,3)
stairs(0:N,[sample_path.ukplot(2,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{h}$ [kwh]')

subplot(5,1,4)
stairs(0:N,[sample_path.ukplot(3,:), nan],'b-')
xlabel('Time, Day')
ylabel('$Q^{c}$ [kwh]')

subplot(5,1,5)
plot(sample_path.ukplot(5,:),'b-')
xlabel('Time, Day')
ylabel('Demand [kwh]')
```