Energy-efficient and reliable control of supermarket refrigeration systems

Michal Kujawski, Waleed Aslam
Intelligent Reliable Systems, IRS4, 2024-06

Master's Thesis

STUDENT REPORT
AALBORG UNIVERSITY

AALBORG UNIVERSITY
STUDENT REPORT

Energy Technology
Aalborg University
http://www.aau.dk

Title:
Energy-efficient and reliable control of supermarket refrigeration systems

Theme:
Master's Thesis

Project Period:
Spring Semester 2024

Project Group:
IRS4

Participants:
Michal Kujawski
Waleed Aslam

Supervisor:
Zhenyu Yang

Page Numbers: 110

Date of Completion:
June 3, 2024

Abstract:

Thanks to large consumption of energy and thermal capacity of food stored in supermarket refrigeration systems, there exists a potential for optimising their use in terms of costs of energy, by exploiting changes in electricity prices. While moving the system's load according to the prices can yield nontrivial savings, it has to be done in a way that respects operating constraints, especially food temperature. For that reason, 2 model predictive control schemes under varying electricity prices, including their reliability under temperature sensor fault, are presented in this thesis. As supermarket refrigeration systems are not suitable for first-principle modelling, data-driven methods, namely novel dynamic mode decomposition and neural networks are investigated for obtaining prediction models of compressors' power and evaporation temperature for use in the controllers. To obtain the synthetic data for training models and perform simulation studies of the controllers, digital twin simulation models provided by Danfoss were used. Finally, the results demonstrate that the developed neural network model can be successfully used with the developed control schemes to decrease costs of energy by between 6.8% and 14.1% depending on the scenario.

Contents

List of Figures

List of Tables

Summary

Supermarket refrigeration systems hold a potential for economic optimisation of their operation, as thermal capacity of food that they store allows for moving load in time according to periods when the electricity price is lower, thus decreasing the overall cost of energy. However, this precooling strategy does not necessarily mean that less energy is used.

In fact, cooling more at some time point to cool less at a later time can decrease coefficient of performance of the system, as the efficiency of compressors will drop as they have to deliver more power for cooling, while the strategy to minimise only the overall energy consumption would be to limit compressor power such that temperature of food always stays at upper permissible limit. Thus, "energy-efficiency" in the title of the thesis is treated a bit lightly, meaning that we want to use energy flexibility of the system to improve its cost efficiency.

Nevertheless, it is still a worthwhile goal to utilise cheap energy prices, as it benefits both supermarket owners and electrical grid, that will require more balancing, as the share of renewable energy increases, that can be provided through implementing load-shifting strategies in supermarkets. Throughout literature study, the authors found out that model predictive control (MPC) is a suitable way of achieving this goal, as it offers ways of optimising cost of system's operation, while maintaining its constraints like temperature of store food etc..

Prediction model used by this scheme plays a pivotal role in it being successful or not and so does choice of modelling approach to obtain it. Previous studies trying to optimise the system's operation used impractical approaches relying on first-principles modelling, that is not possible in practice, due to substantial part of the system's properties being unavailable, or manipulating cabinet temperatures that results in large amount of signals that need to be predicted. Therefore, data-driven approaches including dynamic mode decomposition (DMD) and artificial neural networks (ANNs) were explored, with focus on manipulating compressors rack's power and evaporation temperature, as they are few central signals that affect both system's overall power and cooling.

Ideally, DMD could be used to obtain powerful linear state-space representations even for nonlinear system, that allow for fast computations by MPC. However, due to implementation and time limitations, only ANN approach was applied to the predictive controllers, to demonstrate validity of our modelling procedure. Upon obtaining the model, 2 MPC schemes were designed, economic MPC that decided on the optimal power consumption based on electricity prices on its own and evaporation temperature reference-tracking MPC that was operating in 3 modes, namely minimum, normal and maximum energy consumption that were commanded by the grid to the system within 10 minutes interval, both of which achieved savings in terms of energy costs i.e. at most 6.8% and 14.1% respectively. We claim that these savings would have been more, if it had not been for short time constant of cabinet temperature in the digital twin simulation model provided by Danfoss, which used as a tool for for obtaining training data for the data-driven model and conducting simulation study on the developed controllers, as it did not allow for much precooling.

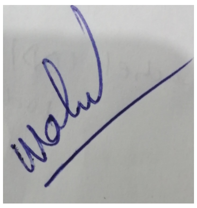
Lastly, reliability of the system was also considered, as the developed methods, among other things, rely heavily on accurate measurements of ambient air temperature. A method for addressing the sensor's bias inflicted by sun exposure available in the literature was validated on the available digital twin model and its application to our solution was justified by investigating impact of bias on it, that rendered it completely unusable, owing to cabinet temperature lower constraint violations caused by providing much higher temperature than actual to the prediction model, that led to increased cooling. By applying the mentioned method for bias compensation, the performance of MPC schemes could be recovered.

In the overall conclusion of the thesis, it is stated that while our development was successful, there are some areas to improve in the future, mainly time constant of the digital twin model, to increase the possible cost savings and modify the controllers accordingly, and simplifying data driven model, for instance by switching to DMD, to make the controllers less computationally expensive.

Preface

We would like to thank our supervisors Zhenyu Yang (AAU Esbjerg) and Roozbeh Izadi-Zamanabadi (Danfoss) for their support towards our development in this thesis, as well as team at Danfoss that provided digital twin simulation models without which training of the presented data-driven models as well as simulation study of developed controller methods would not be possible.

Michal Kujawski & Waleed Aslam,
03.06.2023



Chapter 1

Introduction

Refrigeration systems as a whole are major users of electric energy, accounting for approximately 20% of its consumption globally [1, 2]. Supermarket refrigeration systems contribute substantially to this number, as together with heating, ventilation and air conditioning of supermarkets account for 3-4% of overall electricity consumption in industrialised countries [3]. In Denmark alone, 4500 supermarkets consume 550 000 megawatt-hours of energy every year and currently could be used together to account for 75% of primary operating reserves of the electric grid [4].

This is due to the fact that, owing to their size, supermarket refrigeration systems store large amount of energy, which also makes them suitable for manipulating their operation to achieve cost savings, as, for one thing, this storage allows for accumulating enough energy in form of "coldness" to be used later and [5], for another thing, nontrivial cost savings can be achieved by doing so appropriately [4, 6], creating an incentive for the owners of such systems to invest in solutions optimizing their use. Therefore, it is interesting to investigate ways of utilizing energy capability of these systems in more cost-efficient manner.

There are 2 main approaches to economic optimisation of the supermarket refrigeration system i.e. improving the system through extending its design with new components that reduce operating costs by themselves and adjusting its operation according with changing external factors like electricity prices or ambient air temperature that affects the energy consumption. The former can involve heat exchanger that recovers excess heat produced by the system for district heating, ice tank for extra cooling on the condenser side to reduce peak energy consumption in hot climates or photovoltaic panels for powering compressors.

While all of them are beneficial on their own, the question of how they can be used in the best way arises. For instance, one can ask what the best time to sell excess heat could be or when ice tank should be charged or discharged to achieve the best peak energy consumption reduction. This leads to the point that the system requires the latter approach to its optimisation in any case. Because of this reason, this thesis focuses on improving the system through an optimal control problem formulation, given that supermarket refrigeration systems are dynamic systems.

As proper control of the system in terms of cost of operation is a prerequisite for applying extensions to the system's design, we state that problem initiating our research is that supermarkets are in need of a way to optimising economic operation of their refrigeration systems to enable costs savings both for nominal operation and one while utilising new energy-efficiency-related components.

Chapter 2

Problem analysis

2.1 State of the art

In a broader context of energy cost optimisation, demand side management (DSM) is a set of approaches that balance electrical grid by managing its consumer side through altering consumers' load profiles and patterns of energy consumption [7]. Demand response is a prominent DSM method that focuses on consumers actively changing their consumption based on dynamic electricity prices [8].

Thermal systems are especially suitable for such strategies and considerable number of publications on the topic has been seen for building heating, ventilation and air conditioning in particular [9, 10, 11], where a commonly used tool is Model Predictive Control (MPC). However, the previous research in supermarket refrigeration systems focused mainly on the performance of the system from operation rather than energy perspective, for instance to desynchronise switching of on-off expansion valves to avoid negative impact on compressors [12, 13, 14], while still commonly resorting to MPC.

Nevertheless, there has also been a recent interest in applying MPC to optimise costs of energy consumption of supermarket refrigeration systems. Different MPC approaches to managing energy consumption of supermarket refrigeration systems are propose in thesis by Shafiei [6]. Hovgaard et al. [5] proposed economic MPC for direct load control of cold room participating in Smart Grid, where direct load control scheme is utilised such that the cold room follows power reference decided by the grid by varying compressor power, while maintaining cold room temperature constraints. Also, interesting observation is made that the goal is to minimise the cost of energy rather than energy itself, as minimisation of energy occurs when the system operates at upper temperature constraint at all times, which does not allow for absorbing cheap, or even free energy from its overproduction, making it possible to decrease cooling at a later point. As an alternative approach Hovgaard et al. [4] developed EMPC for indirect load control, where EMPC was optimising energy consumption locally in the supermarket refrigeration system, managing its consumption on its own. The limitations of this work was simulation model which was first-principle and simplified, which was used mainly for demonstration potential of energy-costs savings of capabilities of EMPC. In [15] a methodology of stochastic EMPC for supermarket refrigeration system was also created to account for uncertainties in electricity production and ambient air temperature predictions.

As for modelling of refrigeration systems, prior work was mostly on modelling of steady-state operation of the system for static optimisation [16, 17, 18]. Grey-box modelling of domestic refrigerator was also presented [19], while the mentioned work by Shafiei included both white-box modelling and subspace identification approach [6]. As the authors are not aware of any publication on modelling dynamics of supermarket refrigeration systems for energy cost optimisation through dynamic mode decomposition or artificial neural networks, these developments are considered as one of the contributions of this thesis.

2.2 MPC approach

Given the literature study above, we can see that MPC is particularly popular in the research area of optimisation of thermal systems' control and it is for some good reasons. They are given as a justification of using this approach in our thesis and we also present our strategy for applying it in our case.

The MPC approach was chosen as a preferred control method, due to its 2 properties: intrinsic way of introducing constraints and ability to predict future. The ability to enforce constraints is most crucial, as it is not desired to go beyond the safety bounds of cooled cabinet temperatures that would jeopardise food safety. This is very important, as the most profitable operation will occur while allowing the temperature to get close to these bounds. This is also mainly the reason why there is room for optimisation: currently used fixed setpoint of suction pressure works well for different cooling loads imposed on system by ambient air temperature such that enough cooling is always delivered, even in the worst case. However, this does not lead to the best use of the system when it is not being under full-load condition, as in such a case departing from the fixed setpoint is not allowed, which goes against idea of varying it to match the actual conditions. Nevertheless, thanks to constraint-enforcing ability, the MPC approach is capable of maintaining the temperature in the acceptable range during the ever-changing conditions.

To achieve that, another useful property of MPC is still needed i.e. capability of incorporating forecasts. Unlike PID control, it is capable of utilising extra information that is already available for the supermarket refrigeration system i.e. ambient air temperature and price forecasts. As MPC tries to see the future impact of its actions on the manipulated variables through simulation of the system, it is also capable of incorporating such anticipative information into its predictions, thus resulting in control action that from perspective of PID would be non-causal, like responding to disturbance before it actually occurs. This leads to ensuring that even if sudden change in ambient air temperature, which changes the cooling load occurs, the system will still stay within constraints.

The strategy taken by us in using MPC in this thesis is to utilise it as supervisor that decides references to compressor suction pressure, which determines the cooling available to the system to minimise energy costs for given ambient air temperature and price, while maintaining cabinet temperature constraints, which is illustrated in figure 2.1. The benefits of applying MPC in a supervisor manner is that the local controller guarantees stability of the system and the MPC can provide the references at much slower sampling rate than the one of local controller, both of which are important when applying data-driven black-box models. This need will become apparent later when discussing how the prediction model for MPC scheme can be obtained for the refrigeration system.

2.3 Supermarket refrigeration system

To contribute to reader's understanding of the project and knowledge about the system, major components of the supermarket refrigeration system, as seen in figure 2.2 and their purpose are described below. In addition, possibility of manipulating control of each part of the system will be mentioned.

The explanation of the system begins with its low temperature, low pressure side, or simply the part that absorbs the heat. Rack of compressors is the part of the system that generates cooling required to remove that heat, by controlling pressure of the refrigerant at its inlet, which at the same time is outlet of evaporators, components that exchange heat with air in cooled cabinets and evaporate the refrigerant. The evaporation temperature of refrigerant is affected by the suction pressure of the compressor, where lowering the pressure lowers the evaporation temperature and increases temperature difference between refrigerant and air in cooled cabinets, resulting in allowing for lowering their temperature or, equivalently, increasing available power for absorbing heat from the system. This means,

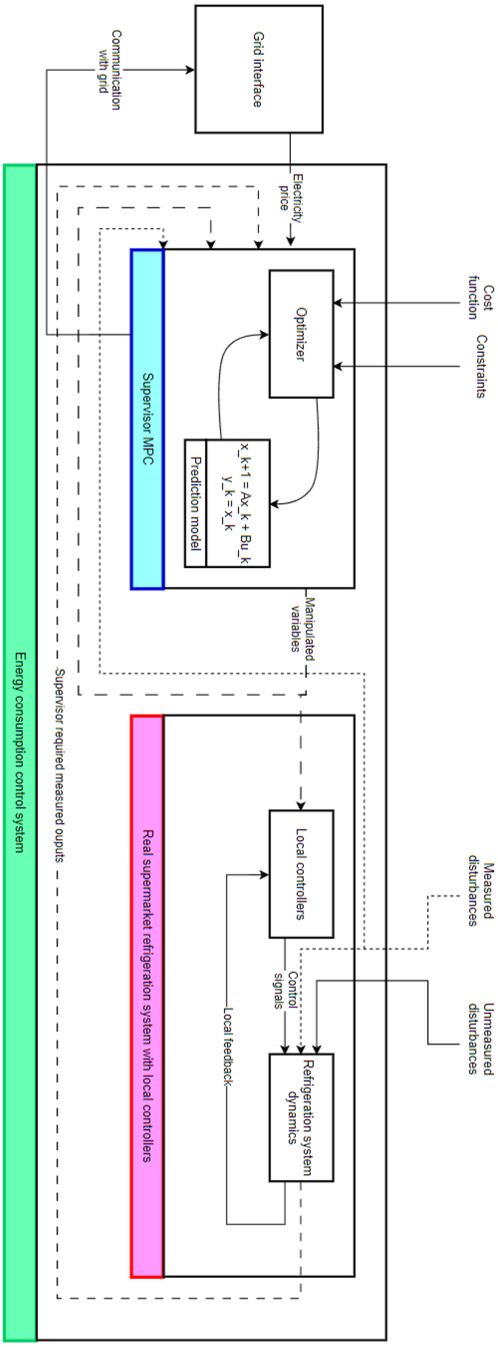


Figure 2.1: Generic diagram of EMPC for optimization of supermarket refrigeration system energy consumption [20]

we can lift suction pressure of the compressor to decrease its power at the cost of raising temperature in the cabinets, due to lowering cooling power. This is an approach that is taken in this thesis to manipulate energy consumption of the system, which is explained throughout it.

Important thing to notice at the cool side of the system is that each cabinet has different temperature requirements based on food that is being stored inside of it. It is reasonable to state that diary products or meat have lower temperature requirements than cooled fresh vegetables. This is assured by electronic expansion valves that control the temperature inside the cabinets through manipulating flow of the refrigerant. This way, different temperatures are achieved, even though the evaporation temperature is the same for all of them. However, an extra set of compressors is present for freezers, as they have much lower temperature requirements creating a need for lowering the evaporation temperature. As it would be impractical to manage this range of temperatures, extra compressor rack allows for separation of pressures for the freezing and cooling cabinets. These 2 separate parts of the system are called low temperature (LT) and medium temperature (MT) respectively. At this point, it can also be noted that manipulating cabinets' temperature references could be used instead of changing compressors' suction pressures directly to mange the system's power consumption, as they would change the need for cooling directly, so compressor could just match the pressure to this need. However, the opposite approach (temperatures following the available cooling delivered by compressors) is more practical, as managing each cabinet's temperature would require a lot of processing, given that in practice there are tens of them in the system.

During compression, compressors move the the gas refrigerant to their outlets and increase its pressure, which is a desired effect. Lifting the pressure allows for easier heat dissipation, as temperature rises with pressure leading to larger temperature difference between refrigerant and surroundings, which causes more heat transfer. However, there is some optimal configuration of the pressure at which heat is exchanged that results in best coefficient of performance (COP) for a given ambient air temperature. Thus, supermarket refrigeration systems use a special algorithm for calculating the reference to pressure controller for maintaining the right pressure, which should not be altered [22]. This is achieved through high pressure valve that drops the pressure of the refrigerant. The cooling at the right pressure is achieved at gas cooler, which is installed outside of the supermarket, due to amount of heat that needs to be removed, because of which ambient air temperature decides how easy it is to get rid of it. In order to control refrigerant's temperature, it uses fan to cause ambient air circulation through it.

After passing through high pressure valve, low temperature refrigerant can be

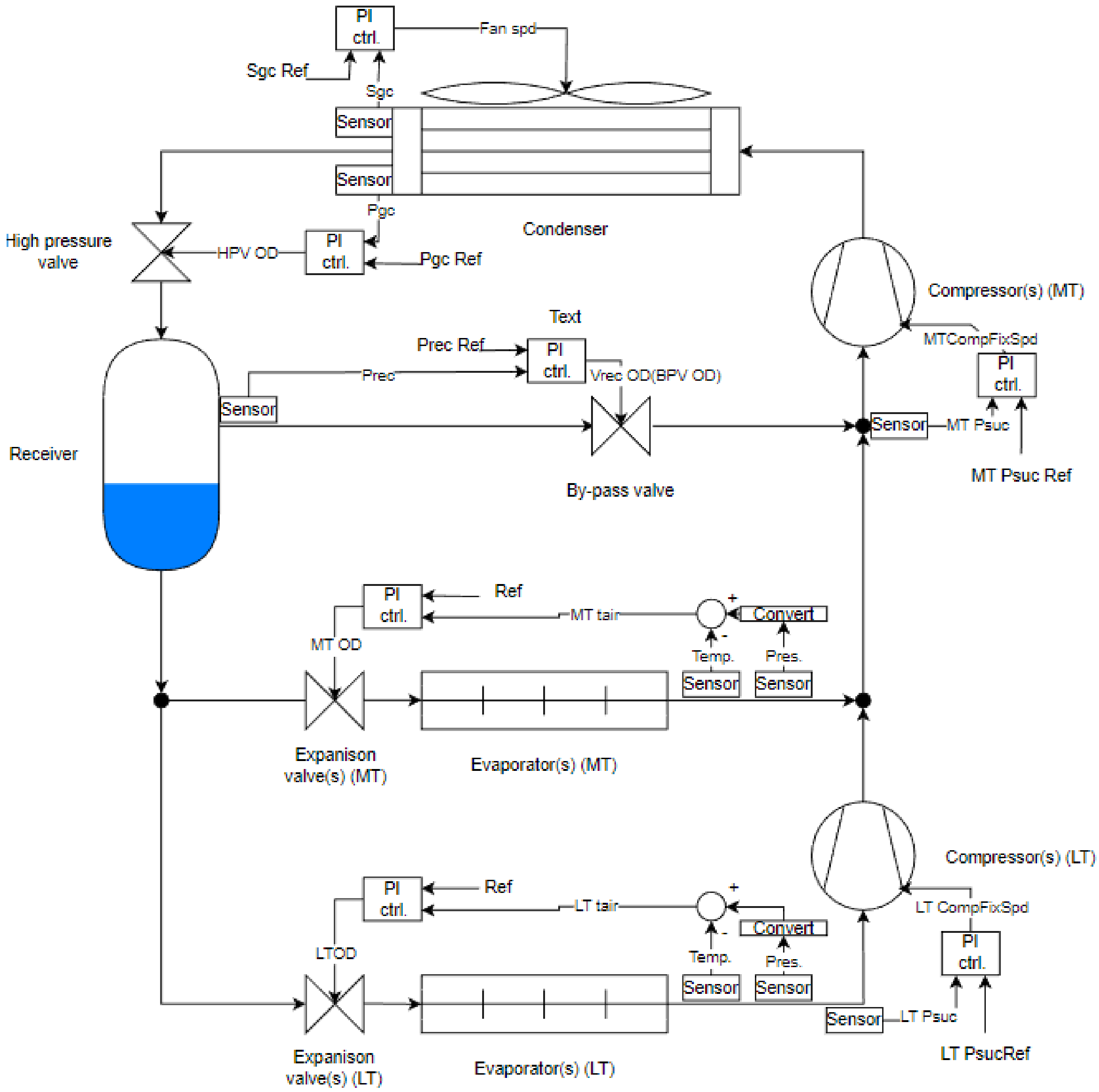


Figure 2.2: Supermarket refrigeration system components diagram [21]

reused. Nevertheless, it has to pass through a receiver, which purpose is to both keep the excess refrigerant whenever there is lower demand for cooling and to separate gas and liquid phase of the refrigerant, before going into evaporators. As it is best to cool with liquid refrigerant and to maintain right pressure at the receiver, a by-pass valve is installed that leads directly from receiver to the MT compressor rack, skipping all of the evaporators. Once liquid refrigerant in the receiver enters evaporators, its cycle through the system repeats.

Based on this explanation we can conclude that the only two parts of the system that can be manipulated for managing its power are evaporators and compressor, where compressors are feasible for real implementation, so signals related to their operation should be investigated.

2.4 Data-driven modelling approach

As seen in the previous section, the supermarket refrigeration system is complex, so there is an inherent difficulty in obtaining all of the parameters needed for first-principle modelling with sufficient accuracy. To make it worse it becomes almost impossible given that every supermarket refrigeration system is custom-made, as most often it is constructed by different local suppliers resulting in lack of standard for its modelling. This naturally calls for data-driven modelling.

This section does not focus on any specific method yet, however, the overall approach is presented, especially the reasoning behind choosing particular inputs and outputs to be used with any of the later considered methods. It was already suggested before that the model should be simple and avoid impacting local control loops directly to account for stability of the system. Based on the previous section, it appears that signals related to rack of compressors are potentially a good choice when it comes to keeping amount of signals in the model to the minimum. This is due to the fact they describe the overall cooling of the system well, due to the compressors being central components delivering cooling.

The following 2 signals were considered as output of the model: compressors power and evaporation temperature, as they both define necessary information about the cooling of the system i.e. power determines overall energy consumption of the system as they account for 80% of it, while evaporation temperature defines the minimum temperature for all of the cabinets for a given ambient air temperature. Therefore, these 2 signals can be used together to achieve 2 goals: improvement of energy utilisation and maintaining of the cabinet temperatures.

Initially, coefficient of performance (COP) was also considered as an output, but in the end it was unnecessary. On one hand, it is already optimised through gas cooler pressure reference generator. On the other hand, we might even be interested in destroying COP, as we might be interested in increasing the cooling, thus decreasing efficiency of the system, when the electricity prices are low. As COP did not serve any other purpose than to be optimised itself, it was removed from the modelling approach.

As for the inputs, ambient air temperature, suction pressure and gas cooler pressure were selected. Ambient air temperature obviously has an impact on how easy it is to reject heat from the system and is the main factor driving the energy consumption, so it must be included. Suction pressure is know to affect both power and evaporation temperature, as lower pressure lowers down evaporation temperature resulting in more cooling and increases power, as more refrigerant is sucked by the compressor. It may seem that these 2 signals are sufficient and gas cooler pressure is not needed to be added, as it is obtained based on ambient air temperature through its reference anyway, however, from out experience the methods data-driven methods could not perform well without it. The reason might be that somehow it makes it easier for them to capture compression ratio, which also decides how hard compressors need to work given specific suction pressure. Consequently, gas cooler pressure was also included.

The scheme presented in this section should be sufficient for producing simple yet accurate models for managing energy cost-efficiency of the system while maintaining the constraints.

2.5 Reliability considerations

Due to its size and complexity, supermarket refrigeration system can naturally be subject to different faults. Thus, it is interesting to investigate if the control systems developed in this thesis can be affected by any sort of fault.

As these methods rely on accurate and fault-free measurements of ambient air temperature, just like in case of calculation of gas cooler pressure reference, a common fault that can negatively impact their performance is slowly-varying bias in ambient air temperature sensor caused by mistakenly placing it in a spot exposed to sun during the day. Because of the fact that the amount of bias added on top of the measurement can reach up to 10 degrees Celsius [22], it can be easily seen that it is a problem relevant to the aforementioned methods, as such a difference of temperature is large enough to vary the the season perceived by the model from spring/autumn (for instance 15 degrees Celsius peak) to summer (25 degrees Celsius with bias added). This in turn will drive the solution of MPC/EMPC and consequently the system away from optimal solution and possibly to the infeasible region, causing cabinet temperature violation.

Therefore, this important consideration of reliability of the proposed control methods should be also taken into account during the development carried out in the thesis with respect to finding out how much this fault affects the system and possibly applying a compensation method to address it.

2.6 Task statement

Given the analysis above, the following research question for the thesis together with the required tasks are posed:

How can energy cost-efficiency of supermarket refrigeration systems be improved with model predictive control manipulating suction pressure reference?

Scope:

- The thesis shall focus on optimisation of the default supermarket refrigeration system through MPC, extensions to the system are optional.
- The considered system uses CO2 as a refrigerant.

Objectives:

- A simple but highly accurate data-driven model has to be developed for predicting signals essential to cooling and power consumption caused by it.
- MPC needs to be developed such that it minimises cost of energy usage in the system based on the developed model and predictions of varying price and ambient air temperature.
- Possible reliability issues of the control schemes with respect to ambient air temperature sensor should be investigated and addressed.
- The ability of the developed system to save cost of energy should be evaluated.

Limitations:

- Data for training data-driven model will be synthetic output of a digital twin model.

- Only MT stage of the system is considered (limitation of the model).
- No lifting of suction pressure during the night is considered (both simulation model and time limitation).
- Solution will be trained / evaluated on a limited range (10-35 degrees Celsius) of ambient air temperature (limitation of simulation model).
- Perfect predictions of ambient air temperature and electricity price are assumed.
- Real price data will be modified to have more negative prices, which leads to less realistic comparison (limitation of the simulation and time)
- Same price data will be used for different scenarios for ease of comparison; the scenarios will span 1 week each.

Chapter 3

Simulation process

Previous chapter explained that supermarkets are too custom to have a standard white- or grey- box model. In order to train and test data-driven models to be used by MPC instead, a lot of data needs to be acquired. However, it is impractical to perform these tests on a real setup, due to a large number of different scenarios that must be tested, which would be costly and time-consuming, considering time scale (order of hours) at which refrigeration systems operate.

Fortunately, this can also be performed on accurate simulation models. Danfoss provided the authors with 2 such models, which are be briefly explained. As preparation of data and code for simulations took substantial amount of time, this effort is also documented here.

3.1 Simulation models

Each of the models that were introduced to the authors have their upsides and downsides. The following section describes their complexity and argument is made for using faster yet less 'realistic' model.

3.1.1 Digital Twin

The initially provided model was a Digital Twin of one of typical Danish supermarkets. The simulation file includes both supermarket refrigeration system's dynamics 'hidden' inside functional mock-up unit (FMU), which was exported from Modelica/Dymola simulation environment where the dynamics were originally modelled, as well as default control systems, built in Simulink, used for managing temperature and pressure references across the refrigeration system. A screenshot

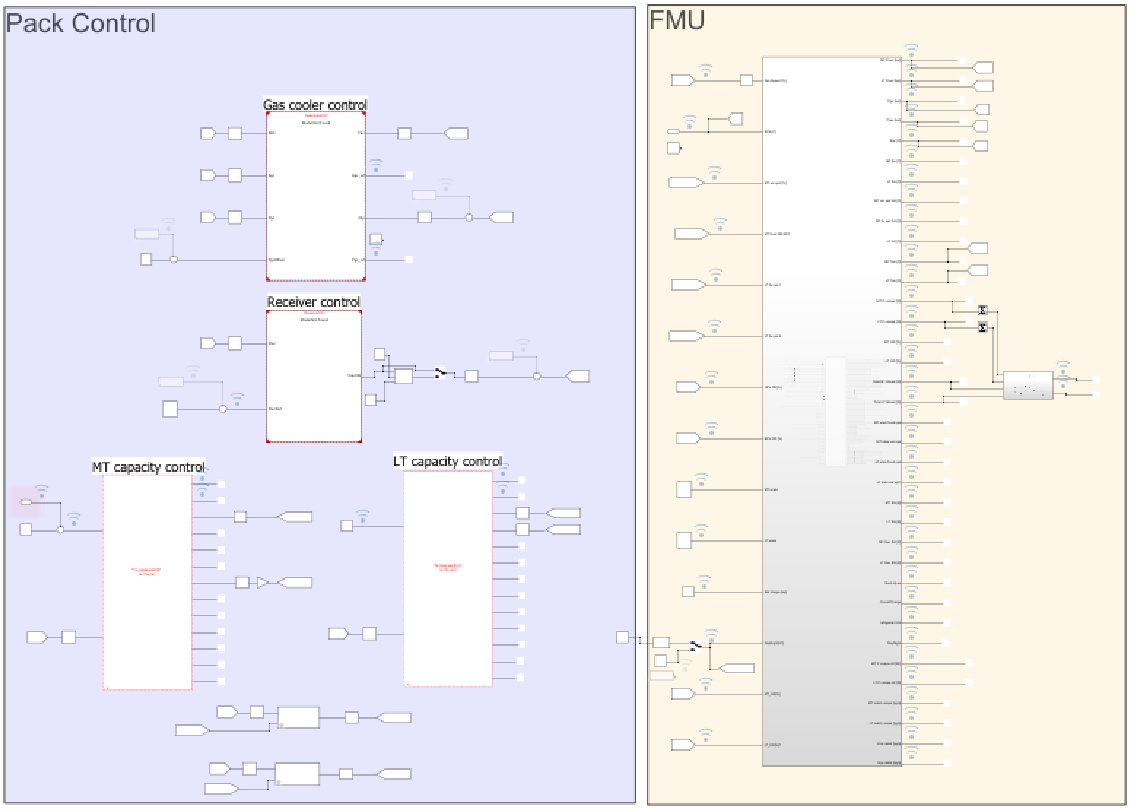


Figure 3.1: Main block in Digital Twin

of Digital Twin can be seen in figure 3.1.

The advantage of Digital Twin is high accuracy of data and realistic reconstruction of the real system i.e. every single major component of the system like expansion valve or compressor is simulated by the model as a separate part. This includes 7 MT expansion valves, 4 LT expansion valves and 2 compressors in each (MT and LT) compressor rack, among other components, which in total generate over 60 signals.

The main disadvantage of the model is simulation speed. Due to being modelled using Modelica and large number of components, simulating 1 day of the system's operation takes between 4 and 5 hours of real time on a modern office laptop. Because of using FMU instead of Simulink blocks, it is also impossible to study the system's inner workings by observing how calculations are being made such that better understanding of its operation could be gained. Nevertheless, the control systems can be freely inspected and manipulated in order to affect system's excitation.

The Digital Twin was used extensively by the authors to obtain data at the start of the project, which unfortunately was causing a substantial bottleneck in the process of developing models, where multiple scenarios had to be obtained, due to its simulation speed. As the demand for simulation work grew over time, a faster model was introduced to the authors, which is described next.

3.1.2 Fast model

The 2nd model, here referred to as a "fast model", represents system similar to the system Digital Twin simulates, however, it was optimised for speed at the cost of some details in the model. For instance, it represents all components in MT stage by single evaporator, expansion valve and compressor. It also does not contain freezers (LT stage). Nevertheless, it still remains quite complex as depicted in figure 3.2.

The execution speed of the model is indeed a considerable upside. As claimed by the authors of the fast model, it runs 100 times faster than the Digital Twin, which is in fact true, as the fast model simulates one day of operation in a time range of 2.5-3 minutes on the same computer as in case of the Digital Twin. This served as a very useful upgrade, because it allowed for checking different tests, designs and ideas almost instantaneously. For example, the Digital Twin required 5 days of running the simulation continuously to obtain simulation of 1 month of operation. Meanwhile, the fast model can simulate year of operation within less than a day.

On the other hand, increasing execution speed of the model was to certain extent caused by reduction in number of components included in it. It should be noted however, that from perspective of developing supervisor for improving energy efficiency is not a major issue, as it does not have to take into account some details, owing to supervisor considering the system as a whole. As an example, refrigerant mass flow is important variable linked to COP, yet it does not matter if it is coming from multiple evaporators / expansion valves or just one bigger component representing all of them. Similarly, compressor power models power of all compressors in the compressor rack, so they can too be represented by a single component.

Due to the fact that fast model already has an ice tank model connected to it (which can be easily disabled to obtain simulations without its effect) that could be used as an extension and as the simulation of Digital Twin would not allow for finishing simulations in the right time, the authors continued working with the fast model and abandoned the Digital Twin for further simulation work.

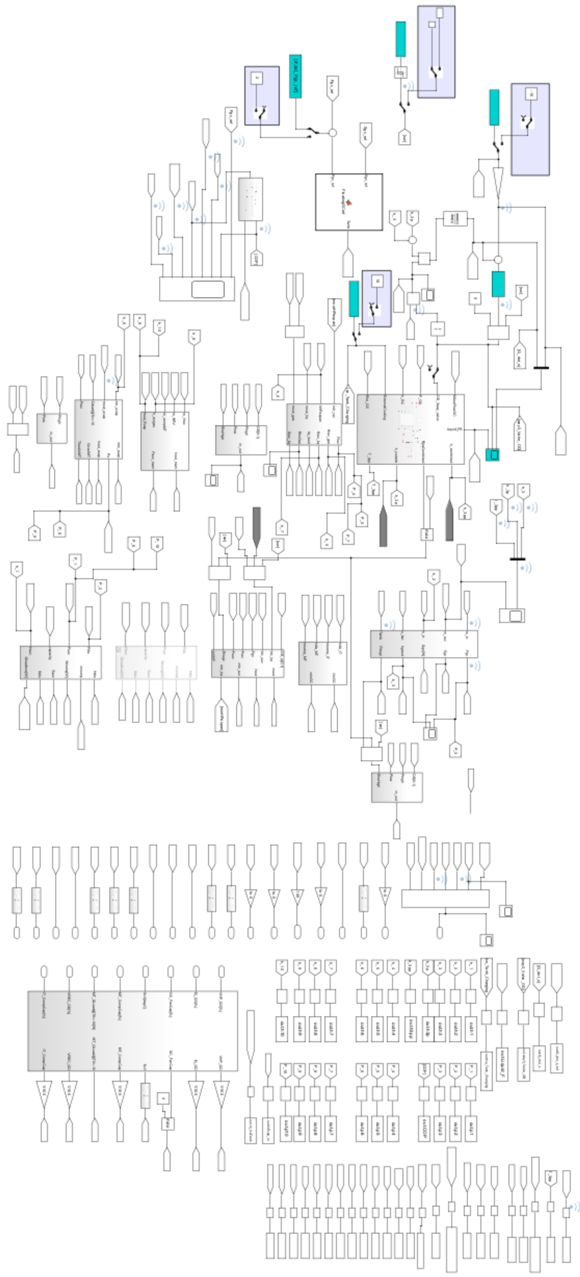


Figure 3.2: Fast model and its complexity in terms of internal signals

3.2 Simulations plan and organisation

As the amount of simulations and variation of signals used in them were extensive, the simulations had to be carried out in a systematic and organised manner. In this section, approach to achieving this with a simulation plan and MATLAB scripts is presented.

3.2.1 Simulations plan

In order to have sufficiently rich data as for both examples of operating conditions and dynamic excitation a lot of simulations had to be run. To avoid running them in a chaotic manner, it was decided how the setup of simulations in terms of inputs should be beforehand.

To begin with, ambient air temperature data for different seasons were needed, as it has substantial impact on how the system behaves. Initially, the data from consecutive days of selected months were considered. However, it turned out not be a good idea, as the data may not be sufficient to train a network based on seasonal data only, as a sudden spike of temperature throughout one day as compared to the others in a testing dataset would cause the network to perform poorly for that period (this is discussed more in-depth in the next chapter).

To resolve this, the temperature data for 5 specific temperature ranges for the network development were initially selected manually instead, such that network would get the data of similar temperatures rather than consecutive days. To make sure network learns the correct relationship between its outputs and the temperature, several temperature profiles were prepared for each range.

Next factor was suction pressure. In order to provide the network with sufficient amount of operating ranges, simulations were run at different references of suction pressure with several realisations of PRBS applied on top of them. To get the full scope of operating conditions, simulations were run for each combination of temperature range and suction pressure. Each of this combinations had 10 variants called days. Each day contained one combination of PRBS with temperature profile.

In addition, there were few other conditions taken into consideration. Temperatures had to be considered for different climates, so that the system could be developed and tested for different locations in the world, resulting in the increase of the number of datasets. Another potential option was running simulations with and without ice tank.

In total, a large number of simulations was acquired. The system was simulated for 4 temperature ranges from 2 different climates, resulting in 400 simulations (4 temperature ranges x 10 suction pressure references x 10 days = 500 runs), which resulted in large quantity of data. Initially, 50 simulations would occupy 6 gigabytes of disk space, so unnecessary signals were not logged afterwards to avoid problem of having too much data. By doing so, the final 400 simulation runs took "only" 22.3 gigabytes of disk space. This was further reduced by preprocessing, which took only signals used for modelling and downsampling them, shrinking down original size of the full data set to 2 megabytes of data of the resulting data set.

3.2.2 Preparing environment for simulations

To carry out this large amount of simulations a programmatic approach was required to avoid repetitive entering of simulations inputs and running them as in manual approach, that would obviously take a lot of time to do and would lead to high risk of errors.

The core functionality is already implemented in function sim in MATLAB/ Simulink, which is capable of managing passing appropriate inputs to the simulations and saving them each in separate files (it has to be noticed that given the amount and length of simulations one can easily run out of memory in workspace, so saving to files is preferred). An alternative version of sim function called parsim also seamlessly implements the same functionality on multiple processor cores, which allows for speeding up the simulation process when a lot of separate simulations are run with no added effort, assuming access to Parallel Computing Toolbox.

Nevertheless, regardless of use of either sim or parsim, some extra functionality needs to be built on top these functions. This involves proper understanding of how data is passed to these functions and programming a loop that can insert them in an organised manner, according to the simulation plan. In addition, input data also had to be acquired or generated. For one thing, PRBS needed to be created. This was done with prbs function in MATLAB, however, it had to be further processed to be an appropriate signal with required properties like sampling time or being injected at specific time in the simulation (for instance, after it settled from a start up transient).

For another thing, ambient air temperature data had to be acquired. In order to get realistic representation of the temperature, weather measurement data was sought for online. However, most data bases had an issue of either having raw, unprocessed data, storing it in hard-to-process format or made it difficult to easily extract the required data from a bigger dataset. Fortunately, after an extensive search, Iowa Environmental Mesonet website was found, which provides an ever-growing weather data archive from airports around the world under the link: https://mesonet.agron.iastate.edu/request/download.phtml. This web page offers a user interface for specifying what kind of data should be imported and how it should be formatted, which is very useful in conjunction with the fact that data from any country can be obtained, allowing for developing and testing of methods presented in this work for different climates.

Considering the above, a lot of effort was required so as to get the simulations running, also taking into account that designing environment for simulations was an iterative process, in which the simulation environment was reconfigured as new issues occurred and the demand for new datasets was growing.

3.2.3 Simulations data organisation

Last aspect to be covered is how data was organised in the files. The problem here was that parsim function saves consecutive simulations datasets in files with same name, which differ only with a number indicating simulation run number added at the end of the name.

The solution to this was using the knowledge of order in which the simulations were run and set of conditions that specific run was associated with. A loop was created that went through all of the simulation dataset files and renamed them using the information that was used for generating the data for each simulation. The dataset files naming format is explained in figure 3.3. The colored letters stand for values of different indicators that carry information about how the simulation was configured. They include city for which data were obtained, whether or not ice tank was connected to the system (left for future work), temperature level profile used, level of suction pressure and day i.e. specific realisation of the temperature profile and PRBS. The values of numeric indicators are integers in the range given by square brackets.

This naming is useful not only for others to use / recreate the datasets, but also allows for easy extraction of specific datasets that should be used for designing neural network. This can be achieved by running a loop that extracts the dataset files based on the specifiers in their names.

The extent of simulation tasks presented in this chapter shall be useful for whoever would like to recreate the results, as it provides their complexity and guideline how to complete them.

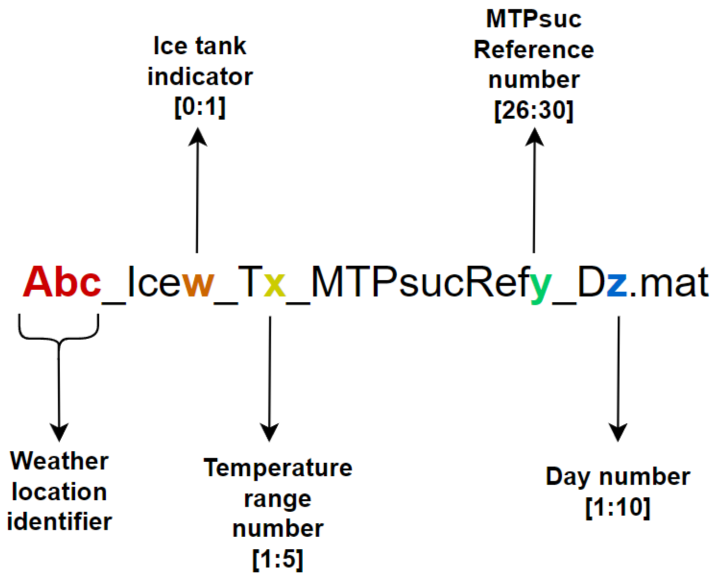


Figure 3.3: Dataset naming format

Chapter 4

High-level modelling of supermarket refrigeration system

In order to optimize the cost of system's operation with MPC, an internal model of system's dynamics is required for predicting the system's behaviour for optimal input selection. To make the model as simple as possible, its acquisition was carried out at the system level, considering main factors that affect overall power and cooling capability of the system. This was required to minimise computational burden while employing machine learning models, as explained in the problem analysis. The data driven methods considered for the modelling purpose are presented below, together with the full development of modelling approach for supermarket refrigeration systems.

4.1 Considered modelling methods

4.1.1 Dynamic Mode Decomposition

Initially Dynamic mode decomposition (DMD) methodology was adopted by authors in [20] to model the dynamics of considered supermarket refrigeration system, however, due to some limitations of the DMD methodology (described at the end of this section) for the considered system the authors also considered alternative methodologies (e.g. ANNs) for modelling the system dynamics.

Dynamic mode decomposition (DMD) has become popular in recent years, due to its ability to model the dynamics of linear and non-linear systems in an innovative way. The technique has several advantages over more complex modelling methods such as artificial neural networks [20]. Some of these are described below.

- One of the main advantages of the DMD technique is the simple way of identifying state space model for multiple input multiple output (MIMO) system. The data needs to be arranged in the form of a matrix to perform singular value decomposition (SVD) which is the basic step of DMD.
- DMD can be used to model the dynamics of both linear and non-linear systems alike. For linear systems, the DMD technique produces the model of system dynamics precisely, however, for non-linear systems the technique requires that state representation of the considered system should be extended to estimate the Koopman operator. This representation uses the linear combination of non-linear states instead of non-linear combinations of linear states which enable the controllers (e.g. MPC) to compute faster as compared to the case of non-linear representation.

DMD was originally developed as a dimensionality reduction technique for fluid dynamics. Using the concept of proper orthogonal decomposition (POD) it estimates the eigenvalues and eigenvectors of A matrix of a dynamic system

$$x_{k+1} = Ax_k$$

based on the provided data. The POD is performed using the singular value decomposition (SVD) technique. In the case of linear systems the DMD eigenvalues represent the eigenvalues of a linear system whereas in case of non-linear systems it approximates the eigenvalues of Koopman operator [23].

Dynamic mode decomposition with control (DMDc)

In actuated systems (systems with control inputs) e.g. the supermarket refrigeration system considered in this project, simple DMD cannot produce an accurate model from input output data, because the dynamics and modes obtained by simple DMD are affected by external input. However, there is a way to incorporate the effect of control input (actuation) with an extension of DMD which is known as DMDc. This method is an extension of DMD to approximately obtained matrices A and B of a discrete time state space model with an assumption of full state feedback and no feed forward, so matrices C and D are identity and null matrices.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{4.1}$$

$$\mathbf{y}_k = \mathbf{x}_k \tag{4.2}$$

Extensions to Dynamic Mode Decomposition (DMD)

DMD and DMDc may encounter limitations in modelling of dynamical systems due to linearity constraints. However, several extensions to DMD have been developed to address non-linear behavior [20].

1. **Extended DMD (eDMD):** eDMD is an enhanced version of standard DMD algorithm with additional non-linear versions of state in the state vector and these non-linear states are used to approximate the Koopman operator [24]. Nonetheless, one big issue that arises in implementing eDMD is the choice of non-linear extended states. If the chosen non-linear states do not approximate the Koopman invariant subspace then the resulting eigenvalues might become misleading.

2. **Sparse Identification of Eigenvalues** Sparse identification of non-linear dynamics (SINDy) is an approach that selects measurement functions similar to eDMD and allows elimination of measurements that are not related to the Koopman Operator. This technique performs well for lightly damped eigenvalues near the origin of pole zero map.

3. **Hankel Alternative View of Koopman (HAVOK):** In this technique, data matrices are constructed by stacking the time shifted versions of the states and these time shifted states inherently approximate the Koopman invariant subspace. The process is simplified because non-linear measurements of states are avoided in this case.

4. **Artificial Neural Networks (ANNs) and Autoencoders:** ANNs and Autoencoders can approximate the Koopman Operator by finding coordinates of non-linear system that evolves linearly in time and can be mapped back to the original state.

5. **Linear and Nonlinear Disambiguation Optimization (LANDO):** LANDO is useful for capturing the behavior of a dynamical system around some operating point because it relies on local linearization of system using DMD algorithm.

DMDc implementation for modelling of power consumption

DMDc was applied for obtaining model of compressor power consumption as the output. Inputs to the model were ambient temperature, compressor suction pressure and gas cooler pressure. However, DMDc was not able to produce good results. Initially it was speculated that this was due to non-linear behaviour of system, hence, authors also tried neural state space and LSTM neural network for modelling, which are presented in the next section. Later in the project, it was found that the main issue was not having enough excitation on one of the main input signal i.e. compressor suction pressure. Also, compressor power and suction pressure had non-linear relationship which could be potentially solved with other DMD methodologies like extended DMD (eDMD). However, it was decided to proceed with neural networks instead of extended DMD version because of time constraints since results from the neural networks were quite satisfactory.

4.1.2 Neural State-Space

A state-space model serves as a representation of a dynamic system, incorporating both a state equation and an output equation.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{4.3}$$

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}) \tag{4.4}$$

The state equation 4.3 explain the system behaviour over time, where $\mathbf{x}$ and $\mathbf{u}$ represent vectors corresponding to the plant state and input, respectively. Meanwhile, the output equation 4.4 describe the relationship between the system states and the measured outputs. For the case of power consumption modelling of supermarket refrigeration system considered in this project, the input vector consisted of three inputs namely ambient temperature (SC3), compressor suction pressure (MTPsuc) and gas cooler pressure (Pgc). The system had two states i.e. compressor power (MTPcomp) and evaporation temperature (EvapT) and two outputs i.e. compressor power (MTPcomp) and evaporation temperature (EvapT).

Typically, the state equation is comprised of a set of first-order ordinary differential equations (ODEs) or difference equations, often derived from fundamental principles and also known as white-box modelling. However, deriving accurate analytical equations for complex systems such as supermarket refrigeration systems may become challenging because of system complexity and lack of comprehensive knowledge of system dynamics for various components. Furthermore, it becomes impractical to derive equations for each supermarket setup because physical configuration of components might change according to the size of each supermarket. For such cases, data-driven modelling has emerged as a valuable alternative. One of the prominent approach is known as neural state space (NSS) which uses neural networks to represent both state and output equations of a non-linear system. A simple neural network known as multilayer perceptron (MLP) is used by neural state space (NSS) to approximate the state and output equations of a non-linear system. MLP is a simple feedforward neural network consisting of multiple layers of interconnected neurons or nodes as shown in the figure 4.1

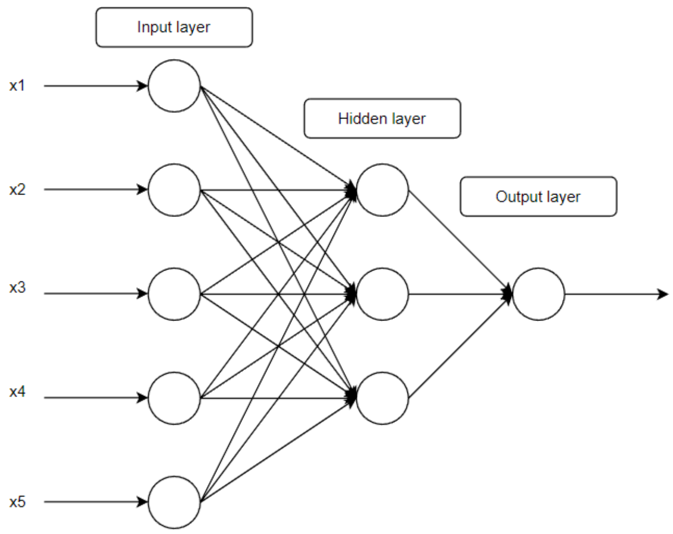


Figure 4.1: Multilayer Perceptron

In MLP, the first layer is the input layer which receives the input data. There can be multiple hidden layers between input and output layers to perform transformations on the input data through weighted sums and activation functions. Each neuron in a hidden layer takes inputs from all neurons in the previous layer and passes its output to all neurons in the next layer and the output layer produces final output of the network. The output layer can be of regression type or classification type depending on the application. Connections between neurons are associated with weights which represent the strength between the connections. Additionally, each neuron has a bias term, which allows the network to capture offsets or shifts in the data.

Weights and biases are adjusted during the training in an iterative manner to minimize the difference between the network's predictions and the actual target values. Activation functions allows MLP to capture non-linear dynamics in the data. Commonly used activation functions are sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax. Back-propagation is used for the training the network weights and biases. The network output is compared to the true target values, and an error signal is computed. This error signal is then propagated backward through the network, and the weights and biases are adjusted using optimization techniques such as gradient descent to minimize the error.

Discrete time neural state space model (sampling time Ts=600s) was created in the MATLAB using idNeuralStateSpace function and then MLP network with the following specifications was created to be utilized by this model.

- Number of states=2
- Number of inputs = 3
- Number of hidden layers = 2
- Size of hidden layers = [64 64]
- Activation function = tanh
- Weights initializer method = glorot
- Bias initializer method = zeros

Data-Preprocessing

Following steps were used to preprocess the data required for the training and validation of neural state space in MATLAB.

The data was complied from different simulation runs to create a comprehensive dataset for training neural network effectively. After that, to remove the effect of intermediate non-linearties, the data was downsampled to 10 minutes time interval.

The data was then normalised, as normalisation has multifaceted benefits in the training of neural network. It facilitates the converge during the training and mitigates the risk of dominating certain features due to large magnitudes. It also safeguards against the numerical instability that may arise during the training. Due to consistent data range, smooth weight updates are ensured during the training.

After normalisation, the data was divided into 3 distinct sets namely training, validation and test. The segmentation is necessary to avoid over-fitting of the network during the training.

Network Training

The network was trained using nlssest function of the MATLAB used for training of neural state space models. Following options were set for the training of the network.

- Update Method = adam
- Number of Epochs = 200
- Minibatch size = 50

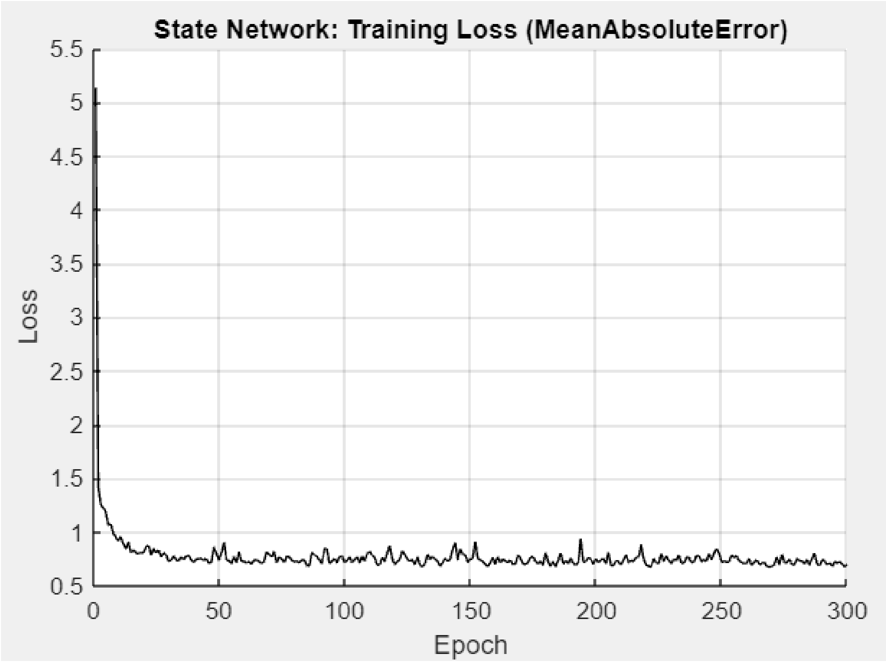


Figure 4.2: Mean absolute error - NSS training

- Learn rate = 0.002

Figure 4.2 shows the mean absolute error (training loss) of NSS trained on an example of data from the month of May and plotted against number of epochs.

Validation

The trained network was validated with different data sets that was not used for training but from the same range of ambient temperature conditions i.e. month of May. The validation plot is shown in figure 4.3.

The fit percentage of 93.9 % shows a very good network fit. Nevertheless, to check the relationship between compressor power and suction pressure, the network was tested with ramp increase and ramp decrease in suction pressure. An increase in suction pressure was causing increase in the power consumption which was contrary to the physical behaviour of the system. Hence even with the good fit of 93.9 % the model is capturing the overall trend in the power consumption which is due to changes in ambient temperature conditions. The issue is described in more detail in the section 4.2.

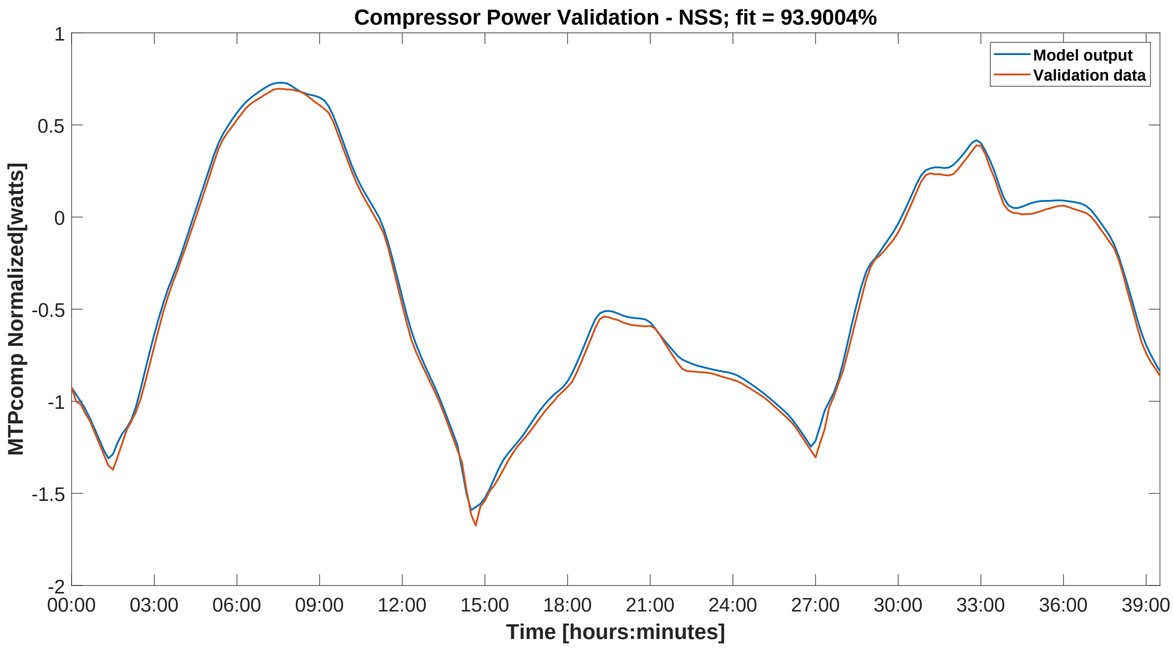


Figure 4.3: Validation of trained NSS network

4.1.3 Long-short Term Memory Neural Network

LSTM is subset of recurrent neural networks (RNNs) family and possesses the properties of capturing long term dependencies in a time series data. By solving the problem of exploding and vanishing gradient that is inherent to RNNs when learning long term dependencies, LSTM can effectively preserve and utilize the information for longer periods of time. This unique property of LSTM makes it one of its kind to make accurate predictions by capturing complex temporal patterns in time series data [25].

LSTM Cell architecture

A typical LSTM cell is composed of an input gate, output gate and forget gate [26]. A cell acts as a memory block and the 3 gates regulate the flow of information associated with the cell. The memory blocks or LSTM cells are connected in a recurrent manner to make the LSTM architecture to maintain the state over time and control the information flow through non-linear gating units.

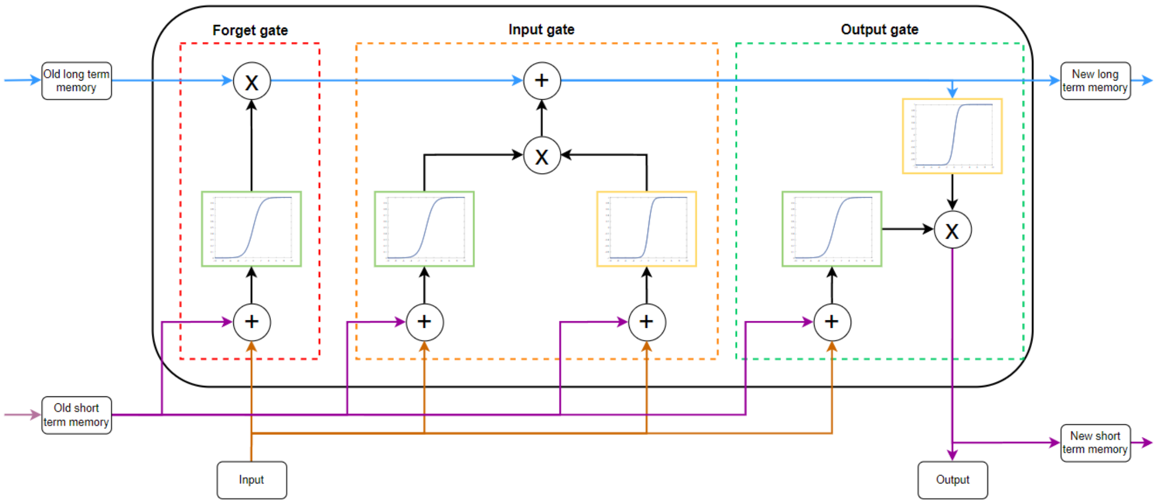


Figure 4.4: LSTM Cell Architecture [21]

Forget Gate

This step determines which information should be retained from the previous cell states. Activation values of the forget gate are calculated based on the current input and past output and cell states.

Input Gate

The purpose of the input gate is to combine the current input with the LSTM output and cell state from the previous time step. Weights are associated with each input, output and cell state for combination and a bias component is added at the end.

Output Gate

This step calculates the output of the LSTM unit based on input at the current timestamp and output and cell state at the previous timestamp. A typical LSTM structure is shown in the figure 4.4

Currently, LSTM neural network is considered as one of the most powerful tools for modelling and prediction of timeseries data. Other applications of LSTM include natural language processing, text recognition, computer vision and image and video captioning.

LSTM implementation in MATLAB

LSTM network with the configuration shown in listing 4.2 was implemented in MATLAB. The network consisted of sequence input layer, an LSTM layer of 50 nodes, a dropout layer to drop 2 percent of training data to avoid over fitting and a regression layer at the output. Options used for training the network are also shown in the listing 4.2.

Data prepossessing was carried out in the same manner as for neural state space implementation. Data was complied from different simulation runs and then down-sampled to 10-minute interval. The data was then normalized and partitioned into training, validation and test data sets.

Listing 4.1: LSTM implementation in MATLAB

```
%% LSTM
numResponses = 2;
featureDimension = 3;
numHiddenUnits = 50;
maxEpochs = 300;
miniBatchSize = 5;

Networklayers = [sequenceInputLayer(featureDimension) ...
    lstmLayer(numHiddenUnits) ...
    dropoutLayer(0.02)...
    fullyConnectedLayer(numResponses) ...
    regressionLayer];

options = trainingOptions('adam', ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'GradientThreshold',20, ...
    'Shuffle','once', ...
    'Plots','training-progress',...
    'ExecutionEnvironment','parallel',...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',50,...
    'L2Regularization',1e-3,...
    'LearnRateDropFactor',0.5,...
    'Verbose',0,...
    'ValidationData',{inDataValN outDataValN});
```

Results obtained from the LSTM implementation were also not satisfactory and are explained in detail in the following section.

4.2 Modelling issues

Regardless of the data-driven modelling method applied, the authors experienced some problems obtaining good model. In fact, the problems were not within the methods, but with insufficient data having been used for the training of models. This section discusses what the 2 main problems were and how they were diagnosed and addressed.

4.2.1 Causes of problems with modelling of the system

Typically, a real supermarket refrigeration system is never at rest, as it never reaches completely steady-state operation [14], due to many local control loops interfering with each other. Assuming that this is also the case in both digital twin models was misleading, as in fact the fixed suction pressure references are kept tightly by their controllers not allowing for much variation, as investigated later. This has led to 2 major issues in modelling, regardless of the method used.

First of them was lack of input excitation. As the measured suction pressure i.e. input to the data-driven models, was constant for nearly all the time, it did not carry any dynamic information and the impact of this signal on the outputs could not be retrieved by the modelling methods applied, leading to incorrect identification of impact of suction pressure on the system, including reversed sign. To counter this problem, a small excitation signal based on pseudo-random binary sequence (PRBS) was applied to the suction pressure reference to introduce dynamics to the signal and consequently to the system (however, it is expected that in the real system this would not be necessary, owing to the mentioned variations in signals).

Another issue was that after applying PRBS type of signal, not enough operating points were provided to the model, because of the suction pressure reference being fixed, resulting in the obtained models reconstructing long time patterns (due to ambient air temperature) in data well and not capturing faster dynamics (due to suction pressure). In addition, the modelling methods used could not encapsulate the entire range of system behaviour for the normal operating range (25-34 bars of suction pressure). To address this, new simulation runs were generated, where the suction pressure reference was varied to the respective levels, such that there were enough reference levels covering the whole range.

As this was a highlight discussion of the final conclusions on the issues that were preventing obtaining good models, a more detailed explanation of observations that helped to diagnose these obstacles in modelling is provided next.

4.2.2 Attempts to improve modelling with LSTM NN

The former issue, lack of excitation, could be diagnosed much easier than lack of suction pressure levels examples, as one could easily see from a time series plot of suction pressure that it remains almost constant. After addressing this with PRBS-based signal, the latter problem was not considered until some iterations in the modelling process were done, as it also manifested itself in the mentioned difficulty in capturing fast pressure dynamics, which caused the authors to focus too much on this phenomenon itself.

To be more specific, it was expected that having enough excitation around an operating point, the model should perform well at least for the area around that point. Nonetheless, that was not the case, which made the authors initially believe that the DMDc and NSS approaches were insufficient to model this system, which in reality is not true, as demonstrated later in this section. For this reason, in an attempt to verify it, LSTM NN was used in the following steps.

The network performed quite well in capturing both long- and short-term dynamics right from start as seen in figures 4.5 and 4.6, except for one problem - there was a systematic bias introduced by the network. To resolve this an extra feedback term was added to the LSTM network. It was implemented by delaying the whole output signal in each data set offline by one sample and providing it as an extra input to the network, and deleting the first samples of other inputs and the power output to match dimensions with the new signal. This allowed for obtaining a perfect reconstruction of power output for testing data sets, see figures 4.7 and 4.8, that were close to the operating point (MT suction pressure = 26.5 bar and ambient air temperature corresponding to typical profiles in October).

Nonetheless, training data containing fixed reference with small PRBS added on top imposes a large limitation on network's ability to generalise for other values of this condition. In order to test the correctness of the response in terms of sign, ramp inputs with small slopes were input to the digital twin (after settling from starting up). For each experiment, either of the considered inputs (ambient air temperature or suction pressure) were kept constant at their mean values, when the other one was tested. The final values of the ramps added on top of the mean signals were +/- 3 degrees Celsius for ambient air temperature and +/- 0.5 bars for suction pressure. The result are presented in the figures 4.9 and 4.10. It can be seen that the network performs well for ambient air temperature ramps, however

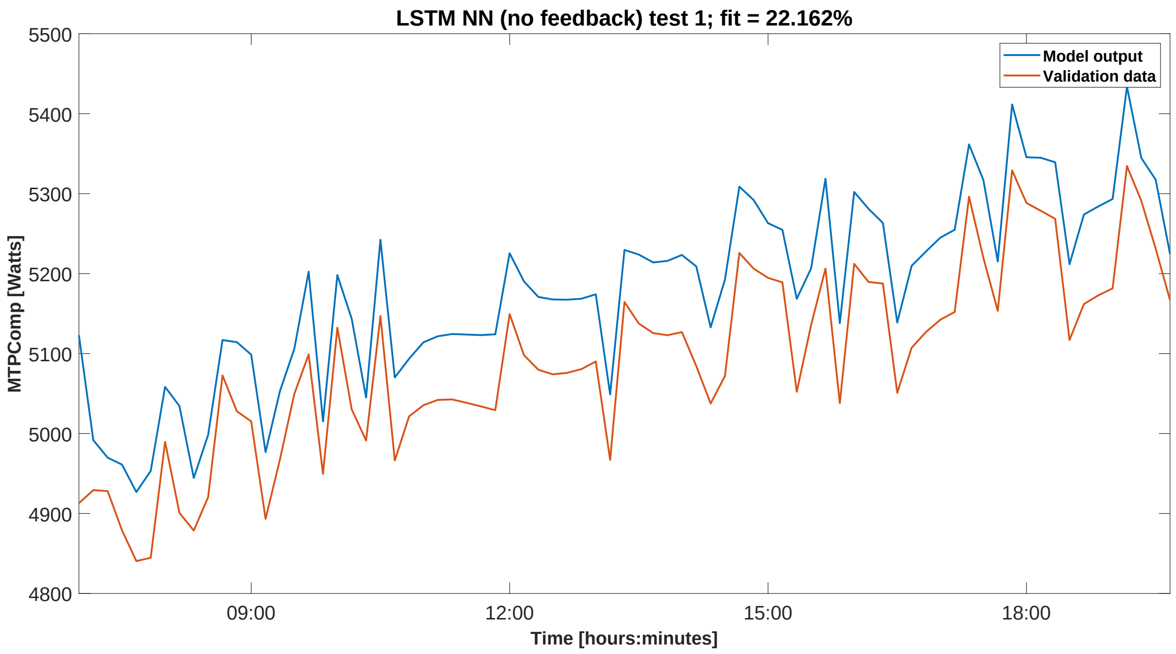


Figure 4.5: LSTM NN tested on cold October day data

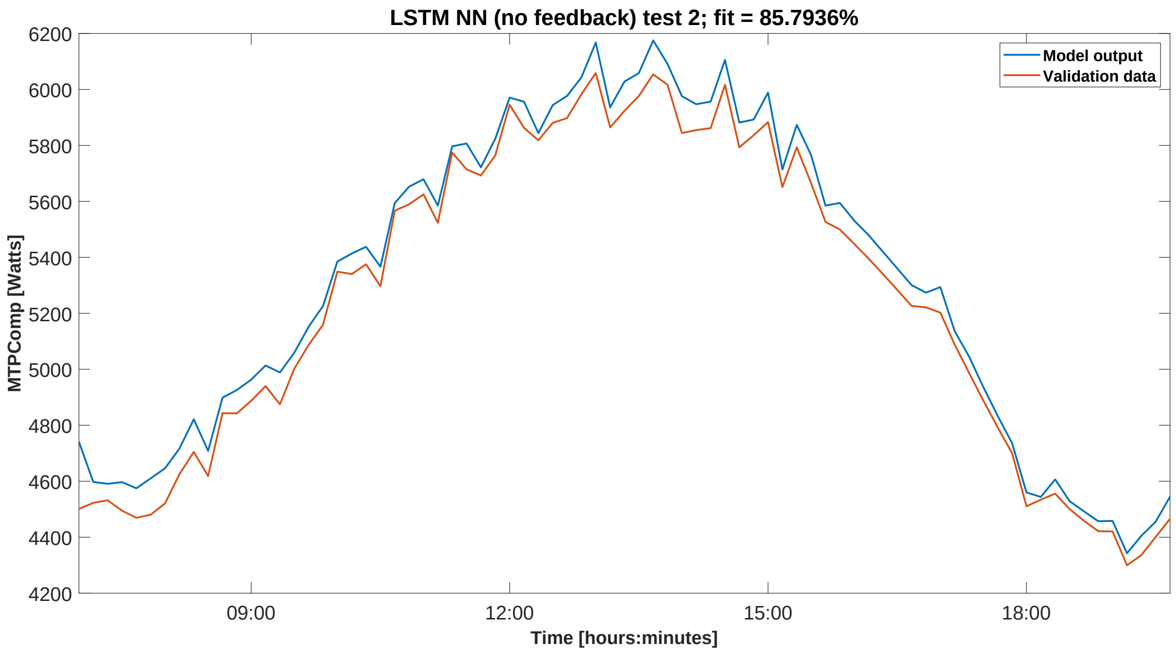


Figure 4.6: LSTM NN tested on warm October day data

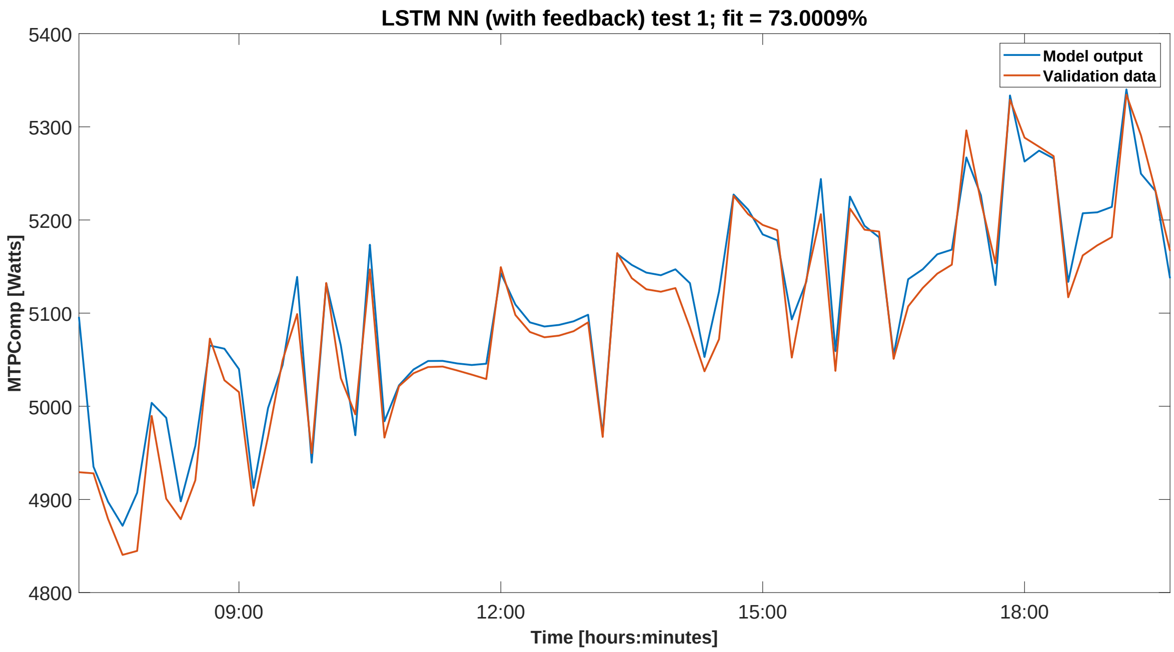


Figure 4.7: LSTM NN with feedback tested on cold October day data

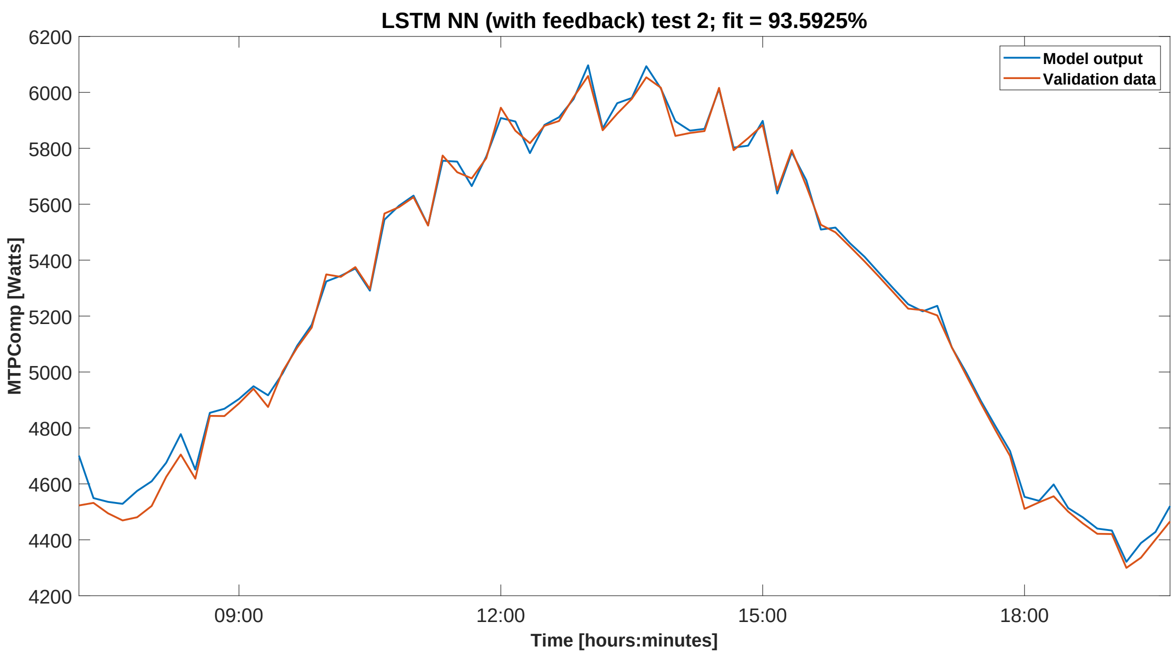


Figure 4.8: LSTM NN with feedback tested on warm October day data

the performance of the model is severely deteriorated for suction pressure ramps.

To check if the issue lied in the network or data, the negative suction pressure ramp data set was appended to the training data set. This resulted in improved performance - the network was not able to follow signal exactly, however the overall pattern and direction was preserved (while not affecting the performance for other testing data sets), which can be seen below 4.11. This suggested that more operating points for suction pressure were needed in training data set, in order to vary it far away from the default setpoint, which is the selected strategy for optimizing costs of energy consumption, especially given that the network cannot cope with values deviating as little as up to 2% from the mean of training data.

This development called for training model with more data. From this point onward, the simulation model used was fast model, as even grater amount of data was needed. After obtaining new data from new simulation model, including ramps, the network was trained again leading to results where it performed perfectly for both typical operation over a day and ramp inputs, as demonstrated in figures 4.12, 4.13 and 4.14 (note that differences compared to the previous plots come from the fact of using other model / data sets). This made the authors revisit the bias issue. The question was if with more data the problem would still persist and feedback would be required. As it turned out, the network performed identically for both cases, so the feedback was no longer needed, indicating that larger amount of data with wider range of values was required for obtaining the model.

Finally, it was also investigated if it was other methods or data that were insufficient. Same training and testing data were applied to NSS, which resulted in correct model that predicted output signals closely to the ones obtained from LSTM model. This was important from the perspective of implementing the model in MPC in MATLAB, which is explained in more detail in the next chapter. It is speculated that DMDc could also be applied provided the new data, however, that was not checked, due to time limitations.

As a last remark, it can be noted that a special strategy should be taken when applying this modelling approach in practice. Ideally, the model could be trained on digital twin and then it could be adjusted using transfer learning to be tailored to specific supermarket. This strategy is recommended, as it would be impractical to perform test spanning whole range of suction pressure condition, so in order to cover it, the pretrained network could be adjusted using values from a smaller range of data. Here, reintroducing feedback to the network could be beneficial as it compensates for scenarios with little data available, as seen previously. Nevertheless, this is left as a future work.

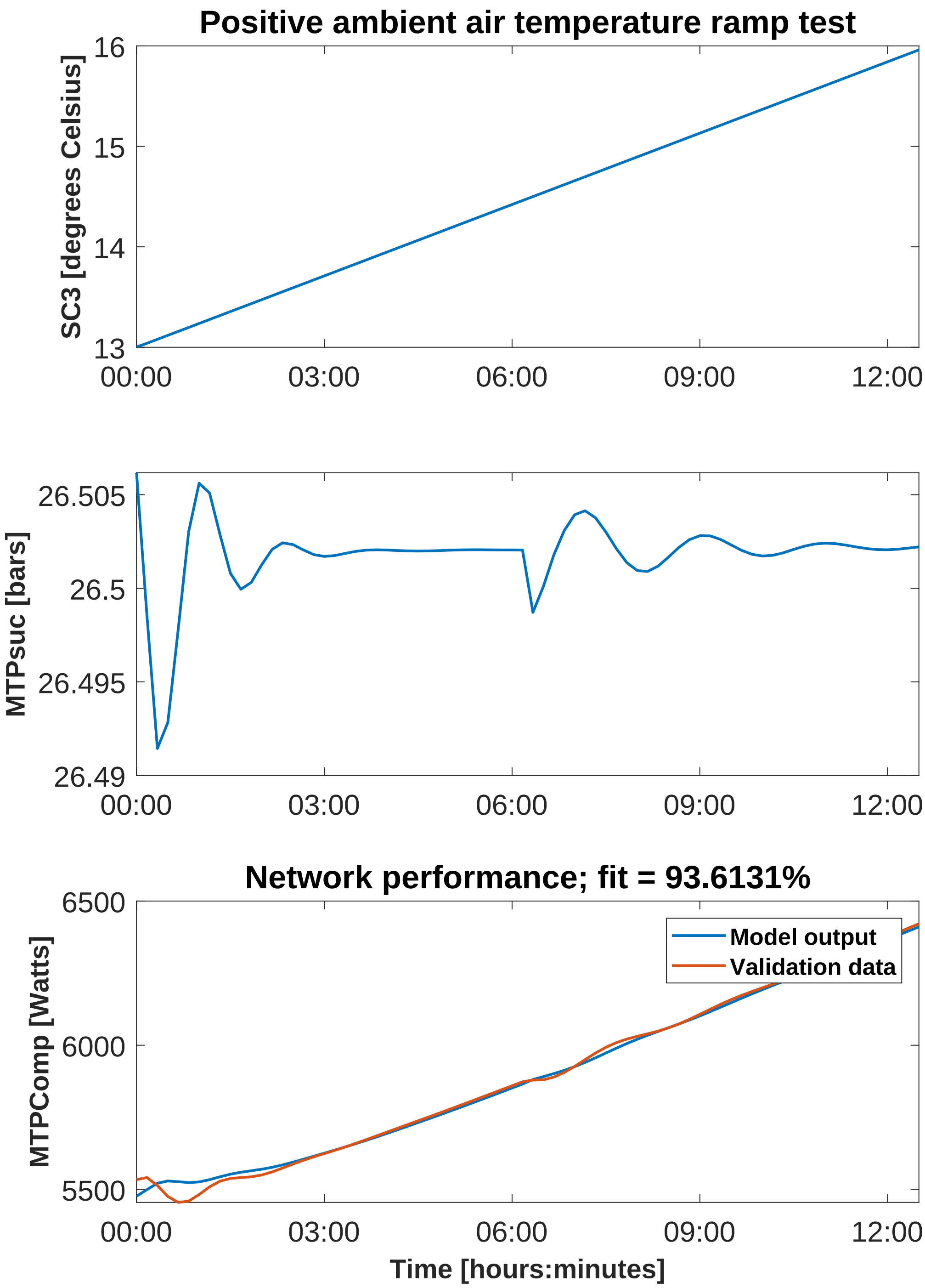


Figure 4.9: LSTM NN tested for increasing ambient air temperature

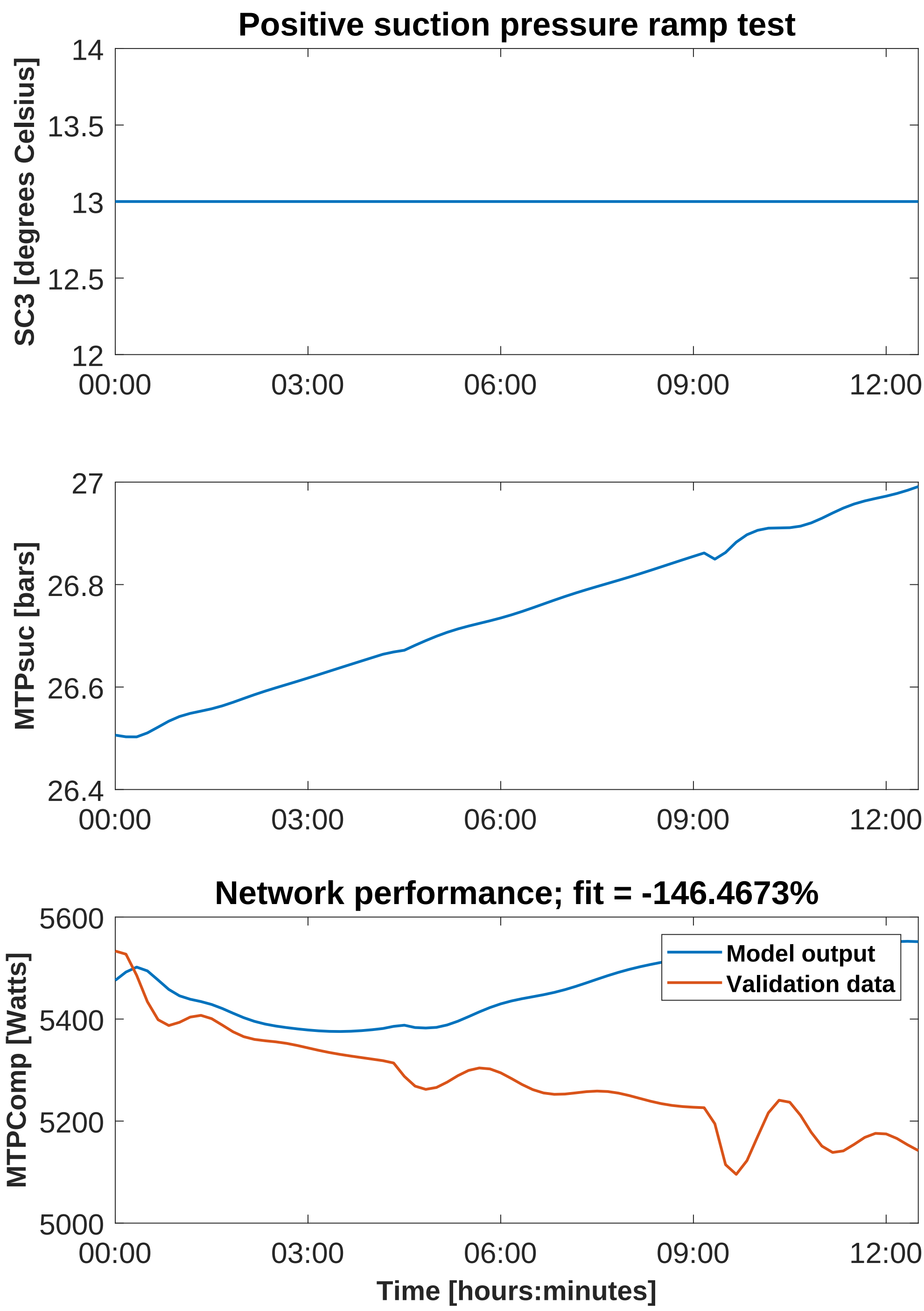


Figure 4.10: LSTM NN tested for increasing suction pressure

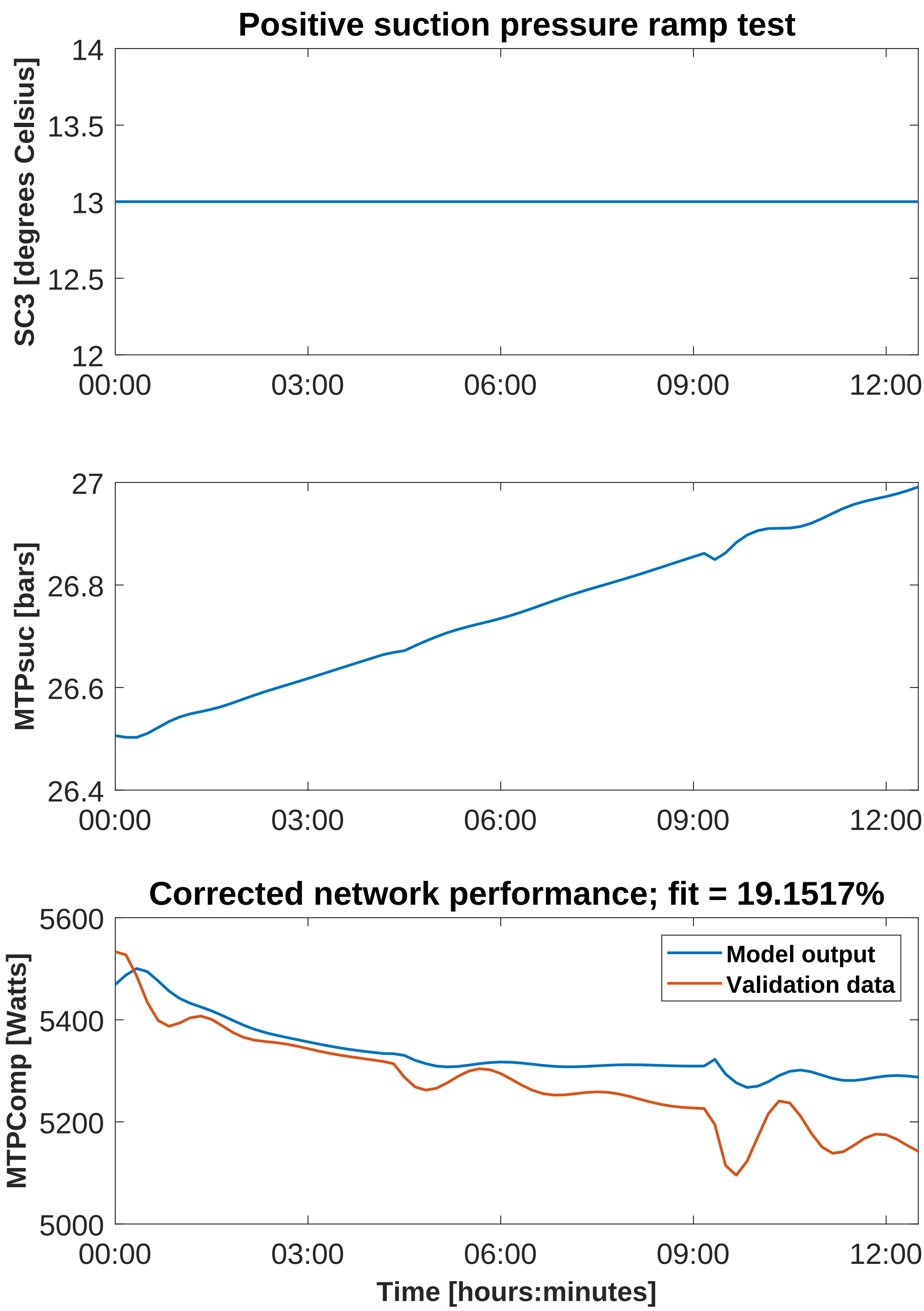


Figure 4.11: LSTM NN trained including negative negative suction pressure ramp tested for increasing suction pressure

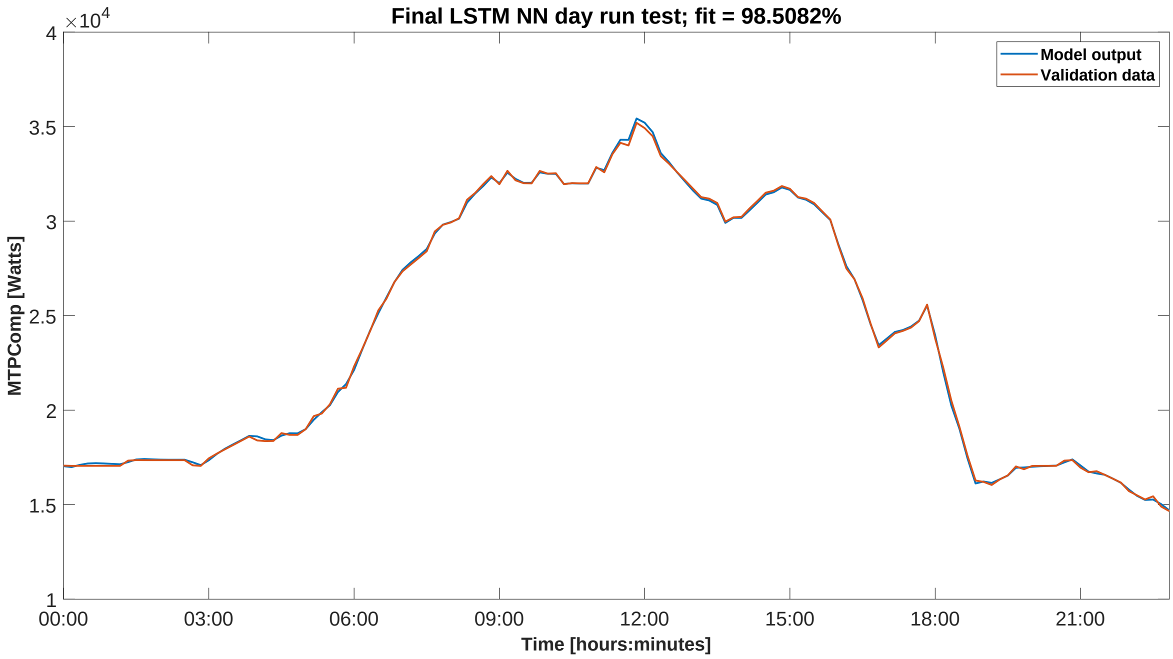


Figure 4.12: Testing of final LSTM NN on summer day data

4.3 Impact of fast model limitations

Despite main problems in modelling being due to authors own mistakes, a lot of problems were contributed through limitations of the fast model used for the most of the thesis work. As they affected its final shape, they are explained in this section, as well as how they were tackled.

4.3.1 Incorrect UAir parameter

One of the important parameters in the fast model is UAir parameter, which serves as a kind of thermal resistance, as the potential cooling load driven by ambient air temperature is divided by it to obtain the actual cooling load affecting the system. Consequently, the higher value of UAir is, the less load is experienced by the system.

The problem with the parameter was that it was set to high in the original version of the fast model, causing values of cabinet temperature to always go negative, even for the suction pressure values that were around the default value that should keep them in between 1 and 5 degrees Celsius. In order to fix that, UAir was tuned such that its temperature was equal to about 3 degrees Celsius for high ambient air temperatures (around 35 degrees Celsius). The right values

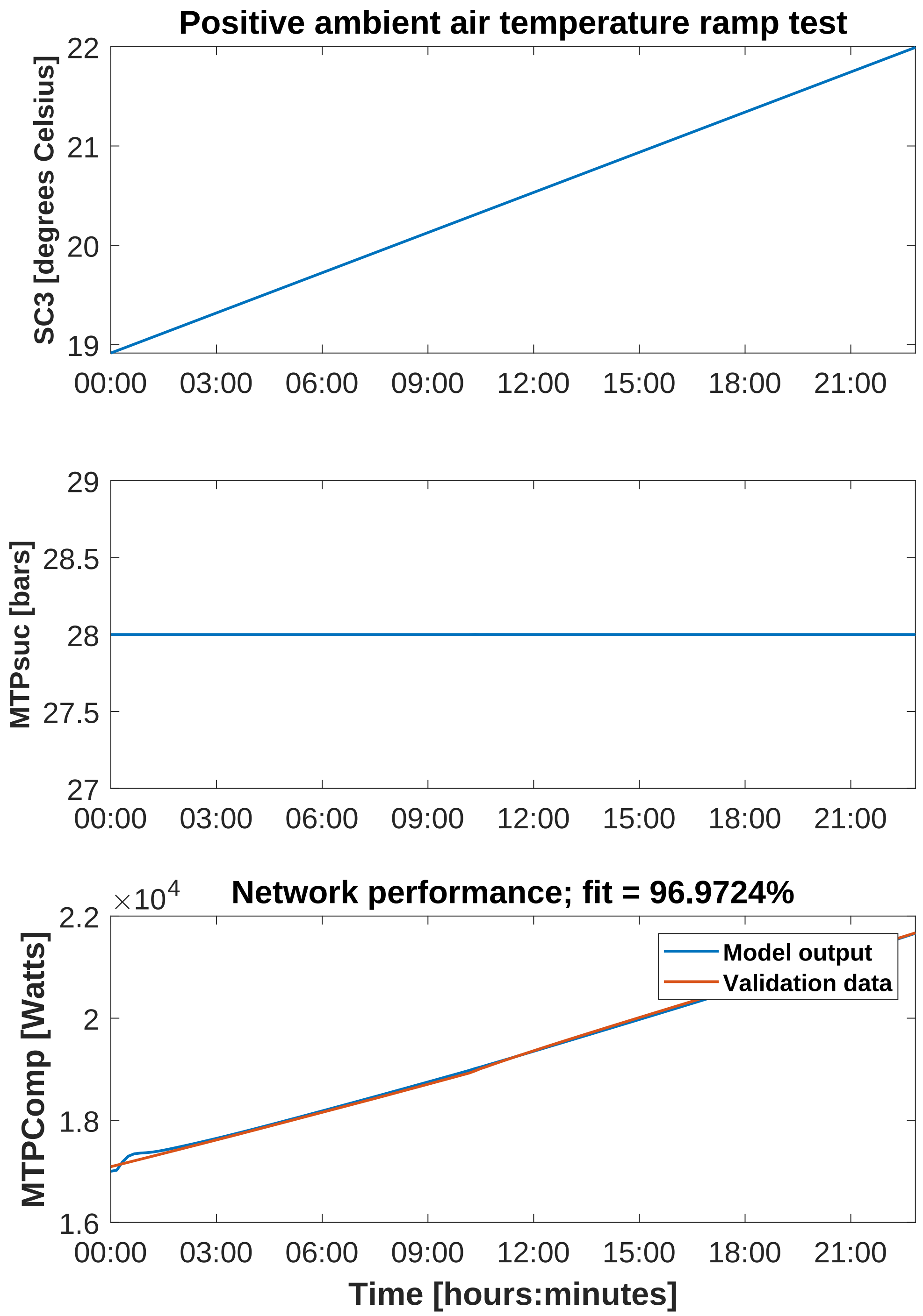


Figure 4.13: Ambient air temperature ramp test for final LSTM NN

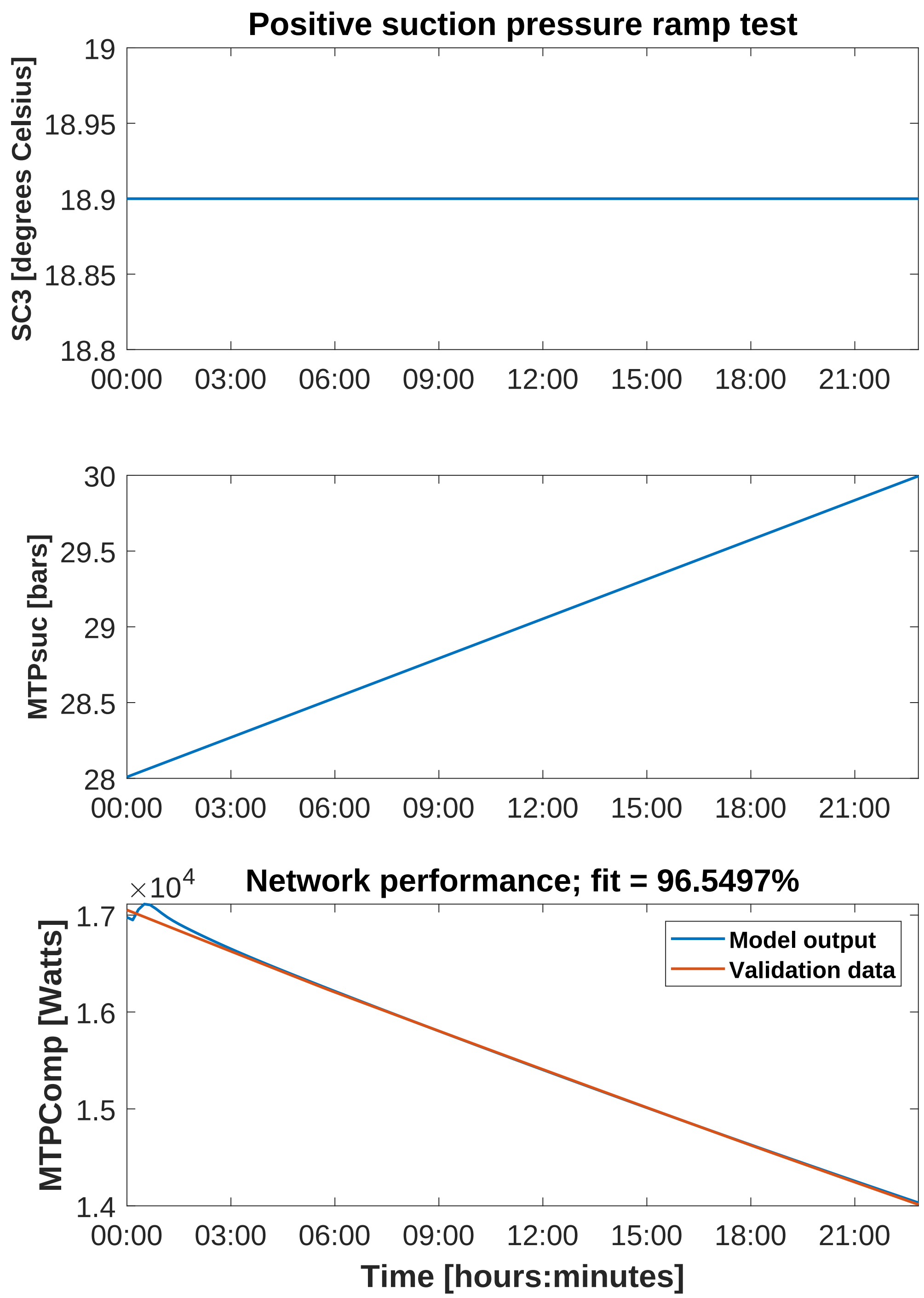


Figure 4.14: Suction pressure ramp test for final LSTM NN

of UAir was found to be 9000 (compared to 20000 originally).

In addition, this problem initially led to authors considering wrong signal as cabinet temperature for later testing of controllers, especially checking if they were not violating constraints. The value that was believed to be cabinet temperature was in fact superheat - a difference between temperature of refrigerant and its boiling temperature, as it stays positive. As the wrong signal was used for evaluating controllers, they also needed to be redesigned. However, it did not take much effort as the workflow was already known and it only took to change the signal considered as cabinet temperature.

After this issue was resolved, the simulations for training and testing models were rerun. However, that was not the only problem on the evaporator side that affected the thesis work.

4.3.2 Lack of cabinet temperature control

In actual operation of supermarket refrigeration systems, both compressors and expansion valves are used, where compressors deliver overall cooling and expansion valves decide how much of it goes to each evaporator, as they have different cooling requirements. This also allows to keep the pressure constant and let the valves close more, when there is less need for cooling.

Unfortunately, the fast model did not have cabinet temperature controller implemented. On of the problems that it caused was that cabinet temperature was operating in open loop and it could not be seen affecting other parts of the system, making some sort of estimation of thermal capacity of the stored food not possible, meaning that the only way to obtain this information would be through providing cabinet temperature directly to the controller, which would not be practical in real scenario with many cabinets. The importance of this information is explained in more detail in chapter 5. Thus, any modelling effort towards this issue was not performed at the current stage.

This also meant that the optimization of energy cost scenario in this thesis was degraded, as the cabinet temperature would vary freely with changes in ambient air temperature, without any action from compressor. To resolve that, one MPC design that tracks tracking the right evaporation temperature to maintain the cabinet one at 3 degrees was set as a baseline for energy cost savings comparisons. This is further elaborated in later parts of the thesis.

It shows that the fast model had some serious limitations and these were not the only ones. However, no more issues were noted on the evaporator side.

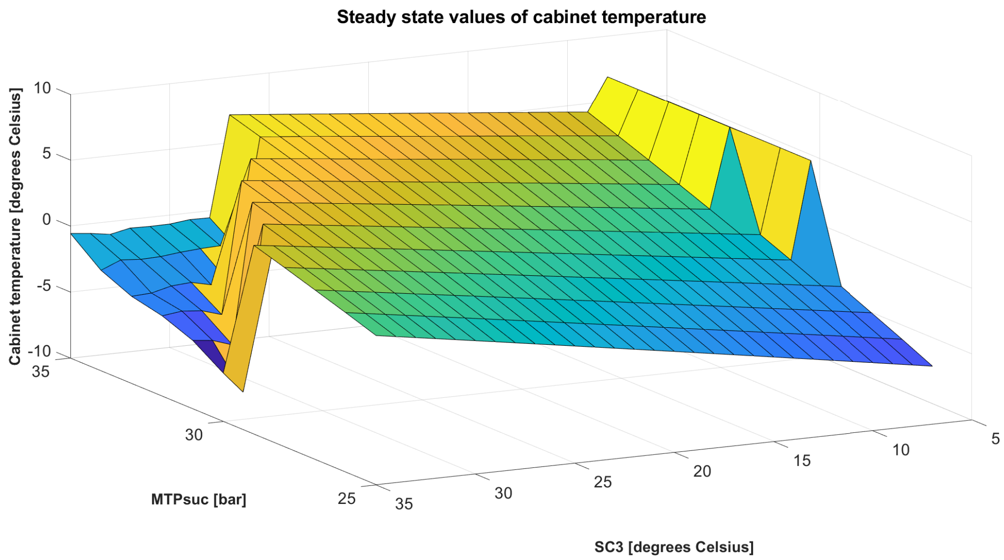


Figure 4.15: Fast model breaking down for high suction pressures outside of ambient air temperature operating range

4.3.3 Operating envelope

Aside from issues on evaporator side, the simulation model also had issue at the system level. To be specific, there was certain operating envelope for which the model was performing correctly and outside of each it was producing erroneous results.

This can be observed in figure 4.15, where it can be seen that the model breaks down for high pressures for both above and below some maximum and minimum ambient air temperature (35 and 9 degrees Celsius respectively), where the results of simulations are inconsistent in these regions. This is not as much of a problem for high temperature case, as high suction pressure levels would lead to loss of cabinet temperature beyond upper constraint. Nonetheless, the opposite case of low temperatures and high suction pressures is problematic, as in this case the pressure must be lifted to avoid drop of cabinet temperature into negative region (however, note that that would not be so much of an issue, if cabinet temperature control was manged with expansion valve!).

Solution to this issue was simply to limit range of ambient air temperature conditions for which the system was tested. This unfortunately meant, that the region in which most energy savings could be observed.

4.4 Final network used: NSS

After addressing the issues presented in section 4.2, authors were able to successfully obtain good models for both neural state space (NSS) and LSTM. The modelling part of the project was concluded here with both NSS and LSTM options available for model predictive control (MPC) implementation in the next phase of the project. However, implementation of LSTM model with MPC in the MATLAB had some bottlenecks as explained in the section 5.2.1. Hence, it was decided to choose NSS as the final network to be used as internal model of MPC. Listing **??** shows the specifications and figures 4.16 and 4.17 show the architecture and analysis of final NSS network used with MPC.

Listing 4.2: Final NSS used for MPC

```
%% Final NSS Model
Ts=600;
obj = idNeuralStateSpace(2, NumInputs=3, Ts=Ts);
obj.StateNetwork = createMLPNetwork(obj, "state", ...
    LayerSizes=[64 64 64 64]);
options = nssTrainingOptions("adam");
options.MaxEpochs =500;
options.MiniBatchSize =20;
options.LearnRate = 0.002;
nss = nlssest(inDataTrainN,outDataTrainN,obj,options, ...
    "UseLastExperimentForValidation",true);
```

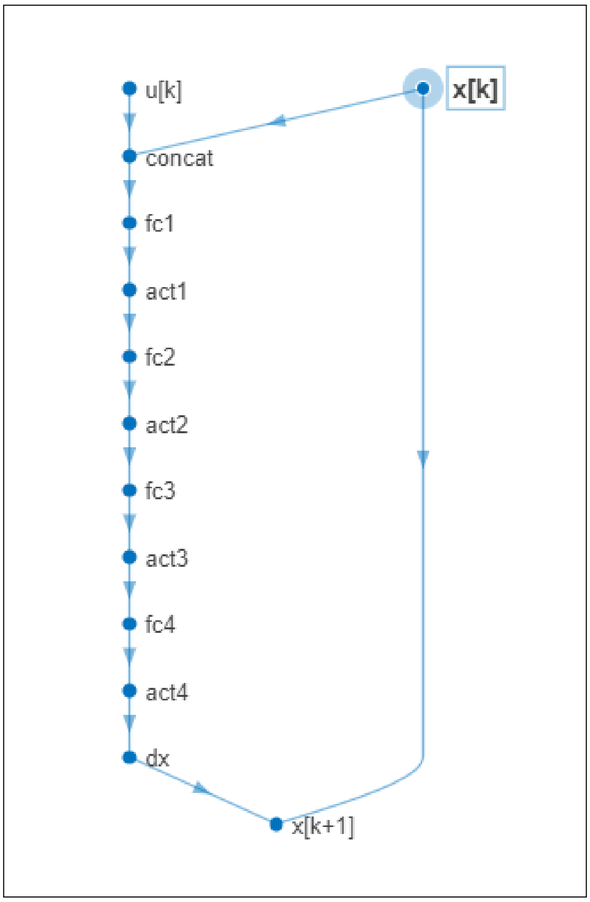


Figure 4.16: Final NSS architecture

ANALYSIS RESULT					
	Name	Type	Activations	Learnable Proper...	States
1	x[k] 2 features	Feature Input	2(C) × 1(B)	-	-
2	u[k] 3 features	Feature Input	3(C) × 1(B)	-	-
3	concat Concatenation of 2 inputs along dimensi...	Concatenation	5(C) × 1(B)	-	-
4	fc1 64 fully connected layer	Fully Connected	64(C) × 1(B)	Weights 64 × 5 Bias 64 × 1	-
5	act1 Hyperbolic tangent	Tanh	64(C) × 1(B)	-	-
6	fc2 64 fully connected layer	Fully Connected	64(C) × 1(B)	Weights 64 × 64 Bias 64 × 1	-
7	act2 Hyperbolic tangent	Tanh	64(C) × 1(B)	-	-
8	fc3 64 fully connected layer	Fully Connected	64(C) × 1(B)	Weights 64 × 64 Bias 64 × 1	-
9	act3 Hyperbolic tangent	Tanh	64(C) × 1(B)	-	-
10	fc4 64 fully connected layer	Fully Connected	64(C) × 1(B)	Weights 64 × 64 Bias 64 × 1	-
11	act4 Hyperbolic tangent	Tanh	64(C) × 1(B)	-	-
12	dx 2 fully connected layer	Fully Connected	2(C) × 1(B)	Weights 2 × 64 Bias 2 × 1	-
13	x[k+1] Element-wise addition of 2 inputs	Addition	2(C) × 1(B)	-	-

Figure 4.17: Final NSS analysis

Chapter 5

MPC designs for energy cost savings in supermarket refrigeration systems

As the MPC is crucial part of the thesis, it is explained in more detail in this chapter. It follows up with explanation of how it was set up and tuned, with final formulations being given at the end of the chapter.

5.1 Model Predictive Control

Model predictive control (MPC) is an advanced method of process control for complex systems and can satisfy specific constraints while controlling the system. MPC has been adopted in many industries (e.g. power systems, oil refineries and chemical plants) for process control, because of its ability to anticipate future events and optimization of control algorithms over a finite time horizon [27]. This unique property is lacking in traditional controllers such as PID, which only react to deviations from set reference points.

Model predictive control (MPC) is conceptually related to classical LQ regulator, but has difference in some key points. MPC minimizes the cost function over two receding horizons while LQ regulator minimizes the cost function over infinite horizons. Infinite horizon of LQ regulator facilitates the calculation of constant state feedback gain while MPC algorithm predicts the future outputs and optimize the future control over two horizons [28].

The dynamic model of the plant is required to set up the MPC controller. The model can be derived from basic principles or through system identification. As explained in section 4 of this report, the dynamic model of the considered supermarket refrigeration system was obtained using system identification through neural state space (NSS). The use of data driven models with MPC is increasingly becoming popular due to the availability of large amounts of data and insufficient knowledge of exact system configurations, which could leads to bottlenecks when performing modelling through first principles. The model is used by MPC to predict how changes in the setpoint or control elements affect the control objectives and process constraints.

MPC solves an online optimization problem at each time step to find the optimal control action that minimizes the error between setpoints and at the same time respecting the constraints. This is done by considering both current and future timesteps data. The real time optimization is achieved by MPC iterative approach in which it implements the current action and then recalculates for the next timestep.

Figure 5.1 illustrates the fundamental concept of MPC implementation for set point regulation. An optimization problem is solved at each time step k over the subsequent prediction horizon Hp by using the predicted output of the plant obtained from dynamic model of the system. The future inputs are adjusted for the next Hc samples (known as control horizon) in such a way that error between output and reference is minimized. The first value from the predicted control input is applied to the plant and the optimization problem is repeated in a recursive manner by updating the measured output and advancing the prediction and control horizon by one step [29]

5.1.1 Receding Horizon

The basic idea of predictive control in MPC is the concept of 'receding horizon', which involves the process of continuously updating and optimizing control actions based on future predictions.

Maciejowski [27] explained the idea of 'receding horizon' by taking an example of single input single output (SISO) plant in discrete time. The output of the considered system is $y(k)$ at time step k and it is expected to track the set point trajectory $s(t)$. Instead of driving the plant output $y(k)$ directly to the set point trajectory $s(k)$, a reference trajectory is defined with time constant $Tref$ which starts from the current output $y(k)$ and adapts to the current conditions at time k as shown in figure 5.1. An internal model predicts the plant behavior over the prediction horizon by presuming an input trajectory $u(k+i|k)$. The controller selects the input such that plant behavior is optimized by assuming that the model is strictly proper which means that output $y(k)$ only depends on the past inputs and not on the current input $u(k)$. The aim of the input trajectory is to bring the plant output to the setpoint trajectory at the end of the prediction horizon $k+Hp$.

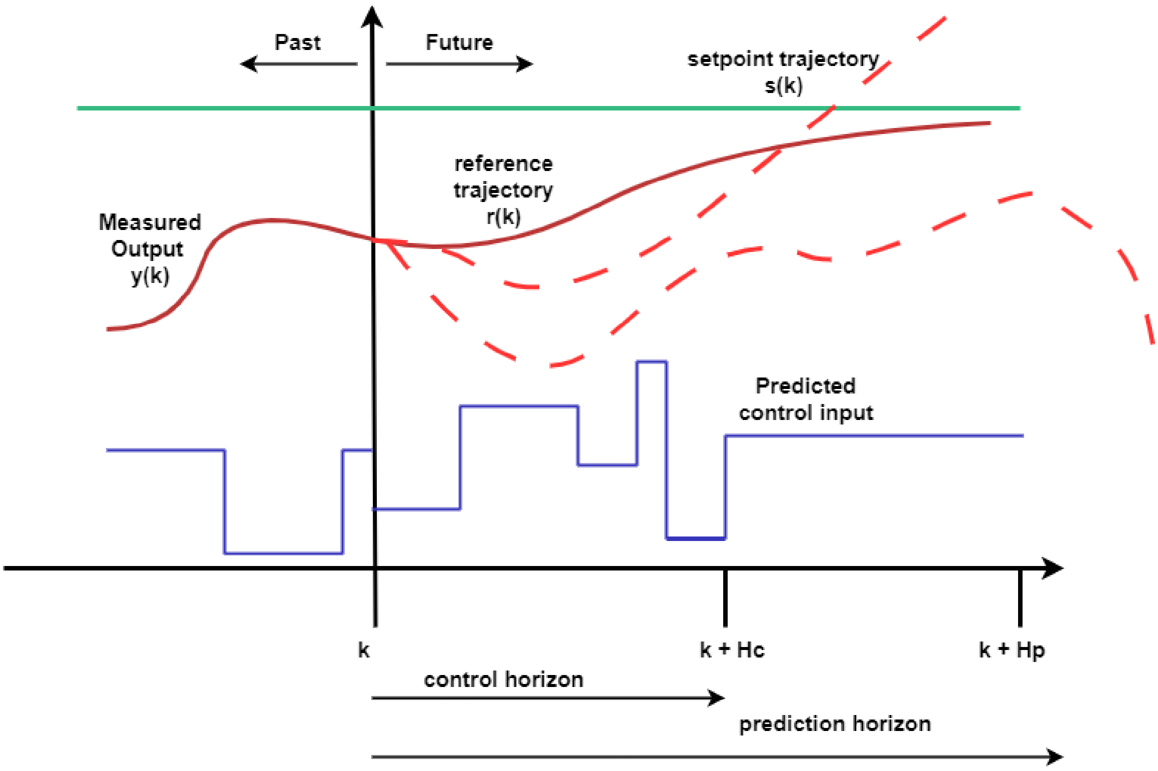


Figure 5.1: A typical MPC implementation for setpoint regulation [29]

This can be achieved by multiple input trajectories, however, the aim is to vary it in first few steps and keeping it constant afterwards and the simplest form is to keep the input constant over the horizon.

Once the input trajectory is selected then only the first element is applied to the process and the same process of measuring output, predicting future behavior, and determining the input trajectory is repeated at each time step. The term 'receding horizon' is used because prediction horizon moves forward with each time step. MPC effectively handles disturbances and changes in plant behavior by continuously updating predictions and control actions based on the most recent measurements and forecasts, maintaining optimal performance over time.

5.1.2 Cost function

The performance of MPC depends on the cost function which is being minimized. The standard MPC cost function is [27]

$$J(k) = \sum_{i=0}^{H_p} ||\hat{y}(k+i) - r(k+i)||_Q + \sum_{i=0}^{H_u} ||\Delta\hat{u}(k+i)||_R \tag{5.1}$$

where

$$||x||_P = x^T P x$$

$$Q, R >= 0$$

$$Hp >= 1$$

$$Hu <= Hp$$

Q, R are semi positive weight matrices, H_p is the prediction horizon and H_u is the control horizon. The objective of the cost function is to minimize the error between output and reference while minimizing the changes in the control signal at the same time. Weight matrices Q and R define the priorities between the signals.

The formulation of cost function is done in terms of $\Delta u(\hat{k})$ instead of absolute values of $u(\hat{k})$ because as long as output error is zero and control signals are steady the absolute values are irrelevant.

5.1.3 Constraints

One of the useful properties of MPC that distinguishes it from other controllers is the setting up of constraints on the inputs and outputs. The constraints on the outputs are due to safety, while the constraints on the inputs are mainly because of saturation of actuators and slew rate.

Constraints can include lower and upper bounds on the input and output values as well as rate of change manipulated variables. Constraints can be implemented as hard or soft constraints. Hard constraints are defined as the constraints which the optimization problem must satisfy, however, if its not possible to satisfy hard constraint then the optimization problem might become infeasible [30] and in this case there will be no change in the manipulated variables by the controller.

Hence, it is recommended to only use the physical limits of manipulated variables on the plant as hard constraints. The rate of change of manipulated variables should correspond to the physical limit of rate of change of variables in the plant. It is generally not recommended to set bound constraints on the output variables unless unavoidable. The best practice is to setup the output reference and set a heavy weight on its violation to keep the output close to the reference. It is also a good practice to leave the output variable unconstrained for some prediction horizon steps.

For the supermarket refrigeration system considered in this project, table 5.1 shows the summary of constraints on the manipulated variables and outputs.

Variable	Type	Min	Max	Rate Min	Rate Max
MTPsuc	MV	25 bar	34 bar	-0.5 bar	+0.5 bar
*Cabinet Temperature (EvapTemp)	MO	2 deg C	4 deg C	-inf	+inf
* The output of the NSS model is the evaporation temperature (MO), while the constraint is setup on the cabinet temperature from 2 -4 degree celsius. The difference between cabinet temperature and depends nonlinearly on suction pressure and ambient air temperature.					

Table 5.1: Summary of constraints for Economic MPC

5.1.4 Constraint Softening

In practice, disturbances and prediction errors are inevitable [30]. So despite the controller making its predictions, the violation in the constraints still can occur. All hard constraints may not be satisfied when optimal manipulated variable is used in the plant.

Hard constraints on MV bounds alone cannot lead to in-feasibility. The same is true for the case of hard MV rate constraint alone. However, hard constraint on both of these can lead to an infeasible solution. Except from the non-negativity of slack variable, all other constraints in the MPC can be soft constraints. Soft constraints allows controller to deem an optimal MV even if predicts a violation.

In multistage MPC, constraints can be soften by using slack variable e. The weight on the slack variable decides the amount of violation allowed. The tuning of weights of slack variable is discussed in section 5.3.

5.2 MPC strategy formulation

As discussed in chapter 4, model obtained for the power consumption of supermarket refrigeration system was a non-linear model estimated using neural state space (NSS). Hence, linear MPC method could not be utilized for the optimization purpose. Non-linear MPC was considered for the problem and there were two options to select for the formulation of non-linear MPC.

The first one was generic non-linear MPC controller which can utilize the linear/non-linear prediction model, equality constraints and cost function. The first choice was generic non-linear MPC, however, the authors soon ran into a problem because of limitations of MATLAB. The aim was to use predictions on ambient temperature and electricity price data for minimizing the cost of operation over the prediction horizon. However, the non-linear MPC does not support forecasts on disturbances and parameters on which were crucial for this project.

The second option was to use multistage non-liner MPC. Multistage non-linear MPC in MATLAB support the implementation of prediction on parameters along with disturbances. In multistage MPC cost and constraint functions can be different for each stage. A multistage MPC controller with the prediction horizon of p has p+1 stages where the first stage corresponds to the current time step and last stage corresponds to the last prediction step [31]. Each stage in multistage-MPC has its own decision variables, parameters, cost and constraint functions.

After a thorough discussion on formulation of MPC for cost minimization problem, two strategies came out as a result and it was decided to implement both strategies and then compare the performance. The first one is reference tracking MPC and the second is Economic MPC. Both strategies are explained in detail on the following pages.

5.2.1 Reference tracking MPC

In reference tracking MPC, the objective was to track the provided reference while respecting the required constraints. The goal was to maintain the cabinet temperature at the reference value.

During normal operation the cabinet temperature reference is 3 degrees Celsius. However, during peak hours, the reference would be changed to 3.8 degrees Celsius to maintain the temperature at upper bound in order to reduce consumption while reference would be changed to 2.2 degrees Celsius, when signal is received from the grid to utilise more power.

Cost function for reference tracking MPC

The cost function in case of reference tracking MPC was to minimize the error between reference evaporator temperature and measured output over the prediction horizon. The cost function was formulated as:

$$J_{ref} = \int_{T=0}^{H_p} (RefTevap - Tevap)^2 + w.edt \tag{5.2}$$

where e is the slack variable used and w is the weight on the slack variable.

5.2.2 Economic MPC

Supermarket refrigeration systems are affected by many disturbances that can be predicted with some level of uncertainty e.g ambient temperature. On the other hand, there are several constraints that need to be satisfied by the controller, while also minimizing the cost of operation. Economic MPC (EMPC) is well suited for this type of problem[4].

The term Economic MPC is derived from the applications where the objective is to minimize the cost of operation. The cost function in reference tracking MPC tries to minimize the error between reference and actual output, however, in case of EMPC the standard cost function can be replaced with a custom cost function. The main objective of the MPC implementation was to minimize the operational cost of the system. The economic objective function to minimize the operational cost of supermarket refrigeration was formulated by multiplying the electricity price $e_p(t)$ with the integral of power consumption of the compressor at the given time t. The cost function is computed over the prediction horizon:

$$J_{ec} = \int_{T=0}^{H_p} e_p \dot{W_{comp}} dt \tag{5.3}$$

In economic MPC, the performance of criteria can be combination of linear or nonlinear functions of states, inputs, and outputs [32]. An economic MPC can have the following properties:

- It can use both linear and non-linear models for prediction of states and outputs. As discussed in chapter 4, NSS was selected as model for compressor energy consumption, so the internal model used by economic MPC is non-linear.
- It can use both generic and built in quadratic cost function. The cost function formulated in equation 5.2 was used instead of the built-in quadratic function.
- SQP algorithm was used to solve the non-linear optimization problem to calculate the optimal moves.

5.3 EMPC formulation issues in MATLAB

While implementing EMPC and its formulation using MATLAB / Simulink Model Predictive Control Toolbox, the authors run into several issues that were caused either by their own programming mistakes or limitations and insufficient documentation of the software. As they took time and were of considerable importance, they are documented below.

5.3.1 Problem 1: using LSTM NN as a prediction model

First issue encountered was inability to run LSTM NN as a meaningful prediction model. Initial approach was to use predictAndUpdate function, which makes predictions one sample at a time and updates internal state of the network, for making predictions inside custom prediction model function which was implemented as in listing 5.1.

Listing 5.1: Prediction model implementation with predictAndUpdate

```
function x1 = stateFcn(x,u,price)
```

```

persistent LSTMmodel inM inStd outM outStd
if isempty(LSTMmodel)
    load('T5allPsucTestNet.mat');

    LSTMmodel = MTPcompLSTM;
    inM = inputMean;
    inStd = inputStd;
    outM = outputMean;
    outStd = outputStd;
end

u = (u - inM) ./ inStd;

[LSTMmodel,x1] = predictAndUpdateState(LSTMmodel,u);
x1 = denormalise(x1, outM, outStd);

end
```

Unfortunately, this kind of implementation is not compatible with the way the toolbox makes predictions. It expects that the provided function is of the state-space form meaning state of the system and the next time step is given as a function of state and input vectors at the current time step. On the contrary, the predictAndUpdate function works as an input-output relation, not making how internal state of the model is affecting the state at the next time step visible from outside of the function. Thus, despite having the same effect as predict function, it is not suitable for implementation in MPC in MATLAB either, as meaningful predictions on the state at the next time step cannot be made by the toolbox without knowing how internal state at the current time step affects it.

Fortunately, if NSS is used, MATLAB has an inbuilt generateMATLABFunction function for converting it automatically from a neural network into a nonlinear state-space equations using its weights and biases is available. Similar work could be done with LSTM NN, however, authors of MATLAB did not expect that LSTM NN could be used as prediction model in MPC, as only NSS can be converted from network to state-space format, so one would have to code own function to do so for LSTM NN. This could be done by either hard coding, which would take time and would be impractical whenever small change in the network is done during development, or by creating custom function for conversion, yet, it would take even more time. Due to time limitation, focusing more on concepts than specific implementation and making modelling adjustments allowing obtaining same results with NSS, that was not done.

It should be also noted that a side issue was resolved by using utilities for NSS, as initial MPC would increase simulation time significantly (it would take few minutes to proceed next 600 seconds). This happened because of MPC solver not being able to find optimum solution, resulting in taking maximum number of iterations, i.e. 120, of performing optimization over the whole prediction horizon. After proper implementation of NSS, it only took few seconds to solve MPC for one day of operation (while using NSS as a plant model as well as in case it was applied on digital twin, it did not increase simulation time significantly).

5.3.2 Problem 2: normalisation of signals used by EMPC

After selecting NSS as a prediction model and converting it into appropriate format, the authors run into another issue, which was normalisation of the data. As the network operates on normalised data, feeding data directly from digital twin caused the EMPC not to work.

To be specific, this led to the EMPC not producing any new output as it kept initial value over the whole simulation. The solution to this mistake was simple, normalising input data (feedback) to the controller and denormalising its output data (reference input to the suction pressure controller). However, it should be noted that it was realised that it could cause another issue, which was cost function misbehaving due to operating on values normalised with z-score. For example, if the EMPC could decide to decrease suction pressure to decrease power (which is the opposite to the actual relationship), as the cost function defined in terms of normalised values would perceive all suction pressure values below mean (28 bar) as less "expensive", owing to them being negative. This was resolved, by denormalising values in the custom cost function before applying its final computation.

Nevertheless, another overlooked issue was noticed, which was EMPC not respecting cooled cabinet temperature. This was caused due to the fact that decrease of power an increase in COP are equivalent objectives (which were the only outputs of the network at the time), none of which accounted for "quality" of cooling. This resulted in the suction pressure to be risen to upper constraint at all times, not respecting the issue of variation of temperature of the cabinet with cooling load (affected by ambient air temperature), which requires the suction pressure to be continuously varied.

This had to be reflected with additional constraints that take into account the cabinet temperature, which development in described in the next section.

5.3.3 Problem 3: temperature-preserving constraints

The main idea in maintaining cabinet temperature was to construct such constraints that suction pressure remains in the bounds that keep the cabinet temperature within desired limits (2-4 degrees Celsius), despite operating close to these limits. 2 approaches to this were initially tested i.e. obtaining cabinet or evaporation temperature as a static function of suction pressure and ambient air temperature, and using evaporation temperature as an extra output of the prediction model.

First approach came from the idea of investigating dependency of temperatures on the evaporator side on other signals. It was found that these signals depend on suction pressure and ambient air temperature, which is expected, as they are all related to cooling. If this dependency is modelled as static function that converts the inputs to the temperature, then it is possible to limit suction pressure correctly. Due to ambient air temperature not being under our control, we can consider it as a fixed value for a specific time step and find the minimum and maximum pressure for which temperature constraint is not violated, as the static function becomes a function of suction pressure only then and direct conversion to the temperature occurs.

Figures 5.2 and 5.3 illustrate such dependencies. These plots were created from reduced number of total of 31 9steady state values (or mean values after transient removal) of respective temperatures from simulation runs overall containing all combinations of 11 fixed suction (25:35 bar) pressures with 29 fixed ambient air temperatures (9:35 degrees Celsius). The range of temperature for this plot was minimally limited to exclude values that were out of digital twin's operating envelope.

In the beginning, despite not being practical in reality, direct relationship between the cabinet temperature and suction pressure / ambient air temperature was constructed, to test if the idea works. However, the controller was ignoring this constraint and always lifted suction pressure to the maximum specified as upper constraint on suction pressure only, which was implemented only for having extra check on correctness of the method used. Thus, the temperature was not maintained (note that initially it was confused with superheat, however, the controller could not manage this relation either).

Then the approach was changed to include evaporation temperature as an extra predicted signal in the model such that the suction pressure is chosen not to exceed certain bounds that maintain cabinet temperature in the desired range. The evaporation temperature was chosen specifically, because of avoiding resorting to direct measurements of cabinet temperature. The advantage of that is if the method is

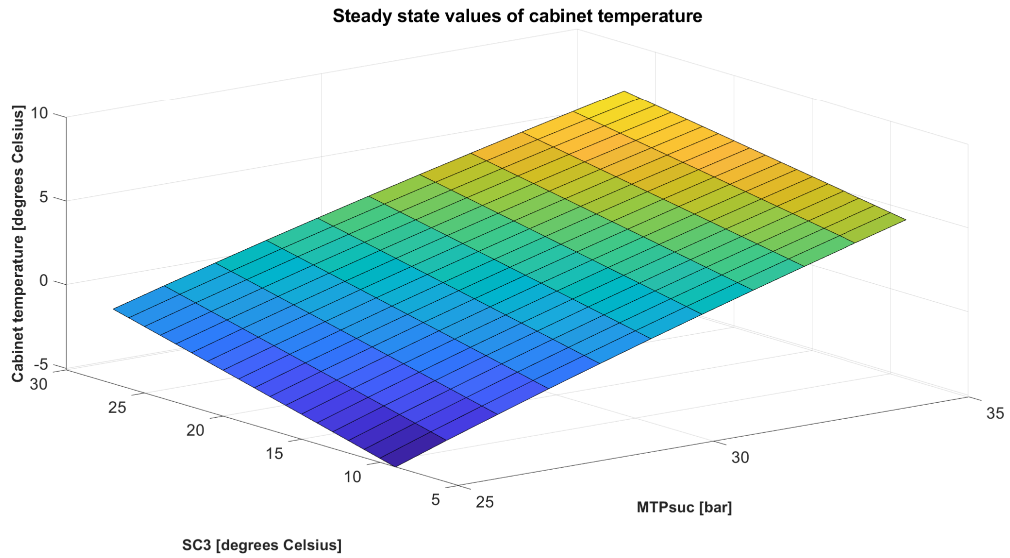


Figure 5.2: Cabinet temperature as a function of ambient air temperature and suction pressure

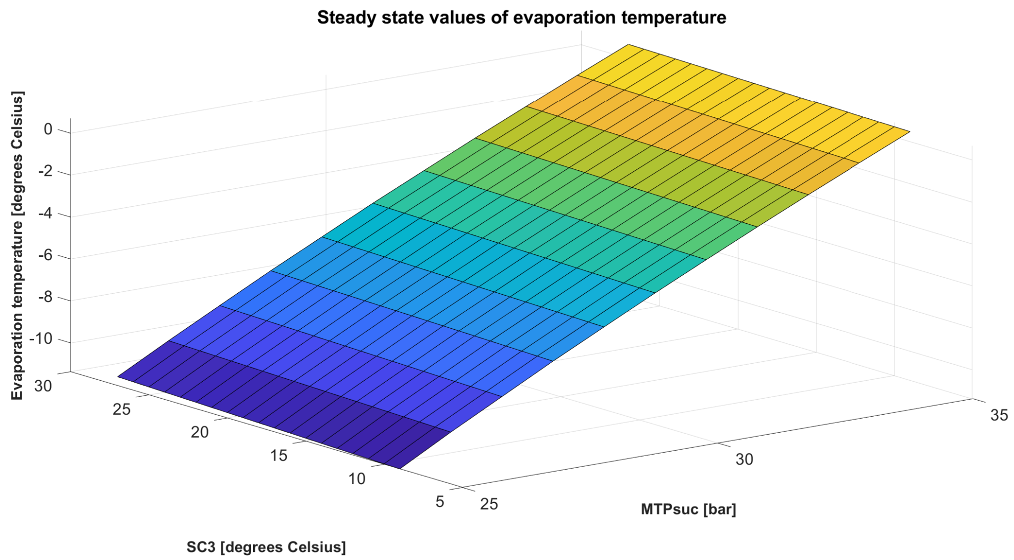


Figure 5.3: Evaporation temperature as a function of ambient air temperature and suction pressure

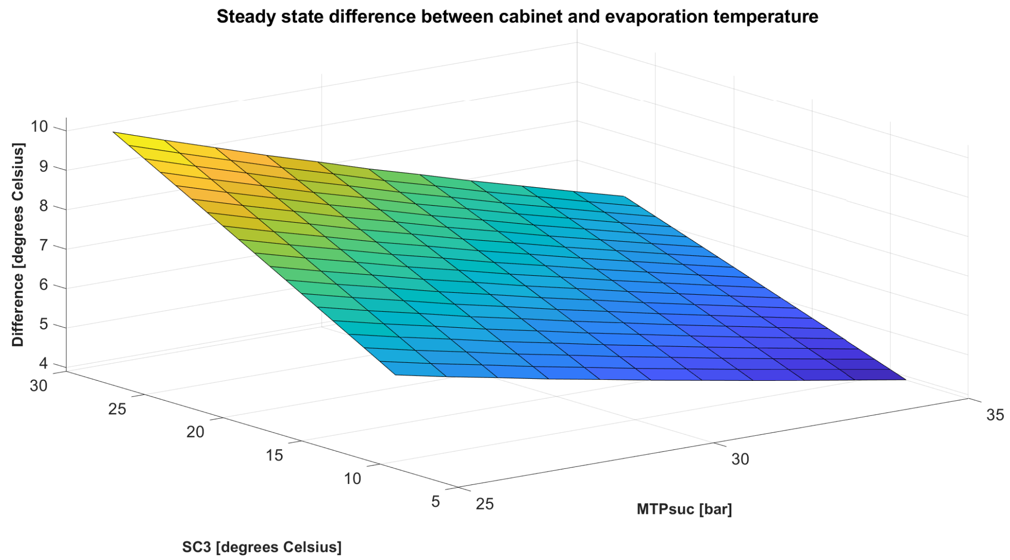


Figure 5.4: Difference between evaporation temperature and cabinet temperature as a function of ambient air temperature and suction pressure

applied on a real system, there would not be problems with scaling, as all the evaporators have same evaporation temperature applied to them.

Nevertheless, the previous work on static modelling of relationships between signals had not gone futile, as there was a need for another static function, namely the difference between evaporation and cabinet temperatures, which also changes with suction pressure and ambient air temperature. This value is required in order to translate evaporation temperature into final value of cabinet temperature. This could be done under assumption that we are not interested about value of cabinet temperature specifically, so long as it is bounded for bounded inputs and we are not interested in its value during transients to new setpoints (instead we are interested if its final value does not violate the constraint). The former comes from the fact that the system is inherently stable, due to local controller, while the latter is justified that in reality the step response of the system will be approximately of slow first-order model.

The plot of differences between the temperatures can be seen in figure 5.4. Low order (second order for suction pressure and first order for ambient air temperature) polynomial was fit with MATLAB fit function to the data points as seen in figure 5.5. The function was implemented for 2 constraints as seen in listing. The normalised values of inputs are first converted to their denormalised versions, like in cost function, to correspond to the variables in the fit polynomial and then they are applied to the function.

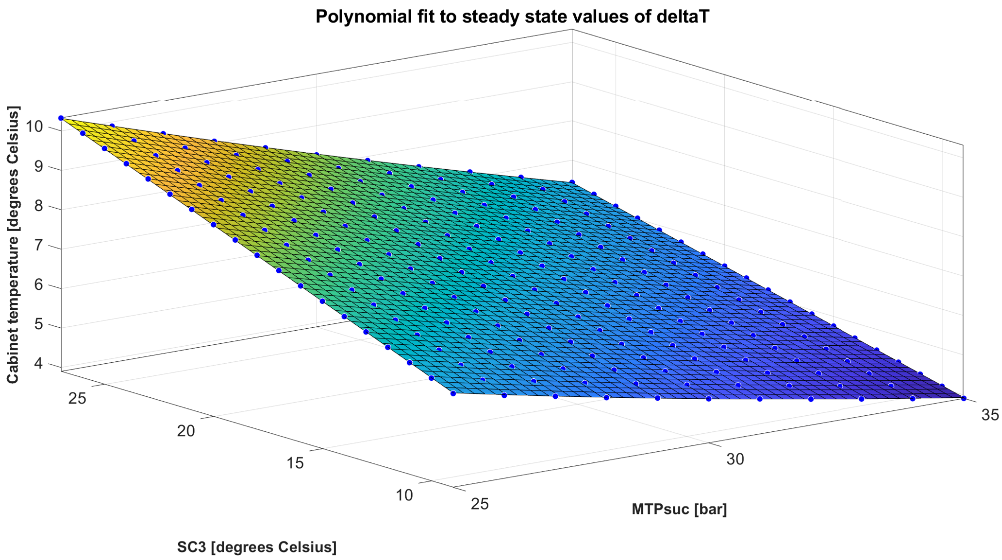


Figure 5.5: Polynomial fit to the difference function

Listing 5.2: Final implementation of temperature-preserving constraints

```
function C = myIneqConFunctionwSlack(stage, x, u, dmv, e,
    ref)

persistent outM outStd inM inStd
if isempty(outM)
    load("netInfo.mat")
    outM =outputMean;
    outStd = outputStd;
    inM = inputMean;
    inStd = inputStd;
end

p00 =       11.62;
p10 =     -0.3888;
p01 =      0.4357;
p20 =    0.003815;
p11 =   -0.008486;

%denormalising values
Evap_temp=x(2)*outStd(2)+outM(2);
SC3 = u(1)*inStd(1)+inM(1);
```

```
MTPsuc = u(2)*inStd(2)+inM(2);

dT = p00 + p10*MTPsuc + p01*SC3 + p20*MTPsuc^2 + p11*MTPsuc
   *SC3;

C = [(-Evap_temp + 273.15 - dT + 2 -e(1));
     (Evap_temp - 273.15 + dT -4 - e(2))];

end
```

The implementation was successful by approaching the problem this way. As a last remark, it should be added that in real system this difference or constraint value could be obtained through an estimator providing single value from measurements coming from multiple evaporators, as it is not possible to acquire steady state values for the whole range of pressures and temperatures for the real system.

5.4 Tuning

MPC controller design involve many parameters that strongly affect the performance and robustness of the controller [33]. MPC design requires tuning of cost function weights, setup of rates of change of manipulated variables and defining control and prediction horizons for optimal performance. This section outlines the steps taken to tune reference tracking MPC and Economic MPC for energy optimization problem for supermarket refrigeration systems. Performance of MPC was evaluated by selecting different weights of slack variable, rate of change of manipulated variable and prediction horizons. Effect on MPC performance is depicted in the graphical format on the following pages. The best combination of weights, prediction horizons and rate of change were selected for the final design for each type of controller.

5.4.1 Temperature reference-tracking MPC

In this section, the effects of weights of slack variable for constraint softening, rate of change of MV and selection of different horizons are discussed for reference tracking MPC.

Constraint softening and tuning of slack variable weight

In the first step, softening of constraints was required to enable the controller to solve optimization problem faster. When the controller was tested without introducing slack variables, it led to very slow execution of simulation in the beginning. This was because at the start the controller starts in the infeasible region and it took it a long time to come to feasible region in the absence of soft constraints.

After introducing the slack variable, the next step was the selection of weight on the slack variable for constraint violation. The smaller weight allows the controller to violate the constraint more while a large weight discourages the controller to violate the constraint.

Figure 5.6 shows the effect of weight on the slack variable in case of reference tracking MPC. The prediction horizon was fixed at 12 steps (2 hours) for all the cases of weights. It can be seen in the figure that the effect of very small and very large weight on the slack variable is the same in this case. This is because of the absence of any sudden change in the disturbance (Ambient temperature in this case) and controller is able to maintain the output very close to the reference value. However, to depict the effect of slack weight value, a simulation run was performed with a sudden drop in the value of ambient temperature (which can happen in reality). It can be seen from the figure 5.7 that small values of weights allows the controller to drop the cabinet temperature (Ther Air) below minimum required value i.e. 2 deg C. However, for larger weights on the slack, the controller maintains the output (Ther Air) above 2 deg C.

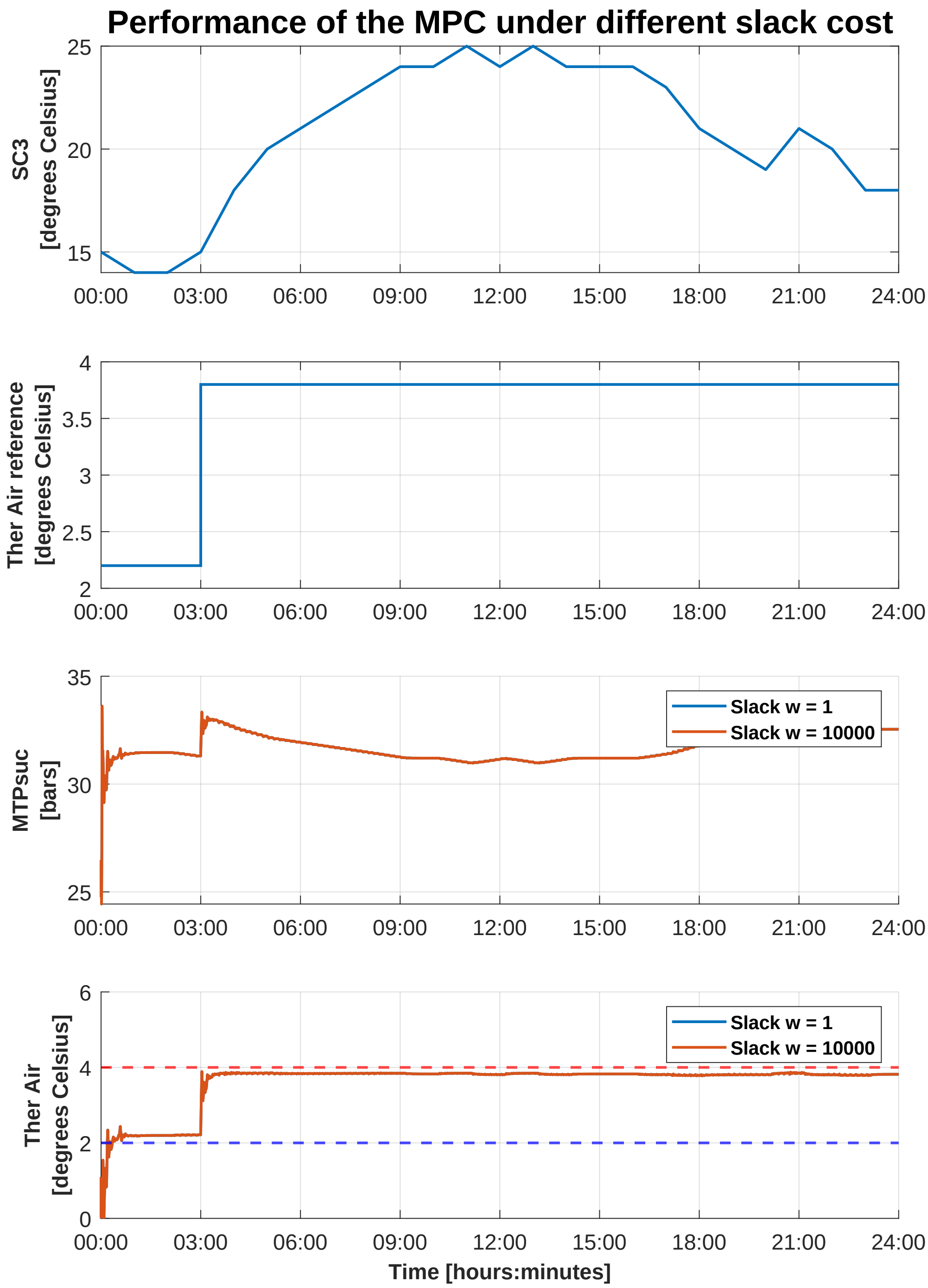


Figure 5.6: MPC constraint softening using slack variable

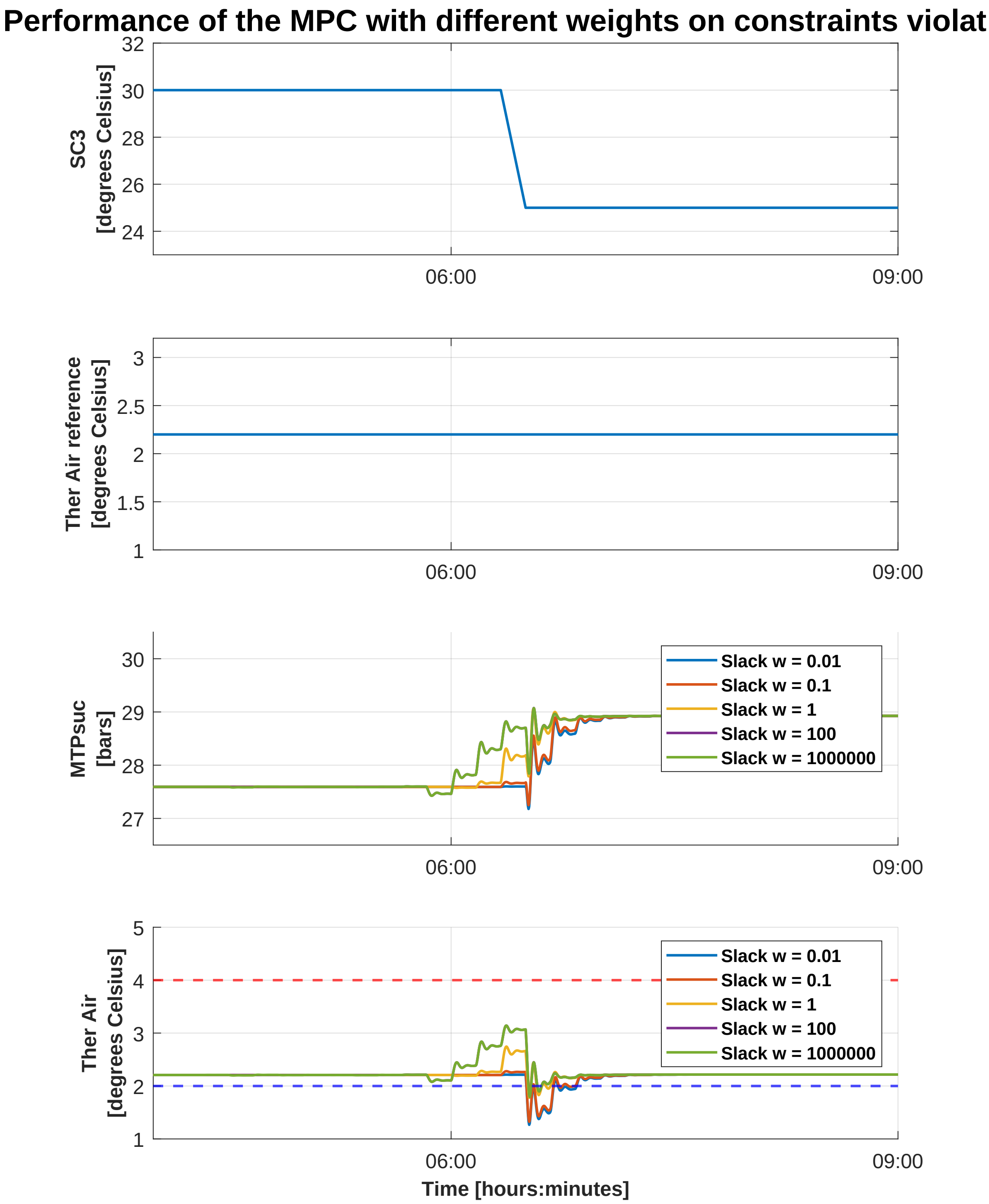


Figure 5.7: MPC performance with different slack under temperature drop

Cost on the rate of suction pressure change

Cost on the rate of change of manipulated variable (MV) is required to prevent the controller to take aggressive control actions. However, limiting the controller action is the trade off between performance and aggressiveness. Figures 5.8 and 5.9 show the effect of different values of weights on rate of manipulated variable (MTPsuc). A smaller weight allows the controller to take more aggressive action while a large weight prevents it to take aggressive action. Aggressive behaviour is not desirable since in practice actuators have slew rate and cannot change the value instantly. On the other hand, imposing a large weight causes the controller to track the reference value with large delay which is undesirable. It must be noted that figures 5.8 and 5.9 show effect of different weights on the rate of manipulated variable in case of non-linear MPC because implementation of weights on the rate of manipulated variable was not successful in MATLAB for the case of multistage non-linear MPC. However, the problem was solved using the hard constraint on the rate of manipulated variable (MTPsuc) and is shown in the figure 5.10.

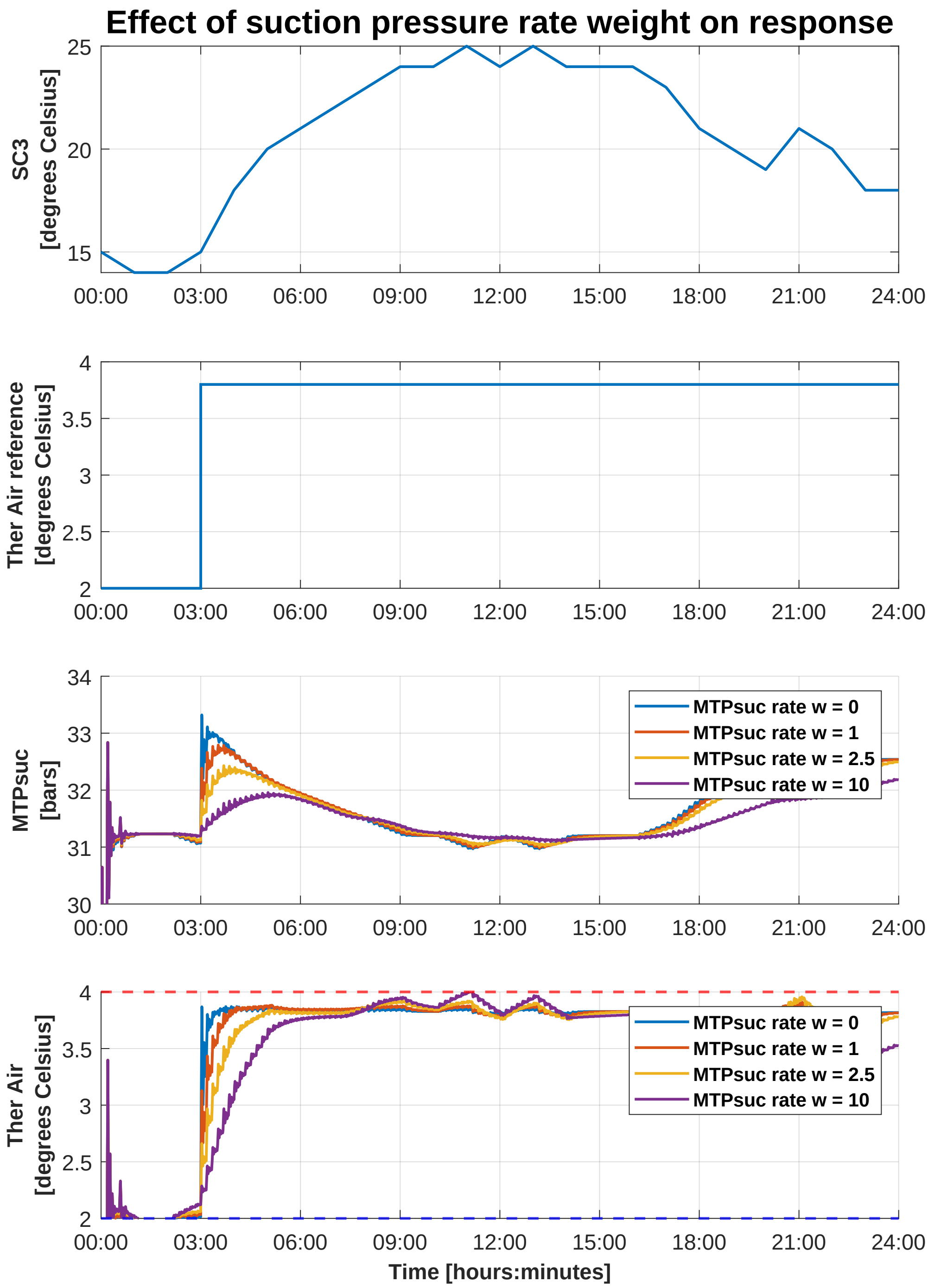


Figure 5.8: Effect of suction pressure rate weight on response

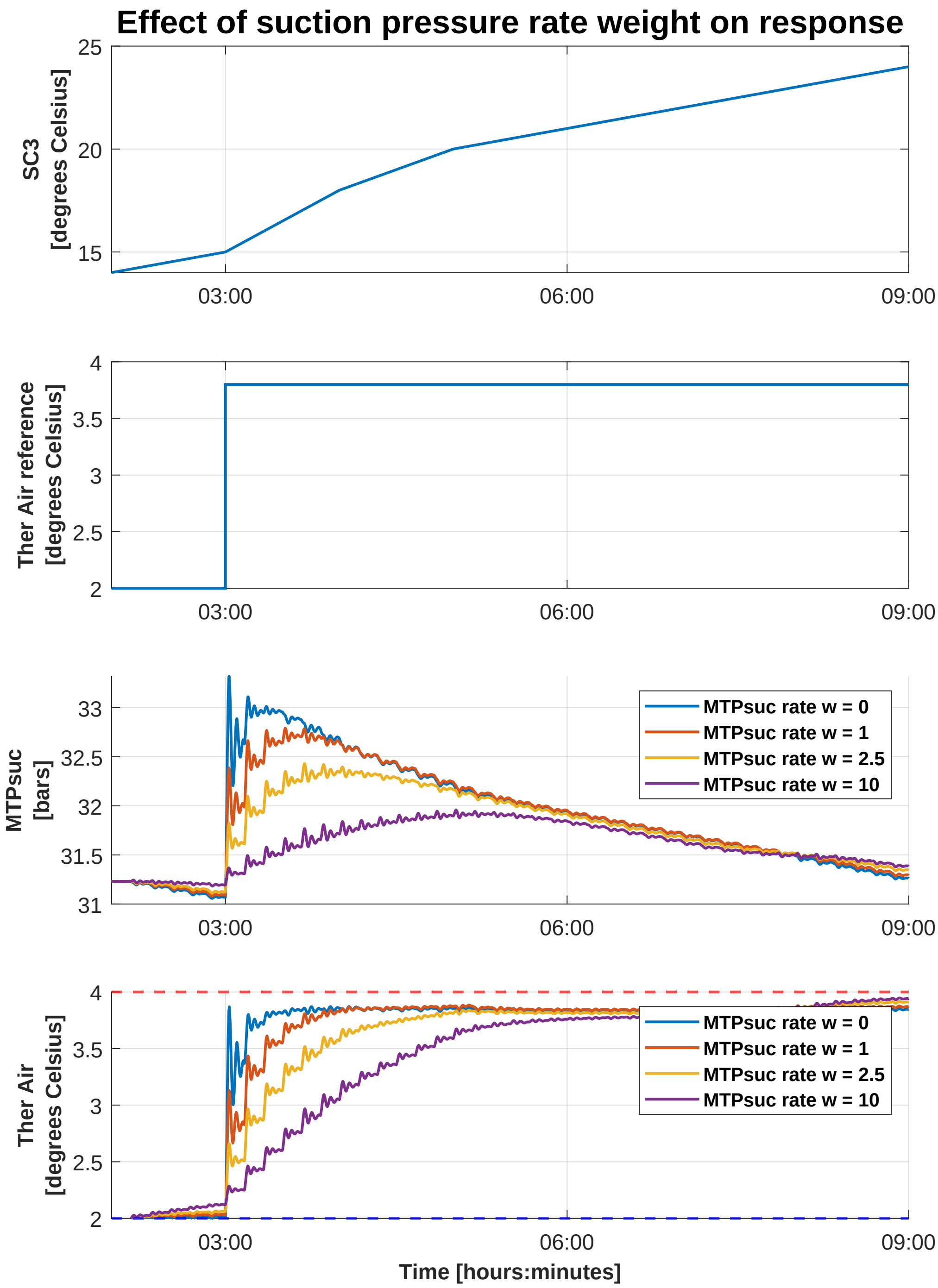


Figure 5.9: Effect of suction pressure rate weight on response (Zoomed in)

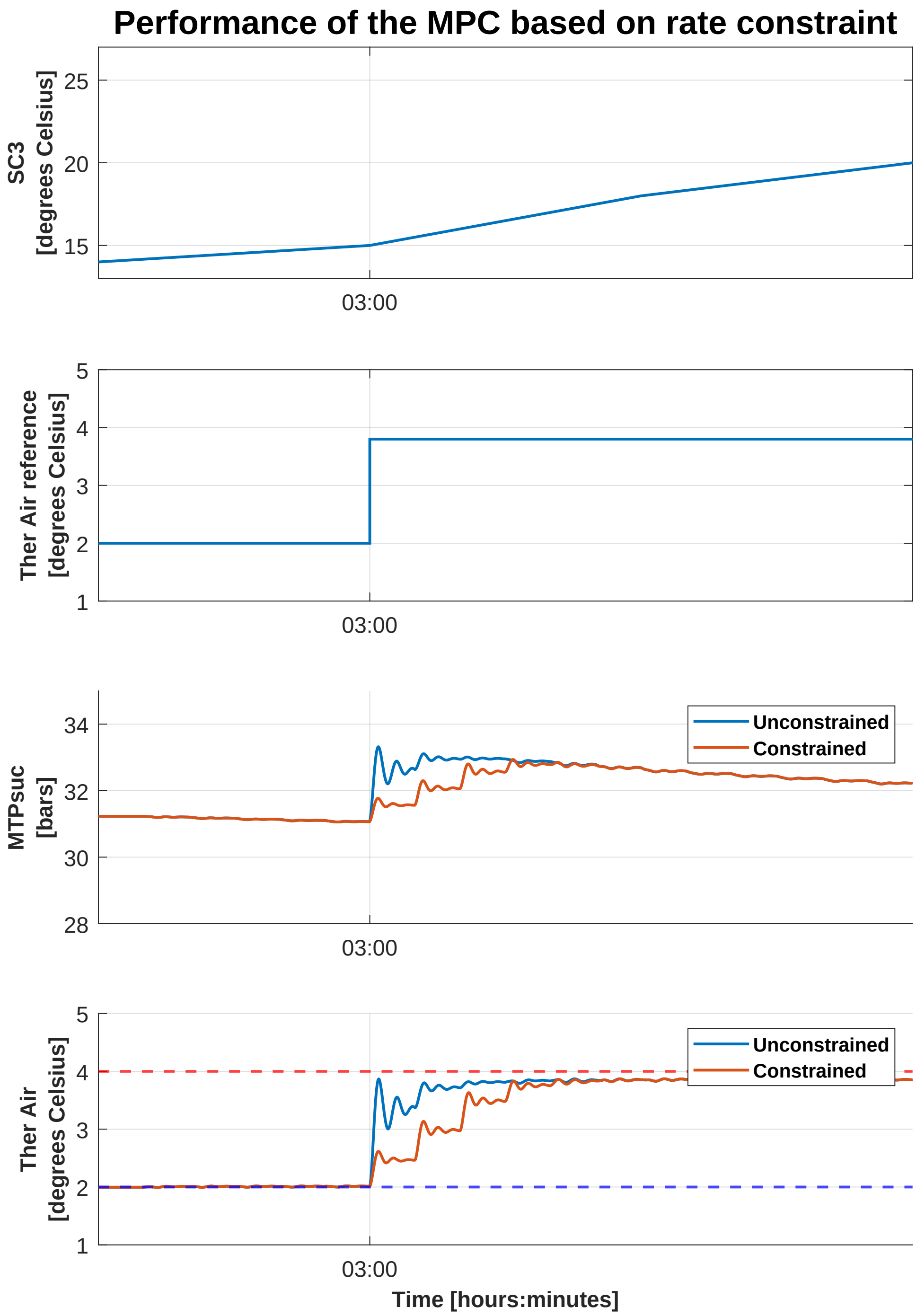


Figure 5.10: Performance of MPC based on rate constraint

Tuning of horizons

An MPC controller find the optimal moves by predicting the output using the internal plant model and predictions on disturbances. Prediction horizon is the future control interval that MPC must evaluate by prediction when optimizing the manipulated variable. As the prediction horizon increases, the computation cost of MPC also increases. Also, increasing the prediction horizon after certain steps do not improve the MPC performance while increase the computation cost.

In case of reference tracking MPC, the effect of different prediction horizons can be seen in figure 5.11. It can be seen that even with the smallest value of prediction horizon i.e. 2 steps, MPC can accurately track the reference and also caters for sudden change in disturbance. Also, there is no noticeable difference between small and large values of perdition horizon in this case. Hence, utilizing large values of prediction horizon are not needed in this case.

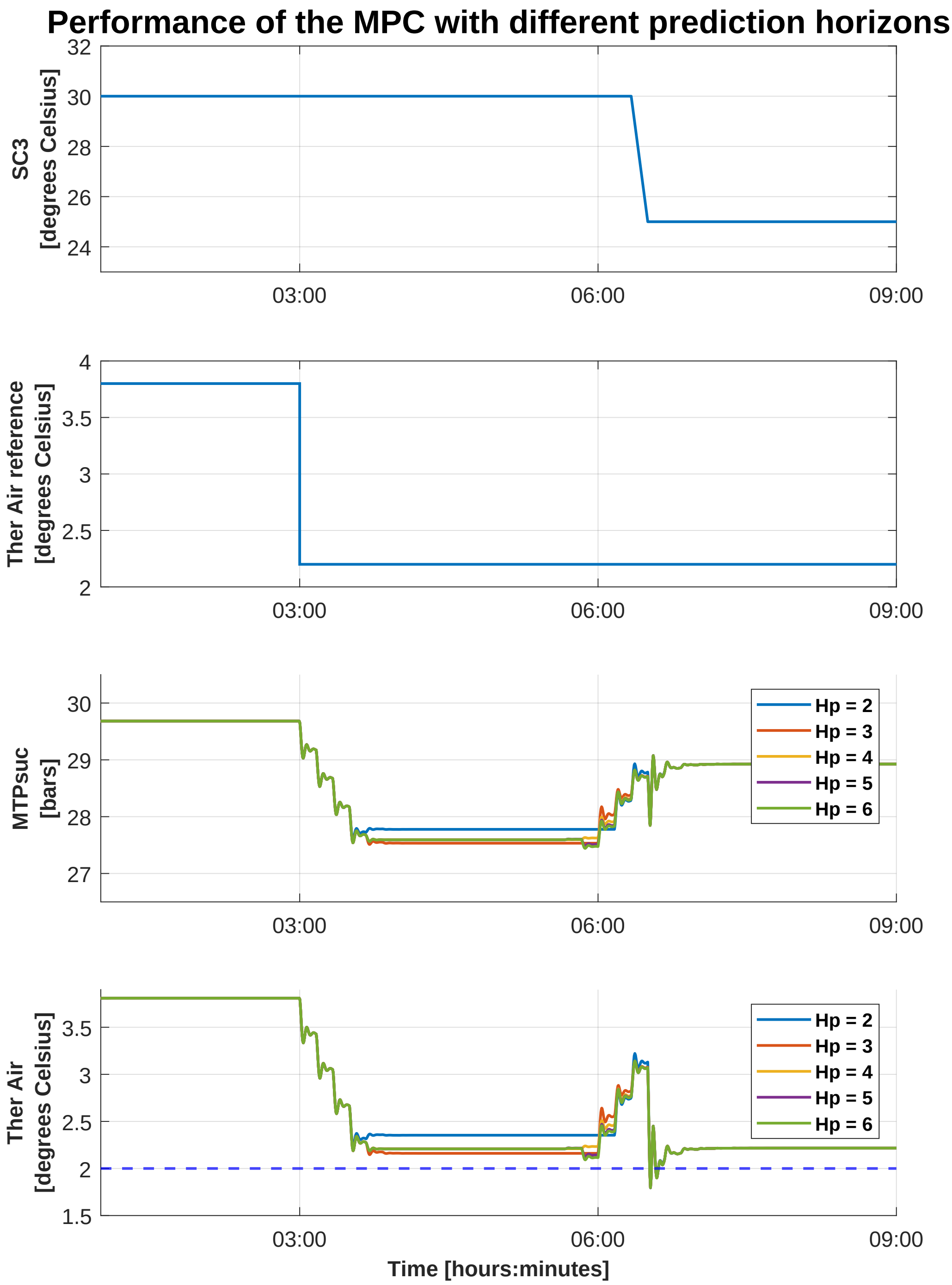


Figure 5.11: MPC performance with different horizons

5.4.2 Economic MPC

In this section, tuning of Economic MPC is presented. First the weights on the slack variables were tuned. Then constrains were set on the rate of manipulated variable (MTPsuc) and last horizon tuning was done by taking price data for one day.

Tuning of weights on slack variable

Slack variable is used to soften the constraint on the cabinet temperature bounds. The bounds are from 2 - 4 degree . For the tuning of slack variable, ambient temperature profile with temperature range from 15 to 25 C was selected and data for the price was fixed to a constant value. Figure 5.12 shows the behavior of cabinet temperature for different values of slack variable weight. A low value of slack variable weight allows for more constraint violation and it can be seen from the figure that cabinet temperature reaches 6 degrees. Hence, a higher value of 10^6 was selected as the weight of slack variable.

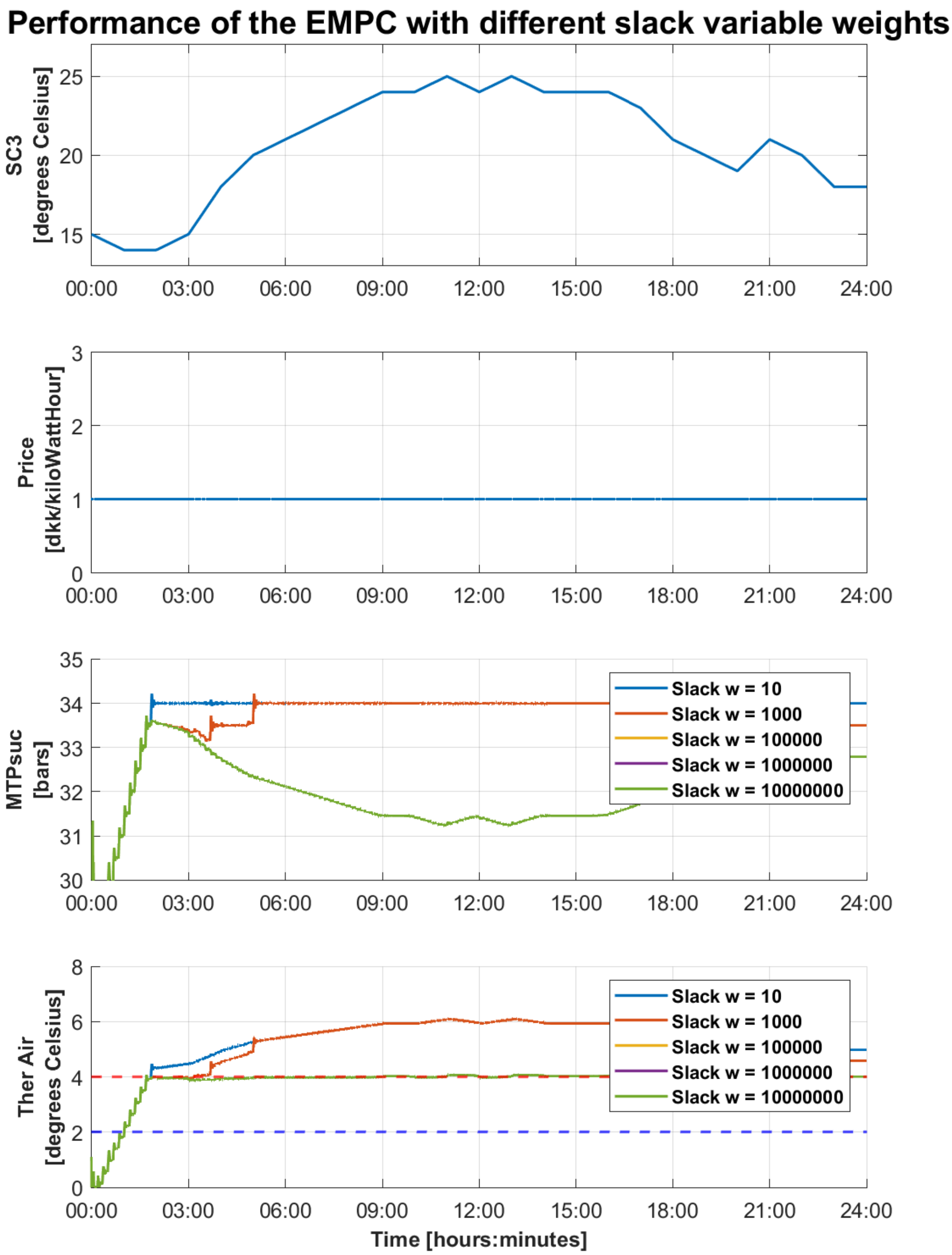


Figure 5.12: EMPC performance with different slack weights

Constraining rate of change of suction pressure

A hard constraint of $\pm 0.5 bar$ was imposed on the rate of suction pressure to avoid controller aggressive behaviour.

Selecting horizons for load shifting

Selection of proper prediction horizon in economic MPC could lead to significant cost savings. First a simulation run was performed as benchmark for the ambient temperature profile as shown in the figure 5.13. The total cost of operation for one day was calculated using the price data as depicted in the figure. Then same ambient temperature condition and price data was used to tune MPC prediction horizon. Figure 5.13 shows the suction pressure and cabinet temperature (Ther Air) for different prediction horizons. Price savings are more significant with longer prediction horizons. However, computational cost of MPC increases with large prediction horizon.

5.5 Summary of tuning of MPCs and final formulations

After analysing the effects of constraints softening, weights of slack variables, rate of change of manipulated variable and prediction horizons in the section 5.3, the following final formulation were made for Tref and EMPC.

5.5.1 Reference-tracking MPC

General formulation, cost and inequality constraint functions of final Tref MPC are provided below.

General Formulation

Sample time : 600 seconds
Prediction horizon : 1 hour (6 steps)
State 1 : MT compressor power (MTPcomp)
State 2 : Evaporation temperature (EvapT)
Manipulated variable : MT suction pressure (MTPsuc)
Measured disturbance 1 : Ambient temperature (SC3)
Measured disturbance 2 : Gas cooler pressure (Pgc)

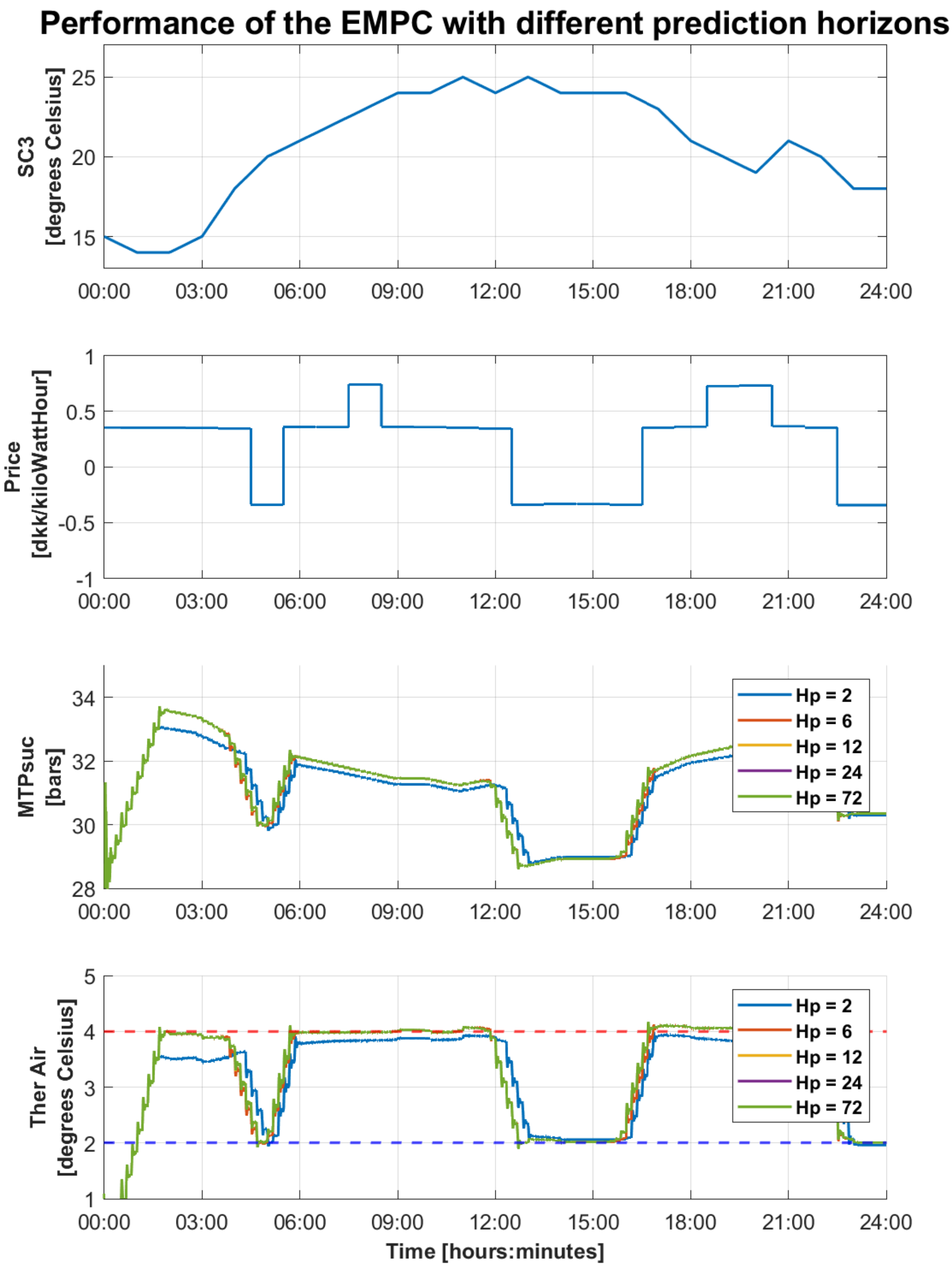


Figure 5.13: EMPC performance with different horizons

Parameter : Electricity price (DKK/KWh)

Cost functions

Cost function for first stage :

$$J = \sum_{i=0}^{H_p} (Tevap_{ref} - Tevap)^2 \tag{5.4}$$

Cost function for remaining stages :

$$J = \sum_{i=0}^{H_p} (Tevap_{ref} - Tevap)^2 + 100(e_1 + e_2) \tag{5.5}$$

Inequality constraint function

$$dT = 11.62 - 0.38MTPsuc + 0.43SC3 + 0.0038MTPsuc^2 - 0.0084MTPsuc.SC3 \tag{5.6}$$

$$-Tevap + 273.15 - dT + 2 - e_1 \geq 0 \tag{5.7}$$

$$Tevap - 273.15 + dT - 4 - e_2 \leq 0 \tag{5.8}$$

5.5.2 Economic MPC

General formulation, cost and inequality constraint functions of final EMPC are as follows:

General Formulation

Sample time : 600 seconds
Prediction horizon : 1 hour (6 steps)
State 1 : MT compressor power (MTPcomp)
State 2 : Evaporation temperature (EvapT)
Manipulated variable : MT suction pressure (MTPsuc)
Measured disturbance 1 : Ambient temperature (SC3)
Measured disturbance 2 : Gas cooler pressure (Pgc)
Parameter : Electricity price (DKK/KWh)

Cost functions

Cost function for first stage :

$$J = \sum_{i=0}^{H_p} e_p P_{comp} \tag{5.9}$$

Cost function for remaining stages :

$$J = \sum_{i=0}^{H_p} e_p P_{comp} + 1000000(e_1 + e_2) \tag{5.10}$$

Inequality constraint function

$$dT = 11.62 - 0.38MTPsuc + 0.43SC3 + 0.0038MTPsuc^2 - 0.0084MTPsuc.SC3 \tag{5.11}$$

$$-Tevap + 273.15 - dT + 2 - e_1 \geq 0 \tag{5.12}$$

$$Tevap - 273.15 + dT - 4 - e_2 \leq 0 \tag{5.13}$$

Chapter 6

Reliable MPC operation under temperature sensor fault

This chapter verifies the methods and results from [22] for bias compensation on the fast model and discusses impact of not applying them in case bias is present on the proposed MPC scheme for energy-cost minimization.

6.1 Bias issue

In order to investigate impact of bias on performance of the developed methods, bias needed to be generated to be inserted into the simulation model. Fortunately, authors of [22] also provided methods for constructing realistic bias. Once that was implemented, its effect on the performance of the MPC schemes was checked.

6.1.1 Bias modelling

The bias consists of 3 terms i.e. term caused by sun itself, term reflecting cloudiness and night time bias, where the sun component reflects change of sun intensity over day, cloudiness term accounts for variations in the intensity, due to clouds, and night bias corresponds to negative bias observed during night hours, each of them having a random component. Combining all of them leads to a realistic and unpredictable bias. The reconstruction of these terms is explained in the following subsection.

Fixed design parameters of the bias terms were taken directly from [22]. However, there were also some tuning parameters that allowed for design flexibility depending on the season and location of the supermarket. These included peak and spread of the sunlight term. It was tuned such that the term lasts for about 12 hours and its peak is between hours of 12 and 13.

6.1.2 Performance of controllers under fault

To illustrate the effect of the bias on the performance of the controllers, simulations were run with the same setup as in the runs for evaluating economic performance of the controllers for ease of comparison.

Example simulation run for EMPC can be seen in figure 6.1. It can be spotted that it has issue maintaining desired temperature whenever large value of bias appears, often leading to violation of cabinet temperature lower constraint, as the MPC schemes perceive the ambient temperature much higher than actual one and try to unnecessarily provide more cooling to the system that lowers the temperature.

Nonetheless, EMPC was still more to be more robust to the bias than MPC, as it has less violations of the constraint. This could be due to the fact that in MPC, the evaporation temperature reference is also affected by the bias, which always drives it below the constraint, even when required cabinet temperature is desired to be at the higher limit. EMPC only experiences this when it decides to lower down the temperature based on prices, however, the performance is still degraded.

This shows that the bias consistently degrades the performance whenever present in large quantity in the system. Therefore, it should be compensated for not only to preserve optimal COP, but also to make the operation of MPC schemes reliable, such that they do not violate constraints and preserve their performance.

6.2 Implementation and verification of bias compensation

As shown in previous section, there is a need for implementing solution for getting rid of the bias along with the developed MPC schemes. To do so, procedure from [22] is replicated and the results are verified to check its applicability to the fast simulation model.

6.2.1 ANN for bias estimation

The bias compensation consist of 2 steps i.e. creating a residual signal indicating the amount of bias present in the measurements and a compensator that subtracts

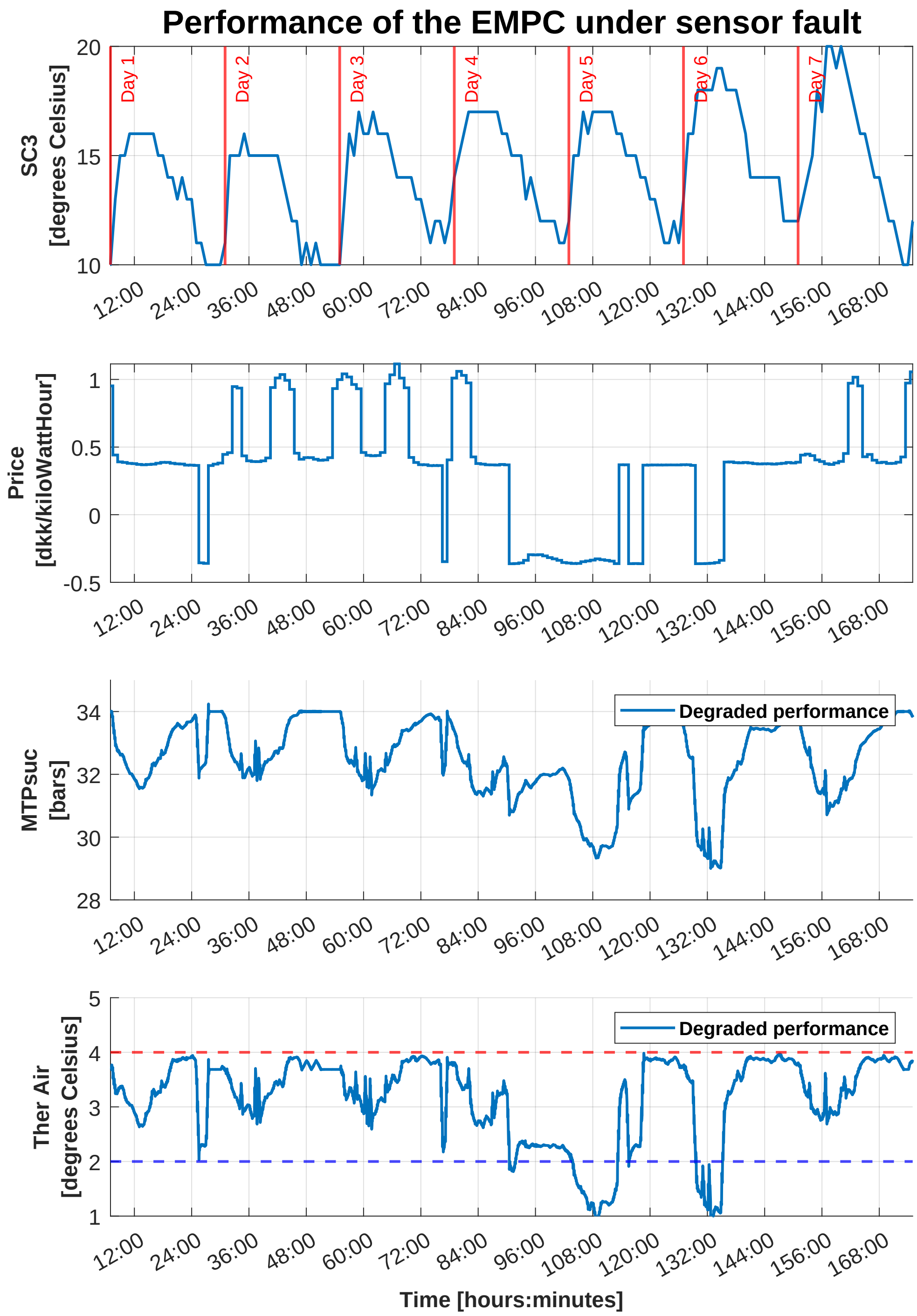


Figure 6.1: EMPC performance under ambient air temperature sensor fault

this bias from the ambient air temperature measurements. To generate the residual, a feedforward ANN that can determine difference between true value and bias measurement was used in the original work [22].

The training data were acquired for the network in a test lasting 12 hours in simulation under ambient air temperature typical for summer nights in order to get data of length equivalent to 2 nights. The simulation was done for the system without MPC in place, to first test if we are able to reproduce the results for plain system without any modifications that could potentially complicate the implementation.

Initially the network was trained using same setup as in the original work. Nevertheless, 2 issues were spotted. One of them was use of suction pressure in the model. It was expected that if the suction pressure deviates away from the fixed reference present in the training data, the ANN will not perform well, which was the case. The solution to this was not using the suction pressure, which did not affect the accuracy of the model much. Alternatively, the residual generation model could be train during operation of MPC that adjusts the pressure continuously, however that was not done, because of time limitations.

The second issue was more unexpected, as it turned out later that the bias compensation does not work well for temperatures below the ones seen in the training of the ANN. It was resolved by training the network based on autumn night temperatures, but in real life scenario this could be easily done by retraining the network once in a while.

In the end, the only difference as compared to the original paper was dropping out the suction pressure from the model's inputs. The trained network was not capable of reconstructing the perturbation signal perfectly (regardless of excluding suction pressure or not). However, that is not an issue, as it is demonstrated in the next part of this section that while the bias is being compensated good results can still be obtained, as the residual stays low, thus allowing for better performance of the ANN.

6.2.2 Verification of bias compensation on fast model

The only way to truly test if the network estimates the bias correctly was through its combination with the compensator, as it cannot handle large values of bias. The compensator in form of integral control was needed to make the residual generator respond to derivative (which is small, given bias changes slowly) of the bias present in the estimation of the ambient air temperature rather than to the remaining bias itself.

Thanks to the integrator, the changes in the bias with respect to time could be correctly estimated by the residual generator, due to their small size, while the compensator could recover the remaining bias in the estimation of the ambient air temperature through ntegration of the resiidual signal. It should be noted here, that the input to the network is the estimate of the temperature instead of biased measurements, as we are interested in rejecting the bias remaining in the estimate.

The main problem within compensator was to tune its gain. As explained by the authors of the compensation idea, its tuning is a trade-off between rejecting bias and modelling error, where high gain makes the system less sensitive to bias, while increasing impact of the modelling error. The opposite is true for low gain, where less modelling error is experienced, but the system contains more bias [22]. This was in fact true once we attempted to tune the gain of compensator and posed an issue, as it was observed that even small deviations from the right value of the gain led to estimate either go unstable (for gains above optimum) or containing large quantities of bias (for gains below optimum). The optimal value for the gain was found to be 0.0025.

2 issues should be discussed here. First is that the compensator's gain was tuned manually, while to apply it to the real system it might need to be adjusted. When the gain as initially tuned, the value of UAir was still incorrect one. Interestingly, after that value was changed, the gain had to be retuned despite using the same network and signals, which indicates a need for automatic adjustment of the gain, based on the system. Another issue is linked to the first one, as it considers the sensitivity of the compensator to the selection of gain. Some way of robustifying compensator to the choice of gain should be implemented, such that the system still performs for some non-ideal values of the gain, as it might be not selected exactly, while adjusting it to the system automatically. However, challenges posed by these issues are considered as a future work.

The compensator was implemented in Simulink, as depicted by figure 6.2. It includes measurements from high pressure side of the system and input normalisation block, to meet the network's input requirements. The results of 2-days test of the compensator on the default system that is not controlled by either of MPC solutions. The effectiveness of the compensation method while either of MPC solutions is active is documented in results chapter, where comparison of the performance under no bias with the performance of present but compensated bias is made.

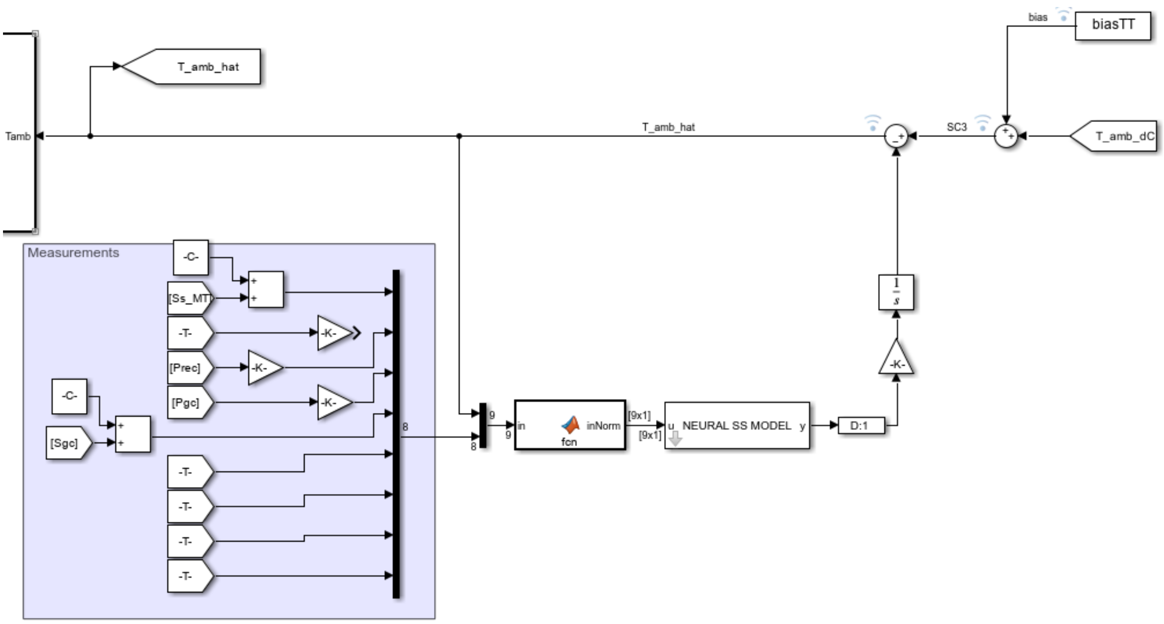


Figure 6.2: Compensator implementation in simulink

Chapter 7

Results

In this chapter results of MPC implementation are shown in terms of changes in overall operational cost, power consumption and maintaining of required cabinet temperature. As discussed in chapter 5, two types of MPC implementation were performed to compare the implementation complexity and performance; the first one being temperature reference MPC (Tref) and the second one being Economic MPC (EMPC).

The performance of both type of MPC implementation is compared with a Baseline system implementation for one week of simulation run. Controller for maintaining the cabinet temperature was not implemented in the fast model provided by Danfoss. Hence, it was decided to implement cabinet temperature control by MPC. The nominal temperature value of cabinet temperature is 3 degrees Celsius, with minimum and maximum range from 2 to 4 degrees Celsius. The baseline was designed to maintain the nominal temperature range however, Tref and EMPC were designed to utilize the information from price and ambient temperature predictions to minimize the cost of operation.

7.0.1 Evaluation based on ambient temperature profiles

Energy consumption of supermarket refrigeration systems depend highly on the ambient air temperature. Hence, it was necessary to simulate the system for different ambient temperature profiles and compare the results. Hourly data of ambient temperature was obtained from [34]. Figures 7.2 and 7.1 show the profiles of ambient temperature data for hot and moderate climate regions. To analyze the performance of system for different ambient temperature conditions, three week of ambient temperature data in the following range was used to run the simulations and compare results.

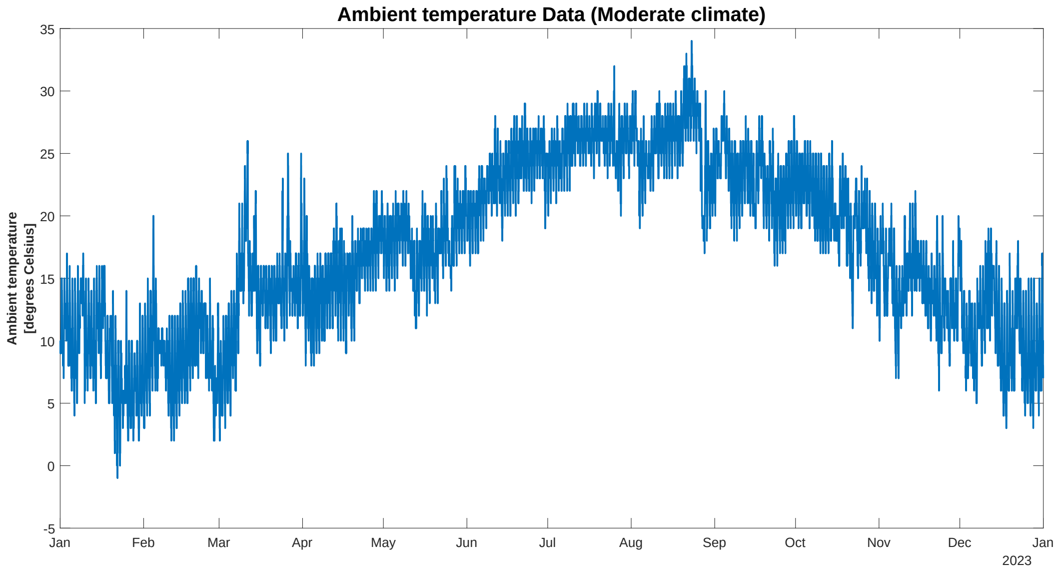


Figure 7.1: Ambient temperature Data (Moderate climate)

- 10 to 20 deg C
- 16 to 26 deg C
- 23 to 35 deg C

7.0.2 Comparison of Power, MTPsuc and Ther Air

Figures 7.3, 7.4 and 7.5 show the comparison of power consumption, suction pressure (MTPsuc) and cabinet temperature (Ther Air) for baseline, Tref and EMPC for 3 different ranges of ambient temperature. Following inferences can be drawn from the graphs.

- Cabinet temperature is maintained between the limits i.e. 2 to 4 deg C in all the cases of implementation. This is the primary objective of refrigeration system.
- Behaviour of suction pressure is significantly different in all the 3 implementations based on price and SC3 data.
- Based on the changes in the MTPsuc, profiles of power consumption are different for all the 3 implementations.

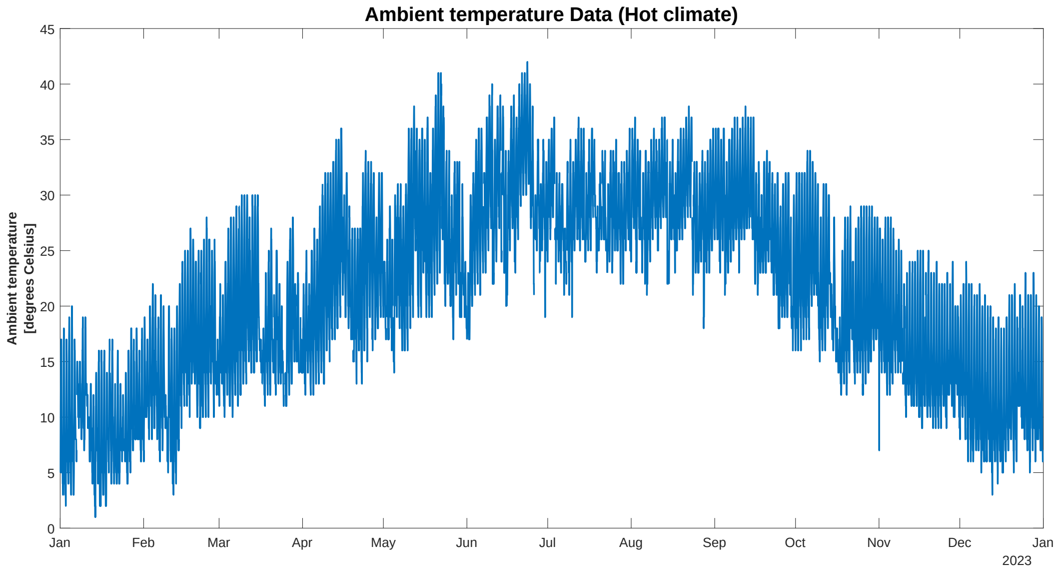


Figure 7.2: Ambient temperature Data (Hot climate)

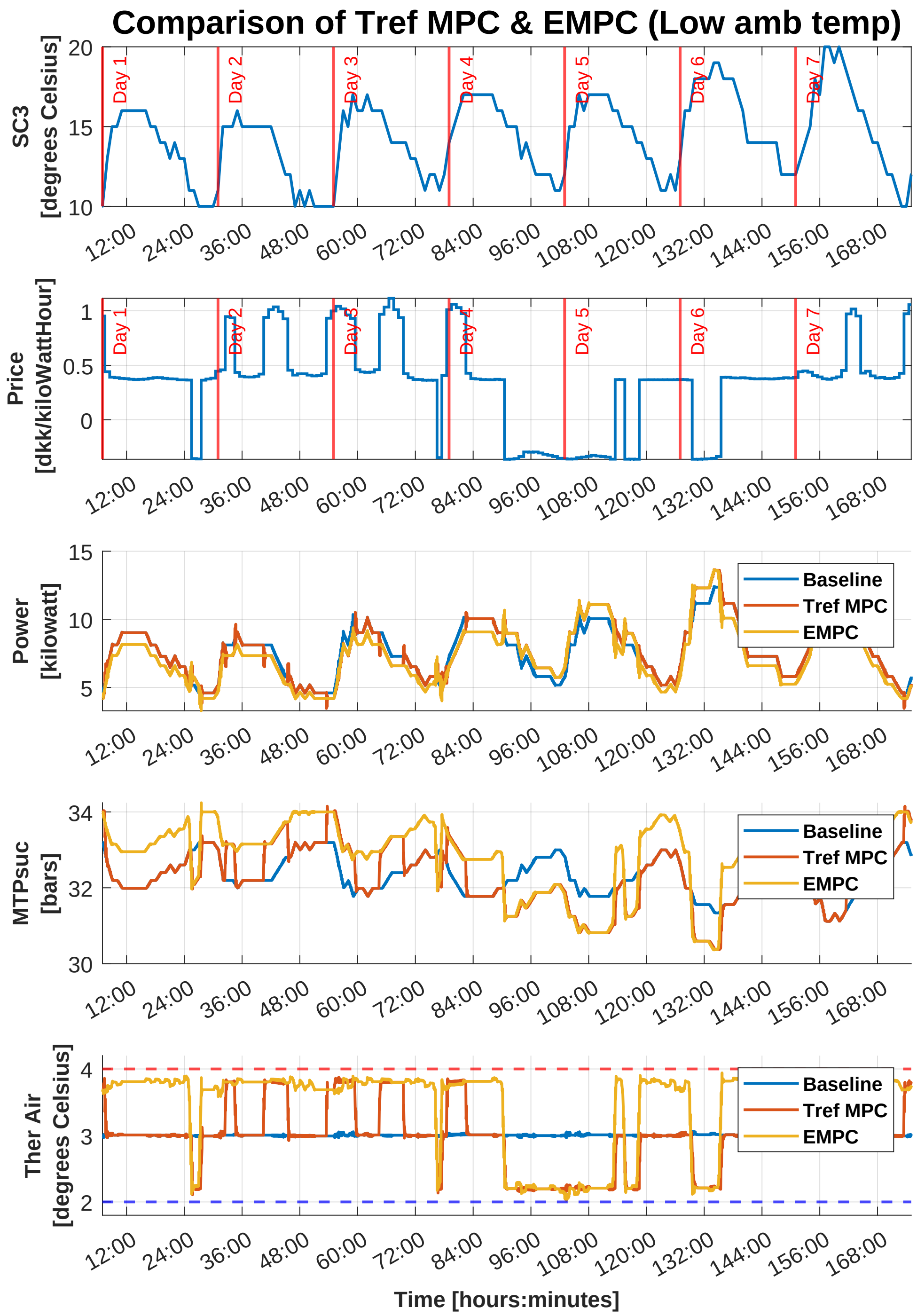


Figure 7.3: Comparison of Tref MPC and EMPC for 10 to 20 C amb temp

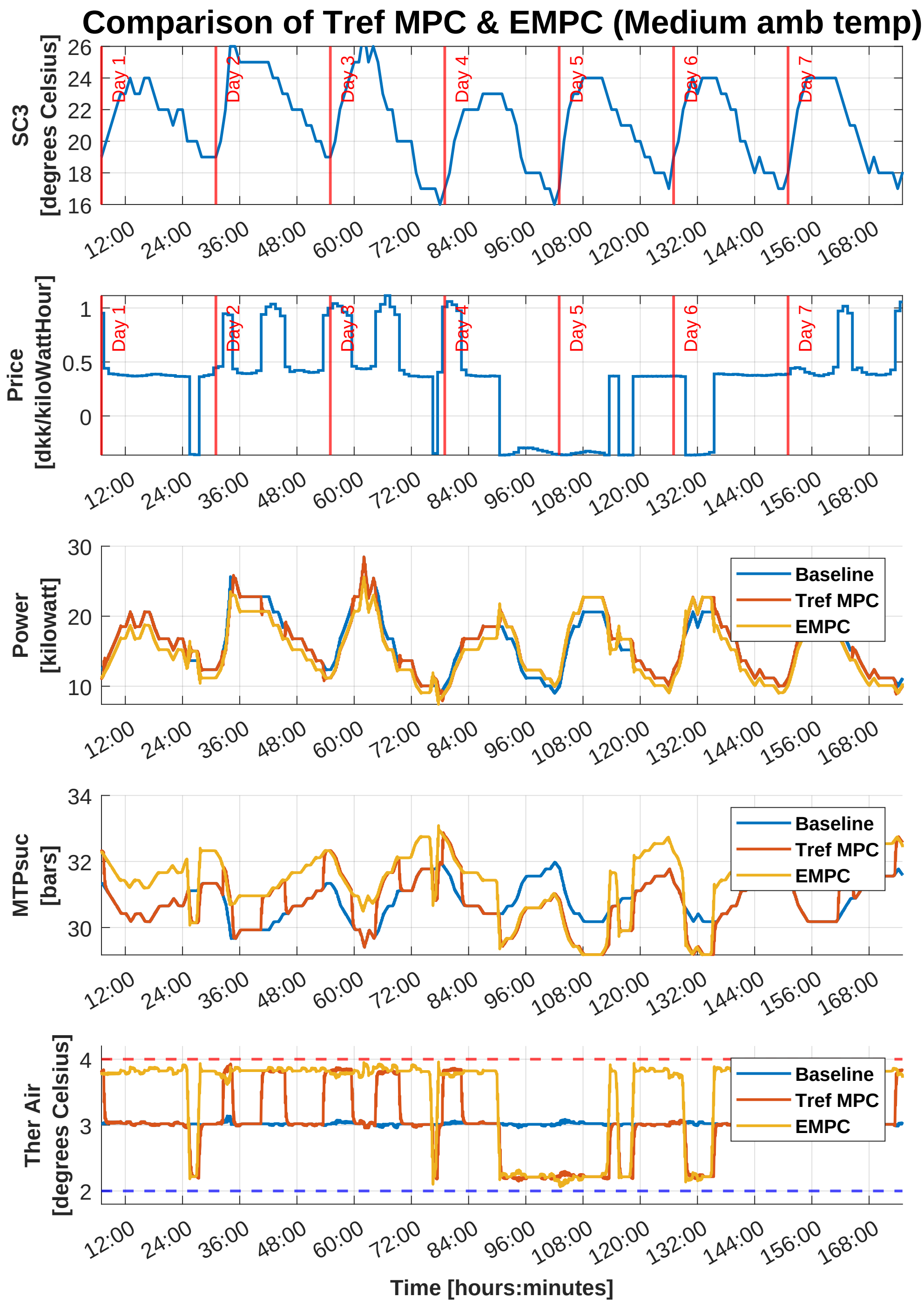


Figure 7.4: Comparison of Tref MPC and EMPC for 16 to 26 C amb temp

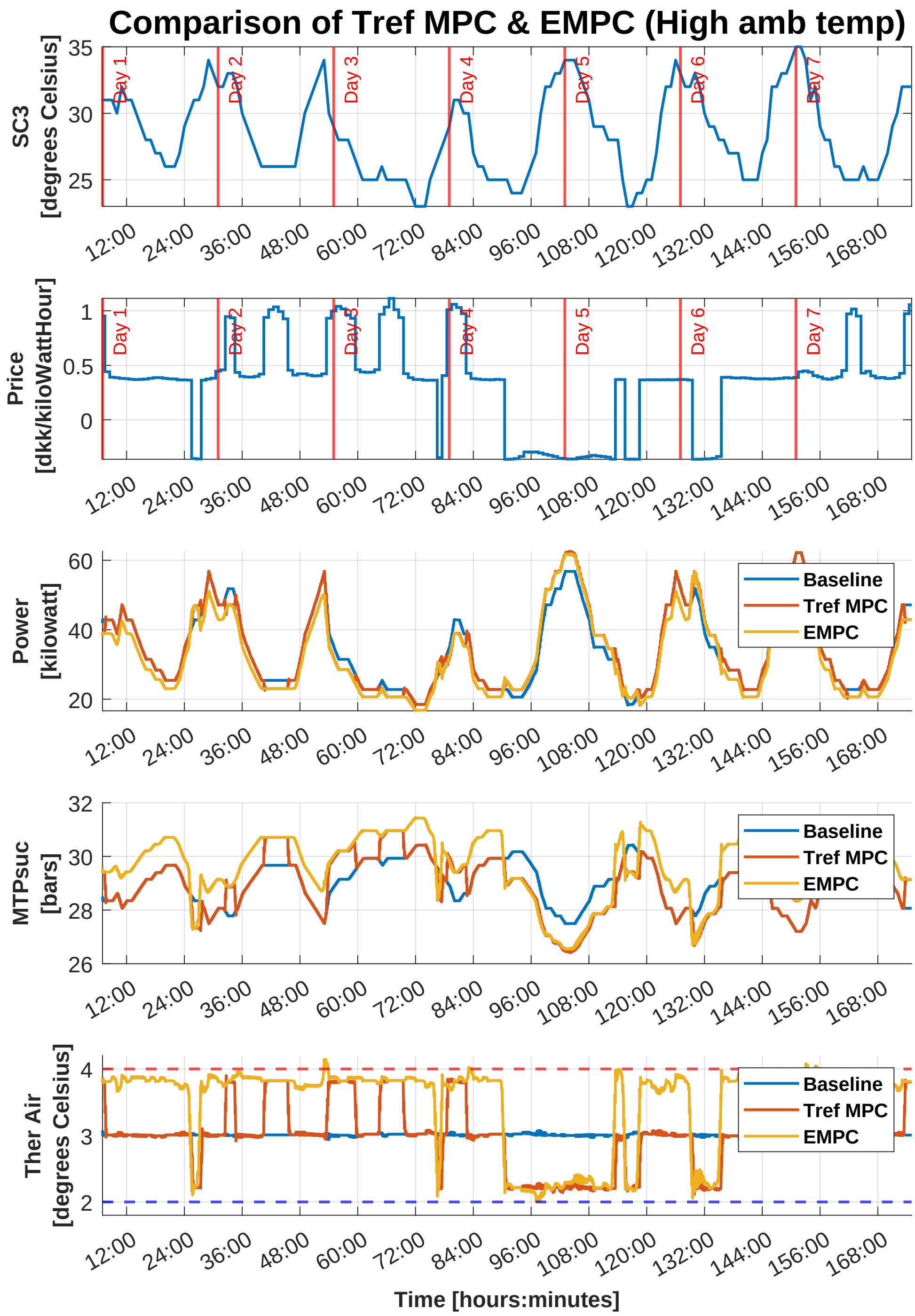


Figure 7.5: Comparison of Tref MPC and EMPC for 23 to 35 C amb temp

7.0.3 Power consumption behaviour in response to electricity price data

Tref and EMPC were designed to reduce the overall cost of operation for the system. A closer view is provided in the figure 7.6 where power data is plotted for different modes. The first mode is energy saving mode in which power is expensive and goal of MPC is to reduce the power consumption in order to save cost. The second mode is energy consuming mode, in which there is excess energy in the grid and goal of MPC is to consume the electricity. The third mode is the normal operation mode.

Subplot 1 in the figure 7.6 depicts that when the energy is expensive, both Tref and EMPC save energy by consumping less power. This is achieved by raising the suction pressure (shown in figures 7.3, 7.4 and 7.5) keeping in view the limits of cabinet temperature. Similarly in subplot 2, when the electricity price is lower, Tref and EMPC consume more energy by lowering the suction pressure. In normal mode behaviour of Tref is the same as baseline however, EMPC try to save energy even in this mode as well. Hence, when the price is normal, EMPC will still raise the suction pressure whenever it finds the opportunity keeping the cabinet temperature within limits. This can be seen in the figures 7.3, 7.4 and 7.5.

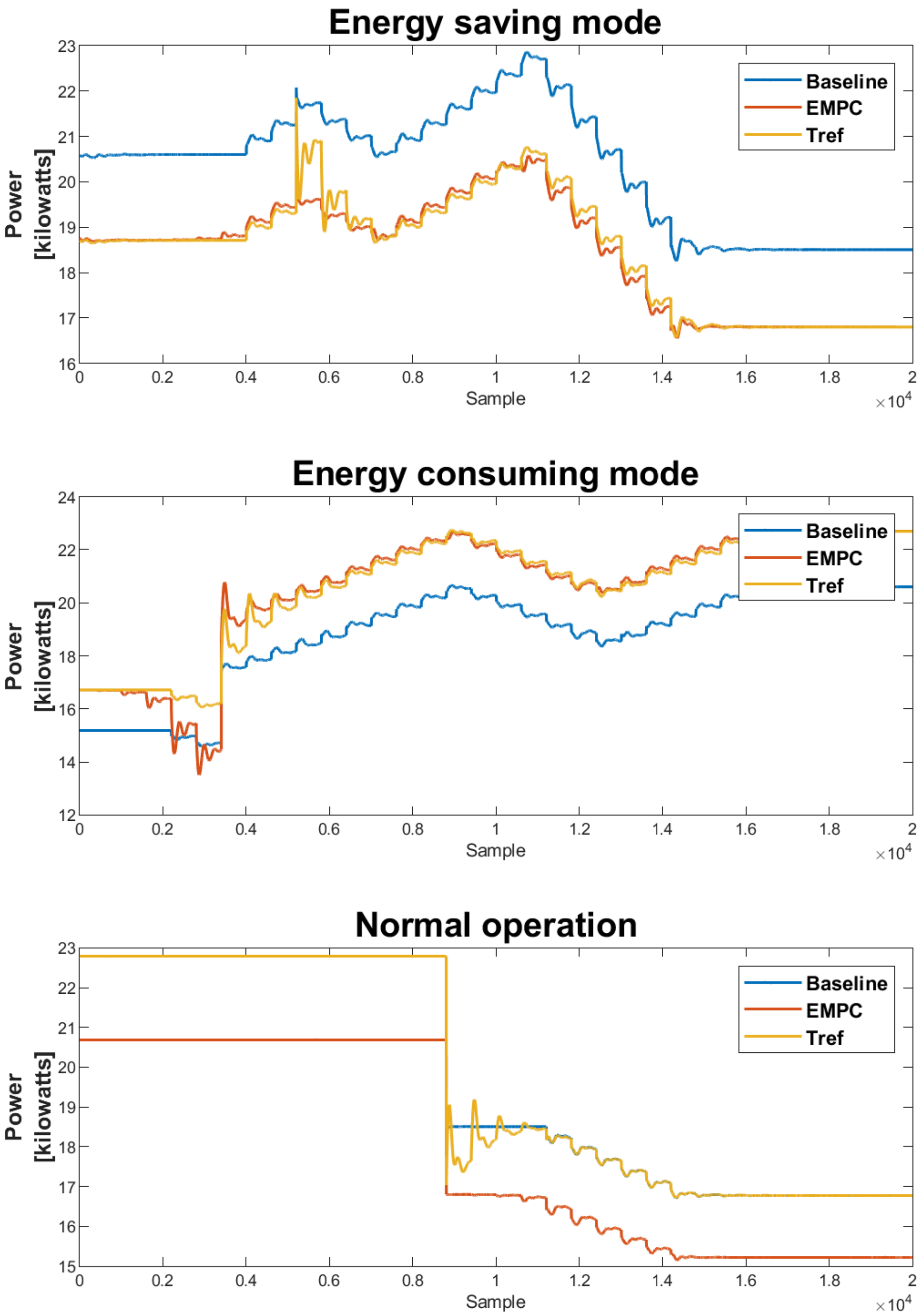


Figure 7.6: Comparison of Baseline, Tref MPC and EMPC for different electricity prices

7.1 Cost Savings

Table 7.7 provides a comprehensive summary of energy consumption, total cost of operation and savings achieved by implementing Tref and EMPC in comparison with baseline for one week of system operation in differnet ambient temperature conditions. Bar graphs in the figures 7.8, 7.9 and 7.10 are also providing an illustrative view of power consumption, cost of operation and percentage savings.

Ambient Temperature (SC3)	Total Energy consumption (KWh)			Total Cost (DKK)			Savings (DKK/Percentage)			
							Tref MPC		EMPC	
	Baseline	Tref MPC	EMPC	Baseline	Tref MPC	EMPC	(DKK)	(%)	(DKK)	(%)
10 to 20 C	1475.3	1478.5	1394.4	425.4	396.3	365.7	29.1	6.8	59.8	14.1
16 to 26 C	3085.6	3087.7	2911.4	920.9	862.6	799.6	58.3	6.3	121.3	13.2
23 to 35 C	6483.8	6527.6	6164.2	1832.9	1717.3	1585.4	115.6	6.3	247.5	13.5

Figure 7.7: Comparison of Baseline, Tref and EMPC

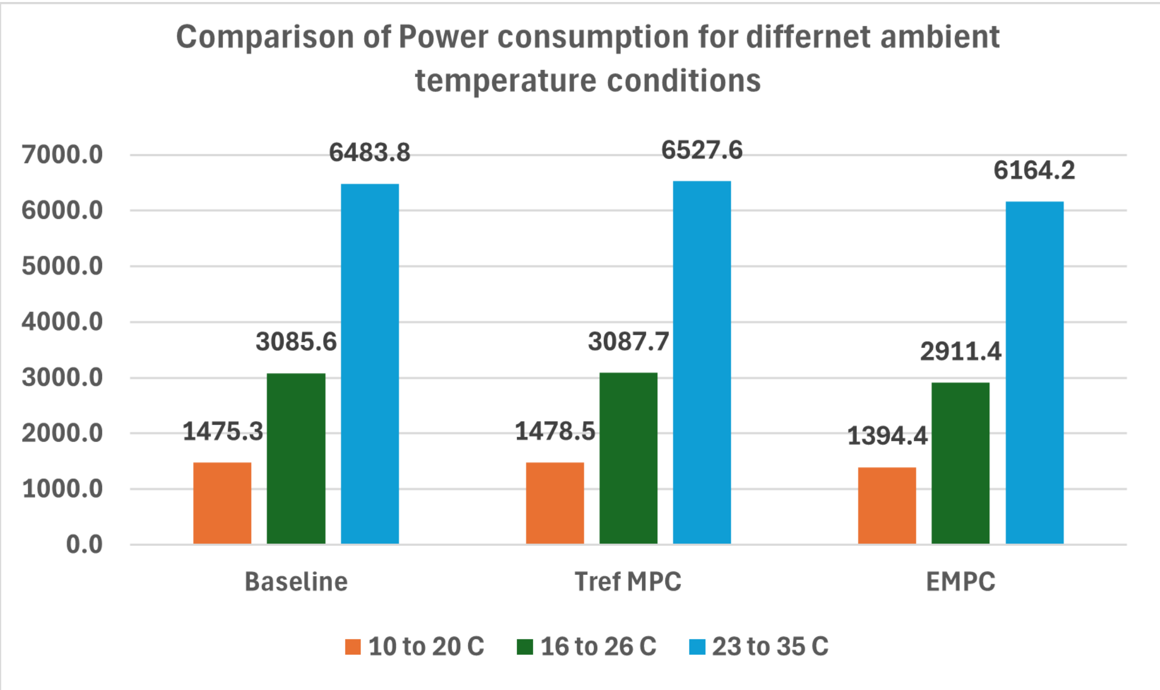


Figure 7.8: Comparison of Power consumption for differnet ambient temperature conditions

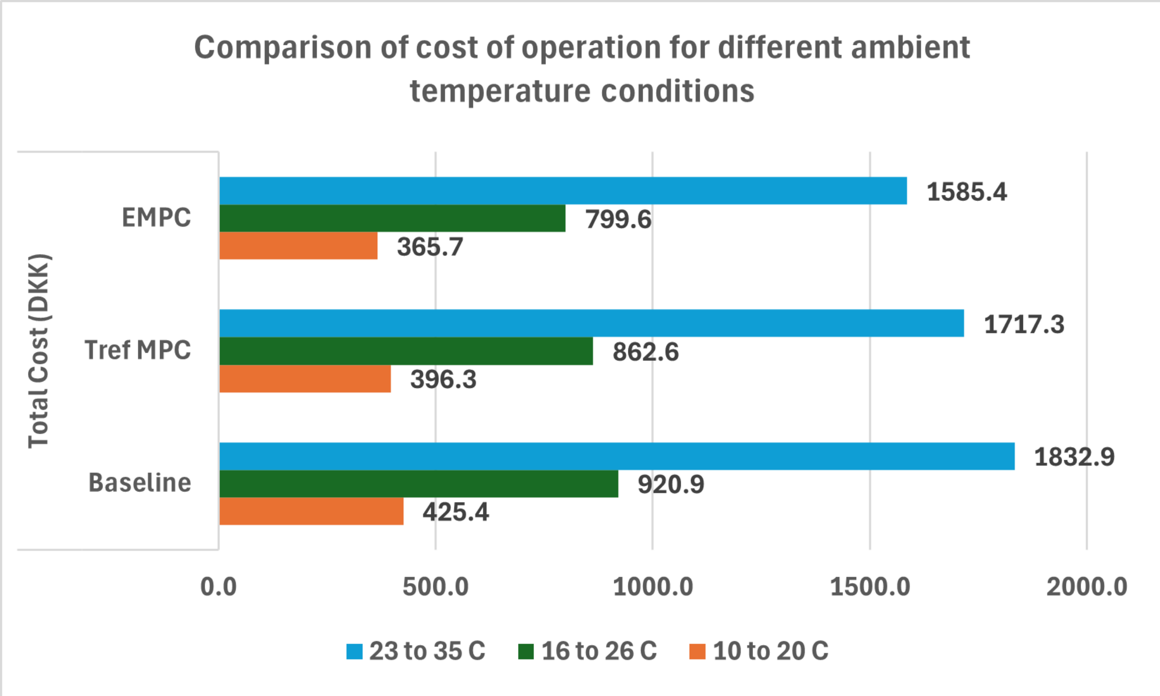


Figure 7.9: Comparison of cost of operation for different ambient temperature conditions

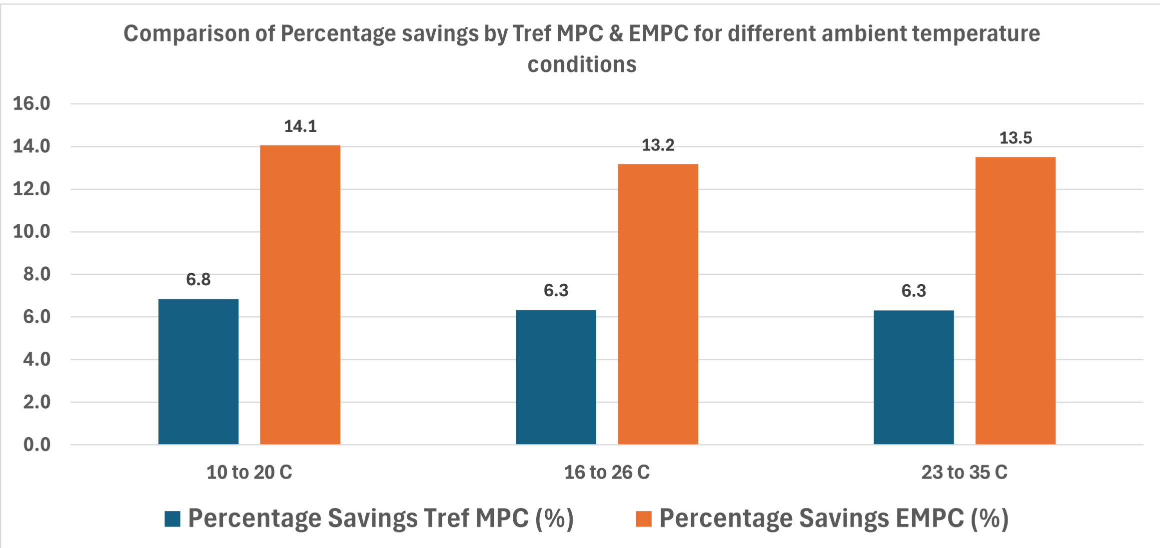


Figure 7.10: Comparison of Percentage savings by Tref MPC and EMPC for different ambient temperature conditions

7.2 Performance of Tref and EMPC under ambient temperature bias

Issue of bias in measuring ambient temperature and how it could effect the performance of the system is discussed in chapter 6. Also the methodology adopted to compensate the bias issue using Tref and EMPC is discussed. This section discuss the impact of bias on the power consumption and cost savings.

Figures 7.11, 7.12 and 7.13 show the ambient temperature with/without bias and after compensation in subplot 1, while power consumption and cabinet temperatures are shown in subplots 2 and 3 respectively for both Tref and EMPC with bias compensation implementation. It can be observed from the subplots 2 and 3 that power consumption remains around baseline system (without bias fault) and cabinet temperature is also maintained within limits.

Table 7.14 show the details of total energy consumption, total cost and savings in the presence of bias fault. It can be observed that significant savings are still achieved even under the presence of temperature sensor fault.

Figure 7.15 and 7.16 compare the savings percentage of Tref and EMPC with and without bias for different ambient temperature conditions. It can be seen that savings are reduced in the presence of bias fault but the system is still able to save appreciable amount of energy in the presence of fault.

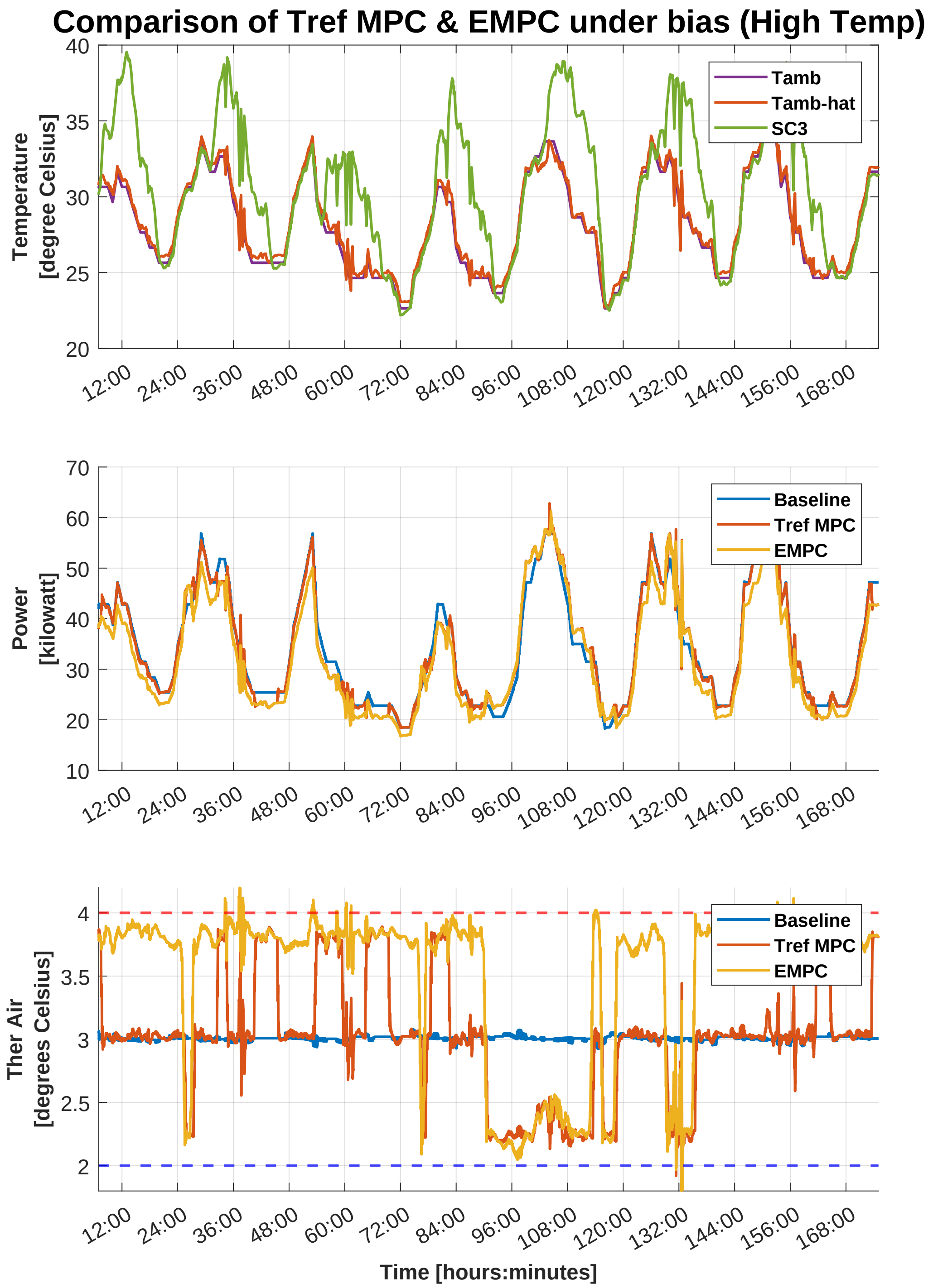


Figure 7.11: Comparison of Tref MPC and EMPC under bias (High Temp)

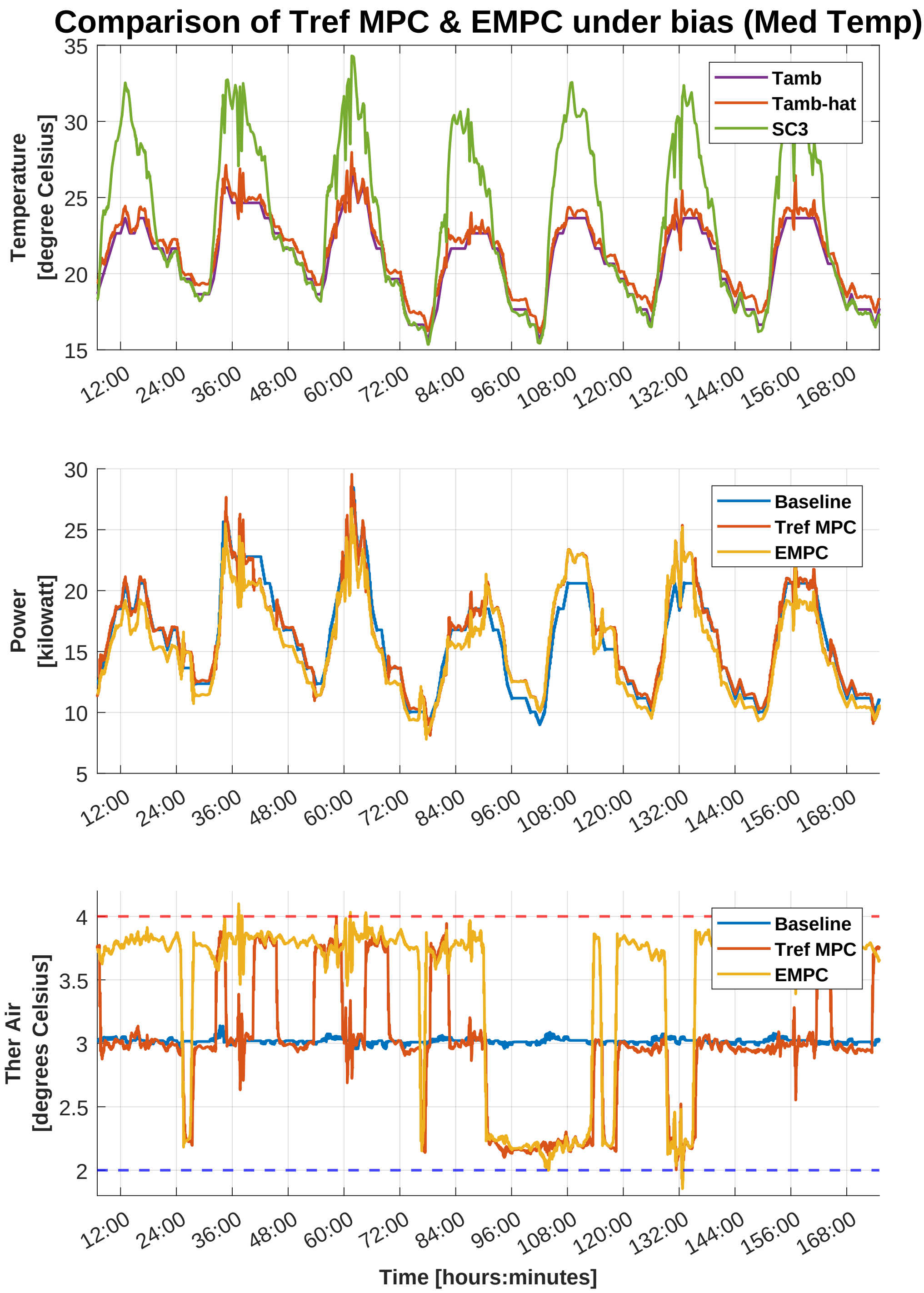


Figure 7.12: Comparison of Tref MPC and EMPC under bias (Med Temp)

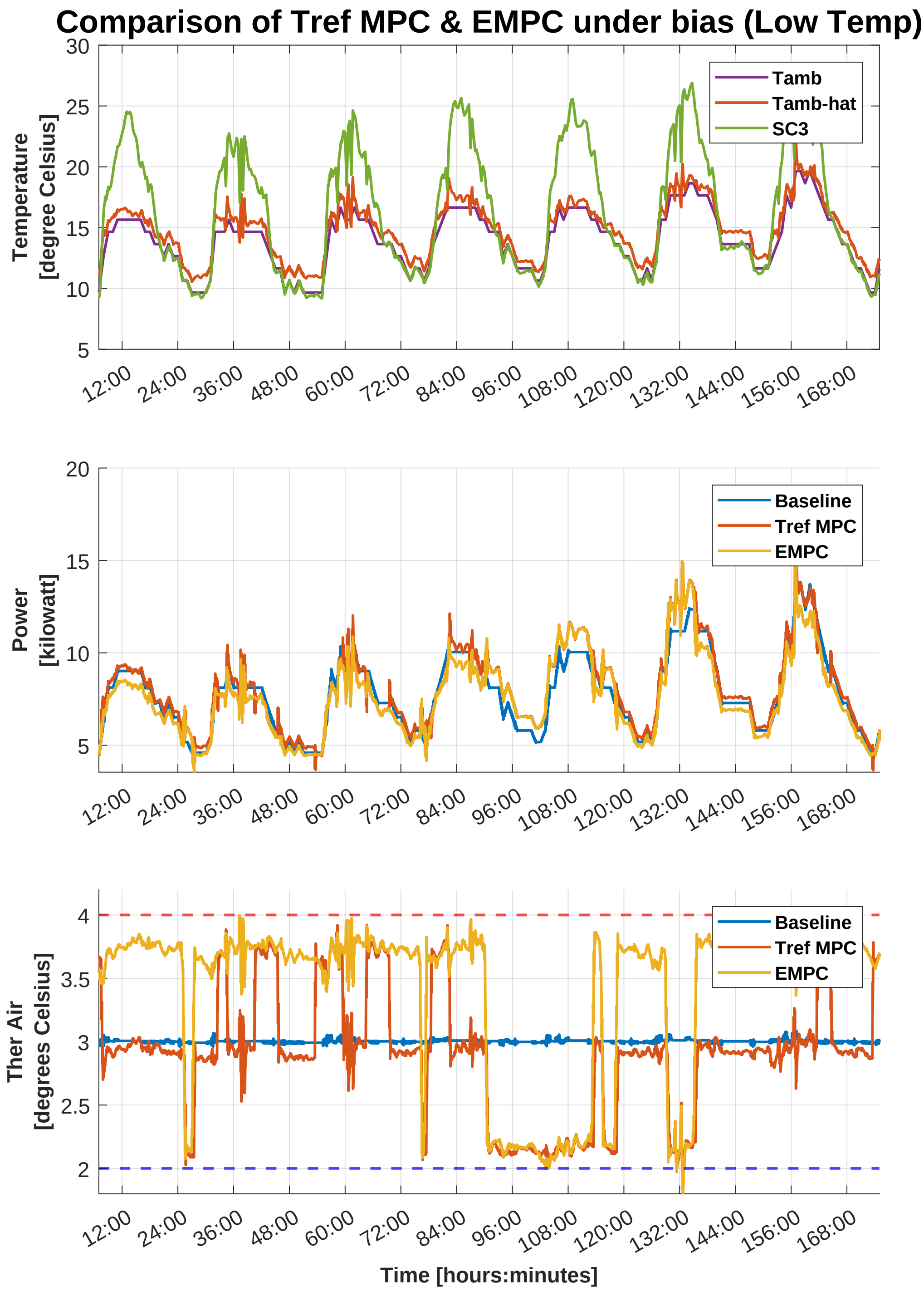


Figure 7.13: Comparison of Tref MPC and EMPC under bias (Low Temp)

Ambient Temperature	Total Energy consumption (KWh)			Total Cost (DKK)			Savings (DKK/Percentage)			
							Tref MPC		EMPC	
	Baseline	Tref MPC	EMPC	Baseline	Tref MPC	EMPC	(DKK)	(%)	(DKK)	(%)
10 to 20 C	1475.3	1527.8	1447.4	425.4	412.6	383.2	12.9	3.0	42.3	9.9
16 to 26 C	3085.6	3126.5	2962.4	920.9	875.2	814.9	45.7	5.0	105.9	11.5
23 to 35 C	6483.8	6509.1	6178.7	1832.9	1719.5	1596.1	113.4	6.2	236.8	12.9

Figure 7.14: Comparison of Baseline, Tref and EMPC in presence of Bias

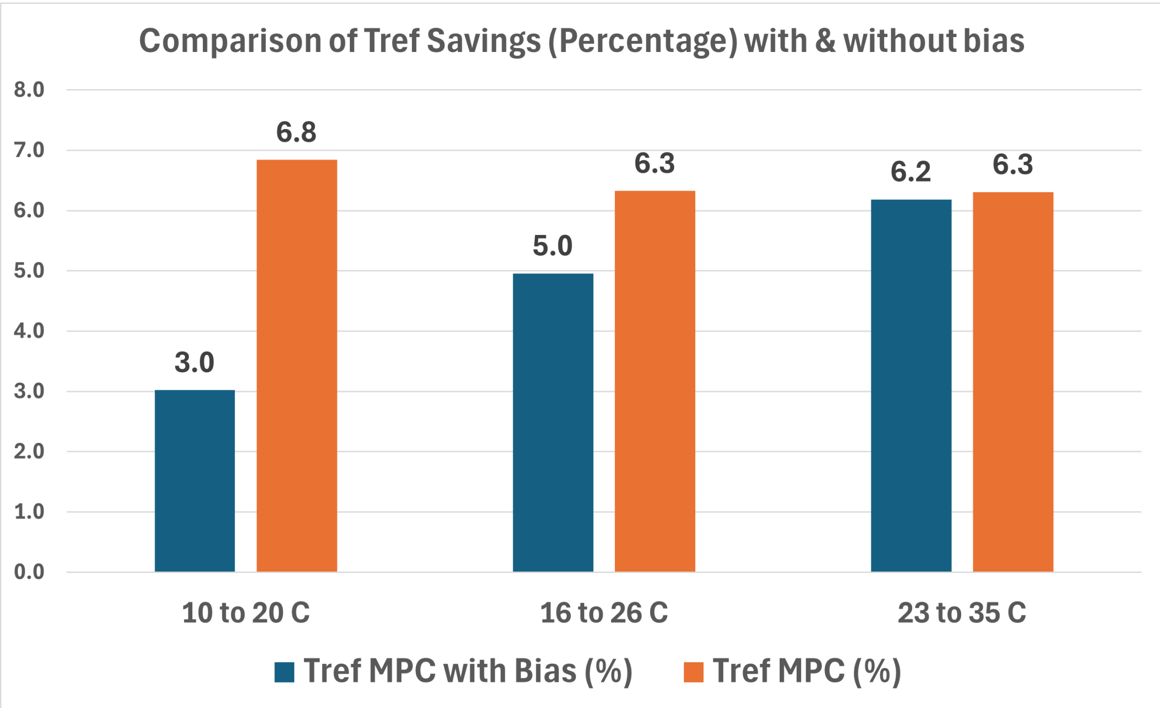


Figure 7.15: Comparison of Tref Savings (Percentage) with and without bias

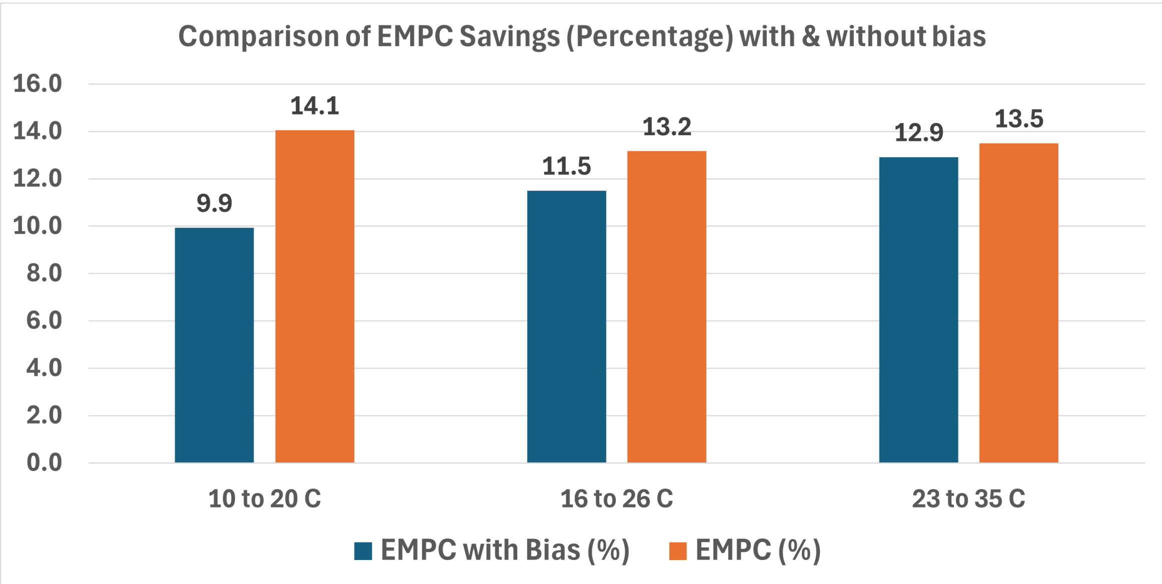


Figure 7.16: Comparison of EMPC Savings (Percentage) with and without bias

Chapter 8

Discussion

With the increase of renewable share in electricity grid, large fluctuations in the energy prices are expected more common now than ever before. Supermarket refrigeration systems are energy intensive systems, but energy demand can be predicted with some degree of uncertainty to few hours ahead. Capitalizing on this opportunity, this project focused on data driven modelling of supermarket refrigeration system that could be used with model predictive control (MPC) in order to save energy consumption.

The project was implemented successfully at the end, however, issues faced during the implementation and bottlenecks that changed the implementation strategy are presented and discussed in the previous chapters. Finally, the successful implementation demonstrated considerable amount of savings that was presented in the result section.

8.1 Discussion of results

Chapter 7 presents the comparisons of results obtained by implementing two different types of MPC configurations. The first one was known as temperature reference MPC (5.1.5) and second one as economic MPC (5.1.6). Both the configurations have advantage on one another in some different manner. The former being simpler to implement while on the other hand resulted in less savings (7.1), the later being computationally expensive but yielded more energy savings. Operational cost of supermarket refrigeration systems is highly depended on the ambient temperature conditions (2.3), hence, system performance was evaluated and compared for three weeks of operation for ambient temperature data representing the broad range of operation i.e. 10 to 35 degree Celsius (3.2.2). The goal was to maintain the cabinet temperature within food safety limits i.e. from 2 to 4 degree Celsius while achieving the savings on the operational cost of the system. This is achieved by maintaining the right suction pressure at the inlet of compressor based on ambient temperature conditions. Graphs (7.0.1) present the comparison of Tref, EMPC and baseline in achieving this cabinet temperature constraint, manipulation of suction pressure and behaviour of power consumption for different ambient temperature conditions and electricity price data. Approach to minimize operational cost is somewhat different for both Tref and EMPC controller (7.0.2).

Significant amount of savings are achieved for both implementation cases (7.1), however, EMPC leads the savings for all tested ranges of ambient temperature. Furthermore, low ambient temperature conditions offer more opportunity to save cost as compared to high ambient temperature conditions (7.1).

System is also equipped with handling temperature sensor bias fault (6.1) and able to produce considerable amount of savings even for faulty temperature sensor, however, there is reduction of savings as compared to healthy sensor scenario (7.2).

8.1.1 Cabinet temperature time constant

Fast model provided by Danfoss had some limitations (2.6), one of which is directly impacted the claim of savings made in the project. The figures would have been significantly larger, since savings achieved using the pre-cooling was not retained by fast model due to short time constant. In reality the time constant is much larger for refrigeration systems. To mitigate the effect of time constant to some extent, the concept of negative price was utilized with care.

8.2 Utilization of IceTank for further improving the cost savings

During the initial formulation of project problem, it was argued to integrate IceTank with MPC to achieve further savings. Unfortunately, due to some constraints, the company was unable to provide IceTank model till the completion of this project and hence it was mutually decided to leave this part for some future project.

8.3 MPC methodology

Two different known configurations of MPC were implemented and tested for cost optimization of supermarket refrigeration systems (5.2). Cost and constraint functions were carefully designed to achieve optimal performance (5.1). Several issues were faced during the implementation (5.3), especially using the MATLAB toolbox for MPC implementation and one of the modelling methods was not tested with MPC due to this limitation. Tuning of both MPC implementations was performed by using the slack variables for constraint softening, implementation of rate of manipulated variable, tuning of prediction horizon and setting up of some hard on the upper and lower bound of manipulated variable (5.4).

8.4 Modelling methodology

The process started with exploring some agile modelling technique like Dynamic mode decomposition (DMD) to obtain simpler model that could speed up the computation requirements of MPC (4.1.1), however, less appreciable results in the start lead to exploring complex modelling methodologies like neural state space (4.1.2) and Long short term memory neural network (4.1.3) which again resulted in unsatisfactory results. Eventually, the problem was identified in the data being used for all of the techniques (4.2), however, much time was already spent on neural networks until this point and it was decided to continue with these two modelling methodologies. Finally NSS was selected because of its compatibility with MPC in MATLAB implementation.

8.4.1 Use of simpler model

If it was not the case with limited time constraint, further exploration of DMD would have been done. The advantages could include obtaining a simpler model to use with MPC which would have been more practical to implement in industrial controllers.

8.5 Perspective and future work

This project has just laid a foundation stone to demonstrate the use of data driven modelling and model predictive control in achieving significant improvements in the energy savings for the case of supermarket refrigeration systems. The future prospects are very bright in this context and a lot can be achieved in this domain. With increasing renewable shares in the electrical grid and available forecasts on the price, supermarket refrigeration systems which offer energy store in terms of cooling can be utilized to optimize energy performance. Some suggestions are presented below.

- Integration of solar panels can be studied with refrigeration systems using MPC. Weather forecast data is available on hourly basis and operational cost can be minimized by selecting the power source appropriately.
- IceTank used in supermarket refrigeration systems provide another way of storing energy. Integration of IceTank with the system and cost savings can be evaluated.
- Excess heat from the condenser can be integrated with district heating and managed properly using MPC.
- Time constant of cabinets can be studied and added in the fast model to obtain more savings.
- Simpler modelling techniques such as DMD can be further explored to optimize the performance of MPC.

Chapter 9

Conclusion

This thesis attempted to improve cost-efficiency of the supermarket refrigeration systems by using their energy flexibility and variation in energy prices. 2 MPC schemes were proposed to optimise system's operation in terms of energy cost: an evaporation temperature reference-tracking nonlinear MPC and nonlinear EMPC.

In order to implement these control strategies, a prediction model of the system's dynamics was first needed. A new data-driven approach was taken, with focus on few central signals related to the compressors that result in smaller amount of signals in the prediction model than in the case where information from all evaporators and cooled cabinets is used for that goal, which in turn decreases computational requirement on the hardware implementing the aforementioned MPC schemes, while maintaining high accuracy of predictions.

To build a data-driven model based on these signals, different methods were explored. This included DMD and 2 ANN architectures i.e. LSTM and NSS. Out of these 3 methods, only NSS was successfully implemented for use in MPC. As the NSS might be too computationally heavy for the industrial control units used in the supermarkets, it was considered mainly for demonstrating validity of the applied modelling approach.

The proposed MPC strategies were evaluated both in terms of cost savings and respecting of the operating constraints on the cabinet temperature that are directly linked to quality of the stored food. Despite operating closely to the constraints, which minimised the cost of energy, they were not violated. The cost savings amounted to 6.8% and 14.1% for the reference-tracking MPC and EMPC respectively.

In addition to this, performance of the controllers under common ambient air temperature sensor bias fault was studied. Owing to unacceptable degradation of performance of both controllers by the fault, a method available in the literature for compensating for bias was applied. As a result of this action, most of the savings under fault scenario could be recovered, especially while the ambient air temperature was high.

We claim that the saving presented in this thesis would have been higher if it had not been for extremely small time constant of cabinet air temperature that was implemented in the simulation model of supermarket refrigeration system used to generate training data for the models and run simulation studies of the proposed control solutions. Once this would be improved, the information of time constant of stored food's temperature could be incorporated into the design of MPC schemes.

Moreover, there are several issues to be resolved if the developed solutions are to be used in the real supermarket refrigeration system. This includes handling of uncertainty in the forecasts in weather and electricity price that the controller's performance heavily depends on, computational simplification of the prediction model, possibly by transitioning from neural models to DMD and some way of ensuring that the controllers tuning is capable of operating in different supermarkets, also the ones having other components for economic optimisation like ice tank.

By and large, we can state that the outcomes of this thesis uncovered potential of the proposed methodology for improving economic efficiency of supermarket refrigeration systems and gave good directions towards necessary improvements to make it applicable in practice.

Bibliography

[1] Mario Pérez-Gomariz, Antonio López-Gómez, and Fernando Cerdán-Cartagena. "Artificial Neural Networks as Artificial Intelligence Technique for Energy Saving in Refrigeration Systems—A Review". eng. In: *Clean technologies* 5.1 (2023), pp. 116–136. ISSN: 2571-8797.

[2] "Demand side management analysis of a supermarket integrated HVAC, refrigeration and water loop heat pump system". eng. In: *Applied thermal engineering* 152 (2019), pp. 543–550. ISSN: 1359-4311.

[3] Ramon Granell et al. "A data-driven approach for electricity load profile prediction of new supermarkets". eng. In: *Energy Procedia* 161 (2019), pp. 242–250. ISSN: 1876-6102.

[4] Tobias Gybel Hovgaard et al. "Model predictive control technologies for efficient and flexible power consumption in refrigeration systems". eng. In: *Energy (Oxford)* 44.1 (2012), pp. 105–116. ISSN: 0360-5442.

[5] T G Hovgaard, K Edlund, and J B Jorgensen. "The potential of Economic MPC for power management". eng. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 7533–7538. ISBN: 142447745X.

[6] Seyed Ehsan Shafiei. *Control Methods for Energy Management of Refrigeration Systems*. eng. 2015.

[7] Peter D. Lund et al. "Review of energy system flexibility measures to enable high levels of variable renewable electricity". eng. In: *Renewable & sustainable energy reviews* 45 (2015), pp. 785–807. ISSN: 1364-0321.

[8] Rahmat Aazami, Kaveh Aflaki, and Mahmoud Reza Haghifam. "A demand response based solution for LMP management in power markets". eng. In: *International journal of electrical power & energy systems* 33.5 (2011), pp. 1125–1132. ISSN: 0142-0615.

[9] Abdelali Agouzoul, Emmanuel Simeu, and Mohamed Tabaa. "Synthesis of model predictive control based on neural network for energy consumption enhancement in building". eng. In: *International journal of electronics and communications* 173 (2024), pp. 155021–. ISSN: 1434-8411.

[10] Cuiling Wang et al. "Cooling seasonal performance of inverter air conditioner using model prediction control for demand response". eng. In: *Energy and buildings* 256 (2022), pp. 111708–. ISSN: 0378-7788.

[11] Fatma Mtibaa. *Real-time data driven model predictive control for efficient energy consumption in smart buildings*. eng. 2023.

[12] Daniel Sarabia et al. "Hybrid NMPC of supermarket display cases". eng. In: *Control engineering practice* 17.4 (2009), pp. 428–441. ISSN: 0967-0661.

[13] D. Sarabia et al. "Hybrid Control of a Supermarket Refrigeration Systems". eng. In: *2007 American Control Conference*. IEEE, 2007, pp. 4178–4185. ISBN: 9781424409884.

[14] C. Sonntag et al. "Hybrid Nonlinear Model-Predictive Control of a Supermarket Refrigeration System". eng. In: *2007 IEEE International Conference on Control Applications*. IEEE, 2007, pp. 1432–1437. ISBN: 9781424404421.

[15] T. G. Hovgaard, L. F. S. Larsen, and J. B. Jorgensen. "Robust economic MPC for a power management scenario with uncertainties". eng. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 1515–1520. ISBN: 9781612848006.

[16] J. M. Belman et al. "Steady-state model of a variable speed vapor compression system using R134a as working fluid". eng. In: *International journal of energy research* 34.11 (2010), pp. 933–945. ISSN: 0363-907X.

[17] J.M. Belman-Flores et al. "Using ANNs to approach to the energy performance for a small refrigeration system working with R134a and two alternative lower GWP mixtures". eng. In: *Applied thermal engineering* 127 (2017), pp. 996–1004. ISSN: 1359-4311.

[18] H.M. Ertunc and M. Hosoz. "Artificial neural network analysis of a refrigeration system with an evaporative condenser". eng. In: *Applied thermal engineering* 26.5 (2006), pp. 627–635. ISSN: 1359-4311.

[19] Fabrizio Sossan et al. "Grey-box modelling of a household refrigeration unit using time series data in application to demand side management". eng. In: *Sustainable Energy, Grids and Networks* 5 (2016), pp. 1–12. ISSN: 2352-4677.

[20] Waleed Aslam Michal Kujawski. "Modelling of supermarket refrigeration system with dynamic mode decomposition for energy optimization". eng. In: *AAU Library* 1.1 (2023), p. 67. ISSN: 123.

[21] Michal Kujawski and Waleed Aslam. *Detection of refrigernat leakage fault in supermarket refrigeration systems*. eng. 2023.

[22] Patrick Haffmans, Roozbeh Izadi-Zamanabadi, and Hossein Ramezani. "A fault-tolerant control strategy to estimate and compensate the temperature sensor bias in supermarket refrigeration systems". eng. In: *ISA transactions* 144 (2024), pp. 490–500. ISSN: 0019-0578.

[23] PETER J. SCHMID. "Dynamic mode decomposition of numerical and experimental data". eng. In: *Journal of fluid mechanics* 656.August (2010), pp. 5–28. ISSN: 0022-1120.

[24] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition". eng. In: *Journal of nonlinear science* 25.6 (2015), pp. 1307–1346. ISSN: 0938-8974.

[25] Kamilya Smagulova and Alex Pappachen James. "A survey on LSTM memristive neural network architectures and applications". In: *The European Physical Journal Special Topics* 228.10 (2019), pp. 2313–2324.

[26] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. "A review on the long short-term memory model". In: *Artificial Intelligence Review* 53.8 (2020), pp. 5929–5955.

[27] Jan M. Maciejowski. *Predictive control : with constraints.* eng. Harlow: Prentice Hall, 2002. ISBN: 9780201398236.

[28] Rasmus Løvschall Kristiansen. *Model predictive control of Type 4 wind turbines*. eng. 2023.

[29] Seyed Ehsan Shafiei and Andrew Alleyne. "Model predictive control of hybrid thermal energy systems in transport refrigeration". In: *Applied Thermal Engineering* 82 (2015), pp. 264–280.

[30] *constraints*. `https://se.mathworks.com/help/mpc/ug/specifying-constraints.html`. Accessed: 2024-05-02.

[31] *multistageMPC*. `https://se.mathworks.com/help/mpc/ug/nonlinear-mpc.html`. Accessed: 2024-05-01.

[32] *EconomicMPC*. `https://se.mathworks.com/help/mpc/ug/economic-mpc.htm`. Accessed: 2024-05-01.

[33] Willy Wojsznis et al. "Practical approach to tuning MPC". eng. In: *ISA transactions* 42.1 (2003), pp. 149–162. ISSN: 0019-0578.

[34] *AmbientData*. `https://mesonet.agron.iastate.edu/request/download.phtml`. Accessed: 2024-05-03.