DYNAMIC MODELLING AND HYBRID NON-LINEAR MODEL PREDICTIVE CONTROL OF INDUCED DRAFT COOLING TOWERS WITH PARALLEL HEAT EXCHANGERS, PUMPS AND COOLING WATER NETWORK

by

Johannes Henning Viljoen

Submitted in partial fulfillment of the requirements for the degree
Philosophiae Doctor (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

Oct 2019

SUMMARY

DYNAMIC MODELLING AND HYBRID NON-LINEAR MODEL PREDICTIVE CONTROL OF INDUCED DRAFT COOLING TOWERS WITH PARALLEL HEAT EXCHANGERS, PUMPS AND COOLING WATER NETWORK

by

Johannes Henning Viljoen

Promotor:	Prof. I. K. Craig
Co-promotor:	Dr. C. J. Muller
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Philosophiae Doctor (Electronic Engineering)
Keywords:	Dynamic modelling, cooling tower, cooling water network, optimisation, particle swarm optimisation, hybrid systems, non-linear model predictive control, gradient descent, electricity consumption minimisation

In the process industries, cooling capacity is an important enabler for the facility to manufacture on specification product. The cooling water network is an important part of the over-all cooling system of the facility. In this research a cooling water circuit consisting of 3 cooling towers in parallel, 2 cooling water pumps in parallel, and 11 heat exchangers in parallel, is modelled. The model developed is based on first principles and captures the dynamic, non-linear, interactive nature of the plant. The modelled plant is further complicated by continuous, as well as discrete process variables, giving the model a hybrid nature. Energy consumption is included in the model as it is a very important parameter for plant operation. The model is fitted to real industry data by using a particle swarm optimisation approach. The model is suitable to be used for optimisation and control purposes.

Cooling water networks are often not instrumented and actuated, nor controlled or optimised. Significant process benefits can be achieved by better process end-user temperature control, and direct monetary benefits can be obtained from electric power minimisation. A Hybrid Non-Linear Model

Predictive Control strategy is developed for these control objectives, and simulated on the developed first principles dynamic model. Continuous and hybrid control cases are developed, and tested on process scenarios that reflect conditions seen in a real plant.

Various alternative techniques are evaluated in order to solve the Hybrid Non-Linear Control problem. Gradient descent with momentum is chosen and configured to be used to solve the continuous control problem. For the discrete control problem a graph traversal algorithm is developed and joined to the continuous control algorithm to form a Hybrid Non-Linear Model Predictive controller. The potential monetary benefits that can be obtained by the plant owner through implementing the designed control strategy, are estimated. A powerful computation platform is designed for the plant model and controller simulations.

LIST OF ABBREVIATIONS

APC	Advanced Process Control
CT	Cooling Tower
CTB	Cooling Tower Basin
CV	Controlled Variable
CW	Cooling Water
CWR	Cooling Water Return
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning
DPS	Direct Policy Search
DV	Disturbance Variable
EU	Engineering Units
GA	Genetic Algorithms
GD	Gradient Descent
HLMPC	Hybrid Linear Model Predictive Control
HNMPC	Hybrid Non-Linear Model Predictive Control
ISE	Integral Square Error
KKT	Karush-Kuhn-Tucker conditions
LMPC	Linear Model Predictive Control
LMTD	Logarithmic Mean Temperature Difference
ML	Machine Learning
MV	Manipulated Variable
MINLP	Mixed Integer Non-linear Programming
MPC	Model Predictive Control
NLP	Non-Linear Programming
NMPC	Non-Linear Model Predictive Control
NP-hardness	Non-deterministic Polynomial-time hardness
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PID	Proportional-Integral-Derivative

PSO	Particle Swarm Optimisation
RL	Reinforcement Learning
SGD	Stochastic Gradient Descent
VSD	Variable Speed Drive

ACKNOWLEDGEMENTS

I wish to acknowledge ORYX GTL Ltd. for granting permission to use plant data for fitting the model developed in this research.

I wish to thank Dr. Nelis Muller for his helpful input and advice.

I am honoured and thankful to have been able to work under the guidance of Prof. Ian Craig for many years. His support and input is much appreciated.

I thank my wife Naomi, and kids Hanna, Miriam, Talitha and Daniel for their patient support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

The benefits of applying Advanced Process Control (APC) to process industry plant utilities are becoming more apparent as such plants are coming under increasing pressure to lower their carbon footprints and optimise costs. However, in the advanced modelling and control literature utility plants and cooling water plants in particular have historically not received significant focus [1]. Before applying an advanced control technique, a dynamic model of the plant needs to be available. Therefore the first objective of this work is to contribute to the literature by developing a validated dynamic model for an industrial cooling water plant. The second objective is to contribute to recent control research by applying advanced control techniques to the plant under scope.

This chapter reviews the problem and scope to be addressed in this research and clarifies the objectives to be achieved. The research approach is defined and a brief overview of the study is included.

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

The process modelling and control literature has historically focused on the core processing equipment in process industry plants such as e.g. reactors and distillation columns. Utility systems such as steam, fuel gas, and cooling water systems have as a result received little attention. This problem is exacerbated by a lack of real industry data to base the research on and test techniques against.

Dynamic modelling techniques have developed to a mature state, and the various unit operations found in a cooling water network, have been modelled dynamically. However, the literature on holistic dynamic cooling water network models based on an actual process plant, is limited.

Advanced Process Control (APC) techniques and Linear Model Predictive Control (LMPC) in particular have matured as technologies. Non-Linear Model Predictive Control (NMPC) is a promising newer research field with a growing body of literature. However, NMPC still has limited commercial industrial implementation to date [2]. As far as could be ascertained, APC techniques and in particular NMPC have been applied to an industrial cooling water network to a very small degree.

1.1.2 Research gap

This work addressed the gap in applying advanced modelling and control techniques to process utilities and in particular cooling water network plants. In particular, this work uses actual industrial data for the model construction, validation as well as open and closed-loop simulation.

The research gap addressed by the modelling part of the research is the unique application of advanced modelling, validation and parameter estimation techniques utilising real industrial cooling water network data. The plant model is developed in a state-space format such that it can easily be used for dynamic simulation, control and optimisation.

The research gap addressed by the control theory part of the research is the unique application of advanced model predictive and APC techniques to the model that has been constructed. The designed control strategy enables closed-loop simulation of the plant model with the controller for plant regulation and optimisation purposes.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The cooling water circuit studied (Figure 3.1), is an induced draft system of 3 cooling towers in parallel, which feed 2 pumps in parallel. The pumps then feed a number of heat exchangers in parallel. The process unit operations are non-linear.

1.2.1 Research objectives

The objectives of this research are the following:

1. To synthesise, build, validate with industrial data, and simulate a first principles based, dynamic, holistic, process model for an industrial cooling water network.
2. To design, build, and simulate in closed-loop an APC system for the model that has been constructed. This control system will have the following control objectives:

   (a) Temperature control of the hydrocarbon streams being cooled by the cooling water circuit.

   (b) Given the first control objective, optimise energy usage in the circuit.

1.2.2 Research questions

The research questions are the following:

1. Can a first principles, dynamic, holistic process model for an actual industrial, commercially operating plant's cooling water network, be constructed, validated and simulated?
2. Can an APC technique be applied to the constructed model, and a closed-loop simulation be run yielding beneficial results?
3. Can the hydrocarbon streams that are cooled by the cooling water system, be temperature controlled by the mentioned control system?
4. Can the energy consumption of the cooling water circuit be optimised as a secondary control objective after the temperature control?

1.3 HYPOTHESIS AND APPROACH

The research hypothesis can be divided into the following:

1. A dynamic process model can be built, validated and simulated for the selected cooling water network.
2. APC can be applied to this model and a closed-loop simulation can be constructed.
3. The hydrocarbon streams cooled by the cooling water, can be temperature controlled by the new control system.
4. As a secondary control objective, total energy consumption can be optimised.

The following steps are followed:

1. Survey the literature with the research objectives, hypothesis and research questions in mind.
2. Identify a suitable simulation platform for modelling.
3. Collect plant data.
4. Construct the plant model using appropriate dynamic modelling techniques. Individual unit operations will be modelled, and combined into a holistic model for the cooling water network.
5. Validate the model and do parameter estimation using the plant data.
6. Identify the control objectives of the control system to be built.
7. Design and construct the controller for the plant model utilising advanced control techniques.
8. Simulate the closed-loop system.
9. Publish the research outputs and contributions.

1.4 RESEARCH GOALS

The goals of this research can be summarised in two main areas. These two areas are reflected in the way the different chapters in this thesis has been divided and structured. The goals are:

1. Literature study, data collection, and model design in order to construct a dynamic model for the cooling water network in scope.
2. Literature study, and controller design in order to construct an advanced controller and subsequent closed-loop simulation with the control objectives as defined in Section 1.2.1.

1.5 RESEARCH CONTRIBUTION

Dynamic modelling has been applied to cooling water networks in the literature, but more in the building and electric power industries, and often not for commercially operating plants. There is scope for more research in dynamic modelling approaches in the process, petro-chemical and oil and gas industries. Furthermore, cooling water networks that have been dynamically modelled, have been limited in scope and do not have the number of unit operations found in a typical, commercially operating process plant. In addition to this, dynamic models often focus on only one process unit operation found in the

cooling water network (e.g. the cooling tower, or a heat exchanger), and do not encompass the entire cooling water network. This work contributes to the literature by building a holistic dynamic model for a commercially operating process plant cooling water network application.

Advanced control techniques, and in particular Model Predictive Control (MPC), have been applied to a very limited degree to cooling water networks in the literature. Since the dynamic models that these control systems are based on have been limited in scope and application (see above), the scope of these controllers have been limited as well, without a significant commercially operating industry application in the process industries. This research contributes to the literature by building an advanced control system for the model constructed for a commercially operating process plant cooling water network application.

The following publications have originated from this work:

1. J. H. Viljoen, C. J. Muller, and I. K. Craig, "Dynamic modelling of induced draft cooling towers with parallel heat exchangers, pumps and cooling water network", Journal of Process Control, vol. 68, pp. 34-51, 2018.
2. J. H. Viljoen, C. J. Muller, and I. K. Craig, "Hybrid non-linear model predictive control of induced draft cooling towers", Submitted to: Control Engineering Practice, March 2019.

1.6 OVERVIEW OF STUDY

Optimisation and process control studies of cooling water networks in the process industries are not common in the literature. One reason for this is the relative scarcity of dynamic models of such systems. Dynamic, first principles modelling is done in this research with the aim of using the resulting model in process control and optimisation studies.

Dynamic modelling has lately been applied by Muller and Craig [3] to cooling water networks with the view to optimise and control the entire process. In [3] a dual circuit cooling water network was modelled including power consumption, with high-level models for the cooling towers, and the plant exchangers. What distinguishes this research from [3] is that a single circuit network is modelled with detailed heat and mass balance modelling of the internals of the 3 cooling towers, as well as

individual models for the plant heat exchangers, and network hydraulics. In addition, in this work variable speed drives are added to the modelling of the cooling tower fans and the cooling water pumps. For the purpose of optimisation, more detailed modelling of the process enables better root cause understanding of process conditions, and enables greater increases in efficiency in areas such as water loss and power consumption in the cooling water circuit. Another optimisation area enabled by deeper levels of modelling, is the ratio of the flow rates to the different parts of the plant as served by the different heat exchangers. For the purposes of control, more detailed modelling will add robustness to the closed loop system by decreasing the difference between the actual plant and the controller plant model during both steady-state and dynamic transient conditions, as well as add degrees of freedom to be used by the controller.

The cooling water network modelled in this research is inherently non-linear, as well as hybrid, since the pumps and cooling tower fans can be in a running state, or can be off. This gives the state of the equipment a Boolean nature. The system therefore has a combination of continuous process state variables, as well as Boolean state variables. This has an effect on the power consumption modelling of the network, as well as the ultimate aim of control and optimisation.

The next chapter (Chapter 2) documents the literature study that was done in order to establish a foundation of methods and techniques from the literature to serve the goals this work. This literature foundation is then used in Chapters 3 and 4 to develop the methods, techniques, models and controllers needed in order to achieve the goals of this work. Chapter 3 focuses on the methods and results of the modelling part of the research, whereas Chapter 4 focuses on the methods and results of the multivariable control part of the work. Chapter 5 contains a consolidated discussion of the results and analysis contained in prior chapters in light of the goals of the work. Chapter 5 also documents conclusions and a summary of the outcome of the research in the light of the goals, literature surveyed, methods applied and results achieved. Chapter 5 further contains a brief list of potential areas of further research that could build on the knowledge contribution of this work.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

The goal of this chapter is to summarise the literature study done in order to successfully accomplish the goals of this work. The literature study can be organised into the following major areas of research:

1. In Section 2.2 dynamic modelling of a cooling water network is addressed. The following major sub-domains will be covered in this part of the literature survey:

   (a) Process unit operation dynamic modelling.

   (b) Numerical methods.

   (c) Parameter estimation.

2. In Section 2.3 multivariable control of a cooling water network is addressed. In this part of the study, the following sub-domains will be covered in the literature survey:

   (a) Non-Linear Model Predictive Control through stochastic optimisation based techniques.

   (b) Non-Linear Model Predictive Control through gradient based optimisation techniques.

3. In Section 2.4 Hybrid Non-Linear Model Predictive Control techniques are covered.

4. In Section 2.5 Reinforcement Learning based adaptive optimal control techniques are covered.

2.2 DYNAMIC MODELLING OF A COOLING WATER NETWORK

A cooling water system is mathematically complex due to the interaction between the thermodynamics and hydraulics [4, 5]. Mathematical modelling of a cooling water network has been done by various researchers, e.g. neural network modelling of a cooling water circuit for a petro-chemical facility was done by Malinowski *et al.* [6]. Multiple authors have done static hydraulic and static thermal modelling (see e.g. [4]). However, dynamic modelling is the goal of this work and the intent of building dynamic models for the unit operations of the cooling water network is important in each subsection in this chapter. Various authors and techniques for solving the modelling part of the research will be surveyed in this chapter.

2.2.1 Equations of state

For initialising streams, and flashing streams when internal energy or volume changes, the Peng Robinson thermodynamic equations of state are often used and apply to the relevant stream compositions [7, 8, 9]. Alternative equations of state are the Van Der Waals equations [9, 10].

The definitions needed for the Peng Robinson equations of state are

$$\begin{aligned} a &= \frac{0.457235 \cdot R^2 T_c^2}{p_c} \\ b &= \frac{0.077796 \cdot R T_c}{p_c} \\ K &= 0.037464 + 1.54226\omega - \omega^2 \\ T_r &= \frac{T}{T_c} \\ \alpha &= \left(1 + K\left(1 - \sqrt{T_r}\right)\right)^2 . \end{aligned} \tag{2.1}$$

In (2.1), R is the universal gas constant, T is the system temperature, T_c and p_c are the temperature and pressure for a compound at its thermodynamic critical point, and ω is the thermodynamic acentric factor for a particular compound.

Based on (2.1), additional definitions are made as part of the Peng Robinson framework as follows:

$$A = \frac{a \cdot \alpha \cdot p}{R^2 T^2}$$
$$B = \frac{b \cdot p}{RT}. \tag{2.2}$$

The definitions from (2.1) and (2.2), are used to set up the dimensionless cubic Peng Robinson equation

$$Z^3 - (1-B)Z^2 + \left(A - 2B - 3B^2\right)Z - \left(AB - B^2 - B^3\right) = 0. \tag{2.3}$$

The roots of (2.3) denote the solid phase, liquid phase, and gas phase of a particular compound, in a particular stream. From the Z value calculated, various other thermodynamic variables can be calculated for the compound. If the equation only has one real root, then the stream is in a single phase. For the application at hand, this will be either a liquid or gas phase [7, 9].

Once Z is known per compound and per phase, the density of the phase and compound in the particular stream can be calculated from [7]

$$\rho = \frac{B}{b \cdot Z}. \tag{2.4}$$

Given a starting internal energy for each compound in a stream if treated as an ideal gas U_{ig}, as well as the symbols as defined earlier in this section, then the non-ideal fluid internal energy U is calculated from [7]

$$\frac{U - U_{ig}}{RT} = -\frac{A}{B\sqrt{8}}\left(1 + \frac{K\sqrt{T_r}}{\sqrt{\alpha}}\right)\ln\left(\frac{Z + \left(1+\sqrt{2}\right)B}{Z + \left(1-\sqrt{2}\right)B}\right). \tag{2.5}$$

2.2.2 Cooling tower modelling

Kloppers and Kröger [11] did a comparison of 3 methods in the literature that are historically used to evaluate and model the steady-state behavior and performance of cooling towers: The Poppe, Merkel, and e-NTU methods. These methods, although helpful, do not provide dynamic models that can be

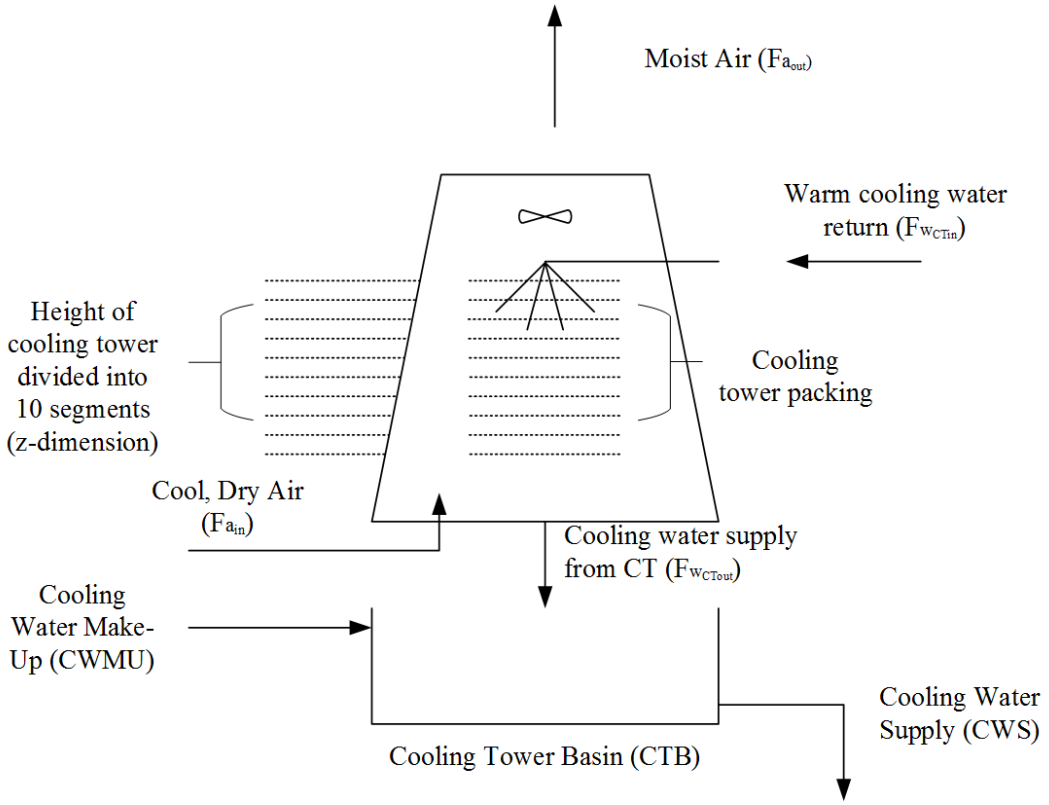

Figure 2.1. Main streams in and out of each cooling tower.

used for control and optimisation. Few dynamic models that describe cooling tower unit operation are available in the literature; see for example the dynamic, first principles models built by Li, Li and Seem [12] and Löfgren [13]. In both [12] and [13] the authors developed dynamic models for a cooling tower by dividing the tower into segments. However in [12] the air side transient mass and energy transfer is neglected, whereas it is included in [13].

For a number of the cooling tower models found in the literature (e.g. in McCabe *et al.* [14]), as well as for practical operation, the wet bulb temperature is a key variable to be calculated [8]. A model for this variable is defined by Stull [15]. Stull shows that the wet-bulb temperature T_{wb} depends on ambient temperature (T_a) and humidity (Y_a) as follows

$$T_{wb} = T_a \arctan\left(0.151977(Y_a + 8.313659)^{0.5}\right) + \arctan\left(T_a + Y_a\right) - \arctan\left(Y_a - 1.676331\right) + 0.00391838 \cdot Y_a^{1.5} \cdot \arctan\left(0.023101 \cdot Y_a\right) - 4.686035. \quad (2.6)$$

Given the wet-bulb temperature, the cooling tower over-all efficiency (η) can be calculated as

$$\eta = \frac{T_{w,in} - T_{w,out}}{T_{w,in} - T_{wb}} \tag{2.7}$$

where T_{wb} is the wet-bulb temperature, $T_{w,in}$ is the temperature of the inlet water flow, and $T_{w,out}$ is the temperature of the water outflow [13, 16].

Fisenko *et al.* [16] constructed a detailed steady-state model for the evaporative cooling of a cooling tower, where it is assumed that all the water flow that happens in the cooling tower, is in the form of droplets. Included in this work is a distribution of droplet radius values, and a relationship between radius, falling speed, water surface tension and air density.

In [3] as well as [17], the mass balance for the cooling tower includes the make-up, evaporation, and blowdown streams in addition to the main in and outflow of cooling water into and from the tower. An assumption about the number of concentration cycles in the system is included in most models (Castro *et al.* [4]).

In [18], the cooling tower outlet temperature is modelled as a function of the flow rates and temperatures of the streams indicated in Figure 2.1, as well as the cooling water thermal effectiveness. In [5], an enthalpy balance is used to model the cooling tower, with the mass transfer coefficient of the packing determined experimentally due to the complex surface geometries of the cooling tower fillings. In [19], Ponce-Ortega *et al.* constructed a still more detailed enthalpy model.

Al-Nimr [20] constructed a simple dynamic model for a cooling tower where both the sensible and latent heat transfer is taken into account. In [12], the dependence of the water density on temperature in each water cell is included in a dynamic model. Ma *et al.* [21] built a simplified dynamic model of the cooling tower, without discussing the details of model derivation. In [22], the heat transmitted by convection to the packing in the tower was included in the dynamic model.

The interface between the water bulk and the air bulk is often assumed to be saturated, i.e. the relative humidity is 100% [23].

Central to developing a dynamic model for the cooling tower is modelling the mass and heat transfers

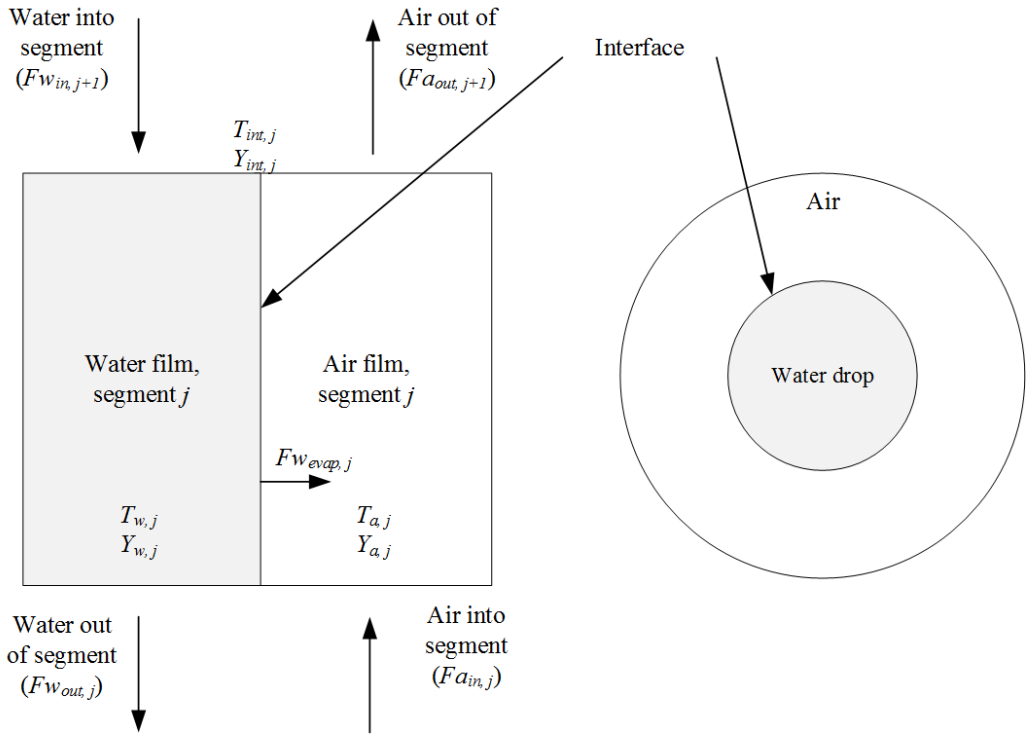

Figure 2.2. The main variables and interface for the j[th] segment of the cooling tower viewed with a vertical cross section from a horizontal perspective.

that occur in the tower. Middleman [24] uses a molar flux equation at low rates of evaporation. In this equation, the mass transfer rate of evaporated water into the air is directly proportional to the difference in the molar fraction of the water vapour at the interface, and the molar fraction of water of the bulk air. Since the dynamic transient of the heat transfer rate to reach steady state in the interface is assumed to be fast (Marques *et al.* [25]), a steady state energy balance equation can be used for the interface of the j[th] segment in the tower as follows:

$$h_w a_{ct} \left(T_{w,j} - T_{int,j}\right) = h_a a_{ct} \left(T_{int,j} - T_{a,j}\right) + h_D \lambda a_{ct} \left(Y_{int,j} - Y_{a,j}\right). \tag{2.8}$$

Here, h_D is the cooling tower air phase mass transfer coefficient, λ is the latent heat of vaporisation of water, a_{ct} is the cooling tower contact area of water surface per volume unit in the tower, $Y_{int,j}$ is the interface absolute humidity of the j[th] segment, $Y_{a,j}$ is the air absolute humidity of the j[th] segment, h_a is the cooling tower air phase heat transfer coefficient, h_w is the cooling tower water phase heat transfer coefficient, $T_{w,j}$ is the water temperature of the j[th] segment, $T_{int,j}$ is the interface temperature of the j[th] segment, and $T_{a,j}$ is the air temperature of the j[th] segment. The symbols h_D, h_a and h_w were chosen this way with subscripts indicating which specific quantity is being referred to, since the same symbols

for heat and mass transfer coefficients were used in some of the most prominent articles on cooling tower modelling: E.g. [11], [25] and [26].

According to [27], Buck's formula can be used to estimate the saturation vapour pressure of water as a function of water temperature. The Arden Buck equations are a group of empirical correlations that relate the saturation vapor pressure to temperature in the case of moist air [28]. Directly over liquid water

$$p_s(T) = 0.61121 \cdot \exp\left(\left(18.678 - \frac{T}{234.5}\right)\left(\frac{T}{257.14+T}\right)\right) \tag{2.9}$$

where p_s is the water saturation vapour pressure in kPa, and T is the temperature in °C [28].

From the Buck equation (2.9), the saturation water vapour pressure at the segment interface can be calculated using the interface temperature T_{int} as calculated from (2.8). From the water vapour pressure, the interface absolute humidity (water mass per mass dry air) can be calculated as [8]

$$Y_{int,j} = 0.622 \frac{p_{wvap,j}}{p_s - p_{wvap,j}}, \tag{2.10}$$

where p_{wvap} is the water vapour pressure at the interface, and p_s is the standard atmospheric pressure. From (2.8) and (2.10), the mass flow rate of water that evaporates from the j$^{\text{th}}$ segment can be calculated as

$$F_{w_{evap},j} = h_D a_{seg}\left(Y_{int,j} - Y_{a,j}\right), \tag{2.11}$$

where $F_{w_{evap},j}$ is the water evaporating rate for the j$^{\text{th}}$ cooling tower segment, and a_{seg} is the contact area of the water surface per segment volume [13]. From (2.11) and Figure 2.2, the mass balance of the water film for the j$^{\text{th}}$ segment is then given by

$$F_{w_{out},j} = F_{w_{in},j+1} - F_{w_{evap},j}. \tag{2.12}$$

From the energy balance for the water film in (2.13), the water temperature in the j$^{\text{th}}$ segment, $T_{w,j}$, can be calculated [25] as

$$\xi_w \rho_w c_w \frac{\partial T_{w,j}}{\partial t} = \frac{F_{w,j}}{a_h} c_w \frac{\partial T_{w,j}}{\partial z} - h_w a_{ct}\left(T_{w,j} - T_{int,j}\right), \tag{2.13}$$

where ξ_w is the volume fraction of water in the jth cooling tower segment, ρ_w is the density of water, c_w is the heat capacity of water, z is the vertical dimension in the tower, and a_h is the horizontal cross-sectional area of the jth cooling tower segment. Similarly from the energy balance for the air film in (2.14), the air temperature in the jth segment, $T_{a,j}$, can be calculated [25] as

$$\xi_a \rho_a c_a \frac{\partial T_{a,j}}{\partial t} = -\frac{F_{a,j}}{a_h} c_a \frac{\partial T_{a,j}}{\partial z} - h_a a_{ct} \left(T_{a,j} - T_{int,j}\right), \tag{2.14}$$

where ξ_a is the volume fraction of air in the cooling tower, ρ_a is the density of air and c_a is the heat capacity of air. From the mass balance in (2.15), the air humidity for segment j, $Y_{a,j}$, can be written as follows:

$$\xi_a \rho_a \frac{\partial Y_{a,j}}{\partial t} = -\frac{F_{a,j}}{a_h} \frac{\partial Y_{a,j}}{\partial z} + h_D a_{ct} \left(Y_{int,j} - Y_{a,j}\right). \tag{2.15}$$

In [16], Fisenko *et al.* used relations to calculate the temperature dependence of mass and heat transfer coefficients as a function of temperature. The relationship between the air mass transfer coefficient, and the air heat transfer coefficient, has been modelled in [26] and [29]. According to [14], the following relation can be used for air-water cooling tower systems:

$$c_a = \frac{h_a}{h_D}. \tag{2.16}$$

2.2.3 Fan modelling

Power consumption efficiency optimisation is an important strategic focus area in the process industry [30]. The power requirements for cooling water networks have been calculated in [3] and [17]. Power consumption modelling with the intent of enabling power optimisation and control is included in this literature study for all unit operations that involve rotating equipment, starting with the modelling of the fans in the system.

Mechanical power, generated by the electric motor of each fan is partly converted to hydraulic work by the cooling tower fan, based on the hydraulic efficiency, and proportional to the pressure drop over the fan and the volumetric flow rate through the fan [31]. In [12], the first two fan affinity laws are shown that relates flow through the fan (F_f), pressure head created (p_f), and power (P_f) to fan speed

(n), from one steady-state operating point (point 0) to another one (point 1) as follows:

$$F_{f1} = F_{f0}\left(\frac{n_1}{n_0}\right) \tag{2.17}$$

$$p_{f1} = p_{f0}\left(\frac{n_1}{n_0}\right)^2 \tag{2.18}$$

$$P_{f1} = P_{f0}\left(\frac{n_1}{n_0}\right)^3 \tag{2.19}$$

In [32] and [33] the power benefits are shown that can be obtained by using Variable Speed Drives (VSDs), especially for fans given the highly non-linear relationship between fan speed and power consumption as seen in (2.19). In [32] this benefit is specifically analysed for a cooling water network, but the modelling performed did not include any dynamics and the optimisation algorithm did not take a multi-variable approach.

In [34], Sen points out that the starting current of induction motors can exceed the steady-state current by 500% to 800%. This translates into very high power consumption during start-up which is a concern as most fans and pumps in industrial plants are driven by induction motors, including the fans and pumps studied in this work (see Chapter 3).

In [35], Szymczyk and Karaskiewicz modelled and simulated the initial transients of 11,500 m^3/h centrifugal induction water pumps after start-up of the pump. The pumps exhibited an under-damped dynamic response for the power consumption. The power consumption overshoot was around 120%, and the pumps reached steady-state in fewer than 30 seconds.

2.2.4 Cooling tower basin modelling

Both [12] and [25] have developed models for the basin holding the water coming down from the cooling tower packing, before the water is pumped out to the plant again as Cooling Water Supply (CWS). Assuming that the volume of the water in the basin stays constant over time, which is a valid assumption as it is level controlled with make-up water in most plants, then the basin mass balance can

be modelled as

$$F_{in} - F_{CWS} + F_{makeup} = -\beta \rho_w V_{CTB} \frac{dT_{CTB}}{dt} \tag{2.20}$$

where F_{in}, F_{CWS} and F_{makeup} are the mass flow rates of the water from the cooling tower into the basin, out of the basin, and the make-up flow respectively, β is the isobaric thermal expansion coefficient of the water, ρ_w is the water density, V_{CTB} is the volume of water in the basin, and T_{CTB} is the water temperature in the basin [12]. The energy balance can be modelled as

$$c_w \frac{dT_{CTB}}{dt} = \frac{F_{in}\left(h_{in} - h_{CTB}\right) - F_{CWS}\left(h_{CWS} - h_{CTB}\right) + F_{makeup}\left(h_{makeup} - h_{CTB}\right)}{\rho_w V_{CTB}} \tag{2.21}$$

where c_w is the water heat capacity, and h_{CTB}, h_{in}, h_{CWS} and h_{makeup} are the specific enthalpy values of the water in the cooling tower basin, the water stream from the cooling tower, the water flowing out from the basin to the plant, and the make-up water respectively. Analogously to (2.21) Marques *et al.* [25] modelled the cooling tower basin energy balance as

$$\rho_w V_{CTB} \frac{dT_{CTB}}{dt} = F_{makeup}\left(T_{makeup} - T_{w_{CTout}}\right) + F_{CWS}\left(T_{w_{CTout}} - T_{CTB}\right), \tag{2.22}$$

where $T_{w_{CTout}}$ is the temperature of the water coming from the cooling tower into the cooling tower basin, and it assumed that the difference in mass flow rate coming into the basin from the cooling tower and leaving to the plant as CWS is made up by the make-up water.

2.2.5 Pump modelling

In [5] and [18] it is recommended to use a second degree polynomial to represent a pump curve. In [36], Wang and Chen use a second degree polynomial to also model pump output differential pressure (Δp_p) as a function of rotational speed (ω) and mass flow rate (F):

$$\Delta p_p = a_{p_0}\omega^2 + a_{p_1}\omega\frac{F}{\rho_w} + a_{p_2}\frac{F^2}{\rho_w^2} \tag{2.23}$$

where a_{p_0}, a_{p_1} and a_{p_2} are constants that can be fitted using the pump curve as per the pump datasheet and (2.23). The steady-state power consumption of the pump ($P_{p_{ss}}$) can be calculated as per (2.24) if

mechanical losses are assumed to be negligible [36]:

$$P_{p_{ss}} = F\frac{\Delta p_p}{\rho_w} \tag{2.24}$$

Li and Li [12] have built more theoretical, first principles based dynamic models for a water pump's mass balance and energy balance.

2.2.6 Hydraulic network modelling

The temperature of the resultant stream flowing from a mixer (assuming no phase changes occur in the mixer), can be modelled as the sum of the temperatures of the input streams weighted by the mass flow rate per input stream [5, 17] as

$$T_m = \frac{1}{F_m}\sum_{i=1}^{n}(F_i \cdot T_i) \tag{2.25}$$

where T_m is the mixer output temperature, F_m is the mixer output mass flow rate, F_i is the i$^{\text{th}}$ input mass flow rate to the mixer, T_i is the i$^{\text{th}}$ input stream temperature to the mixer, and n is the number of streams flowing into the mixer.

The pressure drop over a segment of pipe (ΔP) is calculated in [18] as

$$\Delta P = \left[\frac{32}{\pi^2} f\left(\frac{L}{d}\right)\left(\frac{\rho}{gd^4}\right)\right] V^2 \tag{2.26}$$

where ρ is the liquid density, f is a friction factor, L is the length of the pipe, d is the diameter of the pipe, g is the gravitational constant, and V is the volumetric flow rate [9, 18]. Equation (2.26) can be shortened into

$$\Delta P = K \cdot V^2 \tag{2.27}$$

where K is the applicable coefficient as per (2.26) [18].

For a network of parallel branches of liquid flow, each parallel branch must have equal pressure drop as part of the over-all hydraulic solution of the network [37]. Parallel branches in a heat exchanger,

valve and pipe network can then be solved as a linear system as per [18], using solution algorithms described in [38]. Process equipment in series such as valves, pumps and parallel branch networks will converge to an overall hydraulic pressure-flow operating point as per the resulting overall system characteristic [3].

2.2.7 Control valve modelling

Most authors use a quadratic relationship scaled by valve opening (depending on the valve characteristic), between valve pressure drop and mass flow, where the pressure drop is scaled by the density [3, 8, 39]. The actual volume flow (F/ρ) is proportional through the valve coefficient C_{vCV} to the product of the valve characteristic $f(.)$, the valve opening (V_{OP}), and the square root of the pressure drop over the valve (Δp_{valve}) divided by the specific gravity (g_s).

$$\frac{F}{\rho} = C_{vCV} \cdot f(V_{OP}) \cdot \sqrt{\frac{\Delta p_{valve}}{g_s}} \tag{2.28}$$

The relationship between the valve flow, and the valve travel, $f(.)$, is called the valve flow characteristic [8]. Three very common valve characteristics are quick opening, linear and equal percentage characteristics [8]. Equal percentage flow characteristics are often used for control valves in cooling water networks since Δp_{valve} over the valves change depending on the operating point of the cooling water circuit. Therefore, a more linear installed flow characteristic can be obtained through using the non-linear equal percentage valve characteristic [39]

$$f(V_{OP}) = (R_{EP})^{V_{OP}-1}, \tag{2.29}$$

where R_{EP} is a valve design parameter.

2.2.8 Hammerstein modelling

Hammerstein models are often used to model unit operations in the process industries [40, 41]. The Hammerstein system combines linear dynamics with non-linear steady-state gains [42]. If $f(u,\theta)$ represents the non-linear steady-state gain component, u the input of the model and $g(t,\tau)$ represents the linear time dynamics component with time constant vector τ, then in continuous time the output of

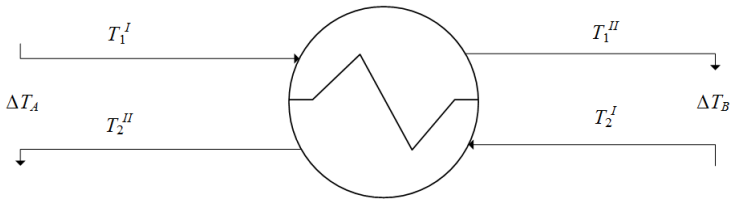

Figure 2.3. Generic cross-flow heat exchanger for heat exchange modelling.

the Hammerstein model is given by [42]

$$\begin{aligned} y(u,\theta,t,\tau) &= f(u,\theta)\, g(t,\tau) \\ &= f(u,\theta)\, \mathcal{L}^{-1}(G(s,\tau)) \end{aligned} \tag{2.30}$$

where $\mathcal{L}^{-1}$ is the inverse Laplace transform of $G(s)$ to emphasise the linearity of the time domain dynamic component. The linear time dynamics $g(t,\tau)$ in (2.30) has a steady-state gain of 1. The Hammerstein approach has the benefit that the modelling can be simplified from models where the dynamic component is also non-linear and integrated as part of the over-all model (see the cooling tower modelling Section 2.2.2 for example). Accuracy could be sacrificed in order to obtain this benefit. If the part of the over-all model being constructed with a Hammerstein approach is less important in terms of the goal of the larger model, it could be an attractive alternative.

2.2.9 Heat exchanger modelling

Various detailed heat exchanger dynamic models have been published in the literature. In [43], a cross flow heat exchanger was modelled dynamically. In [44], a dynamic model of 3 heat exchangers was built. Muller and Craig [3] used a Hammerstein (2.30) approach to modelling the non-linear gain and linear dynamics for heat exchangers in a cooling water network.

Thermodynamic modelling of heat exchangers is assisted by using the concept of Logarithmic Mean Temperature Difference (LMTD) in [3] as well as in [37]. The LMTD is also defined in [8] for different kinds of heat exchangers such as co-current flow and counter-current flow. It is defined in (2.31) with reference to the temperatures and differential temperatures as portrayed in Figure 2.3.

$$\Delta T_{lm} = \frac{\Delta T_A - \Delta T_B}{\ln \frac{\Delta T_A}{\Delta T_B}} \tag{2.31}$$

The differential temperatures are defined as follows:

$$\begin{aligned} \Delta T_A &= T_1^I - T_2^{II} \\ \Delta T_B &= T_1^{II} - T_2^I \end{aligned} \tag{2.32}$$

The heat energy flow in the exchanger is the product of the heat exchanger U factor, the heat exchange area (A), and the LMTD [3, 18].

2.2.10 Numerical methods

In [22], an explicit finite difference method was used to solve the system of developed dynamic Partial Differential Equations (PDEs). The system of differential and algebraic equations developed in this work are of a stiff nature [38] for the cooling tower (see Section 3.3.2 for the modelling drawing on the literature in Section 2.2.2), but not for the rest of the plant.

If the PDEs of the plant system can be rewritten in the form of Ordinary Differential Equations (ODEs) as in

$$\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0, \tag{2.33}$$

then Euler and Runge-Kutta techniques are some of the most common techniques that can be applied to numerically solve the system [38, 45]. Euler's method consist of applying the iterative formula

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \tag{2.34}$$

where h is the step size. Runge-Kutta methods are robust and accurate [45] techniques for the numerical solution of the system in (2.33). The Runga-Kutta solution uses the following equations at iteration

step n of the algorithm

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \\ k_4 &= f(x_{n+1}, y_n + hk_3) \end{aligned} \tag{2.35}$$

where h is the numerical integration step size. The next value for y is given by [9, 38, 45]

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.36}$$

The Runge-Kutta method is a fourth order method, and it can be proven that the cumulative error on a bounded interval is of order h^4 [45].

Stiff systems are sets of differential equations that contain a mixture of very fast dynamic equations and very slow dynamic equations. Single as well as multi-step algorithms are known to potentially have trouble obtaining solutions for such systems [38]. Smaller step sizes can be used to obtain more accurate results. Variable step size approaches are often based on the Klopfenstein modification of classical backward difference formulas called numerical differential formulas. In order to determine the optimum step size and speed up convergence of the implicit corrector formulas, the method depends on the Jacobian, β, of the derivative function f in [38]

$$\dot{x} = f(x,t) \tag{2.37}$$

where

$$\beta = \frac{\partial f}{\partial x}. \tag{2.38}$$

2.2.11 Parameter estimation

Least squares parameter estimation is a common technique used to fit dynamic non-linear process models [8, 46]. The gradient based techniques used for Non-Linear Model Predictive control as documented in Section 2.3, can be modified and used for parameter estimation with a modified objective function.

In [3], a genetic algorithm was chosen for parameter estimation due to the ability of this algorithm to handle non-linear, discontinuous, multi-parameter problems and its likelihood of finding a global optimum. In [47], a novel technique based on the biological DNA mechanism for utilising genetic algorithms to estimate parameters of dynamic systems in the process industries, was developed. See Section 2.3.2 for an overview on genetic algorithms applied to Model Predictive Control.

In [48], Kennedy and Eberhart outlines the concept of using Particle Swarm Optimisation (PSO) for the optimisation of non-linear continuous functions. The PSO algorithm is relatively simple, but very powerful for non-linear optimisation, including for hybrid systems [49]. PSO is an optimisation method that is based on swarm intelligence as found in nature (bee swarms, bird swarms, etc.). It is a collective, iterative method, with an emphasis on cooperation [50].

Each particle stores the speed and position of all the model parameters that are to be optimised. The speed (v_j) of the jth parameter of the ith particle in the population is updated in each iteration for each parameter:

$$v_j = v_j + 2\theta\,(p_j - x_j) + 2\gamma(g_j - x_j) \tag{2.39}$$

where θ and γ are random real numbers between 0 and 1 evaluated for each iteration, p_j is the value of the jth optimisation parameter for the ith particle that has had the best performance in the latest solution space, g_j is the position of the jth parameter for the particle in the entire population that has had the best performance in the latest solution space, x_j is the value of the jth parameter of the ith particle at the time of the iteration.

The position of the jth parameter for the ith particle is subsequently updated with the newly calculated speed in each iteration:

$$x_{j_{k+1}} = x_{j_k} + v_j \tag{2.40}$$

The performance of each particle is evaluated at each iteration by evaluating the objective function of the optimiser with prediction horizon N. In this work, the prediction horizon spans the entire dataset that is used to fit each parameter. The objective function sums the square of the error between the variable being fitted, and the real plant data for the variable, over the length of the prediction horizon.

There is a large number of examples of the usage of PSO for parameter estimation in the literature. One such example for process modelling is given in [51].

With regards to cooling tower model fitting in particular (Section 2.2.2), Marques *et al.* [25] fitted their model utilising cooling tower inlet and outlet temperatures and flow rates.

2.3 MULTIVARIABLE CONTROL OF A COOLING WATER NETWORK

The benefits of applying APC to petrochemical plant utilities is becoming more apparent as such plants are coming under increasing pressure to lower their carbon footprints. Increasing attention is therefore being paid in the literature to the optimisation of utility systems through advanced control and optimisation techniques. Deng *et al.* developed a Dynamic Programming combined with MINLP Model Predictive Control strategy utilising a linear plant model to solve the scheduling problem for a campus central chiller [52]. Yin and Li developed a linear MPC strategy for the energy efficient control for vapour compression refrigeration cycle systems [53]. APC techniques have also been similarly applied to utility systems in [3, 30, 54, 55, 56, 57]. Ambient humidity has been identified as one of the most important disturbance variables to include in the control and optimisation of cooling water systems [4]. In this section the literature is surveyed in order to gather the applicable research to be able to solve the control problem of the cooling water network. The selected control strategy should be able to deal with hybrid non-linear multivariable models as all these elements are present in the cooling water network model.

Linear Model Predictive Control (LMPC) has become the most popular multivariable control strategy in the process industries [58]. LMPC is often called APC in the process industries, however, APC refers to non-linear and other control and optimisation techniques as well in the literature [59]. A LMPC formulation can integrate optimal control, stochastic control, control of processes with dead time, and multivariable control [2]. Conventional LMPC techniques have been used in [21, 25, 44, 60] to control cooling water systems. When the dynamics are linear and finite dimensional, the stage cost is quadratic, and there are no constraints, the objective function reduces to the Ricatti difference equation [61, 62].

LMPC makes use of a linear plant model, and in cases where the plant exhibits highly non-linear behaviour, the controller response will often not be acceptable [2], hence the need for Non-Linear Model Predictive Control (NMPC). This problem is exacerbated if the process spends a lot of time away from the stable design operating point. Significant progress has been made in NMPC research over the last few years [62, 63, 64, 65, 66].

2.3.1 Non-Linear Model Predictive Control

An NMPC controller requires a model of the plant to be controlled. There are a number of methods that can be used to obtain such models. First principle models can be difficult to obtain but are often preferred as they preserve process knowledge, and when used for controller design provide additional benefits for optimisation, constraint forecasting, design and diagnoses [67]. First principle models are therefore often used for NMPC applications. Recent results in the development of efficient large-scale non-linear programming algorithms have led to faster, online realisations of optimisation methods for NMPC, with predictive non-linear dynamic (e.g., first principle) models [65, 66].

The essential goal in NMPC is to compute a control move sequence $\mathbf{u}$ such that the system to be controlled reaches an equilibrium

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.41}$$

where $\mathbf{x}_{k+1}$ is the value of the system state vector at the next sampling instant, $k+1$, as produced by the non-linear function f taking parameters of the current system state $\mathbf{x}_k$, and the calculated current control move $\mathbf{u}_k$ [61, 63]. The control sequence $\mathbf{u}$ produced by a function $\mu(\mathbf{x})$ depends on the system

state as per

$$\mathbf{u}_k = \mu\left(\mathbf{x}_k\right). \tag{2.42}$$

The control sequence $\mathbf{u}$ can be expanded as $\{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$. The list of variables included in each $\mathbf{u}$ vector, are often referred to as Manipulated Variables (MVs) in the literature [3, 1]. For any control sequence $\mathbf{u}$, the solution of (2.41) at time k with initial state $\mathbf{x}_0$ is denoted as $\mathbf{x}_\mathbf{u}\left(k, \mathbf{x}_0\right)$ [63]. The NMPC algorithm will minimise the system cost function [61, 63]

$$J_N\left(\mathbf{x}_0, \mathbf{u}\left(\cdot\right)\right) = \sum_{k=0}^{N-1} l\left(\mathbf{x}_\mathbf{u}\left(k, \mathbf{x}_0\right), \mathbf{u}_k\right) + V_f\left(\mathbf{x}_\mathbf{u}\left(N, \mathbf{x}_0\right)\right), \tag{2.43}$$

where V_f is the terminal value of the objective function, l is the non-terminal cost and N is the prediction horizon. The minimisation is done subject to constraints as in [61, 63]

$$\begin{aligned} &\min_{\mathbf{u}(\cdot)} J_N\left(\mathbf{x}_0, \mathbf{u}\left(\cdot\right)\right) \\ &\text{with respect to } \mathbf{u}\left(\cdot\right) \in \mathbb{U}^N\left(\mathbf{x}_0\right) \\ &\text{subject to } h\left(\mathbf{u}\right) \geq 0, \end{aligned} \tag{2.44}$$

where $\mathbb{U}^N\left(\mathbf{x}_0\right)$ is the set of admissible control sequences, and $h\left(\mathbf{u}\right)$ is the inequality constraint function [63].

In the NMPC literature, the control horizon normally refers to the number of control moves that are calculated in open-loop and sent to the plant, before feedback from the plant is fed into the controller again and the optimisation algorithm is run again [61]. In other words if the NMPC control horizon is m, the system stays in open-loop for m steps of the control sequence $\{\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+m-1}\}$ from (2.42). This definition is different from the definition for control horizon in LMPC [2]. The LMPC control horizon is the number of time steps that the manipulated variables are calculated into the future at each LMPC controller iteration, of which only the first control move is sent to the plant while the system stays in closed-loop. Shorter NMPC control horizons require shorter prediction horizons for the same stability guarantees [68]. In order to save on the required computational cost, shorter NMPC prediction horizons and NMPC control horizons are therefore preferred.

In the case where the plant is simulated, as opposed to a real plant, the NMPC controller will send

control sequences from the non-linear controller to the solver of the dynamics (2.41) of the process or system being controlled. The dynamics solver will in turn send computed state sequences $\mathbf{x}_k$ back to the non-linear controller. In the case of reference following control, this feedback signal will be subtracted from the reference signal $\mathbf{r}_k$ before being sent to the controller as input for the objective function [61].

In the following sub sections of Section 2.3, stochastic NMPC optimisation methods will be discussed, followed by sections on deterministic NMPC methods using non-linear optimisation through gradient based techniques, followed by a section on Reinforcement Learning approaches.

2.3.2 Non-Linear Model Predictive Control through stochastic non-linear optimisation

The field of stochastic optimisation includes algorithms that can also be applied for control engineering purposes. Example of these algorithms are Genetic Algorithms (GA) [69] and Particle Swarm Optimisation (PSO). GA are iterative procedures that normally operate on populations of constant size. Initially population members are generated randomly, and for each one a fitness level is calculated. Then cross-over and mutation between the fittest population members produce children, after which fitness is again calculated, and the procedure repeated [70]. The basic algorithm is outlined below in Algorithm 1.

Algorithm 1 Genetic Algorithm in pseudo code

Produce an initial population of individuals.
Evaluate the fitness of all individuals.
while termination condition not met **do**
  Select fitter individuals for reproduction and produce new individuals (cross over and mutation).
  Evaluate fitness of new individuals.
  Generate a new population by inserting some new "good" individuals and by erasing some old "bad" individuals.
end while

Individuals encode a set of decision variables, like MVs, by concatenating them in a bit string according to the standard binary code, where the interval of interest and desired precision of the decision variables determines the length of the bit string. The objective function to be optimised in GA terminology is a measure of fitness [71].

Numerous examples of GA applied to control design [71], and NMPC [55, 72, 73, 74, 75] exist in the literature.

In Section 2.2.11 the PSO algorithm was described. The PSO algorithm is a very powerful optimiser for a continuous non-linear optimsation problem. Numerous examples exist in the literature of PSO being applied to solve NMPC problems, e.g. [76, 77, 78, 79, 80]. The PSO algorithm applied to the problem of hybrid control will be reviewed in Section 2.4.1.

Stochastic optimisation approaches are very powerful in that the probability of reaching a global minimum for the objective function is high [71], however this cannot be guaranteed. Also, the PSO and GA techniques are stochastic in nature, as opposed to deterministic and will not always reach a minimum at the same speed for different runs of the controller.

2.3.3 Non-Linear Model Predictive Control through deterministic non-linear optimisation

In this section and the next two, convexity will often be referred to. A set $S \in \mathbb{R}^n$ is a convex set if the straight line connecting any two points in S lies entirely in S. Formally, for any two points $\mathbf{a} \in S$ and $\mathbf{b} \in S$ we have [81, 82, 83]

$$\alpha\mathbf{a} + (1-\alpha)\,\mathbf{b} \in S \quad \forall \quad \alpha \in [0,1]\,. \tag{2.45}$$

A function J is convex if its domain S is a convex set (2.45) and if for any two points $\mathbf{a} \in S$ and $\mathbf{b} \in S$ the following is true [81]:

$$J(\alpha\mathbf{a} + (1-\alpha)\,\mathbf{b}) \leq \alpha J(\mathbf{a}) + (1-\alpha)\,J(\mathbf{b}) \quad \forall \quad \alpha \in [0,1]\,. \tag{2.46}$$

It is assumed in the sections that follow that the continuous time plant systems to be controlled are sampled at constant frequency when the algorithms are implemented on digital computers. Moreover, if an asymptotically stabilizing feedback law for the discrete time system induced by the sampled data system can be computed, then the resulting continuous time sampled data closed-loop is also asymptotically stable provided its solutions are uniformly bounded [61].

Cost functions of NMPC problems can include a terminal cost that is valid only at the end of the prediction horison as included e.g. in (2.43) [62]. This terminal cost can be added in order to assist the guarantee of the convergence and stabalisability of the control problem [61]. Furthermore, there could be different constraints on the MVs at the end of the prediction horison as well, called terminal constraints [61].

The dynamic programming principle (see Section 2.5 on Reinforcement Learning and [84]) is important for the analysis of NMPC systems [61]. However, for the actual numerical solution of NMPC problems, the computational cost of dynamic programming approaches usually grows exponentially with the dimension of the state of the system. In contrast, the gradient based techniques described in the sections below scale much more moderately with the space dimension and are applicable to systems with higher state dimensions [61].

Grüne and Pannek [61] have analysed asymptotic stability for the generic NMPC problem without terminal constraints

$$\begin{aligned} \min_{\mathbf{u}(\cdot)} J_N(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \min_{\mathbf{u}(\cdot)} \sum_{k=0}^{N-1} l(\mathbf{x}_\mathbf{u}(k, \mathbf{x}_0), \mathbf{u}(k)) \\ \text{with respect to } \mathbf{u}(\cdot) &\in \mathbb{U}^N(\mathbf{x}_0), \text{ subject to} \\ \mathbf{x}_\mathbf{u}(0, \mathbf{x}_0) = \mathbf{x}_0,\ \mathbf{x}_\mathbf{u}(k+1, \mathbf{x}_0) &= f(\mathbf{x}_\mathbf{u}(k, \mathbf{x}_0), \mathbf{u}(k)) \end{aligned} \tag{2.47}$$

where J is the objective function, l is the cost function, $\mathbf{x}$ is the system state vector, $\mathbf{u}(\cdot)$ is the control sequence to be found through optimisation, k is the time steps, $\mathbf{x}_0$ is the initial system state, f is the system dynamic model, and N is the prediction horizon. Grüne and Pannek [61] list the conditions, including a suitable Lyaponov function [85], under which the nominal NMPC closed-loop system (2.47) with NMPC feedback law μ_N is asymptotically stable on $\mathbb{X}$ (the state space within the state constraints of the system). Included in this analysis are conditions for which the closed-loop objective function at equilibrium, J_∞^{cl}, is bounded [61] .

Grüne and Pannek [61] show that larger prediction horizon values mostly result in more stable and accurate control. However, they also show that closed-loop systems can become unstable if the prediction horizon is too long though, suggesting there is an optimum horizon length. Very long prediction horizons are also more computationally expensive [61].

Numerical calculation of derivatives are important for deterministic numerical optimisation methods, as first order (Section 2.3.5) as well as second order (Section 2.3.4) methods rely on the Jacobian and Hessian vectors and matrices [61, 81]. A popular formula for approximating the partial derivative $\frac{\partial J}{\partial u_i}$ at a given point $\mathbf{u}$ is the forward-difference or one-sided-difference approximation, defined as

$$\frac{\partial J}{\partial u_i}(\mathbf{u}) \approx \frac{J(\mathbf{u}+\varepsilon e_i) - J(\mathbf{u})}{\varepsilon} \tag{2.48}$$

where ε is a small scalar and e_i is the i^{th} unit vector, i.e. the vector of which the elements are all 0 except for a 1 in the i^{th} position [81]. A similar approach can be used to calculate the Hessian matrix. Other approaches to numerical differentiation include graph based approaches [81, 86].

Deterministic numeric non-linear optimisation methods are normally divided into two classes in the literature: Line search methods and trust region methods [61, 81, 82, 87]. In the line search strategy, the algorithm chooses a direction $\mathbf{p}_k$ and searches along this direction from the current iterate $\mathbf{u}_k$ for a new iterate with a lower function value. The distance to move along $\mathbf{p}_k$ can be found by approximately solving the following one-dimensional minimisation problem to find a step length α

$$\min_{\alpha>0} J(\mathbf{u}_k + \alpha \mathbf{p}_k) \tag{2.49}$$

By solving 2.49 exactly, the maximum benefit can be derived from the direction $\mathbf{p}_k$, but an exact minimisation may be expensive and is usually unnecessary. Instead, the line search algorithm generates a limited number of trial step lengths until it finds one that loosely approximates the minimum of (2.49). At the new point, a new search direction and step length are computed, and the process is repeated [81].

For trust region algorithms, the information gathered about J is used to construct a model function m_k whose behaviour near the current point $\mathbf{u}_k$ is similar to that of the actual objective function J [61]. Because the model m_k may not be a good approximation of J when $\mathbf{u}$ is far from $\mathbf{u}_k$, the search for a minimiser of m_k is restricted to some region around $\mathbf{u}_k$. The candidate step p is found by approximately solving the following sub-problem [81]:

$$\min_{\mathbf{p}} m_k(\mathbf{u}_k + \mathbf{p}), \quad \text{where } \mathbf{u}_k + \mathbf{p} \text{ lies inside the trust region.} \tag{2.50}$$

2.3.4 Non-Linear Model Predictive Control through Newton based gradient techniques

In this section NMPC techniques called second-order methods will be reviewed [81]. These methods are often based on Newton's method for finding the roots of an equation, and therefore can also be called Newtonian methods. In this section and the rest of Section 2.3 below and Section 2.4 the objective function J is shown as depending only on the MVs $\mathbf{u}$ and the initial plant state $\mathbf{x}_0$ since all subsequent plant states $\mathbf{x}_k$ are determined by (2.41) given $\mathbf{u}_k$ and $\mathbf{x}_0$. The dependence on $\mathbf{x}_0$ is not shown in the equations but is implied in order to simplify the notation. The NMPC problem utilising gradient techniques can be defined as follows:

$$\begin{aligned} &\min_{\mathbf{u}} J(\mathbf{u}) \\ &\text{s.t. } h(\mathbf{u}) \geq 0 \\ &\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^n \end{aligned} \tag{2.51}$$

where $J(\mathbf{u})$ is the objective function, $h(\mathbf{u})$ is the inequality constraint function, $\mathbf{u}$ is a real valued vector of the optimisation (MV) variables, and $\mathbb{U}$ is the set of allowed MV values within the system constraints [61]. For the purposes of this research there are no equality constraints in the optimisation problem, therefore the discussion in the rest of this chapter is limited to only inequality constraints. It is also assumed that constraints are only applied to MVs and not to CVs or other state variables as well since that is not applicable to the research questions of this work. The set of points in the solution space that are feasible (not violating the constraints) is called the feasible set, Ω. It is very probable that a global minimum of the optimisation problem will lie on the boundary of this feasible set. In order to deal with this an auxiliary function called the Lagrangian, $L(\mathbf{u}, \lambda)$, is defined as

$$L(\mathbf{u}, \lambda) = J(\mathbf{u}) - \lambda^\top h(\mathbf{u}) \tag{2.52}$$

where the vector $\lambda \in \mathbb{R}^{d_h}$ is called the Lagrange multiplier [2, 61, 81, 83], and effectively punishes violations of the defined and considered constraints during optimisation, and d_h is the number of inequality constraints. In order for a local minimum for (2.51) to exist, the Karush-Kuhn-Tucker (KKT)

conditions need to hold:

$$\begin{aligned} \nabla_{\mathbf{u}} L(\mathbf{u}^\star, \lambda^\star) &= 0 \\ h_i(\mathbf{u}^\star) &\geq 0 \\ \lambda_i^\star &\geq 0 \\ \lambda_i^\star h_i(\mathbf{u}^\star) &= 0 \end{aligned} \tag{2.53}$$

where $\mathbf{u}^\star$ are the optimum MV values, $\lambda^\star$ is the optimum Lagrange vector, and i is the index of the i$^{\text{th}}$ element of the λ vector corresponding to the i$^{\text{th}}$ inequality constraint [61, 62, 81, 82, 83]. The problem then becomes one of minimizing (2.52) subject to (2.53).

For the constrained case, as in this work, there are two main second order methods often used to solve this problem, i.e. the active set method, and interior point methods [81]. For the active set method the focus is in particular on a so called working set, $\mathbf{W}_k$, which contain the current active inequality constraints. All of these active constraints are then converted to equality constraints, and the resulting non-linear programming problem is solved [61]. The working set is updated during each iteration of the algorithm [87], along with determining a search direction $\mathbf{d}_k$ (see explanation below (2.56)) as an approximation to the non-linear problem

$$\begin{aligned} &\min_{\mathbf{u}} J(\mathbf{u}) \\ &\text{s.t. } C_i(\mathbf{u}) = 0 \\ &\forall i \in \mathbf{W}_k \\ &\mathbf{u} \in U \subseteq \mathbb{R}^n \end{aligned} \tag{2.54}$$

where $\mathbf{C}(\mathbf{u})$ is a vector containing all the active inequality constraints during the particular iteration of the algorithm. Then the KKT conditions for the overall optimisation problem become

$$\mathbf{M}\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) = \begin{bmatrix} \nabla_{\mathbf{u}} L\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) \\ \mathbf{C}^{\mathbf{W}_k}(\mathbf{u}) \end{bmatrix} = 0 \tag{2.55}$$

where $\tilde{\lambda}^{\mathbf{W}_k}$ is the Lagrangian with zero elements for the indices of constraints that are outside the working set $\mathbf{W}_k$ [61]. When applying Newton's method to this problem, a step of the resulting iteration

is given by

$$\begin{bmatrix} \mathbf{u}^{new} \\ \lambda^{\mathbf{W}_k new} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \lambda^{\mathbf{W}_k} \end{bmatrix} - \left(\nabla_{\mathbf{u},\lambda^{\mathbf{W}_k}} \mathbf{M}\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) \right)^{-1} \begin{bmatrix} \nabla_{\mathbf{u}} L\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) \\ \mathbf{C}^{W_k}(\mathbf{u}) \end{bmatrix} \tag{2.56}$$

where the updated $\mathbf{d}_k$ search direction follows from the change in $\mathbf{u}$ and $\lambda^{\mathbf{W}_k}$, and with

$$\nabla_{\mathbf{u},\lambda^{\mathbf{W}_k}} \mathbf{M}\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) = \begin{bmatrix} \nabla_{\mathbf{uu}} L\left(\mathbf{u}, \tilde{\lambda}^{\mathbf{W}_k}\right) & -\nabla_{\mathbf{u}} \mathbf{C}^{\mathbf{W}_k}(\mathbf{u})^{\top} \\ \nabla_{\mathbf{u}} \mathbf{C}^{\mathbf{W}_k}(\mathbf{u}) & 0 \end{bmatrix}. \tag{2.57}$$

In (2.57) $\nabla_{\mathbf{uu}} L\left(\mathbf{u}, \tilde{\lambda}^{W_k}\right)$ is the Hessian matrix of the Lagrangian.

The other major class of solution methods based on Newton's method as a gradient based NMPC methodology, in addition to the discussed active set methods, is interior point methods [62, 65, 66, 81]. Interior point methods always generate a sequence that lies in the interior of the feasible set Ω. All the inequality constraints are transformed into equality constraints using slack variables [81, 82, 88, 89]. The optimisation problem therefore becomes larger, but the difficulty of dealing with constraints entering and exiting the working set, as in active set methods, is removed [61]. Approaches to solving the problem are classified as either continuation or barrier approaches [81]. The optimisation problem then becomes

$$\begin{aligned} &\min_{\mathbf{u}} J(\mathbf{u}) \\ &\text{with respect to } \mathbf{u} \in \mathbb{R}^{n_z}, \mathbf{s} \in \mathbb{R}^{n_s} \\ &\text{subject to } h_i(\mathbf{u}) - s_i = 0 \text{ and } s_i \geq 0 \\ &\forall i \in \mathbb{I} \end{aligned} \tag{2.58}$$

where $\mathbf{s} \in \mathbb{R}^{n_s}$ are the slack variables that modify the inactive inequality constraints so they become active. $\mathbb{I}$ is the set of inequality constraints in the problem. The Lagrangian for the problem is given by

$$L(\mathbf{u}, \mathbf{s}, \lambda) = J(\mathbf{u}) - \lambda^{\top}(h(\mathbf{u}) - \mathbf{s}). \tag{2.59}$$

The KKT conditions for the problem then becomes

$$\begin{aligned} \nabla_{\mathbf{u}} L(\mathbf{u},\mathbf{s},\lambda) &= 0 \\ \mathbf{S}\lambda - \mu\mathbf{e} &= 0 \\ \mathbf{h}(\mathbf{u}) - \mathbf{s} &= 0 \end{aligned} \tag{2.60}$$

with $1 \gg s_i \geq 0$, $1 \gg \lambda \geq 0$, and $1 \gg \mu \geq 0$. The vector $\mathbf{e} = (1,1,\ldots,1)^\top$ and matrix $\mathbf{S} = \text{diag}(\mathbf{s})$ are introduced to simplify the notation. Applying Newton's method to the above

$$\begin{bmatrix} \nabla_{\mathbf{uu}} L(\mathbf{u},\mathbf{s},\lambda) & 0 & -\nabla_{\mathbf{u}} h(\mathbf{u})^\top \\ 0 & \mathbf{W} & \mathbf{S} \\ \nabla_{\mathbf{u}} h(\mathbf{u}) & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d_u} \\ \mathbf{d_s} \\ \mathbf{d}_\lambda \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathbf{u}} f(\mathbf{u}) - \nabla_{\mathbf{u}} h(\mathbf{u})^\top \lambda \\ \mathbf{S}\lambda - \mu\mathbf{e} \\ h(\mathbf{u}) - \mathbf{s} \end{bmatrix} \tag{2.61}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{W} = \text{diag}(\lambda)$ [65]. By iteratively solving (2.61) and adjusting the changes in $\mathbf{u}$, $\mathbf{s}$ and λ ($\mathbf{d_u}$, $\mathbf{d_s}$ and $\mathbf{d}_\lambda$) through line search (2.49) or trust region methods (2.50), the NMPC problem can be solved [81].

In [65] both the Active Set algorithm (2.56) as well as the Interior Point algorithm (2.61) was used to solve the NMPC problem when controlling a distillation column. In [66] the Interior Point method is used in the background to solve the detailed NLP, while a sensitivity based approach is used online to minimise computational delay for distillation and power plant case studies.

2.3.5 Non-Linear Model Predictive Control through gradient descent

Another line search (2.49) gradient based optimisation approach closely related to the Newtonian methods described above, is the gradient descent method, also called the steepest descent method [81]. This method is a first order optimisation technique. The steepest descent gradient based algorithm is based on the Jacobian gradient vector of the system, and does not require the calculation of a Hessian matrix (as required by the Newtonian methods in Section 2.3.4). Explicit computation of the Hessian matrix of second derivatives can be a cumbersome, error-prone and computationally expensive process [81, 87, 90]. Furthermore, most second-order techniques require that the Jacobian of the Lagrangian must be invertible [61].

For the steepest descent method, the objective function can be decreased by moving in the direction of the negative gradient of the system [86] which is the direction in which the objective function decreases most rapidly [81]:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \eta \nabla_{\mathbf{u}} J(\mathbf{u}_k) \tag{2.62}$$

where $\mathbf{u}_k$ is the Manipulated Variables vector for controller iteration step k, η is the learning rate and $\nabla_{\mathbf{u}} J(\mathbf{u})$ is the Jacobian gradient vector of the objective function. The gradient vector is calculated as the partial derivative of the objective function with regards to each optimisation variable (or MV) in the system [86]. The value of η can be set as a small constant through tuning, or various line search techniques can be used to set the value [82, 86, 91].

Newtonian methods (Section 2.3.4) can converge faster than gradient descent methods as they use additional information about curvature of the objective function as contained in the Hessian of the system [86]. This is very useful near a minimum, but can be harmful near a saddle point (point of zero gradient but not a minimum nor a maximum - inflection point). Steepest descent however, is not attracted to saddle points, unless the gradient points at them [86], which can be a benefit.

Qian [92] shows how the steepest descent algorithm can be made to converge faster and more robustly by adding an exponentially decaying [93] momentum term to the iterative equations. The convergence of the gradient iterations are accelerated by accounting for the history of iterations when computing the ones to come [94].

$$\begin{aligned} \Delta\mathbf{u}_{k+1} &= \gamma \Delta\mathbf{u}_k + \eta \nabla_{\mathbf{u}} J(\mathbf{u}_k) \\ \mathbf{u}_{k+1} &= \mathbf{u}_k - \Delta\mathbf{u}_{k+1} \end{aligned} \tag{2.63}$$

where γ is the exponential decay fraction of the previous step for the momentum term. During each iteration of the control algorithm, the MVs are adjusted by the term calculated as per (2.63). This momentum gradient descent algorithm is also called Polyak's heavy ball algorithm in the literature [95]. Momentum can considerably accelerate convergence to a local minimum [93, 95]. If the objective function is convex, continuously differentiable, and its gradient is Lipschitz continuous with constant L then

$$0 \leq J(\mathbf{u}_2) - J(\mathbf{u}_1) - \langle \nabla_{\mathbf{u}} J(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle \leq \frac{L}{2} \|\mathbf{u}_1 - \mathbf{u}_2\|^2 \tag{2.64}$$

holds for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$, where $\langle \nabla_{\mathbf{u}} J(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle$ is the inner product of the two vectors separated by the comma [94]. If the objective function is also strongly convex with modulus $\mu > 0$ then

$$J(\mathbf{u}_2) - J(\mathbf{u}_1) - \langle \nabla_{\mathbf{u}} J(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle \geq \frac{\mu}{2} \|\mathbf{u}_1 - \mathbf{u}_2\|^2 \tag{2.65}$$

holds for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$ [94]. For the objective function satisfying these requirements, if $\mathbf{u}^\star$ is a minimum at $f^\star = J(\mathbf{u}^\star)$ then $f(\mathbf{u}_k) - f^\star$ associated to the sequence $\{\mathbf{u}_k\}$ in (2.62) converges linearly, and there exists scalar $q \in [0, 1)$ such that

$$\|\mathbf{u}_k - \mathbf{u}^\star\| \leq q^k \|\mathbf{u}_0 - \mathbf{u}^\star\| \tag{2.66}$$

where $k \in \mathbb{N}_0$ and q is called the convergence factor [94] and $\mathbf{u}_0$ is the initial value of the MVs. The optimal convergence factor for this objective function using (2.62) is [95]

$$q = \frac{L - \mu}{L + \mu}, \tag{2.67}$$

and the optimal learning rate is [95]

$$\eta = \frac{2}{L + \mu}. \tag{2.68}$$

Polyak [95] showed by doing local analysis close to the minimum that the optimal convergence factor becomes

$$q = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \tag{2.69}$$

when the momentum term is added as in (2.63). Polyak [95] further showed that this optimal convergence factor for the momentum equation is achieved when

$$\eta = \frac{4}{\left(\sqrt{L} + \sqrt{\mu}\right)^2} \tag{2.70}$$

and

$$\gamma = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2 \tag{2.71}$$

First-order gradient methods usually can start toward the minimum within the fist few iterations, but can then slow down as the optimum is approached. Second-order gradient methods have excellent convergence characteristics as the minimum is approached, but may have starting difficulties [96]. The slowdown of first-order methods is improved substantially by adding the momentum term and algorithm as in (2.63).

For the Steepest Descent Algorithm, constraints can be managed through the Projected Gradient method where the optimisation variables are projected onto the allowed hyperspace at all times usually through an orthogonal method [91]. For box constraints (constant, linear constraints) on the variables to be optimised, this method modifies (2.63) as follows [97]:

$$\mathbf{u}_{k+1} = \Pi\left(\mathbf{u}_k - \Delta\mathbf{u}_{k+1}\right) \tag{2.72}$$

where $\Pi_i\left(u_i\right)$ is the projection function for one element of the $\mathbf{u}$ vector and is defined as

$$\begin{aligned}\Pi_i\left(u_i\right) &= \min\left(h_i, \max\left(l_i, u_i\right)\right) \\ &= \begin{cases} h_i, & \text{if } u_i > h_i \\ u_i, & \text{if } l_i \leq u_i \leq h_i \\ l_i, & \text{if } u_i < l_i \end{cases}\end{aligned} \tag{2.73}$$

where l_i is the lower bound constraint on u_i, and h_i is the upper (higher) bound constraint on u_i [97].

In addition to steepest descent with momentum, there are other first order non-linear optimisation momentum methods used in industry and documented in the literature, e.g. the Nesterov accelerated gradient method [86, 93, 94].

Conjugate gradient first order gradient descent methods have been developed specifically for quadratic objective functions. Perhaps two of the most documented conjugate gradient algorithms in the literature for this class are the Fletcher Reeves and Polak Robiere methods [98]. These methods have originally been developed to smoothen out the trajectory of the steepest descent without momentum algorithm [86].

First order gradient descent methods can also utilise the Lagrangian to manage constraints similar to reviewed in Section 2.3.4 for second order methods as part of the objective function to be minimised, as opposed to using the projected gradient method [86]. This was the approach used by [99] in designing a NMPC controller to control a six wheeled unmanned ground vehicle.

Under the right conditions and convexity, as mentioned in the paragraphs above, gradient based NMPC techniques, including the steepest descent algorithm are guaranteed to converge to a global minimum [61, 82, 87, 91].

In [100] first order gradient descent was used for trajectory tracking for obstacle avoidance of autonomous road vehicles using NMPC. In [98] steepest descent without momentum, and the conjugate gradient methods were used to solve the NMPC problem for trajectory tracking of a four-wheeled omnidirectional mobile robot. In [90] the steepest descent method, as well as a second order method where the Hessian is approximated were used to analyse real-time NMPC theory. In [101] both the steepest descent algorithm, and a Newton based line search algorithm was used to control an autonomous surface craft. The particular algorithm was switched during each iteration step depending on the objective function hyperspace and gradient conditions.

2.4 HYBRID NON-LINEAR MODEL PREDICTIVE CONTROL

The control of non-linear plant models where the MVs are partly Boolean and partly continuous variables, are part of a collection of problems solved through control algorithms using Mixed Integer Non-linear Programming (MINLP) methods [83]. Real plants have both time-driven and event-driven dynamics, and control systems for such plants should preferably be able to accommodate both types [2]. MINLP problems combine the combinatorial difficulty of optimising over discrete variable sets with the challenges of handling constrained continuous non-linear functions [102]. These problems are usually much more complicated and expensive to solve than the corresponding continuous problem on account of the discrete nature of the variables and the combinatorial number of feasible solutions

which thus can exist [87]. The structure of these problems take the form as per (2.74)

$$\begin{aligned} &\min_{\mathbf{u}_R,\mathbf{u}_I} J(\mathbf{u}_R,\mathbf{u}_I) \\ &\text{s.t. } h(\mathbf{u}_R,\mathbf{u}_I) \geq 0 \\ &\mathbf{u}_R \in \mathbb{U}_R \subseteq \mathbb{R} \\ &\mathbf{u}_I \in \mathbb{U}_I \subseteq \mathbb{I} \end{aligned} \tag{2.74}$$

where $J(\mathbf{u}_R,\mathbf{u}_I)$ is the objective function, $h(\mathbf{u}_R,\mathbf{u}_I)$ is the inequality constraint function, $\mathbf{u}_R$ is a real valued vector, and $\mathbf{u}_I$ is an integer valued vector [83]. The elements of $\mathbb{U}_R$ are typically minimum and maximum values for the applicable real valued manipulated variables, and the elements of $\mathbb{U}_I$ are typically minimum and maximum values for the applicable discrete valued manipulated variables.

Each permutation of the discrete state variables could define a different plant state. These different plant states will each have different continuous dynamics as per the hybrid automata of each state [2]. If the number of possible discrete states that the plant can be in is large, (2.74) can be an NP-hard combinatorial problem [102].

2.4.1 Hybrid Non-Linear Model Predictive Control through Particle Swarm Optimisation

The PSO algorithm can be used to solve the continuous manipulated variables in a hybrid control setting as described in Sections 2.2.11 and 2.3.2. In [49] the authors adjusted their PSO algorithm for the Boolean or binary variable case, which can then be used to solve the discrete part of a Hybrid Non-Linear Model Predictive Control (HNMPC) problem. In the binary case, the trajectories of the speed of the particles indicate how the probability that a coordinate will take on a zero or one value, is changing over time. Equation (2.39) remains essentially the same for the binary case,

$$v_t = v_j + 2\theta(p_j - x_j) + 2\gamma(g_j - x_j), \tag{2.75}$$

except that p_j, x_j and g_j are now integers in $\{0,1\}$. v_t as calculated in (2.75) is constrained to yield v_j as a probability that must be constrained to the interval $[0.0,1.0]$ with

$$\begin{aligned} v_j &= S\left(\min\left(V_m, \max\left(-V_m, v_t\right)\right)\right) \\ &= \begin{cases} S\left(V_m\right), & \text{if } v_t > V_m \\ S\left(v_t\right), & \text{if } -V_m \leq v_t \leq V_m \\ S\left(-V_m\right), & \text{if } v_t < -V_m \end{cases} \end{aligned} \tag{2.76}$$

where $S(v)$ is the sigmoid function, and $V_m > 0$ is a hyper parameter of the system used to constrain the absolute value of v_t. The effect of this constraint is to increase the probability of x_j changing state after converging to an optimum [49] since (2.78) below could result in a different value for x_j if v_j is kept away from the limits of the $[0.0,1.0]$ interval. The sigmoid function is given by [86]

$$S(v) = \frac{1}{1+e^{-v}}. \tag{2.77}$$

After the transformation of (2.76), u_j can now be updated for the current iteration as a function of v_j and scalar r sampled from a uniform probability distribution on the real interval $[0.0,1.0]$

$$\begin{aligned} u_j\left(r, v_j\right) &= \int_{-\infty}^{r} \delta\left(s - v_j\right) ds \\ &= \begin{cases} 1 & \text{if } r > v_j \\ 0, & \text{if } r < v_j \end{cases} \end{aligned} \tag{2.78}$$

where $\delta(s)$ is the Dirac delta function and the integral in (2.78) is the Heavyside step function [49].

The algorithm described in (2.75) to (2.78) optimises an objective function where the optimisation variables (MVs) are discrete Boolean variables as opposed to real variables. This algorithm is more effective at finding good local optima than evolutionary discrete algorithms, while evolutionary algorithms (such as Genetic Algorithms) are more effective at jumping to regions far away from the recent local optima found to new regions that might contain global optima [49].

2.4.2 Hybrid Non-Linear Model Predictive Control through tree based methods

Most solution methods for MINLP apply some form of tree search. Two broad classes of methods can be distinguished in the literature - single-tree and multi-tree methods. If the integer variables in the problem are Boolean, the resulting state tree is a binary tree with two branches for every node. The nodes at level m in such a tree correspond to each of the 2^m non-linear optimisation problems that would have to be solved if each node was visited during a brute force overall solution algorithm [2].

Instead of solving all the possible objective function minimisation problems defined by each possible combination of Boolean variables, alternatively branch and bound methods can be used to solve the MINLP problem [2]. An example of a branch and bound tree graph is shown in Figure 2.4. The idea is to establish a partition of the feasible set into smaller subsets and then calculate certain bounds on the costs within some of the subsets to eliminate from further consideration other subsets. In this way the number of optimisation problems to be solved can be reduced.

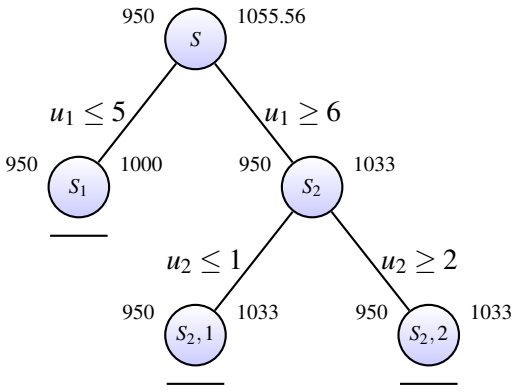

Figure 2.4. An example of a brand and bound algorithm tree graph.

The branch and bound algorithm starts at the root node (S in Figure 2.4) which is defined as the original MINLP problem (2.74) with the integer variables in an initialised state [87]. The algorithm searches a tree of which each node corresponds to a Non-Linear Programming (NLP) sub-problem [83]. The edges of the tree correspond to branching decisions. Both optimality and feasibility of the NLP sub-problems are used to make pruning decisions about the nodes in the tree. A node in the Branch and Bound tree is uniquely defined by a set of bounds (l,h) on the integer variables and corresponds to

the NLP [102]

$$\begin{aligned} &\min_{\mathbf{u}} J(\mathbf{u}) \\ &\text{subject to } c(\mathbf{u}) \leq 0 \\ &\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R} \\ &l_i \leq u_i \leq h_i, \quad \forall i \in \mathbb{I} \end{aligned} \tag{2.79}$$

The upper and lower bound constraints on the root node's integer variables are relaxed to $(-\infty, \infty)$. For each NLP node problem, if the solution u' of $NLP(l,h)$ is feasible but not integral (an integer), then two branches are created for any non-integral variable, e.g u'_i. Branching introduces two new NLP nodes, also referred to as child nodes of $NLP(l,h)$ [83]. In particular, bounds are initialised for the two new NLP problems as $(l^-,h^-) = (l,h)$ and $(l^+,h^+) = (l,h)$ and then the bounds are modified corresponding to the branching variable

$$\begin{aligned} h_i^- &= \lfloor u'_i \rfloor \text{ and} \\ l_i + &= \lceil u'_i \rceil, \end{aligned} \tag{2.80}$$

where $\lfloor u'_i \rfloor$ is the floor function and $\lceil u'_i \rceil$ is the ceiling function applied to u'_i [102]. The two new NLP problems are then defined as $NLP(l^-,h^-)$ and $NLP(l^+,h^+)$ [87].

The upper bound on the optimal solution of the global (2.74) is initialised to be $H \to \infty$. If any node $(NLP(l,h))$ is infeasible, then any node in the sub-tree rooted in this node is also infeasible, and can be pruned from the graph [83, 102]. If the solution $u^{(l,h)}$ is integral, it qualifies as a new incumbent solution if $J\left(u^{(l,h)}\right) < H$, and then $u^\star = u^{(l,h)}$ and $H = J\left(u^{(l,h)}\right)$. Otherwise this node can be pruned from the graph because its solution is dominated by the upper bound. If the lower bound on the optimal value of $(NLP(l,h))$ is dominated by the upper bound then the node can be pruned because there cannot any better integer solution in the sub-tree rooted at $(NLP(l,h))$ [102].

The above description of the branch and bound algorithm is completed by two strategic decisions that are part of the algorithm: The selection of the branching variable at each node, and the selection of which NLP problem to solve next [83, 87]. The ideal choice for the sequence of branching variables to evaluate would be to prioritise branching variables that minimise the size of the tree that is being created and needs to be iterated over by the hybrid controller. However, this is impractical since the

sequence of branching variables is not known a priori. A more realistic goal in selecting a branching variable is to choose a variable that maximises the increase in the lower bound of the objective function at a node [102].

The goal of the node selecting strategy in terms of which NLP problem should be solved next, is to find a good feasible solution quickly in order to reduce the global upper bound, and to prove the optimality of the current incumbent estimated optimal solution $u^\star$ by increasing the lower bound as quickly as possible [102]. Two popular node selecting strategies in the literature are depth first search and best bound search.

In depth first search the deepest node in the tree is selected. One advantage of this strategy is that it keeps the list of open nodes as small as possible. Another advantage is that this strategy minimises the change to subsequent NLP relaxations ($NLP(l,h)$) that are solved, because only the single bound is changed. A major disadvantage of depth first search is that it can exhibit poor performance if no upper bound is found, potentially exploring many nodes with a lower bound that is larger than the solution [102].

Best bound search selects the node with the best lower bound. Its advantage is that for a fixed sequence of branching decisions it minimises the number of nodes that are explored because all nodes that are explored would have been explored independently of the upper bound. On the other hand, the weaknesses of this strategy are that it may require significantly more memory to store the open problems, since normally an integer feasible solution is not found until the end of the search. This last point is particularly relevant for very large problems or if the solution time is limited such as in real-time applications, because the best bound search may fail to produce even a single feasible point [102]. Variations on both depth first search and best bound search have been proposed in the literature [102].

In addition to branch and bound other major algorithms in the literature used to solve the MINLP problem with are: Outer approximation, feasability approach, generalised outer approximation, and generalised cross decomposition [83].

Mayer *et al.* [103] used a linear approach with hybrid branch and bound methods to control a building cooling system supply. Cacchiani and DoAmbrosio [104] developed a branch and bound algorithm to

solve a unit commitment problem of a generation company whose aim is to find the optimal scheduling of a multi-unit pump-storage hydro power station. Although this problem is not a control problem, it is a case of a non-linear constrained MINLP. In [105] deterministic HNMPC with a tree based algorithm was used to control the pressure of two pressure vessels in series taking in air as the process input. In [105] the computational load was found to be a practical constraint to be addressed through relaxation techniques.

HNMPC is a viable strategy with which to solve control problems defined through MINLP models. Heuristic approaches, such as Genetic Algorithms, or Particle Swarm Optimisation, have been shown to be feasible optimisers in HNMPC controllers to control hybrid plants directly [41, 55, 75]. However, these stochastic methods have fewer global optimum guarantees, and less direct optimisation capability. Deterministic non-linear optimisation methods using tree graphs, however, are shown to be able to guarantee global optimality [105], and be able to perform direct optimisation.

2.5 ADAPTIVE NON-LINEAR CONTROL THROUGH REINFORCEMENT LEARNING

An alternative to designing a deterministic controller that will use the developed plant model to optimally control the plant, is to design an adaptive controller that will learn the best way to control the plant in simulation given the plant model. According to [106] the field of adaptive control can be roughly divided into four sub-fields: Model reference adaptive control, Reinforcement Learning (RL) adaptive control, neural networks for control, and fuzzy logic techniques. Of these four alternative approaches, the field of RL has recently experienced a resurgence, and will be briefly reviewed here as an alternative to solving the optimal control problem for the cooling water network.

Similar to how a child learns by interacting with his/her environment, a RL agent learns by interacting with its environment [84]. The field of RL is the sub-field of Machine Learning that studies how to use past data obtained from the agent interacting with the environment to enhance the future manipulation of a dynamical system [107]. RL has historically been focused more on solving optimisation problems where the input variables are discrete [108]. These discrete RL approaches could be utilised to solve the discrete component of the hybrid control problem in this research since it is also an integer

combinatorial optimisation problem. This would be more feasible still if the continuous control problem could also be solved by an RL algorithm.

Advancements have been made in problems with high-dimensional observation spaces, but with only discrete and low-dimensional action spaces [109]. Recent research has started to contribute more to solving problems with continuous and higher dimensional action spaces, similar to the cooling water network control problem addressed in this research [109].

In the RL literature, a number of broad categories can be identified for RL methods that will be able to solve the continuous control problem of this work. These categories are model-based Reinforcement Learning, approximate dynamic programming, and direct policy search [84, 107, 108].

Approximate dynamic programming approaches the RL problem by directly approximating the optimal control cost and then solving this with techniques from dynamic programming [107]. One of the goals of RL is to maximise the objective function normally referred to as R, the reward function [84]. The controller is normally referred as the policy. The control problem phrased in terms of RL and approximate dynamic programming is

$$\begin{aligned} \max_{\mathbf{u}} \mathbb{E}_{\mathbf{e}_k} \left[\sum_{k=0}^{N} R\left(\mathbf{x}_k, \mathbf{u}_k\right) + R_{N+1}\left(\mathbf{x}_{N+1}\right) \right] \\ \text{where } \mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{e}_k\right) \\ \left(\mathbf{x}_0, \mathbf{u}_0\right) = \left(\mathbf{x}, \mathbf{u}\right) \end{aligned} \tag{2.81}$$

where $\mathbf{u}_k$ is the output of the control policy at time k, $\mathbf{x}_k$ is the system state at time k, f is the system dynamics function, $\mathbf{e}_k$ is the external disturbances on the system over which the expectation is taken, $\mathbf{x}_0$ is the initial system state, and $\mathbf{u}_0$ is the initial control policy output, and N is the optimisation horizon [84, 107]. At the end of the optimisation horizon

$$Q_{N+1}\left(\mathbf{x}, \mathbf{u}\right) = R\left(\mathbf{x}_{N+1}\right) \tag{2.82}$$

and then recursively the theory of Dynamic Programming [84] yields

$$Q_k\left(\mathbf{x}, \mathbf{u}\right) = R\left(\mathbf{x}, \mathbf{u}\right) + \mathbb{E}_{\mathbf{e}}\left[\max_{\mathbf{u}'} Q_{k+1}\left(f\left(\mathbf{x}, \mathbf{u}, \mathbf{e}\right), \mathbf{u}'\right) \right]. \tag{2.83}$$

Equation (2.83) is also called the Bellman equation in the literature. Using the Bellman equation the Q-functions can be recursively defined, and then the optimal control output can be computed from any starting time and state $\mathbf{x}_0$ by applying the policy that maximises the right hand side of (2.83). Then for all time, the optimal policy only depends on the current state if the Markov property holds, and is given by

$$\mathbf{u}_k = \underset{\mathbf{u}}{\operatorname{argmax}}\, Q_k(\mathbf{x}_k, \mathbf{u}). \tag{2.84}$$

Approximate dynamic programming methods typically aim to compute and approximate these value functions from data [84], lately using especially deep learning neural network techniques [109], and normally discounting the future rewards [107].

In the direct policy Search class of techniques, the REINFORCE algorithm [107] is established as a primary method in the literature. It can be shown that if R is the reward function, and $\mathbf{u}$ is the action variable which is sampled from a policy probability distribution $p(\mathbf{u}, \theta)$ where θ is the policy parameters, and

$$J(\theta) = \mathbb{E}_{p(\mathbf{u},\theta)}[R(\mathbf{u})] \tag{2.85}$$

then

$$\nabla_\theta J(\theta) = \mathbb{E}_{p(\mathbf{u},\theta)}[R(\mathbf{u}) \nabla_\theta \log p(\mathbf{u}, \theta)] \tag{2.86}$$

is the gradient of J with respect to θ. The REINFORCE algorithm can therefore update the policy parameters θ through doing Stochastic Gradient Descent (SGD) [86] on the data stream that the RL agent will experience as it interacts with the plant model on-line [107]. Implementations of variations of this approach include the policy gradient and pure random search techniques [107].

Related to the above class of techniques, [109] have recently shown how the Actor Critic method [84] can be applied to large, continuous actions spaces. The parametrised (θ^μ) actor function $\mu(\mathbf{x}|\theta^\mu)$ specifies the current control policy by mapping states to specific actions. The critic function $Q(\mathbf{x}, \mathbf{u})$ with parameters θ^Q is learned using the Bellman equation (2.83). The policy gradient can be derived

by applying the chain rule as

$$\begin{aligned}\nabla_{\theta^\mu} J &\approx \mathbb{E}_{\mathbf{x}_k \sim \rho^\beta}\left[\nabla_{\theta^\mu} Q\left(\mathbf{x}, \mathbf{u} | \theta^Q\right)|_{\mathbf{x}=\mathbf{x}_k, \mathbf{u}=\mu(\mathbf{x}_k|\theta^\mu)}\right] \\ &= \mathbb{E}_{\mathbf{x}_k \sim \rho^\beta}\left[\nabla_u Q\left(\mathbf{x}, \mathbf{u} | \theta^Q\right)|_{\mathbf{x}=\mathbf{x}_k, \mathbf{u}=\mu(\mathbf{x}_k|\theta^\mu)} \nabla_{\theta^\mu} \mu\left(\mathbf{x}|\theta^\mu\right)|_{\mathbf{x}=\mathbf{x}_k}\right]\end{aligned} \tag{2.87}$$

where $\nabla_{\theta^\mu} J$ is the policy gradient [109] of the expected reward J that can be used to to update the parameters of the policy μ, where ρ^β is the distribution of the environment where the system state is sampled from in closed-loop.

RL techniques applied to continuous, high dimensional actions spaces are still relatively unexplored in the literature. Established methods often lead to models that are brittle [107]. If the Direct Policy Search (DPS) approaches such as per (2.85) are used, another concern is potentially high variance in the stochastic gradients of the expectation of the reward function [107] as per (2.86). Since non-linear function approximators are used (mainly deep neural networks), convergence of the optimal control problem, such as in (2.84), can no longer be guaranteed [109]. These and other challenges with applying RL to the control problem on large and continuous actions spaces have been addressed in the deep deterministic policy gradient method [109]. However, significant challenges remain and deep RL is a developing approach to the non-linear control problem.

2.6 CHAPTER SUMMARY

Literature has been reviewed in this chapter forming the foundation for methods to be developed in the next chapter that will enable dynamic models to be constructed for all the process unit operations in the plant to be modelled and controlled. These models will then be combined into a holistic model for the whole plant. More emphasis was put on the cooling tower since it is the most important, and the most complex unit operation in the cooling water network. Included in the literature models reviewed in this chapter is the important variables of ambient humidity, and the electrical power consumption of all the rotating equipment in the cooling water network.

NMPC is a newer technique compared to LMPC, but has a growing body of literature dedicated to it. Stability requirements for NMPC has also been documented by various authors. The NMPC problem can be solved using stochastic or deterministic methods. The deterministic class can be divided into different gradient based methods of which first order and second order methods are two

large sub divisions. Convergence criteria, rates and dependencies for the gradient techniques were also surveyed.

Theory as well as applications of NMPC methodologies for continuous non-linear dynamic systems are well represented in the literature. The same is true for hybrid control of linear dynamic systems (HLMPC). Descriptions of HNMPC for hybrid non-linear dynamic systems are much less frequent in the literature. This is even more the case for the field of cooling tower and cooling water systems, which are already under represented in the dynamic modelling and control literature. However, HNMPC techniques have been applied to some domains, and new techniques can be developed by combining methods from the HLMPC and NMPC domains, as well as from the MINLP domain.

Techniques from the growing domain of Reinforcement Learning that could be used to solve the non-linear control problem for the cooling water network, were also surveyed.

In this chapter the necessary literature was surveyed in order to motivate the methods used in this work. In the next two chapters, techniques to solve the research questions building on the methods, techniques and approaches surveyed in this chapter will be developed.

CHAPTER 3 DYNAMIC MODELLING OF A COOLING WATER NETWORK

3.1 CHAPTER OVERVIEW

In this chapter, the literature review documented in Chapter 2 is used as a foundation on which to develop techniques and models to solve the first research question. In particular, a first principles dynamic model is developed for industrial cooling towers and cooling water network.

The actual plant being modelled is described in Section 3.2, the complete process model is developed in Section 3.3 and documented in Section 3.4, and all hyper-parameters are shown. Section 3.5 deals with the methodology and results of the parameter estimation task. Methods and models described in Chapter 2 are referenced where appropriate to provide context.

Open-loop simulation results for various disturbance and input scenarios are described and discussed in Section 3.6. The essence of this chapter has already been published in the Journal of Process Control [1].

3.2 PROCESS DESCRIPTION

The cooling water circuit modelled in this research is graphically depicted in Figure 3.1 and the relevant design data is shown in Tables 3.1, 3.2, 3.3 and 3.4. The cooling capacity is supplied by 3 cooling towers in parallel. Cooling Water Return (CWR) is sprayed into each cooling tower at the top of the tower. At the bottom of the towers, the cooling water falls into a common cooling tower basin.

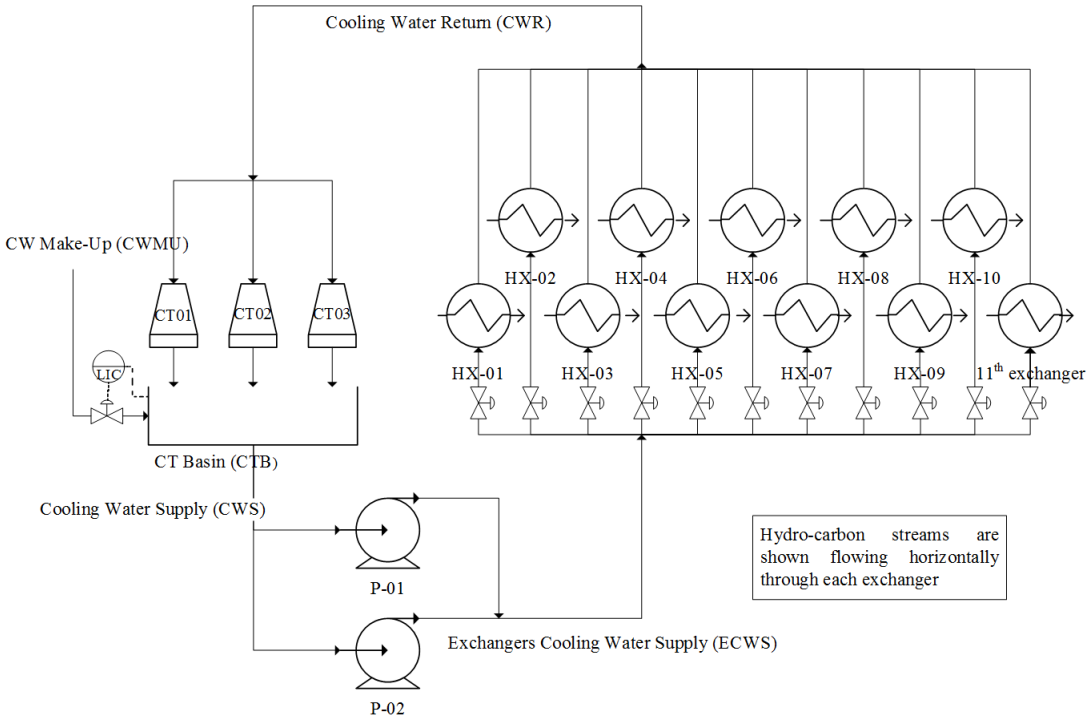

Figure 3.1. The cooling network and associated equipment.

Airflow through the cooling towers is induced by induction motor driven fans at the top of the cooling towers.

Each tower is designed for a maximum cooling water flow rate of 8,700 m^3/h, a heat duty of 101 MW, Cooling Water Return temperature of 45 °C, and a supply temperature of 35 °C. The wet bulb temperature is designed to be 31 °C. Evaporative losses from the cooling tower are designed to be 1.83% of the total cooling tower flow.

The cooling tower fans are 9.1 m in diameter, and rotate at a design speed of 120.1 rpm. The design power consumption of each fan is 137 kW.

Each pump is designed for an operating flow rate of 3,850 m^3/h and an operating discharge pressure of 5.3 barg. Each pump was designed to run at an operating point of 740 rpm at which it consumes 811 kW. However, pumps on the actual plant consume more power than this design power operating point.

133 parallel cooling water heat exchangers are fed with cooling water by the cooling towers and the

pumps. Most of the heat exchangers are used to cool down process hydrocarbon streams with the cooling water coming from the cooling water pumps. However, 73% of all the cooling water flows through the 10 biggest heat exchangers, and 27% flows through the remaining 123 smaller exchangers. After having passed through the 133 heat exchangers, the cooling water flows back to the cooling towers, completing the cooling water circuit. The 123 smaller exchangers were combined into a single 11^{th} exchanger (see Section 3.5) for modelling purposes.

In the real plant this model is based on, each heat exchanger has a hand valve upstream of it. In the model developed here, this hand valve is modelled as a control valve for simulation purposes and it will be included as a Manipulated Variable in the advanced control strategy developed in Chapter 4.

3.3 PLANT MODEL DEVELOPMENT

In this section the modelling work done is documented, with references to Chapter 2 and the literature, for each sub area of the model. Dynamic models for the major process unit operations (cooling towers, pumps, valves and heat exchangers) are derived separately, and the equations are developed to link the different operations.

3.3.1 Equations of state

In order to calculate the correct starting densities (2.4), internal energy (2.5), as well as a molar fraction of the liquid and/or gas phase present in a particular stream with a particular chemical composition at the start of simulations, the Peng Robinson equations of state of thermodynamics (2.3) were used (as described in Section 2.2.1). The application of this in the model developed was restricted to only some of the streams in the model, e.g. the flashing of the air flow into the cooling towers.

3.3.2 Cooling tower modelling

The cooling tower model has been constructed with the following initial assumptions:

1. The cooling tower operates under adiabatic thermodynamic conditions.

Table 3.1. Model variables and parameters.

Symbol	Description	Steady-state value	Units
A	Surface area of heat exchanger	Per exchanger	m^2
a_{ct}	Contact area of water surface per volume unit in the tower	399.9	m^2/m^3
a_h	Horizontal section area of cooling tower	207.4	m^2
a_{p_0}	Pump hydraulic curve constant 0	4,514.46	$Pa{\cdot}s^2$
a_{p_1}	Pump hydraulic curve constant 1	1,926.25	$Pa{\cdot}s^2/m^3$
a_{p_2}	Pump hydraulic curve constant 2	-45,255.62	$Pa{\cdot}s^2/m^6$
a_{seg}	The contact area of water surface per segment volume	116,951	m^2
c_a	Heat capacity of air	1,008.5	J/(kg·K)
C_{vCV}	Control Valve valve C_v coefficient	0.0040572	$m^3/(s{\cdot}\sqrt{Pa})$
C_{vCT}	Cooling Tower water spray nozzel C_v	121.2	$m^3/(s{\cdot}\sqrt{Pa})$
c_w	Heat capacity of water	4,185	J/(kg·K)
F	Mass flow rate	Per equipment	kg/s
$F_{a_{in}}$	Dry air flow rate into a cooling tower segment	Per segment	kg/s
$F_{a_{out}}$	Dry air flow rate out of a cooling tower segment	Per segment	kg/s
F_f	Mass flow rate through cooling tower fan	650.2	kg/s
$F_{w_{in}}$	CW flow rate into a cooling tower segment	Per segment	kg/s
$F_{w_{out}}$	CW flow rate out of a cooling tower segment	Per segment	kg/s
$F_{w_{evap}}$	Water evaporating rate per cooling tower segment	Per segment	kg/s
h_a	Cooling tower air phase heat transfer coefficient	0.6658	$W/(m^2{\cdot}K)$
h_{CTB}	Height of Cooling Tower Basin	13.2	m
h_D	Cooling tower air phase mass transfer coefficient	0.000657	$kg/(m^2{\cdot}s)$
h_w	Cooling tower water phase heat transfer coefficient	64.395	$W/(m^2{\cdot}K)$

Table 3.2. Model variables and parameters.

Symbol	Description	Steady-state value	Units
I_{MaxCTB}	Cooling tower basin maximum CW inventory	10,058,400	kg
k_{HX_i}	Heat Exchanger hydraulic flow coeficient for stream i	Per exchanger	kg/s$\cdot\sqrt{\frac{kg}{m^3 Pa}}$))
k_{P1}	Rotating equipment power transient response state space coefficient 1	1.48	dimensionless
k_{P2}	Rotating equipment power transient response state space coefficient 2	0.52	dimensionless
k_{P3}	Rotating equipment power transient response state space coefficient 3	1.48	dimensionless
n	Fan rotational speed	2.0	Revolutions per second
$p_{a_{out}}$	Pressure of moist air leaving the cooling tower	100212	Pa
p_{da}	Pressure of dry air coming into cooling tower	100,000	Pa
p_f	Pressure head generated by cooling tower fan	211.9	Pa
p_p	Pump discharge differential pressure	736,600	Pa
p_s	Standard atmospheric pressure	100,000	Pa
$p_{w_{CTin}}$	Pressure of CWR coming into cooling tower	250,000	Pa
$p_{w_{CTout}}$	Pressure of CW at the bottom of the cooling tower	100,000	Pa
p_{wvap}	Water vapour pressure at the interface	Per segment	Pa
P_f	Cooling tower fan power consumption	137,006.8	W
P_p	Pump power consumption	932,989.9	W
R_{EP}	Equal Percentage control valve characteristic constant	40	dimensionless
T_a	Cooling tower air segment temperature	Per segment	K
$T_{a_{in}}$	Temperature of dry air coming into cooling tower	298	K
$T_{a_{out}}$	Temperature of moist air leaving cooling tower	301.2	K
T_{int}	Cooling tower interface temperature	Per segment	K

Table 3.3. Model variables and parameters.

Symbol	Description	Steady-state value	Units
T_w	Cooling tower water temperature	Per segment	K
U	Heat exchanger overall heat transfer coefficient	Per exchanger	W/(m^2 ·K)
$V_{a_{seg}}$	Volume of air in one Cooling Tower segment	147.04	m^3
$V_{w_{seg}}$	Volume of water in one Cooling Tower segment	29.3	m^3
Y_a	Cooling tower air absolute humidity	Per segment	kg water per kg dry air
Y_{int}	Cooling tower interface absolute humidity	Per segment	kg water per kg dry air
ΔT_{lm}	Logarithmic Mean Temperature Difference (LMTD)	Per exchanger	K
λ	Latent heat of vaporisation of water	2,257,000	J/kg
ξ_a	Volume fraction of air in the cooling tower	0.50	Fraction
ξ_w	Volume fraction of water in the cooling tower	0.1	Fraction
ρ_a	Density of air at 25 °C	1.1694	kg/m^3
ρ_w	Density of water	1000	kg/m^3
$\tau_{p_{CTin}}$	Dynamic pressure time constant for cooling water return entering the cooling tower	60	seconds
$\tau_{p_{a_{out}}}$	Dynamic pressure time constant for moist air leaving cooling tower	60	seconds
$\tau_{\Delta p_{HX1}}$	Dynamic differential pressure time constant for stream 1 of heat exchanger	60	seconds
$\tau_{\Delta p_{HX2}}$	Dynamic differential pressure time constant for stream 2 of heat exchanger	60	seconds

Table 3.4. Model variables and parameters.

Symbol	Description	Steady-state value	Units
$\tau_{T_{HX2}}$	Dynamic temperature time constant for stream 2 of heat exchanger output	360	seconds
$\tau_{\Delta p_{CV}}$	Dynamic pressure time constant for pressure drop over control valve	10	seconds
ω	Pump rotational speed	12.33	Revolutions per second

2. The water and air streams are divided into 10 sections in the vertical dimension (see Figure 2.1). The water flow changes vertically between sections throughout the tower due to evaporation.
3. The density and specific heat of the water and dry air are constant accross the height of the cooling tower, since the change in temperatures are relatively small.
4. The fill packing is uniformly wet and in thermal equilibrium with the aqueous phase, and covers all 10 sections in the vertical dimension of the tower.
5. Water waste due to drifting is negligible. As per design this is 0.02% of the water flow through the tower. It is assumed zero for the purposes of this work.
6. There is no meaningful delay between the tower, the pumps and the heat exchangers.

The default ambient air temperature ($T_{a_{in}}$) is 25 °C, but it is changed as part of the model fit and step testing that were done. The ambient cooling tower inlet air density and heat capacity is adjusted using the Peng Robinson equations of state.

Similar to [12, 13] the cooling tower is divided into segments. The air mass and energy transfer equations are included in the model developed in this work, whereas the equations in [13] do not contain all the relevant detail for a numerical simulation of the cooling towers modelled in this work. The ambient temperature and humidity variables are both included in the model derived.

Similarly to [16], all heat and mass transfer is modelled as occurring between droplets and the surrounding air flow. The average droplet radius is assumed to be 1 mm, and the total volume of

packing per cooling tower is assumed 1161 m^3 as per the design data. Similarly to [25], the volume fraction in the cooling tower that is water, is assumed to be 10%. The contact area (a_{ct}) for heat and mass transfer can be calculated from these assumed values.

The cooling tower mass balance is modelled with the streams as indicated in Figure 2.1 for each cooling tower. It is assumed that if the cooling tower fan (see Section 3.3.3) is switched off, then the cool dry air flowing into the tower ($F_{a_{in}}$) is zero.

In Figure 2.1, the level in the basin below the cooling tower is shared between all 3 cooling towers, and is modelled as an integrator. The cooling tower height is divided into 10 segments of equal height and volume to be modelled (see Figure 2.1). The design dimensions of the tower were used to size each segment.

Similar to [23] the interface between the water bulk and the air bulk is assumed to be saturated. Each water droplet in the cooling tower is modelled as shown in Figure 2.2. Drops are aggregated into the 10 different segments with key variables describing the characteristics of each segment. The interface between the water and the air phases are modelled as shown in Figure 2.2.

The cooling tower steady-state, as well as the dynamic mass and energy balances for both the water and air side of the tower, are modelled according to (2.8) to (2.16).

The pressure drop over the cooling tower is focused on the water spray nozzle at the top of the tower. Similar to the control valves models documented in Section 2.2.7, the nozzle is modelled with an orifice equation (3.1) relating pressure drop (observed at 2.5 barg in the real plant process data) and flow ($F_{w_{CTin}}$) through the nozzle as

$$\frac{F_{w_{CTin}}}{\rho_w} = C_{vCT} \cdot \sqrt{\frac{p_{w_{CTin}} - p_{w_{CTout}}}{\rho_w}}, \tag{3.1}$$

where C_{vCT} is the C_v of the valve, and ρ_w is the density of the cooling water. A first-order transfer function is used to model the dynamic change of water pressure drop over the tower in response to a change in water mass flow rate (time constant $\tau_{p_{CTin}}$) to create a Hammerstein-like model (see (2.30) in Section 2.2.8).

3.3.3 Fan modelling

The non-linear static gain of the cooling tower fan models are modelled with the fan laws of (2.17), (2.18) and (2.19). The dynamic transient of the pressure head (2.18) created by the fan as a function of fan speed is modelled with a Hammerstein first-order response with time constant of $\tau_{p_{a_{out}}}$.

The cooling tower fan power and pump power were modelled with a similar dynamic power transient to that published by [35] and documented in Section 2.2.3. The power overshoot is important for optimisation and control from a hybrid system perspective, as it will deter fans or pumps from being switched on unnecessarily if power minimisation forms part of the control objectives.

Based on the above, the fan power is modelled as an under-damped second order system with two state variables, $X_{1_{Pf}}$ and $X_{2_{Pf}}$, where k_{P1}, k_{P2} and k_{P3} are constant coefficients, and $P_{f_{ss}}$ is the steady-state fan power consumption. The static and dynamic models are thus combined similar to a Hammerstein model consisting of a static valued non-linear element combined with a following linear dynamic element ((2.30) in Section 2.2.8).

$$\begin{aligned} X_{1_{Pf}} &= P_f \\ \frac{dX_{1_{Pf}}}{dt} &= X_{2_{Pf}} \\ \frac{dX_{2_{Pf}}}{dt} &= -k_{P1}X_{1_{Pf}} - k_{P2}X_{2_{Pf}} + k_{P3}P_{f_{ss}} \end{aligned} \tag{3.2}$$

P_f is then updated by integrating $\frac{dX_{1_{Pf}}}{dt}$.

3.3.4 Cooling tower basin modelling

Utilising (2.20), the mass inventory in the cooling tower basin is modelled through the following mass balance:

$$\frac{dI_{CTB}}{dt} = \sum_{i=1}^{3} F_{w_{CTout},i} + F_{makeup} - F_{CWS}, \tag{3.3}$$

where I_{CTB} is the basin mass inventory of water, $F_{w_{CTout},i}$ is the mass flow rate of water out of the ith cooling tower, F_{makeup} is the mass flow rate of water make-up, and F_{CWS} is the mass flow rate of water

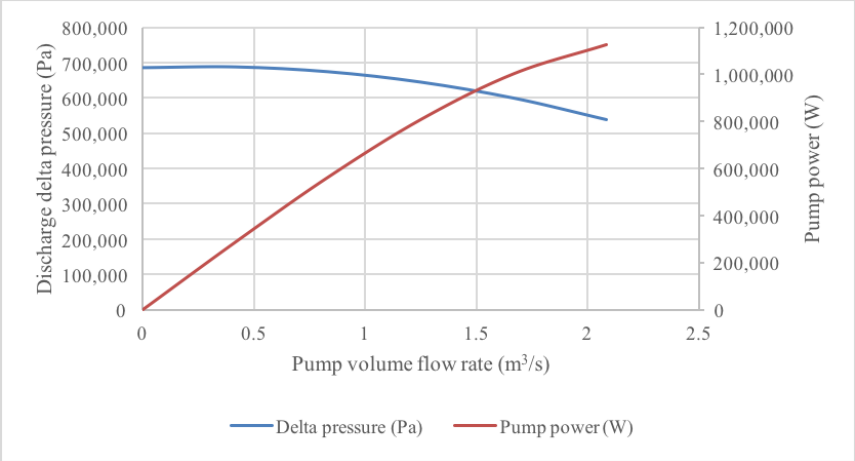

Figure 3.2. Pump curves - Δp_p vs. F and $P_{p_{ss}}$ vs. F.

from the basin as Cooling Water Supply.

Assuming perfect mixing of the water in the basin [25], and combining (2.21) and (2.22) the energy balance of the basin is given by

$$c_w I_{CTB} \frac{dT_{CTB}}{dt} = \sum_{i=1}^{3} c_w F_{w_{CTout},i} T_{w_{CTout},i} + c_w F_{makeup} T_{makeup} - c_w F_{CWS} T_{CTB}, \tag{3.4}$$

where the temperatures of the various streams ($T_{w_{CTout},i}$, T_{makeup} and T_{CTB}) are multiplied by their flow rates ($F_{w_{CTout}}$, F_{makeup} an F_{CWS}) and summed. F_{CWS} is determined by the unit operation down stream of the basin (pumps), which in turn depends on the pressure at the bottom of the basin. This pressure, p_{CTB}, is given by

$$p_{CTB} = \rho_w g h_{CTB} \frac{I_{CTB}}{I_{MaxCTB}} + p_s, \tag{3.5}$$

where ρ_w is the density of water, g is the gravitational constant, h_{CTB} is the design height of the cooling tower basin, and I_{MaxCTB} is the maximum mass inventory of the basin.

3.3.5 Pump modelling

For the pumps modelled in this work, (2.23) and (2.24) yield the pump curves displayed in Figure 3.2 when combined with data sheet information.

The pump power is modelled as a Hammerstein model with a first order dynamic response added as shown in the total system model in (3.35).

Equation (2.23) is solved iteratively with the other unit operations in the hydraulic cooling water circuit, such that each pressure drop converges to its equilibrium value to match the over-all system installed hydraulic curve.

3.3.6 Hydraulic network modelling

An algorithm was developed to model the distribution of mass flow of cooling water through the parallel network of heat exchangers utilising the principles gleaned from Section 2.2.6 and (2.27). The pressure drop (Δp_{e_i}) and mass flow rate (F_{e_i}) through each valve or exchanger (i^{th} element in branch) in each branch of the network, will follow the rule in (3.6).

$$\Delta p_{e_i} = k_i F_{e_i}^2 \tag{3.6}$$

Using (3.6), the pressure drop over the entire applicable branch Δp_j (j^{th} branch) is given by (3.7) where there are n valves or exchangers in the branch.

$$\Delta p_j = \sum_{i=1}^{n} k_i F_{e_i}^2 \tag{3.7}$$

$$\Delta p_j = F_j^2 \sum_{i=1}^{n} k_i \tag{3.8}$$

$$K_j = \sqrt{\sum_{i=1}^{n} k_i} \tag{3.9}$$

$$\sqrt{\Delta p_j} = F_j K_j \tag{3.10}$$

In (3.7), the mass flow is common to all terms and can be removed from the summation to be F_j as per (3.8). The different k values are then summed, and taken the square root of to become K_j for the j^{th}

branch in the network, out of a total of m branches as per (3.9). K_j can be calculated at each iteration of the simulation by using the square root of the pressure drop, and the mass flow as per (3.10).

The different Δp_j values for the network must be the same for all branches at all times, and the sum of the flows F_j through the network will be equal to the known total inflow to the network (F_{total}). With these constraints, a system of linear equations with m unknowns and m equations can be constructed and solved for the different flows through the network (3.11). Once the flows are known, the known K_j values can be used to calculate the pressure drop over the network.

$$\begin{bmatrix} K_1 & -K_2 & 0 & \cdots & 0 & 0 \\ 0 & K_2 & -K_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & -K_{m-1} & 0 \\ 0 & 0 & 0 & \ddots & K_{m-1} & -K_m \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_{m-1} \\ F_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ F_{total} \end{bmatrix} \tag{3.11}$$

3.3.7 Control valve modelling

The valves are modelled as per (2.28) [39]. Equal percentage flow characteristics are used for the control valves (2.29). The equal percentage control value characteristic constant is as per Table 3.2. The pressure drop over the valve is modelled to follow a Hammerstein first-order dynamic transient response with a time constant of $\tau_{\Delta p_{CV}}$ as per (3.39).

3.3.8 Heat exchanger modelling

For each exchanger, (3.12) will apply [3], with Q the heat duty of the exchanger, U the heat transfer rate coefficient, A the total heat exchange area for the exchanger, and ΔT_{lm} the LMTD for the exchanger as per (2.31).

$$Q = U \cdot A \cdot \Delta T_{lm} \tag{3.12}$$

For each exchanger, assuming that no phase changes occur over the exchanger, (3.13) will apply per fluid that is passing through the exchanger [3]. Here F is the mass flow rate of the fluid (stream 1 or 2 as per Figure 2.3), c is the heat capacity of the fluid and ΔT is the temperature difference between the inlet to the exchanger and the outlet for the particular stream. The heat capacity is assumed constant per stream over time since the changes in temperature are relatively small.

$$Q = F \cdot c \cdot \Delta T \tag{3.13}$$

By solving (2.31), (2.32), (3.12) and (3.13) simultaneously, closed form expressions for T_1^I, T_1^{II}, T_2^I and T_2^{II} can be obtained as shown in (3.46) and (3.47). These temperature steady-state equations are given dynamic Hammerstein transients (3.43) by adding a first-order time constant per exchanger ($\tau_{T_{HX1}}$ and $\tau_{T_{HX2}}$), and turning the equations into Ordinary Differential Equations (ODEs).

Pressure losses over the exchanger are divided between bundle losses and nozzle losses according to [18], and can be approximated as a quadratic relationship between flow and pressure drop scaled by a hydraulic coefficient for stream 1 and 2 of the exchanger (k_{HX_i}). Each exchanger's hydraulics are modelled assuming the mass flow rate is proportional to the square root of the pressure drop over the particular fluid path of the exchanger divided by the stream density, as per (3.14).

$$F_i = k_{HX_i} \cdot \sqrt{\frac{\Delta p_i}{\rho_i}} \tag{3.14}$$

The Hammerstein dynamic transients of the pressure drops of the two streams over the heat exchanger, are modelled with first order time constants $\tau_{\Delta p_{HX1}}$ and $\tau_{\Delta p_{HX2}}$ as per (3.42).

3.4 TOTAL SYSTEM MODEL

Pressure drops over the heat exchangers and the cooling towers will result in a particular system resistance curve as experienced by the pumps. The pump flow produced will then be at the intersection of the pump, and the network system curve. The system of equations, as developed for the total plant, will find this intersection through iteration and convergence.

The blowdown flow sent away from the cooling water circuit is on average 0.7% of the total flow in circulation, and is made-up with the make-up flow into the cooling tower basin. For the purposes

of this work, the blowdown rate is assumed to be zero, resulting in the make-up flow in the model focusing on replacing evaporative losses, and not blowdown as well. The make-up flow rate is used in a Proportional-Integral-Derivative (PID) control loop that controls the level in the cooling tower basin.

Numeric integration of the continuous equations are used. The Runge-Kutta 4th order method (2.36) with fixed step size (200 ms) was used to numerically solve the stiff differential equations (section 2.2.10) of the cooling tower models. Euler integration (2.34) was used to solve the rest of the plant's differential equations since these are not stiff and can be solved faster by a simpler method.

Fixed step size methods are preferred, as opposed to variable step size methods like Klopfenstein methods (2.38), since all the unit operations of the cooling water circuit will be solved at the same time step. Numerical integration is simpler with a fixed step size when performed over the simulation horizon for open-loop simulation, and over the prediction horizon in the internal model objective function of the controller (see Chapter 4). The chosen step size is shown through empirical simulation to result in the model converging for all experimental simulations performed, as well as for the internal model for the NMPC controller.

The complete model equations are given below for the model as derived in the sub sections above. State-space and algebraic equations are grouped per section of the plant.

3.4.1 Cooling tower model

In the below equations the subscript j refers to the jth section of the ith cooling tower model. Steady-state values have ss as a subscript.

3.4.1.1 State space model

$$\frac{dp_{CTin}}{dt} = \frac{-1}{\tau_{p_{CTin}}} p_{CTin} + \frac{1}{\tau_{p_{CTin}}} p_{CTinss} \tag{3.15}$$

For each segment j:

$$\frac{dT_{w,j}}{dt} = \frac{F_{w,j+1}T_{w,j+1} - F_{w,j}T_{w,j}}{\rho_w V_{w_{seg}}} - \frac{h_w a_{seg}\left(T_{w,j} - T_{int,j}\right)}{c_w \rho_w V_{w_{seg}}} \tag{3.16}$$

$$\frac{dT_{a,j}}{dt} = \frac{F_{a,j-1}T_{a,j-1} - F_{a,j}T_{a,j}}{\rho_a V_{a_{seg}}} - \frac{h_a a_{seg}\left(T_{a,j} - T_{int,j}\right)}{c_a \rho_a V_{a_{seg}}} \tag{3.17}$$

$$\frac{dY_{a,j}}{dt} = \frac{F_{a,j-1}Y_{a,j-1} + F_{w_{evap}} - F_{a,j}Y_{a,j}}{\rho_a V_{a_{seg}}} \tag{3.18}$$

3.4.1.2 Algebraic equations

$$h_a = c_a h_D \tag{3.19}$$

$$p_{w_{CTin}} = p_{w_{CTout}} + \rho_w \left(\frac{F_{w_{CTin}}}{\rho_w C_{vCV}}\right)^2 \tag{3.20}$$

$$p_{ma_{out}} = p_{ma_{in}} + p_{f1} \tag{3.21}$$

For each segment j:

$$T_{int,j} = \frac{-\left(h_D \lambda a_{ct}\left(Y_{int,j} - Y_{a,j}\right) - h_a a_{ct} T_{a,j} - h_w a_{ct} T_{w,j}\right)}{h_a a_{ct} + h_w a_{ct}} \tag{3.22}$$

$$Y_{int,j} = 0.622 \frac{P_{wvap,j}}{P_s - P_{wvap,j}} \tag{3.23}$$

$$F_{w_{evap},j} = h_D a_{segment}\left(Y_{int,j} - Y_{a,j}\right) \tag{3.24}$$

$$F_{w_{out},j} = F_{w_{in},j+1} - F_{w_{evap},j} \tag{3.25}$$

3.4.2 Fan model

3.4.2.1 State space model

$$\frac{dp_{a_{out}}}{dt} = \frac{-1}{\tau_{p_{a_{out}}}} p_{a_{out}} + \frac{1}{\tau_{p_{a_{out}}}} p_{a_{outss}} \tag{3.26}$$

$$\begin{aligned} \frac{dX_{1_{Pf}}}{dt} &= X_{2_{Pf}} \\ \frac{dX_{2_{Pf}}}{dt} &= -k_{P1}X_{1_{Pf}} - k_{P2}X_{2_{Pf}} + k_{P3}P_{f_{ss}} \end{aligned} \tag{3.27}$$

3.4.2.2 Algebraic equations

$$F_{f1} = F_{f0}\left(\frac{n_1}{n_0}\right) \tag{3.28}$$

$$p_{f1} = p_{f0}\left(\frac{n_1}{n_0}\right)^2 \tag{3.29}$$

$$P_{f1} = P_{f0}\left(\frac{n_1}{n_0}\right)^3 \tag{3.30}$$

$$P_{fan} = P_{fx1} \tag{3.31}$$

3.4.3 Cooling tower basin model

3.4.3.1 State space model

$$\frac{dI_{CTB}}{dt} = \sum_{i=1}^{3} F_{w_{CTout},i} + F_{makeup} - F_{CWS} \tag{3.32}$$

$$\frac{dT_{CTB}}{dt} = \frac{\sum_{i=1}^{3} F_{w_{CTout},i} T_{w_{CTout},i} + F_{makeup} T_{makeup} - F_{CWS} T_{CTB}}{I_{CTB}} \tag{3.33}$$

3.4.3.2 Algebraic equations

$$p_{CTB} = p_{CWS} = \rho_w g h_{CTB} \frac{I_{CTB}}{I_{MaxCTB}} + p_s \tag{3.34}$$

3.4.4 Splitter and mixer around pumps model

The flow through each pump is determined by the pressure increase over the pump, which is defined by (3.36).

3.4.5 Pump model

3.4.5.1 State space model

For each pump:

$$\begin{aligned} \frac{dP_{px1}}{dt} &= P_{px2} \\ \frac{dP_{px2}}{dt} &= -k_{P1}P_{px1} - k_{P2}P_{px2} + k_{P3}P_{p_{ss}} \end{aligned} \tag{3.35}$$

P_{px1} is the current pump power, and is solved by integrating $\frac{dP_{px1}}{dt}$.

3.4.5.2 Algebraic equations

For each pump:

$$\Delta p_p = p_{ECWS} - p_{CWS} \tag{3.36}$$

$$F_p = \rho_w \frac{-a_{p_1}\omega + \sqrt{(a_{p_1}\omega)^2 - 4a_{p_2}(a_{p_0}\omega^2 - \Delta p_p)}}{2a_{p_2}} \tag{3.37}$$

$$P_{p_{ss}} = F\frac{\Delta p_p}{\rho_w} \tag{3.38}$$

3.4.6 Heat exchanger network splitter and mixer model

The flow through each branch in the heat exchanger network, as well as the pressure upstream of the network, are modelled by the algorithm given in Section 3.3.6, and (3.10) and (3.11).

3.4.7 Control valve model

3.4.7.1 State space model

$$\frac{d\Delta p_{CV}}{dt} = \frac{-1}{\tau_{\Delta p_{CV}}}\Delta p_{CV} + \frac{1}{\tau_{\Delta p_{CV}}}\Delta p_{CV_{ss}} \tag{3.39}$$

3.4.7.2 Algebraic equations

$$f(V_{OP}) = (R_{EP})^{V_{OP}-1} \tag{3.40}$$

$$\Delta p_{CT_{ss}} = g_s \left(\frac{F}{\rho_w C_{vCV} \cdot f(V_{OP})} \right)^2 \tag{3.41}$$

3.4.8 Heat exchanger model

For each heat exchanger (i refers to either stream 1 or stream 2 of the heat exchanger):

3.4.8.1 State space model

$$\frac{d\Delta p_{HXi}}{dt} = \frac{-1}{\tau_{\Delta p_{HXi}}}\Delta p_{HXi} + \frac{1}{\tau_{\Delta p_{HXi}}}\Delta p_{HXiss} \tag{3.42}$$

$$\frac{dT_{HXi}}{dt} = \frac{-1}{\tau_{T_{HXi}}}T_{HXi} + \frac{1}{\tau_{T_{HXi}}}T_{HXiss} \tag{3.43}$$

3.4.8.2 Algebraic equations

$$\Delta p_i = \rho_i \left(\frac{F_i}{k_{HX_i}} \right)^2 \tag{3.44}$$

$$K_{HX} = UA \left(\frac{1}{F_1 c_1} - \frac{1}{F_2 c_2} \right) \tag{3.45}$$

$$T_{HX1_{outss}} = \frac{F_2 c_2 \left(T_{1_{in}} + e^{K_{HX}} T_{2_{in}} - T_{2_{in}} \right) - F_1 c_1 T_{1_{in}}}{F_2 c_2 e^{K_{HX}} - F_1 c_1} \tag{3.46}$$

$$T_{HX2_{outss}} = \frac{F_1 c_1 \left(T_{1_{in}} - T_{1_{outss}} \right) + F_2 c_2 T_{2_{in}}}{F_2 c_2} \tag{3.47}$$

3.5 PARAMETER ESTIMATION

A data driven approach to fitting each parameter of the model as derived in the previous sections, is applied. The PSO algorithm, as per (2.39) and (2.40), being a global optimiser, is effective at finding a global optimum for the problem of fitting the parameters in the model to the plant data.

In this work, a population of 30 particles was used. The cooling tower hydraulic flow coefficient (3.20) is estimated from the measured cooling tower inlet pressure data.

In order to increase the accuracy of the individual equipment models, the approach taken here was to fit each piece of equipment separately to plant data where possible, and then combine all the fitted equipment into the total model. The remaining parameters were then estimated. Initial values for all parameters were obtained from design data and equipment data sheets (Tables 3.1 to 3.4). The parameters to be fitted were then identified as those for which plant data is available, as well as the ones that are either not available in the data sheets to begin with, or that are most likely to be different in the actual plant from the data sheet values. This meant that the parameters that could be fitted per equipment were those of the cooling towers, and certain heat exchanger parameters. The parameters

to be fitted with the total model after the individual equipment were fitted were the remaining heat exchanger parameters (see below for more details).

For the cooling towers, in addition to the hydraulic pressure drop coefficient, the parameters to be fitted are the mass transfer coefficient of air (h_D), the heat transfer coefficient of the water (h_w), and the heat transfer coefficient of the air (h_a). These parameters were fitted for the cooling tower, using cooling tower design sheet data, and then adjusted by fitting them to the actual data. The fitted parameters are shown in Table 3.5.

Table 3.5. Cooling tower coefficients fitted to plant data.

Parameter	**Description**	**Fitted to plant data**	**Units**
h_D	Cooling Tower mass transfer coefficient	0.000657	kg/(m^2 ·s)
h_w	Water heat transfer coefficient in cooling tower	64.395	W/(m^2 ·K)
h_a	Air heat transfer coefficient in cooling tower	0.6658	W/(m^2 ·K)

The cooling tower model was fitted by comparing the simulated CWS temperature (T_{CTB} in (3.33)) to actual plant data for two periods, as can be seen in Figure 3.3 and 3.4. The first period is from 12 am at midnight to 12 pm noon. For the second period the simulation was run over a 6-hour period from 9:40 pm. The CWS temperature is the best available variable to validate the model with as it affects the plant the most, is the only important output variable that is measured, and is a key variable used in calculating the cooling tower efficiency with (2.7). The model responds as expected to ambient temperature, humidity, as well as Cooling Water Return ($T_{w,j+1}$ in (3.16) for the top segment) temperature changes. The variation in the ambient temperature ($T_{a,j-1}$ in (3.17) for the bottom segment) was 12 K, for the ambient humidity ($Y_{a,j-1}$ in (3.18) for the bottom segment) it was a 40% relative humidity change, and the CWR temperature varied 5 K over the testing period. The R^2 coefficient of determination for the cooling tower fit is 0.76.

Since more instruments providing plant data are available for the 10 bigger exchangers, each of these exchangers were modelled as separate process equipment. The 123 smaller exchangers were combined into a single 11$^{\text{th}}$ exchanger in order to simplify the model (see Table 3.7). For the exchangers where hydrocarbon inlet and outlet stream temperatures and flow rates are historised, the U value was fitted for each exchanger using the applicable hydrocarbon heat capacity values. Heat exchanger heat exchange

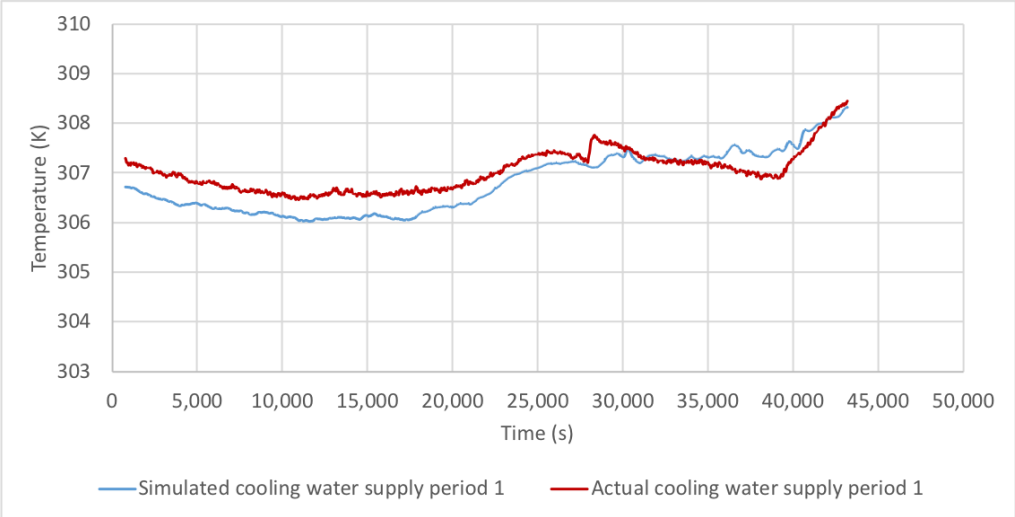

Figure 3.3. Cooling water supply temperature (T_{CWS}) simulation vs. actual for period 1.

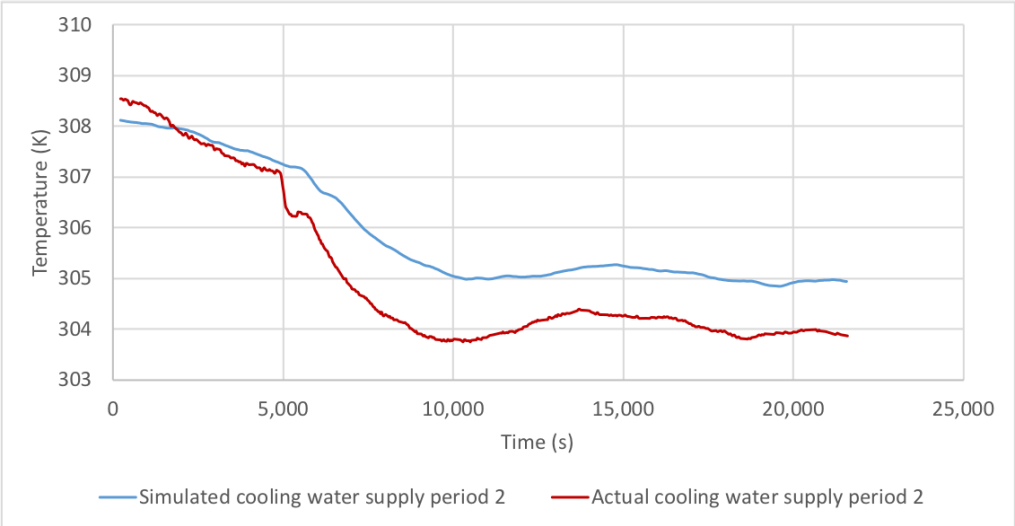

Figure 3.4. Cooling water supply temperature (T_{CWS}) simulation vs. actual for period 2.

surface area values (A in Table 3.1) were obtained from the data sheets of each exchanger. The dynamic temperature time constant for the hydrocarbon side was also fitted where temperatures are historised (see Table 3.6). The R^2 coefficient of determination for each heat exchanger fit is also included in Table 3.6. For exchangers where historised measured data were not available, datasheet information was used (see Table 3.7). In [5], the flow rate to each exchanger was measured by an electromagnetic flow meter, and this data was used to validate the model that was developed. The same approach is taken in this research: Measured cooling water flow rates to the individual heat exchangers are not available, so an electromagnetic flow meter was used to measure cooling water flow rates by hand.

These measurements were used to calculate the ratios between flows to each exchanger, and the total measured flow rates were scaled accordingly to determine the cooling water flow to each exchanger. These flow rates were used to fit the k_{HX_2} hydraulic coefficients on the cooling water side of each exchanger (3.44).

Table 3.6. Heat exchanger parameters fitted to design, and more complete plant data.

Exchanger	**UA design**	**UA fitted**	$\tau_{T_{HX1}}$ **fitted**	k_{HX_2} **fitted**	R^2 **of fit**
HX-06	226,233	187,535.3	269.65	2.06	0.99
HX-07	125,457	75,624	208.64	0.97	0.99
HX-04	534,570	535,954	216	8.91	0.96
HX-09	NA	535,287	81.136	12.96	0.99
HX-05	81,760	35,928	538.1	4.99	0.23

Table 3.7. Heat exchanger parameters fitted to design, and limited plant data.

Exchanger	**UA design**	k_{HX_2} **fitted**
HX-01	142,639	10.95
HX-02	142,639	13.10
HX-03	533,096	5.01
HX-08	301,030.1	11.25
HX-10	149,693.4	26.93
11^{th} HX	514,076.2	79.17

Simulated and actual process temperatures for the exchangers where hydrocarbon temperatures and flow rates are measured, are shown in Figures 3.5 to 3.8. Actual and simulated data are compared for between 10 and 24 hours per exchanger. The models for exchangers HX-06, HX-07, HX-04 and HX-09 follow the actual data very closely (within 0.5 K over the operating envelope shown).

For exchanger HX-05 there is an error of 1.3 K in the worst case (Figure 3.9). The hydrocarbon flow rate into HX-05 is not measured but is calculated from a mass balance of other streams that can predict the HX-05 flow rate. The lower accuracy for the HX-05 model could be due to an inaccuracy or unmeasured disturbance in this upstream mass balance model. However, the HX-05 model is considered accurate enough for the purposes of this work.

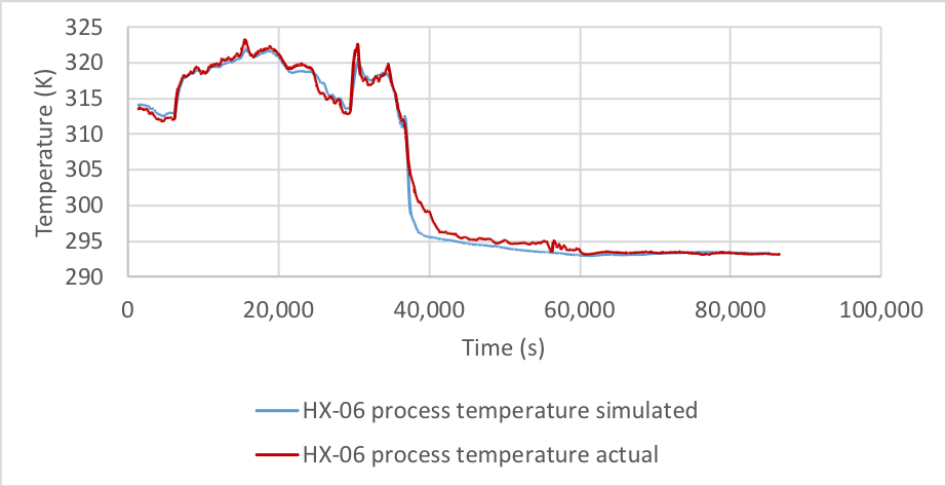

Figure 3.5. HX-06 process outlet temperature (T_{HX06}) simulated and actual data.

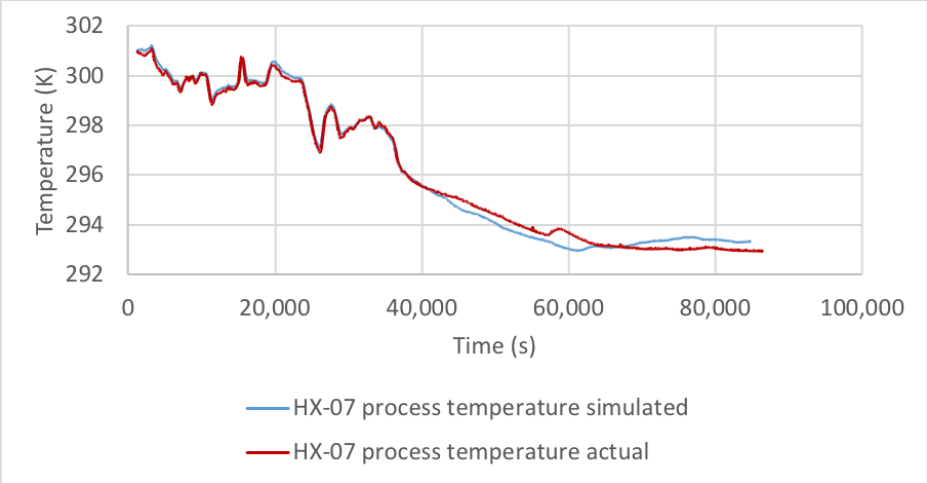

Figure 3.6. HX-07 process outlet temperature (T_{HX07}) simulated and actual data.

3.6 PLANT MODEL SIMULATION

In this section, the models developed in Sections 3.3 and 3.4 are simulated in open-loop to observe the response of the model outputs to steps in the model inputs. With a view to Chapter 4 on the the closed-loop simulation results and the designed controllers, the most important measured model outputs are called Controlled Variables (CVs). Input variables that can be readily actuated are called Manipulated Variables (MVs), and the environment variables that are measured disturbances are called Disturbance Variables (DVs).

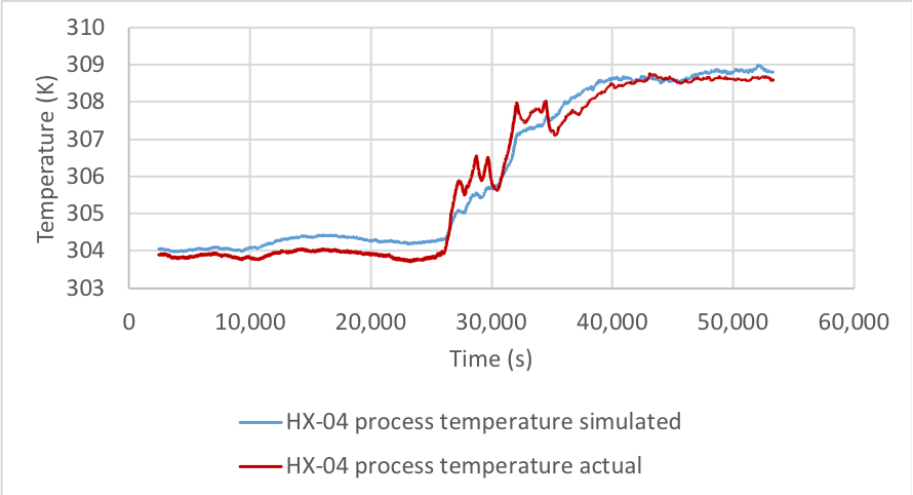

Figure 3.7. HX-04 process outlet temperature (T_{HX04}) simulated and actual data.

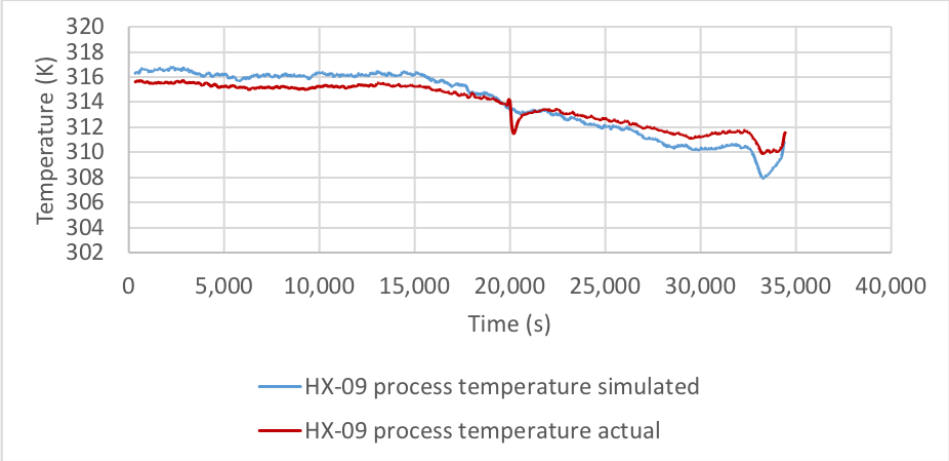

Figure 3.8. HX-09 process outlet temperature (T_{HX09}) simulated and actual data.

The MVs and DVs are stepped one at a time while the remaining model inputs are held constant. The model is run with a sampling period of 200 milliseconds. A high sampling frequency is needed due to the stiff nature of the cooling tower state space model (see Section 3.3.2 and Section 3.4) in order to ensure stability and convergence. The MVs and DVs stepped in this section, and the CVs for which responses were recorded are shown in Table 3.8.

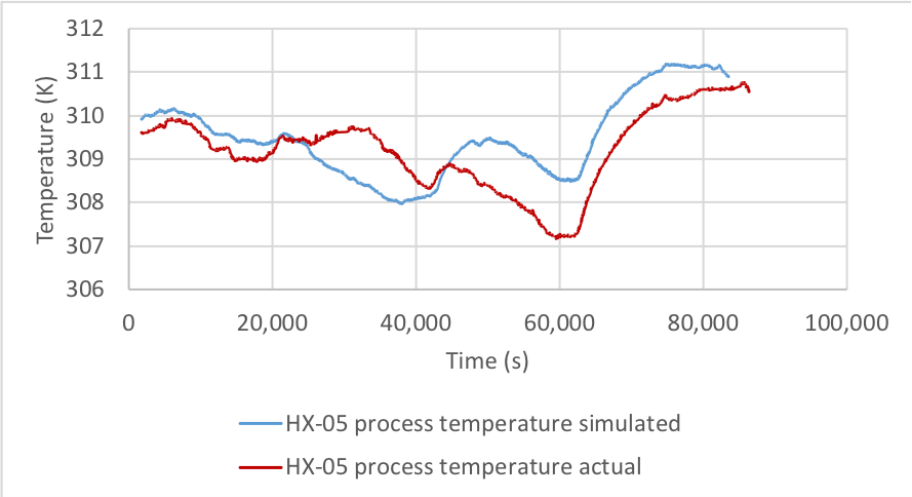

Figure 3.9. HX-05 process outlet temperature (T_{HX05}) simulated and actual data.

3.6.1 Fan speed MV steps

[110] shows that decreases in the L/G ratio (Liquid to Gas ratio, in this research water to air flow ratio) results in an increase in the efficiency of the cooling tower. The air flow in the model used in this work, can be manipulated by changing the fan speed (n) according to (2.17). Four equal steps of 0.6 rps were made every 4 hours in the cooling tower fan speed from 2.5 rps to 0.1 rps as shown in Figure 3.10. The resulting impact on cooling tower fan power (P_f) is shown in Figure 3.11, and the impact on process temperature outlet of HX-03, cooling water return temperature and cooling water supply temperature are shown in Figure 3.12. HX-03's output temperature prediction showed a particularly large gain when responding to steps elsewhere in the plant, and was therefore chosen for these figures. The impact of changes in the fan speed on the plant temperatures at the normal operating speed of 2 rps, is much less than when the speed moves to below 0.7 rps. This suggests that the plant could be optimised by running the fans at lower speed where less power is consumed. The oscillation of the fan power in Figure 3.11 is not clearly visible since it reaches steady-state within 40 seconds.

3.6.2 Ambient temperature DV steps

Figure 3.13 shows the impact of steps in the DV ambient temperature ($T_{a_{in}}$) on process temperature (HX-03; see Figure 3.1), Cooling Water Return temperature, and the Cooling Water Supply temperature.

Table 3.8. Variables used in plant model step tests.

Variable name/description	**Variable in model**	**Variable type**
Cooling tower fan speed	n	MV
Pump speed	ω	MV
Valve opening at HX-02	V_{OP_2}	MV
Ambient temperature	$T_{a_{in}}$	DV
Ambient humidity	$Y_{a_{in}}$	DV
Cooling tower fan power	P_f	CV
Pump power	P_p	CV
Heat exchanger 2 outlet temperature	T_{HX02}	CV
Heat exchanger 3 outlet temperature	T_{HX03}	CV
Heat exchanger 6 outlet temperature	T_{HX06}	CV
Cooling water return temperature	T_{CWR}	CV
Cooling water supply temperature	T_{CWS}	CV
Total mass flow rate in circuit	F_{CWS}	CV
Make-up flow rate	F_{CWMU}	CV
CW flow rate through HX-02	F_{valve_2}	CV

The step increases are simulated to occur every 4 hours, starting at 283 K and going up to 313 K. This DV impact is also reflected in the validation data in Figure 3.3.

3.6.3 Fan shut down (Fan speed MV)

The 3 cooling tower fans were shut down (fan speed n set to zero) one by one in a hybrid control fashion and the impact on the plant temperatures simulated (see Figure 3.14). The lack of cooling capacity with fewer towers available, increases non-linearly as the fraction of cooling capacity lost with each fan trip, increases. The settling time also increases non-linearly with the linear drop in cooling tower fans running, with the total cooling water network temperature dynamics becoming much slower as less fan draft is available.

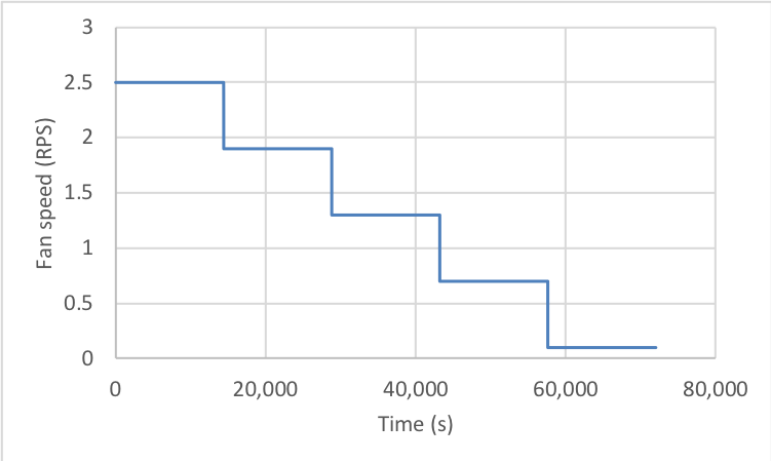

Figure 3.10. Cooling tower fan speed (n) as stepped.

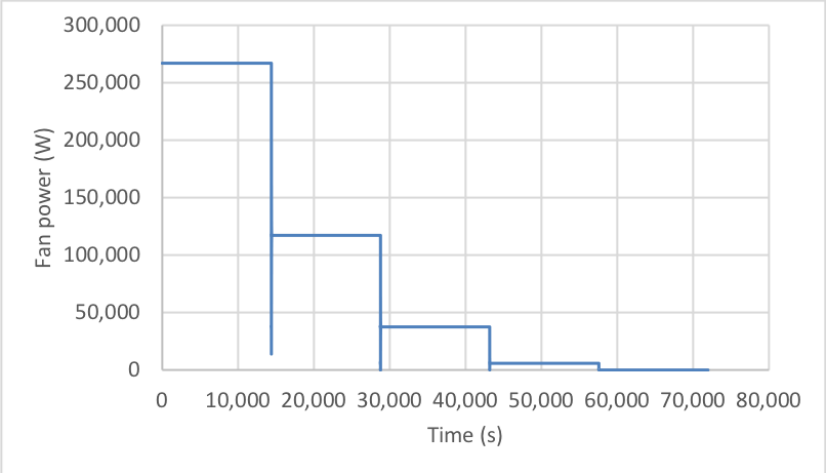

Figure 3.11. Cooling tower fan power (P_f) response when speed stepped.

3.6.4 Pump speed MV steps

The pump speed (ω) was stepped hourly in increments of 1 rps from 10.33 rps to 14.33 rps, and the response of the pump power (P_p), total flow through the circuit (F_{CWS}), and the plant temperatures were simulated (see Figure 3.15 to Figure 3.17). These limits were chosen to be within a window of 15% around the normal operating point of the pump speed. The non-linear nature of the pump power consumption to pump speed relationship is evident. Further, it can be seen from Figure 3.17 that the pump speed is a viable MV along with the other MVs for control and optimisation purposes. Larger pump speeds result in higher heat transfer rates in the heat exchangers (depending on each heat exchanger's operating point) resulting in cooler hydrocarbon streams. In addition, it can be seen that

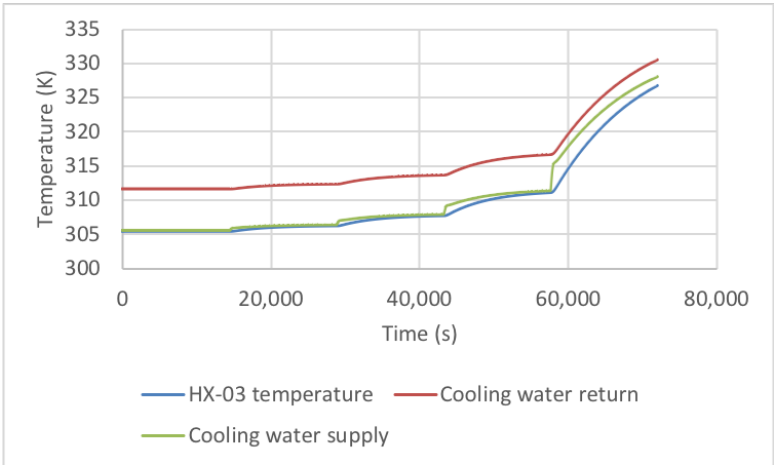

Figure 3.12. Plant model temperatures (T_{HX03}, T_{CWR}, T_{CWS}) response to stepped cooling tower fan speed.

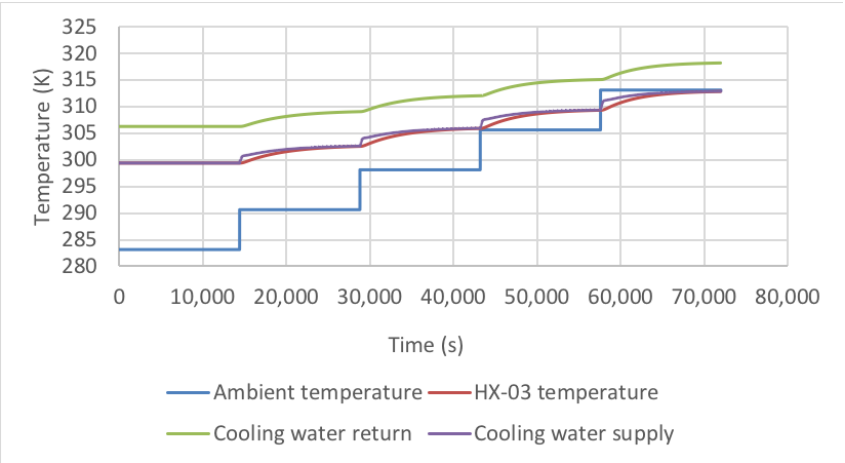

Figure 3.13. Ambient temperature ($T_{a_{in}}$) stepped and plant model temperatures (T_{HX03}, T_{CWR}, T_{CWS}) response.

larger flow rates and speeds result in incrementally less increase in heat exchange, than lower flow rates and pump speeds. This reflects the results from the cooling tower air fan speed and resulting air flow's incremental impact on cooling in the tower. At lower pump speeds the cooling water network temperatures are more sensitive.

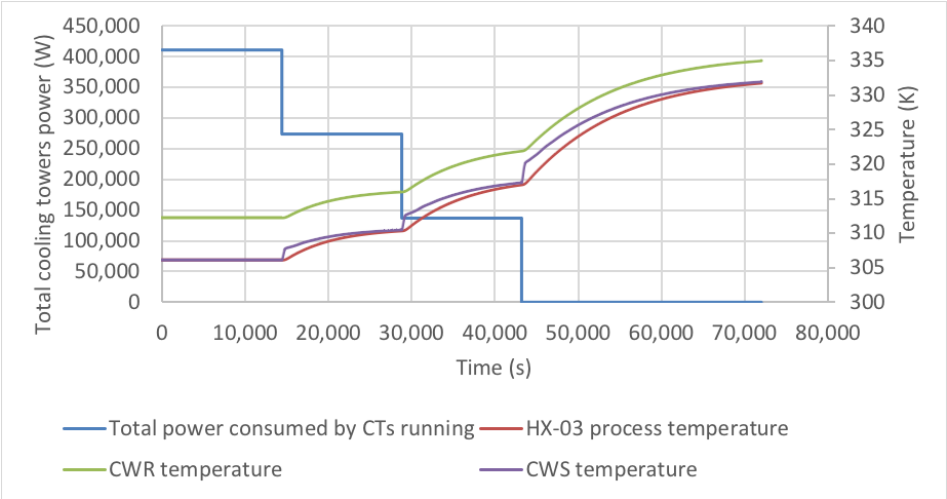

Figure 3.14. Cooling tower fans shut-down one-by-one and power response ($P_{f_{total}}$) and subsequent plant model temperature (T_{HX03}, T_{CWR}, T_{CWS}) response.

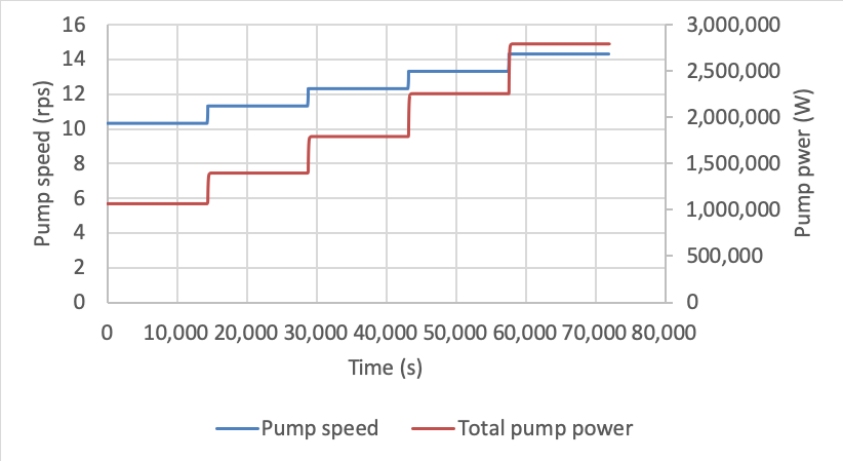

Figure 3.15. Pump speed (ω) stepped, and pump power (P_p) response.

3.6.5 Pump shut down (Pump speed MV)

The two running pumps were reduced to one running pump by tripping out one pump (setting pump speed ω to zero), and the resulting response of the total flow through the circuit, pump power, and plant temperatures are shown in Figure 3.18 and Figure 3.19. This is the natural extension of the pump speed simulations, where in this case the reduction in heat duty in the heat exchangers are more severe. This results in the pumps being turned into a potential hybrid control MV, just like the cooling tower fans, as an extension of its continuous MV control capability.

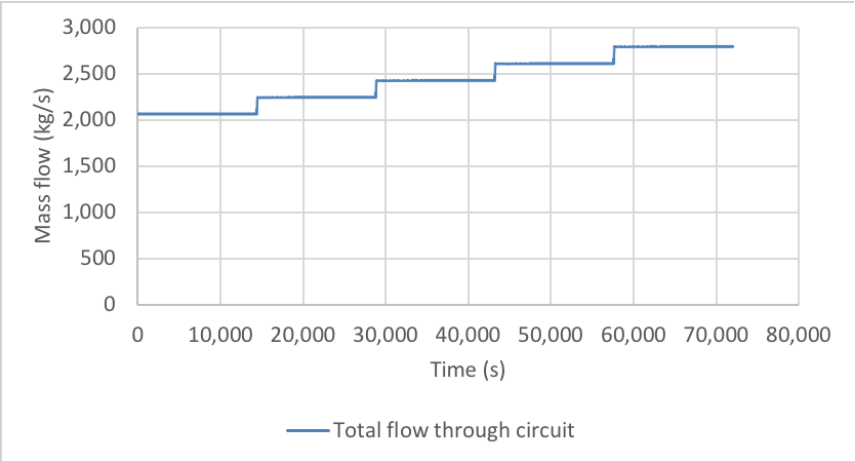

Figure 3.16. Total mass flow rate (F_{CWS}) through circuit response to stepped pump speed (ω).

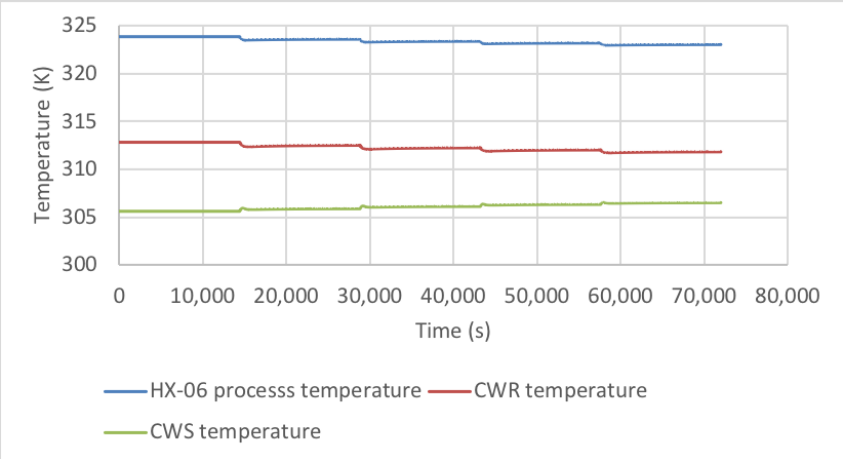

Figure 3.17. Plant model temperatures (T_{HX06}, T_{CWR}, T_{CWS}) response to stepped pump speed (ω).

In Figure 3.18 the mass flow rate transient takes 168 seconds in simulation to reach the new steady-state value, but then overshoots slightly and takes around another 190 seconds to reach steady-state.

3.6.6 Ambient humidity DV steps

The ambient relative humidity DV ($Y_{a_{in}}$) was stepped down from 100% to 0% every 4 hours (see Figure 3.20), and the effect of these changes on the response of plant temperatures are shown in Figure 3.21. At 100% humidity the evaporative cooling in the cooling towers is at minimum, and it is at maximum for this DV at 0% humidity. The change in cooling in the process streams follows the drop in evaporation due to the ambient humidity. The loss of cooling water in the network due to the extra

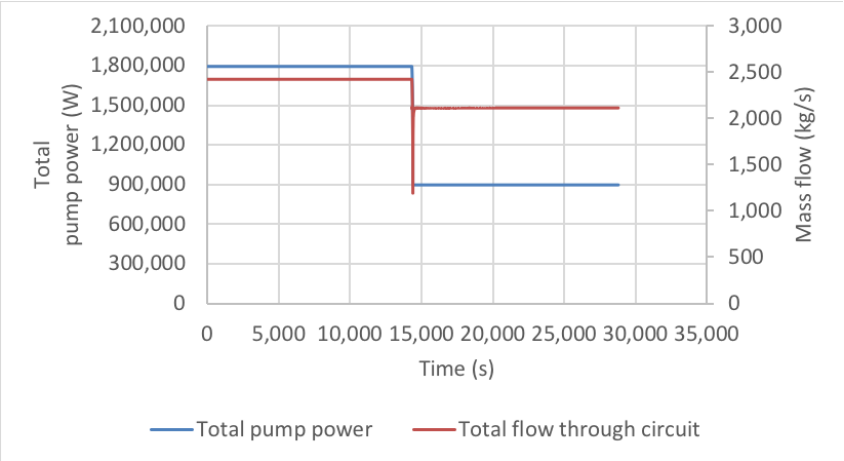

Figure 3.18. One of the two pumps tripped power response (P_p), and subsequent total flow through circuit (F_{CWS}) response.

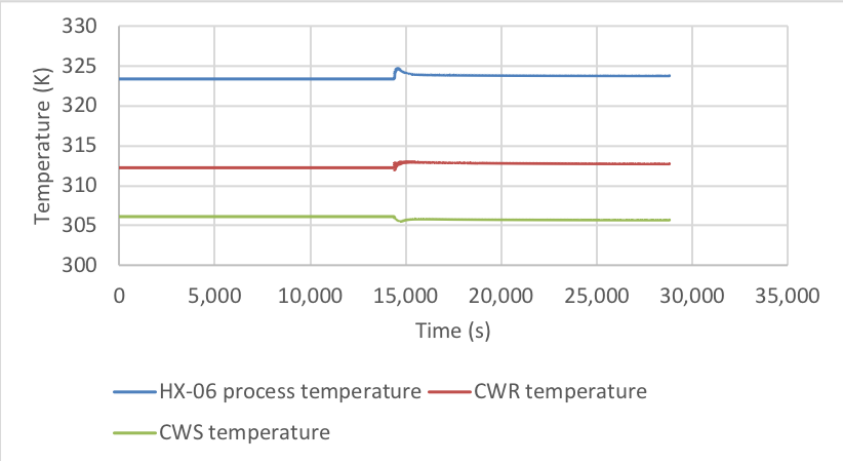

Figure 3.19. Plant model temperatures (T_{HX06}, T_{CWR}, T_{CWS}) response to one pump tripped scenario.

evaporation following the drops in humidity, is replaced with additional make-up water flow (F_{CWMU}) controlled by the PID level controller, as seen in Figure 3.20.

3.6.7 Cooling water control valve MV steps

The control valve MV controlling cooling water flow to the 2nd heat exchanger (V_{OP_2}) was stepped from turn down (1%) to 100% open in 10 linear hourly steps, and the response of the valve flow, total circuit flow, another process temperature (HX-06 process temperature - T_{HX06}) and other plant temperatures are shown in Figure 3.22 to Figure 3.24. The linearisation achieved with the equal

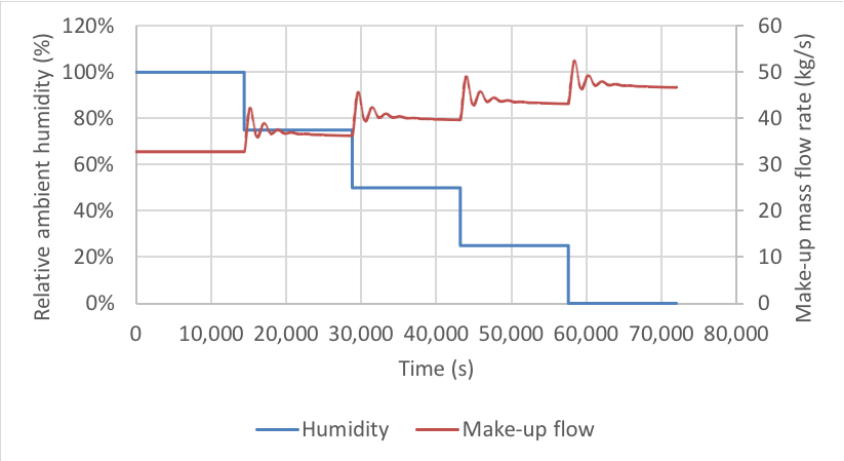

Figure 3.20. Ambient humidity ($Y_{a_{in}}$) steps and make-up flow (F_{CWMU}) response.

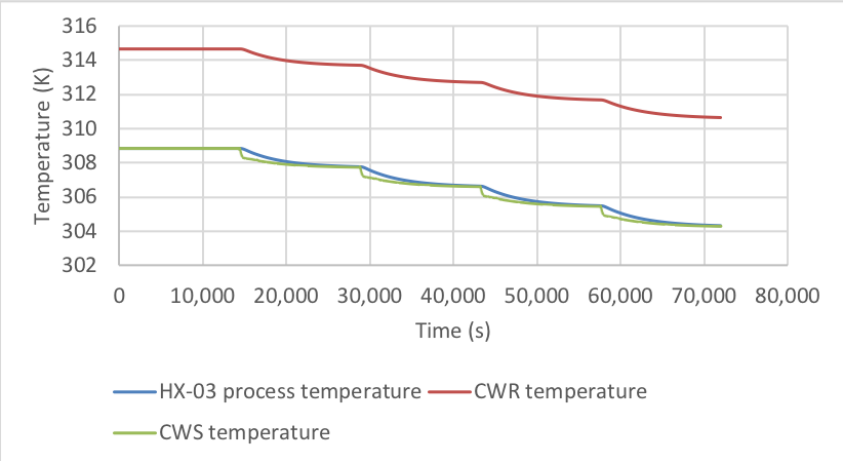

Figure 3.21. Plant model temperature (T_{HX03}, T_{CWR}, T_{CWS}) response to ambient humidity ($Y_{a_{in}}$) steps.

percentage characteristic is evident. Nevertheless, temperature controllability of the hydrocarbon stream of HX-02 (T_{HX02}) is much greater, and quite effective, below 50% valve opening.

The impact on the rest of the system temperatures is much less, since only one valve was stepped. The HX-06 process temperature increases very slightly by a total of 0.3 K from 1% to 100% for the valve. The CW flow through HX-06 decreases by around 4% over the full stepping sequence, which partly explains the small drop in temperature for the outlet process flow of that exchanger.

The total flow through the circuit increases (Figure 3.23) with the valve being stepped open, since the total resistance of the circuit from the outlet of the pumps, is decreased with increased valve opening.

Less pressure drop over the circuit results in the pumps moving down their operating curve towards higher flow.

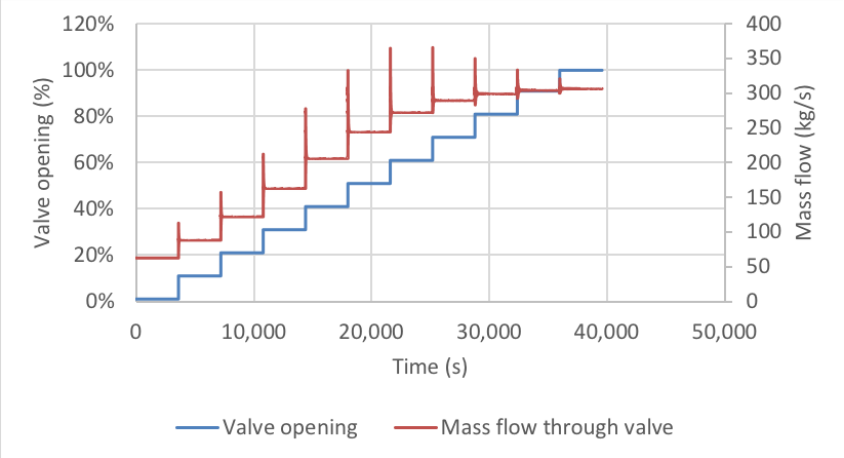

Figure 3.22. CW flow valve (V_{OP}) to one exchanger stepped, and mass flow through valve (F) response.

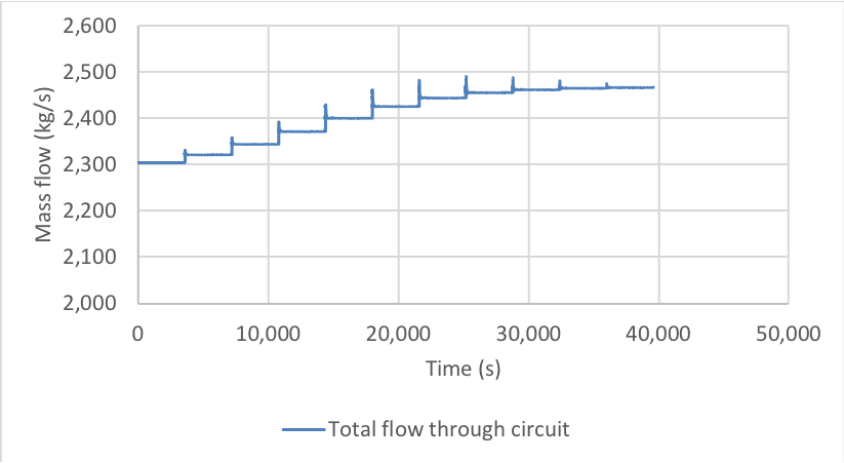

Figure 3.23. Total mass flow through CW circuit (F_{CWR}) response to CW exchanger flow valve (V_{OP}) steps.

3.7 CHAPTER SUMMARY

This chapter provided a detailed description of the design of the plant model, as well as justification for various modelling choices over alternative available solutions. This included the design of the parameter estimation techniques employed.

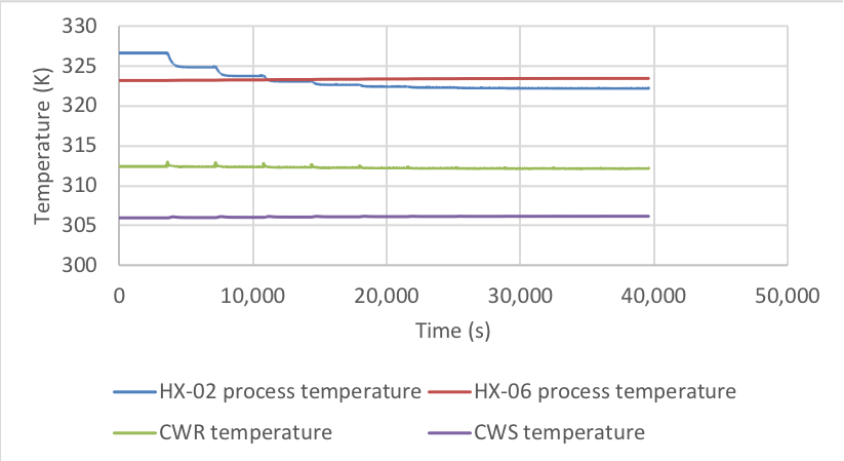

Figure 3.24. Plant model temperature (T_{HX02}, T_{HX06}, T_{CWR}, T_{CWS}) response to CW flow to exchanger valve (V_{OP}) steps.

The simulation framework was described, and detailed open-loop step testing performed on the plant model in simulation documented. The model response was discussed, and potential MV design options for the controllers to be developed in Chapter 4 were surveyed.

CHAPTER 4 MULTIVARIABLE CONTROL OF A COOLING WATER NETWORK

4.1 CHAPTER OVERVIEW

In this chapter multivariable control strategies are developed for the cooling water network model described in Chapter 3. NMPC and HNMPC are applied to control and optimise the cooling water network. The non-linear optimisation technique applied as part of the optimisation and control algorithm is gradient descent for the continuous MVs [61, 92], and graph traversal for the discrete Boolean MVs [83]. When the discrete MVs are not used, the controller is referred to as a continuous NMPC, and when they are, as a hybrid HNMPC. Controller design choices are motivated based on the literature reviewed in Chapter 2.

Closed-loop simulations are performed for the different control strategies and the dynamic plant model developed in Chapter 3. Three "controllers" are introduced - a base case divided into two sub base cases with constant MVs in Section 4.2, a continuous NMPC with only continuous MVs in Section 4.3, and a hybrid HNMPC with continuous and discrete MVs also in Section 4.3. The performance of these controllers is then tested and compared in simulation to ascertain how well they are able to reject disturbances typically found in a real cooling water network in Section 4.4. Four process scenarios are considered, i.e. a power minimisation scenario, a plant load disturbance scenario, and changes in ambient temperature and humidity. Numerical comparisons of the controller performance of the NMPC, HNMPC and base case controllers for the different control scenarios are included.

The monetary benefits achieved by the NMPC and HNMPC controllers are estimated by simulating the closed-loop system when exposed to actual ambient conditions. The controller minimises the

power consumption while at the same time maintaining control of the temperatures of the hydrocarbon streams, which results in significant cost savings for the plant owner.

4.2 COOLING WATER NETWORK CONTROL BASE CASE

4.2.1 Plant operation and disturbances

The plant on which this work is based does not automatically control the hydrocarbon stream temperatures at the exit of each cooling water heat exchanger as shown in Figure 3.1. This results in potentially significant fluctuation of the process temperatures as will be shown for the base case scenarios in this work. Significant deviation from setpoint negatively affects the down stream processing equipment (e.g. reactors, distillation columns, etc.) that were designed to operate at a constant hydrocarbon stream temperature.

The cooling water plant is subject to disturbances in the form of process plant load changes, and ambient weather conditions changes. Typical 24 hour (midnight to midnight) ambient temperature and humidity fluctuations that the cooling water plant is exposed to, are shown in Figure 4.1 and Figure 4.2.

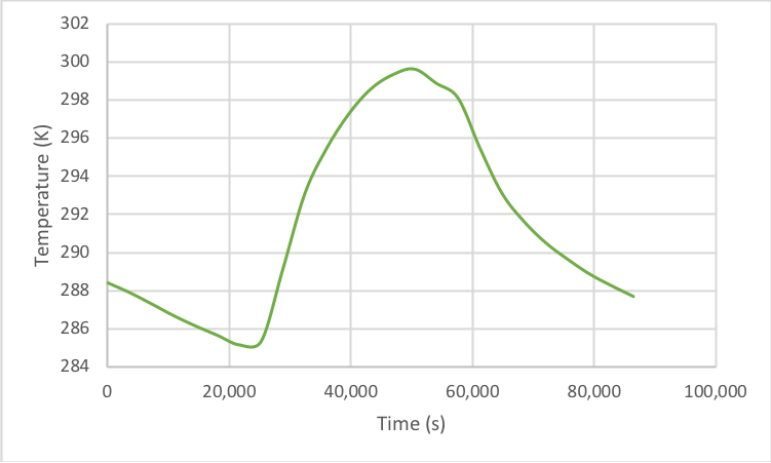

Figure 4.1. Typical 24 hour ambient temperature variation.

A fairly common plant disturbance occurs when the facility is operated at 80% load during start-up or shut-down transients, or when maintenance or catalyst change-out is being carried out on one process train, or when the facility is being curtailed due to upsets at the feedstock supplying company. This

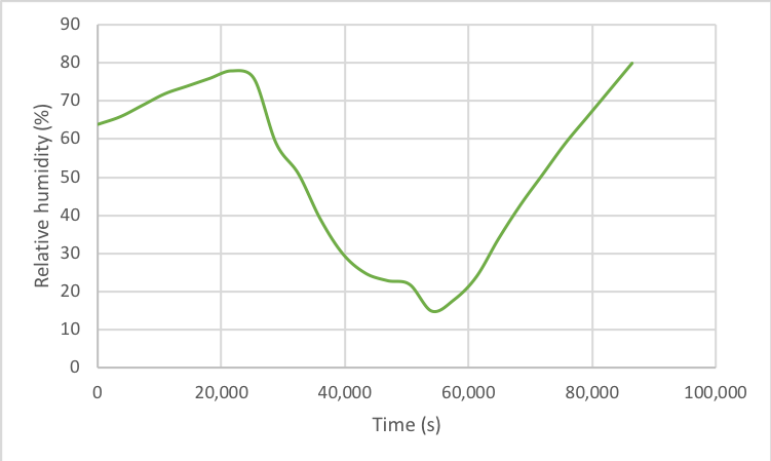

Figure 4.2. Typical 24 hour ambient relative humidity variation.

scenario is illustrated in Figure 4.3 that shows a simulated change in the hydrocarbon total mass flow load. After 4 hours (14,400 seconds) the plant load is restored.

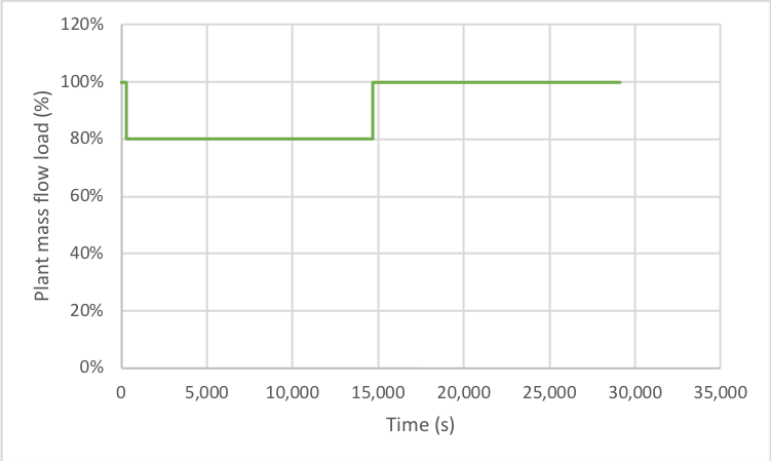

Figure 4.3. Plant hydrocarbon mass flow rates during plant load disturbance scenario.

4.2.2 Base case

The base case describes a scenario when the plant is operated without any advanced control and/or optimisation. The valves and speeds of the pumps and fans are not actuated, and the Boolean MVs used to switch the fans or pumps on or off are not active. The only feedback control in the cooling water network is the PID level control of the cooling tower basin, where the level is controlled through make-up water. Level control is required as this loop is open-loop unstable.

In order to illustrate the power minimisation and then the temperature control capabilities of the control strategies developed in this work separately, the base case is divided into two sub base cases: Base Case A, and Base Case B.

4.2.2.1 Base Case A

In Base Case A the fan and pump speeds run at their design values. Base Case A can therefore serve as a benchmark to compare the plant with APC implement to, where the control strategies exploit the opportunity to decrease the speeds to optimal values while maintaining temperature control. The plant design speed for the cooling tower fans is 2 rps and for the pumps it is 12.33 rps [1]. The cooling capacity generated at these speeds is only required during the warmest summer days when the ambient temperature and ambient humidity are at their highest. For the typical ambient conditions that the cooling water plant is exposed to during evenings and most days the cooling capacity generated at these speeds is more than needed, resulting in more electric power consumption than is optimal for a given cooling requirement.

The uncontrolled plant in Base Case A experiences hydrocarbon temperature fluctuation due to the typical ambient temperature and humidity variations shown in Figure 4.1 and Figure 4.2. This is shown in Figure 4.4 for the temperature of the hydrocarbon outlet of HX-01.

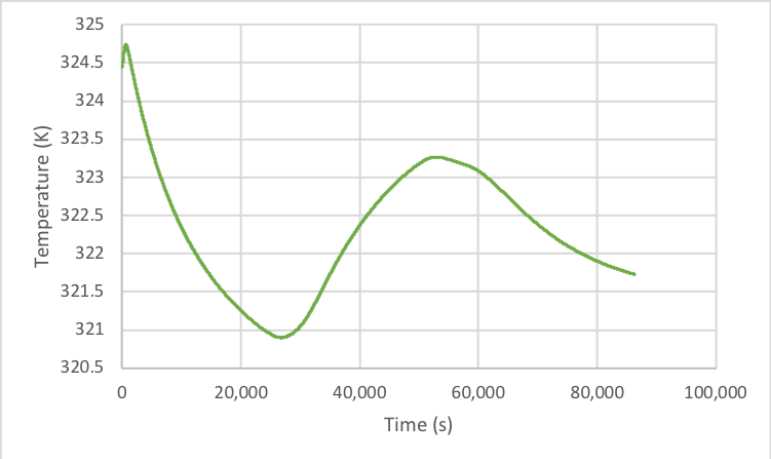

Figure 4.4. Base Case A: Uncontrolled HX-01 outlet temperature.

4.2.2.2 Base Case B

In Base Case B, the starting speeds of the fans and pumps are reduced to 1.2 rps and 9.75 rps respectively from that of Base Case A. This enables the control action to be less focused on power minimisation at typical ambient conditions, and more focused on temperature control.

For the uncontrolled plant in Base Case B, the hydrocarbon temperature fluctuates significantly from steady-state due to the process disturbances the cooling water plant is exposed to. This is evident from the fact that the plant load disturbance scenario shown in Figure 4.3 results in the hydrocarbon temperature changes seen in Figure 4.5.

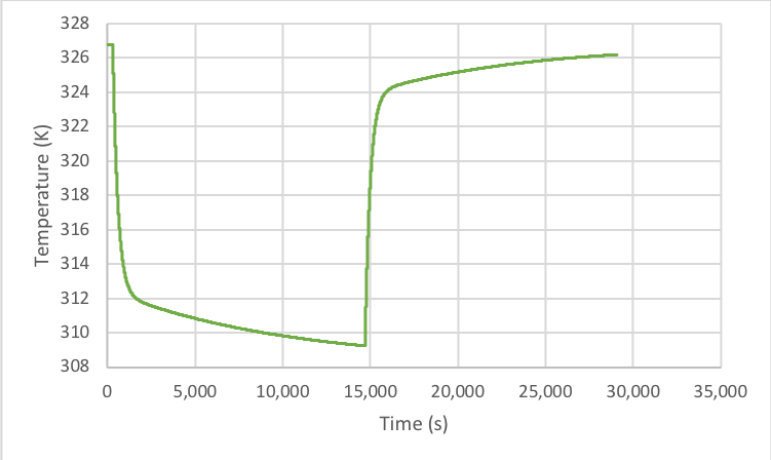

Figure 4.5. Base Case B: Uncontrolled HX-01 outlet temperature.

4.2.2.3 Monetary loss

Typical electricity rates for industries in the Arabian Gulf where the plant is located are USD 0.048 per kWh [111]. At this rate, and for the design fan and pump speeds as indicated above (Base Case A), the daily power consumption of the plant is 52,826 kWh, which translates into a daily power bill of USD 2,536.

4.2.3 Making the cooling water network amenable to Advanced Process Control

The uncontrolled plant does not have enough sensors and actuators for continuous NMPC nor HNMPC. Only the hydrocarbon process outlet streams of HX-04, HX-05, HX-06, HX-07 and HX-09 have temperature measurements (Chapter 3 and [1]). The other heat exchangers will need temperature measurements installed in order to enable APC strategies for the cooling water network.

Al-Bassam and Alasseri [32] and Saidur *et al.* [33] show that energy consumption can be reduced when using Variable Speed Drives (VSDs). Continuous NMPC will require VSDs for the cooling tower fans and for the cooling water pumps. In addition, the fans and pumps will require some switch gear to enable on and off discrete switching of these equipment by the hybrid component of the HNMPC controller (see Section 4.3.1).

The hand valves that feed each cooling water heat exchanger with CWS, will need to be turned into control valves for the continuous NMPC controller to function. This will require different valve hardware as well as positioners and actuators for each valve. The plant changes described above are assumed to be in place in what follows in order to demonstrate the benefits that can be achieved when using APC. Economic benefits assessment of APC technologies in general was surveyed in [59].

4.3 COOLING WATER NETWORK CONTROL DESIGN

In this section, two APC strategies, NMPC and HNMPC, are developed for the cooling water network model described in Chapter 3. The purpose of these controllers is to demonstrate the benefits that can be obtained over the two base cases described in Section 4.2.2.

4.3.1 Control objectives and framework

There exists an opportunity to minimise power consumption (see e.g. [30]), and to control the process temperature by utilising some or all of the MVs. For this purpose, an advanced control scheme can be developed (see e.g. [55]). The control objectives are disturbance rejection for hydrocarbon process stream temperatures, power consumption minimisation for the cooling tower fans, and power consumption minimisation for the pumps.

The simulation results in Section 3.6 show that a number of MVs can be used to control the process outlet temperatures of the heat exchangers. These MVs include the cooling tower fan speeds (Figure 3.12), the pump speeds of the CWS pumps (Figure 3.17), and the valve openings of the cooling water flow to each exchanger (Figure 3.24). For control strategies in this work, the hand valves upstream of each heat exchanger are modelled as equal percentage control valves as per (2.28).

The CVs for the system are the hydrocarbon process stream temperatures at the outlet of each heat exchanger (T_{HX_i}), the power consumption of each cooling tower fan motor (P_{f_i}), and the power consumption of each pump motor (P_{p_i}).

Three control strategies are used. The MVs applied depend on the chosen control strategy:

1. Base case consisting of two sub base cases: No part of the plant is actuated, and the MVs are kept constant (see Section 4.2.2).
2. Continuous NMPC: The continuous MVs are included in an NMPC controller. In this scenario, the 11 hand-valves upstream of the cooling water side of each heat exchanger, are exchanged for positioner controlled control valves and included as continuous MVs in the controller (V_{OP}). The 3 cooling tower fan motors are assumed to be fitted with VSDs enabling the fan speeds (n) to be continuous MVs in the controller. The 2 cooling water pump motors are assumed to be fitted with VSDs enabling the pump speeds (ω) to become continuous MVs.
3. HNMPC: The 3 cooling tower fans on/off states, and 2 pump motor on/off states, are included as Boolean MVs in addition to the continuous MVs of control strategy 2.

The measured DVs of the system are the ambient temperature and humidity (Figure 4.1 and Figure 4.2), as well as the hydrocarbon mass flow rates and hydrocarbon input temperatures to the heat exchangers.

The capabilities of the NMPC and HNMPC controllers will be compared using the following process scenarios:

1. Power consumption minimisation.
2. Plant load disturbance rejection.

3. Ambient temperature disturbance rejection.
4. Ambient humidity disturbance rejection.

The industrial cooling water network on which this work is based is not currently automated. This work shows what can be achieved if the temperatures and power consumption are controlled. The MVs (control valves, fan speeds and pump speeds) do have explicit operating ranges on the real plant and therefore the MV constraints are included in the control problem and designed controllers. Engineering unit CV ranges are defined as an order of magnitude of simulated range without explicit limits being available in order to be able to do normalisation of engineering units. The limits of the continuous CVs and MVs in engineering units are as per Table 4.1.

Table 4.1. Continuous CV and MV ranges.

Parameter	**Minimum**	**Maximum**	**Units**
Heat exchanger temperatures (11 CVs)	0	400	K
Cooling tower fan power (3 CVs)	0	300000	W
Pump power (2 CVs)	0	3000000	W
Cooling tower fan speed (1 MV)	0.1	2.5	rps
Pump speed (1 MV)	1	14	rps
CW control valves (11 MVs)	0	1	fraction

4.3.2 NMPC

The controllers used in this work are chosen from the NMPC and HNMPC techniques surveyed in Sections 2.3 and 2.4. These techniques are suitable as the plant in question is non-linear with large and continuous actions spaces. For control problems of this kind the techniques from Reinforcement Learning are less mature (Section 2.5). Chapter 3 and [1] indicate that each discrete mode of the plant model has a convex hyperspace (2.45) for the continuous MVs. This is indicated by the shapes of the curves of the CV responses to equal sized steps in the MVs from steady-state. This implies that gradient based NMPC techniques from Section 2.3 can be used to apply NMPC to the plant using the plant model developed as documented in Section 3.4.

NMPC problems have been shown in Section 2.3.3 to theoretically have stable deterministic closed-loop control solutions for the types of continuous control problems studied in this work (2.47). The Newton-based active set method, (2.54) to (2.56), and interior point method, (2.58) to (2.61), were applied to design two NMPC controllers utilising these methodologies. The developed plant model given in Section 3.4 was used as the internal model of the controller, and also as the plant in the closed-loop simulation. The Hessian matrix that needs to be calculated at each iteration of the controller was found to be especially computationally costly to calculate as indicated in the literature cited in Section 2.3.4. This is due to each entry of the matrix requiring the ODEs that describe the plant model to be integrated into the future up to the prediction horizon in order to calculate the objective function cost. The Hessian matrix can also be less well conditioned depending on the objective function hyperspace, a problem that can be averted by using first order methods. (Sections 2.3.4 and 2.3.5).

Instead of the Newtonian methods mentioned above, the first order gradient descent methods of steepest descent (2.62), and steepest descent with momentum (2.63), were applied to the NMPC problem. Both methods resulted in stable closed-loop control, but the steepest descent with momentum method had superior performance with significantly faster convergence of the most important CVs (as predicted by the literature in Section 2.3.5 and in particular (2.64) to (2.71)) to their setpoints in each control case simulated (see Section 4.4 for detailed results). Empirical closed-loop tuning of the exponential decay momentum factor γ and learning rate η were performed (see Table 4.3).

The constraints on the MVs for the control problem (see Table 4.1) are constant, linear constraints ("box constraints"). Therefore the projected gradient method, (2.72) and (2.73), is suitable for handling the constraints in the NMPC algorithm and was implemented. The Lagrangian approach (2.52) to constraint handling could also have been applied with the steepest descent algorithm. However, it is a more complicated and computationally expensive approach and is not needed since only the MV constraints are important in the NMPC problem to be solved.

The one-sided-difference approximation (2.48) was used to calculate the derivatives of the Jacobian vectors and Hessian matrices for all the gradient based numerical non-linear optimisation algorithms used.

4.3.3 HNMPC

For the plant controlled in this work, the 3 cooling tower fans and the 2 cooling water pumps can be turned into Boolean MVs if the motors of the rotating equipment can be switched on and off by the HNMPC controller. Due to the induction motors driving the fans and pumps exceeding their steady-state current by 500 to 800 percent during start-up (see Section 2.2.3), a deadband is introduced into the controller if the power is included for minimisation in the objective function, which will limit unnecessary switching of the equipment.

A continuous PSO NMPC controller was augmented with discrete PSO control, as per (2.75) to (2.78), to become an HNMPC PSO controller. However, the Boolean MVs at times stayed in one discrete state longer than needed, and at other times switched Boolean state after an optimum had already been found. The performance of the Boolean MVs under the control of this HNMCP PSO controller was found to be inferior to the graph traversal algorithm approach mentioned below.

The steepest descent with momentum NMPC controller designed in Section 4.3.2 for the continuous MVs, was combined with a graph traversal approach to optimise the discrete states that the plant can be in. This design resulted in an HNMPC controller based on first order gradient descent with momentum for the continuous MVs, and a tree based MINLP discreet MV controller (Section 2.4.2) for the Boolean MVs.

The possible states that the plant can be in from the perspective of the hybrid controller, is shown in the tree graph of Figure 4.6. The graph is shown as a tree where each level of the tree shows the permuted possible states of 1 of the 4 Boolean MVs. The leaves of the tree in the lowest row when traced up to the root node, represent all the 16 possible permutations of states the controller can put the plant in. The 3 cooling tower fans can all be off as a discrete plant state, or can be switched on one by one. In addition, at least one of the two pumps needs to run since with no pumps running no cooling water would circulate in the network. If one pump is chosen to always be on, and each cooling tower is assigned a fixed role in the permutations of cooling tower states, the number of effective discrete plant states to be optimised between can be limited to 8 states as per Table 4.2. As per Section 2.4, the MINLP problem to be solved by the HNMPC controller (2.74) can be an NP-hard problem, therefore any consolidation of plant discrete states that can be done as per Figure 4.6 will increase the controller

viability and reduce computational cost. The reduced number of effective states are shown as the red nodes in Figure 4.6.

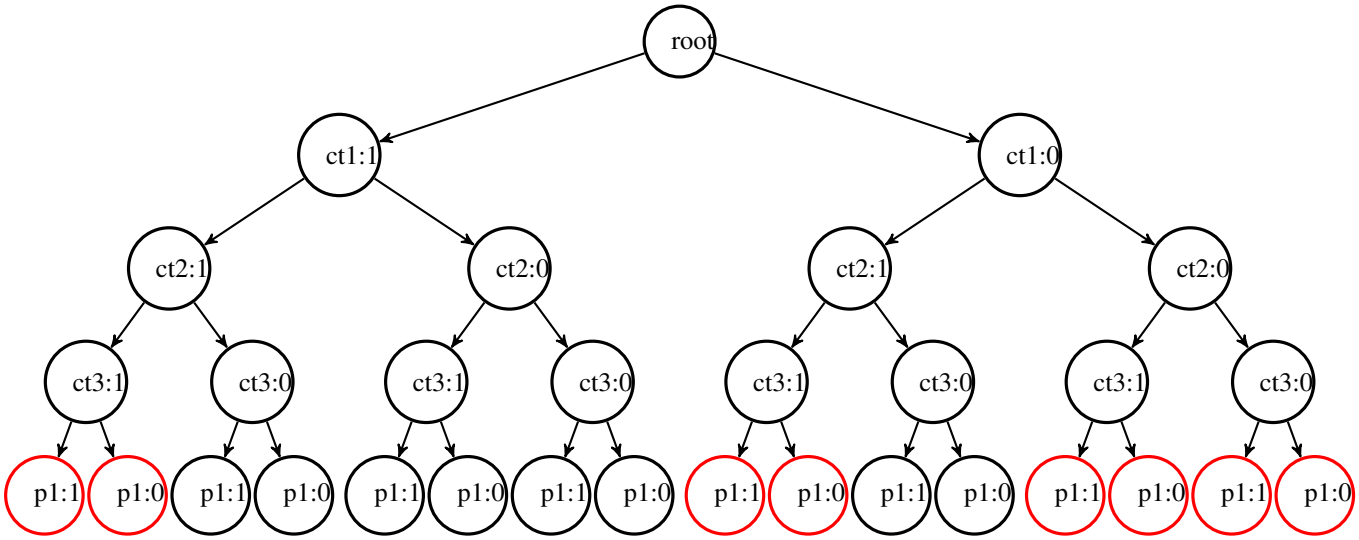

Figure 4.6. Hybrid controller state graph (Short hand: ctx:1 means cooling tower x fan is on, and ctx:0 means it is off. px:1 means pump x is on, and px:0 means it is off).

Table 4.2. Discrete plant states as determined by Boolean MVs.

State nr	CT 1	CT 2	CT 3	Pump 1
1	1	1	1	1
2	0	1	1	1
3	0	0	1	1
4	0	0	0	1
5	1	1	1	0
6	0	1	1	0
7	0	0	1	0
8	0	0	0	0

The number of possible plant states to be optimised between and for which the NLP problem has to be solved each time the controller is run is low enough so that a brute force traversal of the graph nodes can be performed (Section 2.4.2). If this is viable, then the need for a more complex and expensive algorithm like branch and bound (2.79) can be averted. This approach was found to be viable for the HNMPC controller implemented.

During each iteration of the HNMPC controller, the objective function (4.1) is calculated for each plant state which is modelled as a red node on the plant state graph being traversed (Figure 4.6). The best node as per the objective function (4.1) is chosen, and the hybrid MVs switched accordingly given the current state of the continuous MVs. Then the gradient descent step for the continuous MVs is calculated and implemented with (2.63). After this, the iteration step is completed, and the algorithm will start again with traversing the tree nodes at the next controller iteration. This high-level algorithm is portrayed in pseudo code in Algorithm 2.

Algorithm 2 HNMPC controller algorithm

```
Initialise plant model.
Initialise discrete MVs.
Initialise continuous MVs.
for the simulation period do
    Integrate plant model state equations one step.
    if controller waiting period complete according to controller fequency then
        for all effective discrete plant states do
            Evaluate objective function for plant state.
        end for
        Select discrete MVs of plant state with highest objective function.
        Calculate continuous MV Jacobian gradient of objective function.
        Calculate the new continuous MV values with the gradient.
        Apply the projected gradient method for constraint handling.
        Implement gradient descent continuous MV step for the calculated gradient.
    end if
end for
```

A logic block is inserted between the model and the NMPC controller to take a single continuous MV from the NMPC controller for fan speed, and duplicate it across all 3 fans. The same is done for the 2 pumps. This approach simplifies the NMPC design. Individual motors can still be shut-down and started up by the Boolean MVs of the HNMPC controller independent from each other.

The objective function to be minimised by the HNMPC controller, is a weighted sum of the 3 classes of CVs in the controller (see Section 4.3.1):

$$J = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{11} Q_{T_{HX}j}\left(T_{HX_j} - T_{SPHX_j}\right)^2 + \sum_{j=1}^{3} Q_{P_{fj}} P_{f_j}^2 + \sum_{j=1}^{2} Q_{P_{pj}} P_{p_j}^2\right) \tag{4.1}$$

where the setpoint targets T_{SPHX_j} are subtracted in each summation from the heat exchanger outlet process temperatures T_{HX_j}. The fan power consumptions are represented by P_{f_j} and the pump power consumptions by P_{p_j} [17]. $Q_{T_{HX}j}$, $Q_{P_{fj}}$ and $Q_{P_{pj}}$ are the weights for the heat exchanger hydrocarbon stream outlet temperatures, and the fan and pump power error terms respectively. The objective function is calculated and summed over a prediction horizon (N) of 45 minutes simulation time, each time the HNMPC and NMPC control algorithms execute. In order to minimise computational cost, shorter prediction horizons and shorter HNMPC and NMPC control horizons are preferred for a certain level of stability [68]. Therefore, the HNMPC and NMPC control horizons are 1 control move for the prediction horizon of 45 minutes. With this design, feedback from the plant is obtained after each control move that the controller algorithm sends to the plant.

The frequency of the controller executing one iteration was set to 10 minutes simulation time. The numerical plant model is iterated every 200ms due to the stiff nature of the plant model differential equations [1]. No terminal constraint, as per (2.43), was added to the objective function.

For the continuous-only NMPC scenarios, the same objective function (4.1) is used, but the Boolean MVs are not available for manipulation.

The power variables are always minimised in the objective function and its contribution to the objective can be set by tuning the weights of the respective terms in (4.1). The HNMPC controller will not allow the MVs to move outside their limits (2.73), whereas temperature CVs can but are penalised more in the objective function the further away from the setpoint targets they move.

Empirical tuning of the controller tuning constants was performed through simulation runs of the closed-loop system for all scenarios and control cases with the goal of achieving tight control but also stability. Other parameter values experimented with either compromised on tightness of control, or stability of the closed-loop response. The maximum step size allowed for the continuous MVs during one controller iteration, was limited to 10% of the engineering unit range of the MV. The purpose of

this limitation is to prevent severe disturbances to the plant due to potentially very aggressive action of the controller. Tuning constants and optimisation algorithm parameters were used in the controller as per Table 4.3.

Table 4.3. Tuning parameters of NMPC and HNMPC controllers.

Parameter	**Symbol**	**Value**	**Units**
Momentum decay	γ	0.0925	
Learning rate	η	0.0625	
Objective weight HX temp.	$Q_{T_{HXj}}$	1	
Objective weight Fan power	$Q_{P_{fj}}$	0.00001	
Objective weight Pump power	$Q_{P_{pj}}$	0.0001	
Controller prediction horizon	N	45	Minutes
Controller running interval		10	Minutes
Controller NMPC control horizon		1	Control move
Maximum continuous MV change per iteration		10	% of EU range

The same tuning values for all controller parameters were used for both the NMPC and HNMPC controllers. A high-level portrayal of the plant and designed control system indicating the various MV and CV signals between the plant equipment and NMPC/HNMPC controller are shown in Figure 4.7. The 11 parallel heat exchangers and 11 control valves are shown as 1 exchanger and valve to save space on the drawing.

The HNMPC controller is implemented in a C++ simulation platform [112] that is able to control and optimise the plant model (see Section 3.4) with its stiff partial differential equations in real time. See Appendix A for more information about the platform, and Section 4.4.7 for details on the timing performance of the simulations performed in various scenarios.

4.4 CLOSED-LOOP SIMULATION

The NMPC and HNMPC controllers are simulated in closed-loop in the subsections that follow. The different scenarios are discussed separately with each control strategy simulated for each scenario. Subsequently the scenarios and strategies are compared and conclusions drawn.

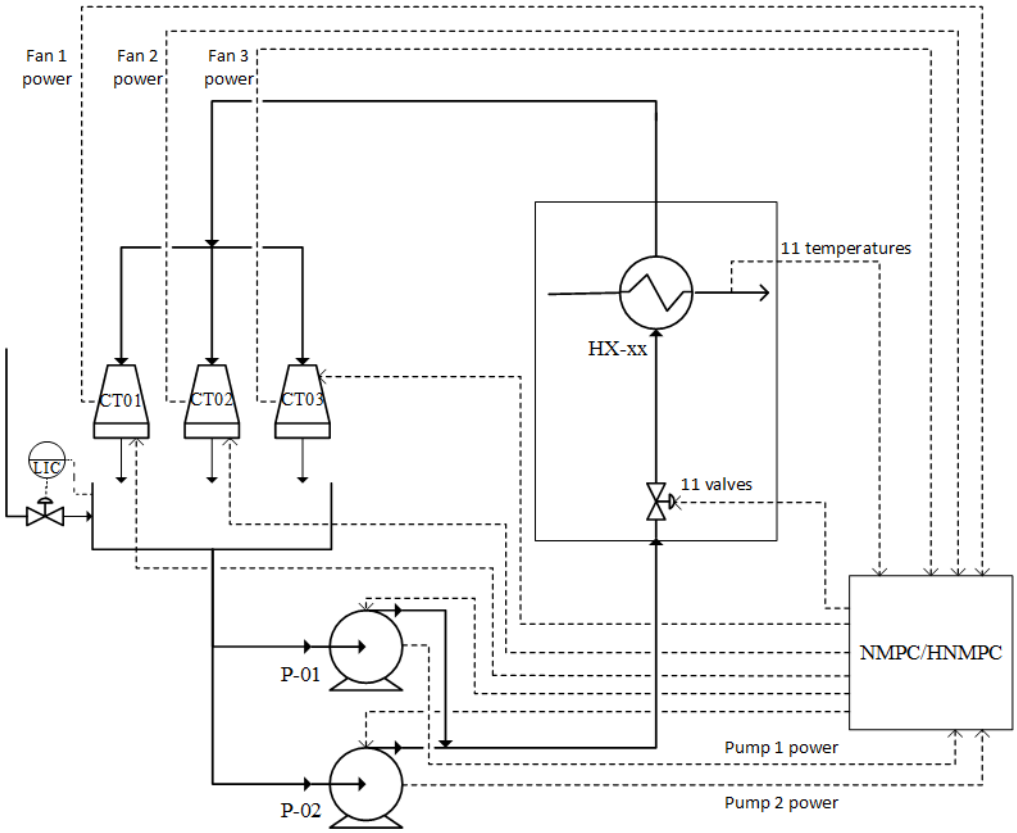

Figure 4.7. Overall control system design with NMPC/HNMPC, MV and CV signals. HX-xx is a place holder for the 11 heat exchangers.

All simulations are started with the plant running with an ambient temperature of 25 °C, and an ambient humidity of 50%. The power minimisation scenario is run over 24 hours of simulation time. The plant load disturbance, ambient temperature disturbance and ambient humidity disturbance rejection scenarios are started with 5 minutes of simulation time with the plant at steady-state before the disturbances are applied, followed by 8 simulation hours of disturbance rejection control.

4.4.1 Power minimisation scenario

The design figures documented in Section 3.2 for the cooling tower fan speeds and cooling water pump speeds are higher than what is required for the ambient conditions the plant is typically exposed to. The continuous NMPC controller is able to exploit this opportunity when excess cooling capacity is being generated by ramping down the fan and pump speeds while maintaining the temperatures.

In this scenario the simulation is started with the fans and pumps running at their design speeds as per Base Case A in Section 4.2.2. The plant is exposed to a typical 24 hour ambient temperature and humidity cycle as per Figure 4.1 and Figure 4.2. The hydrocarbon stream temperatures at the outlet of HX-01 are shown in Figure 4.8 for the uncontrolled base case and when NMPC and HNMPC are used. The NMPC and HNMPC controllers are able to keep the temperature at setpoint, and have the same effect since fast discrete control action is not required due to the slowly varying disturbances.

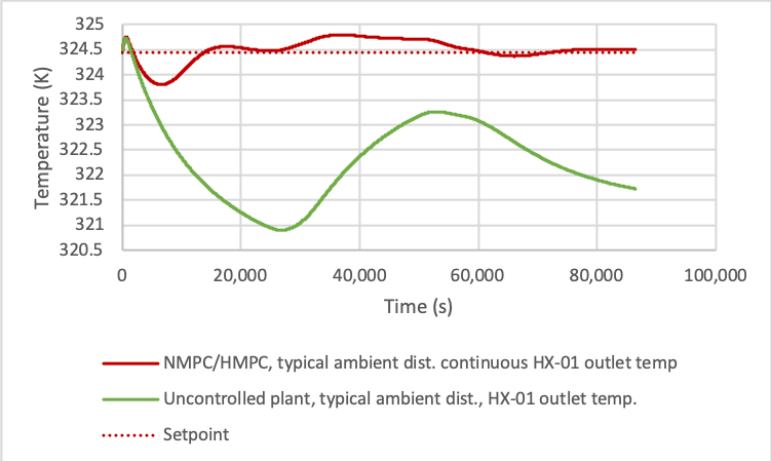

Figure 4.8. HX-01 outlet hydrocarbon temperature for power minimisation scenario.

All the hydrocarbon temperatures at the outlet of the 11 heat exchanges are shown in Figure 4.33 for the power minimisation scenario. The NMPC and HNMPC controllers are able to keep all the temperature CVs to setpoint while rejecting the ambient conditions disturbances.

The controllers decrease the fan speeds as per Figure 4.9 in order to minimise fan power as shown in Figure 4.10 while controlling temperature. Due to the cubed relationship between fan power and speed [1], a 40% reduction in speed results in almost an 80% reduction in fan power.

The pump speed is also reduced by the controller as per Figure 4.11 in order to minimise pump power as shown in Figure 4.12.

Cooling water valve movement is negligible in this scenario since the ambient conditions change very slowly and adjusting cooling capacity in the circuit as a whole is needed. This cooling capacity adjustment is mainly done by the fan and pump speeds as opposed to the cooling water valves. As expected, the controllers focus on the fans and pumps as this is where the most power is consumed.

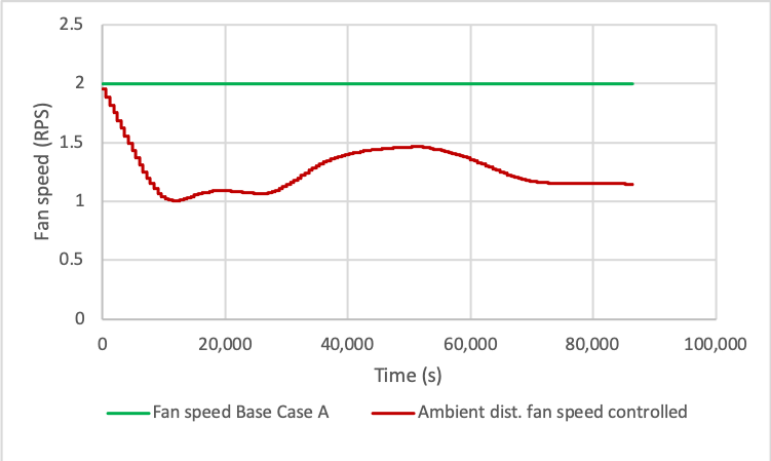

Figure 4.9. Cooling tower fan speed for power minimisation scenario.

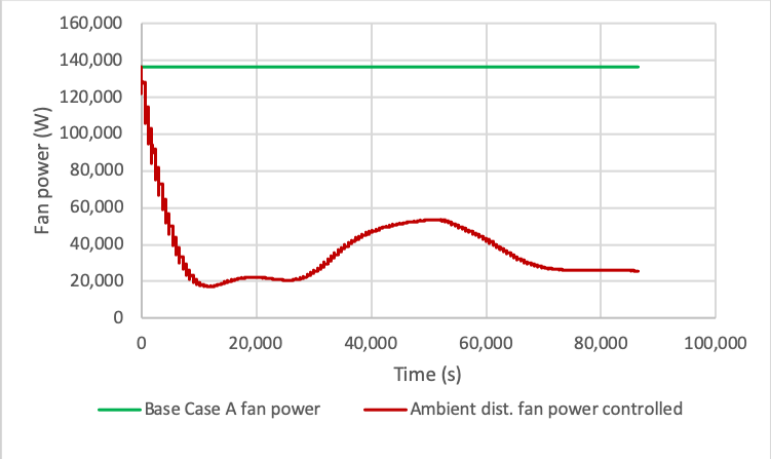

Figure 4.10. Cooling tower fan power for power minimisation scenario.

4.4.2 Plant load disturbance scenario

Base Case B as described in Section 4.2.2 is used here to compare the NMPC and HNMPC controllers. The plant load disturbance applied is shown in Figure 4.3. The simulation is continued for another 4 hours after the plant loads have been restored to give the temperature time to settle.

The temperature disturbance rejection performance of the NMPC and HNMPC controllers are compared to Base Case B in Figure 4.13. The NMPC and HNMPC controllers optimise the cooling capability as much as possible, and limit the undershoot of the hydrocarbon temperatures. The HNMPC can contain the undershoot slightly better than the NMPC. When the plant load is restored, the controllers bring

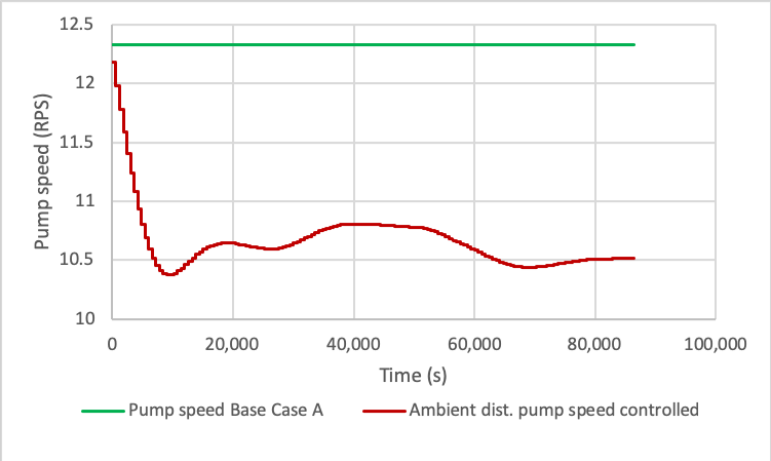

Figure 4.11. Cooling water pump speed for power minimisation scenario.

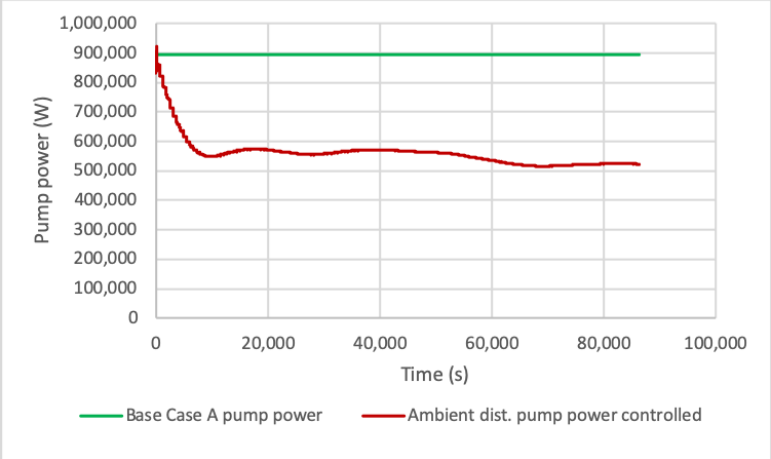

Figure 4.12. Cooling water pump power for power minimisation scenario.

the temperature back to setpoint.

Similar to Figure 4.13, the controller performance in terms of temperature disturbance rejection in the continuous control case is shown in Figure 4.34, but for all the hydrocarbon outlet temperatures. Similar to the outlet temperature of HX-01, most other heat exchanger outlet temperatures drop when the disturbance happens, but are then brought back to setpoint by the controller action before the disturbance is removed. However, the outlet temperatures of HX-03 and HX-04 increase, instead of decrease in response to the disturbance, and has to be brought down to setpoint by the control action, as opposed to the rest of the temperature CVs that need to be increased back to setpoint. The reason for this is that these temperature setpoints are low and are very close to the CWS temperature coming

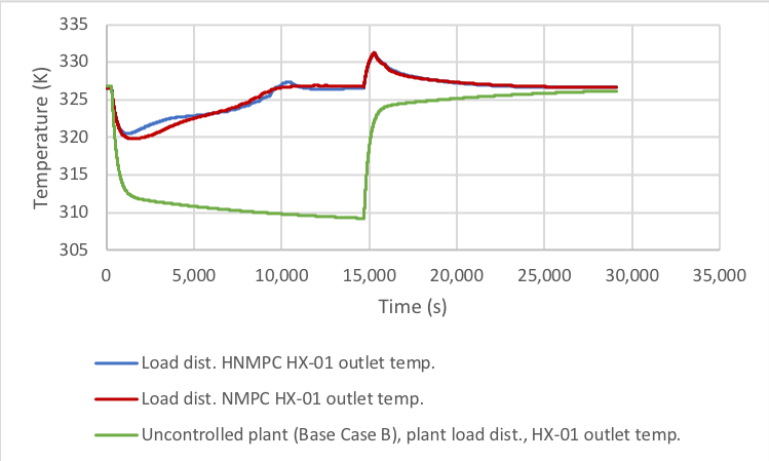

Figure 4.13. HX-01 outlet hydrocarbon temperature, with plant load disturbance.

into the heat exchangers from the cooling water pumps exchangers. When the cooling capacity of the network is then reduced by the controller reducing the fan speeds and pump speeds the result for these two CVs is too little cooling instead of too much cooling like for the other CVs. The valve action and other controller MV action will take this into account to limit the increase in the temperature of these 2 CVs.

All heat exchanger hydrocarbon outlet temperatures as controlled by the hybrid controller for the plant load disturbance scenario, are trended in Figure 4.35.

The cooling tower fan power is minimised by the controller while still keeping the temperature control as the first priority (Figure 4.14). Under continuous control the controller is limited to moving the speed to not more than 10% of its range per control iteration. The speed is ramped down as fast as possible, and then ramped up again to limit temperature overshoot when the plant load is restored (Figure 4.15). After the temperature is restored to setpoint, the speed is still ramped down somewhat to minimise power consumption. Only the fan power for one of the three fans is shown in the case of NMPC control, since the fan speed and power will be the same for each fan in the NMPC case.

Under HNMPC control, all 3 cooling tower fans are switched off by the controller as soon as the plant load drops (Figure 4.15) thus minimising cooling capacity and saving power (Figure 4.14). Fan 2 is switched on again after around 2,500 seconds, followed by fan 3 at around 3,000 seconds. Fan 1 is switched on at 9,000 seconds, and switched off again at 11,000 seconds. When the plant load is

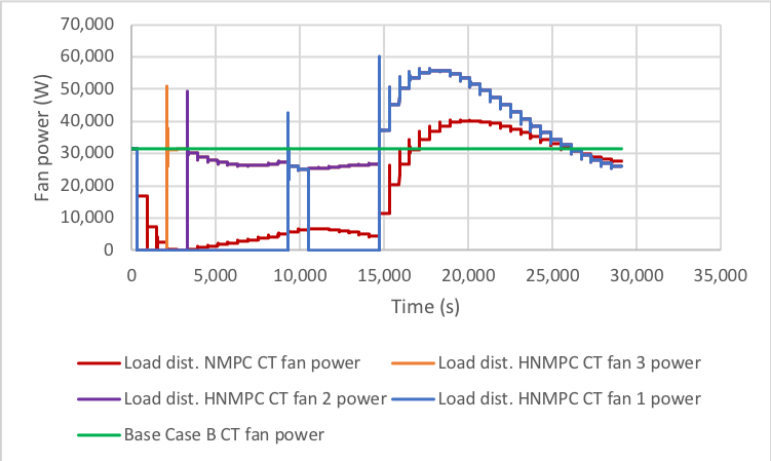

Figure 4.14. Cooling tower fan power with plant load disturbance.

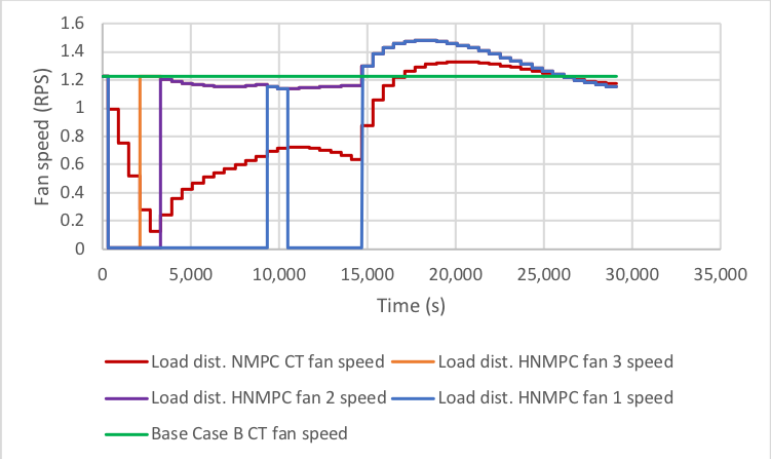

Figure 4.15. Cooling tower fan speed with plant load disturbance.

restored, fan 1 is switched on again and all fans are ramped up and later down to limit temperature overshoot, and save power.

Under NMPC control the pump power is minimised (Figure 4.17) by turning the pump speed down (Figure 4.16). When the plant load returns, the speed of both pumps is ramped up to prevent hydrocarbon temperature overshoot.

Under HNMPC control, pump 1 is shut down immediately (Figure 4.16) when the plant load is reduced, thus minimising cooling capacity, as well as pump power (Figure 4.17). Pump 2 is kept on (Figure 4.16) to supply cooling water to the plant but turned down to save power (Figure 4.17). The power

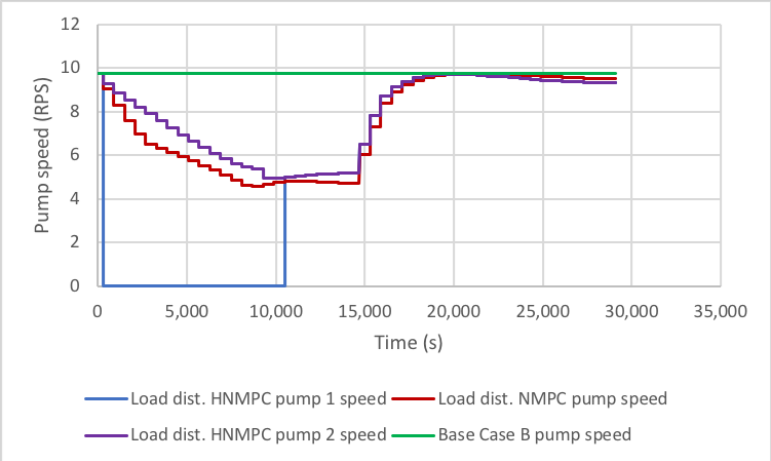

Figure 4.16. Cooling water pump speed with plant load disturbance.

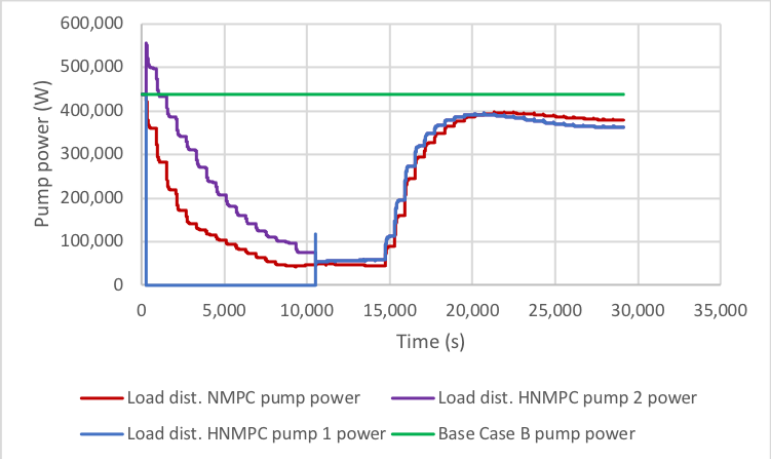

Figure 4.17. Cooling water pump power with plant load disturbance.

consumption in pump 2 drops initially when pump 1 is switched on again, as less cooling water is flowing through it when the other pump takes a lot more cooling water as it starts up.

The cooling water valve movements during both continuous and hybrid control are very similar, and are shown for valve 1 in Figure 4.18. The control valves move a lot more in the plant load disturbance scenario than the other scenarios since the effect of this disturbance on the temperature CVs is very significant with faster dynamics. This is due to the disturbance not being in the ambient conditions but very close to the CVs in the plant, thus effecting the CVs faster compared to the other disturbance scenarios.

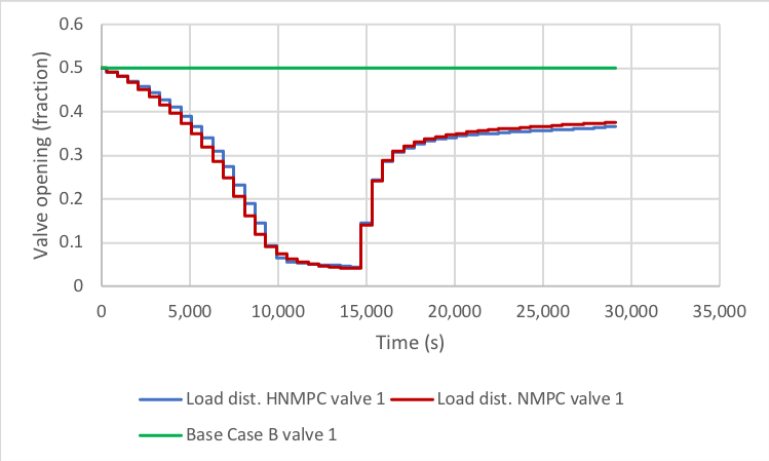

Figure 4.18. Cooling water valve 1 movement under continuous and hybrid control with plant load disturbance.

The cooling water valve movements for all the cooling water valves are shown in Figure 4.36 for the continuous control case only. The control action taken by the NMPC controller as shown in the trends is very interesting. The controller focuses the valve action on the temperatures that are affected the most by the disturbance - the outlet temperatures of HX-01, HX-02, HX-11 (see Figure 4.34). These temperatures drop by as much as 8 K from setpoint due to the disturbance before the controller can turn them around, and the associated valve openings are reduced significantly to assist the other main MVs in controlling these temperatures. However, as menioned above, the outlets of HX-03 and HX-04 actually increase in temperature and need more cooling, not less, from the controller, which is why especially valve 4 is opened more. Valve 9 is also opened by the controller since temperature HX-09 is also quite low, and will otherwise overshoot the setpoint due to the other MV actions.

Temperature of HX-08 and HX-10 are very close to setpoint by the end of this first 4 hour simulation period, but still below setpoint by less than 0.6 K. Therefore the associated valves 8 and 10 are slowly ramped down by the controller at this time to push temperatures up slightly. When the load is then restored and disturbance removed from the plant, the openings of these valves are reduced by the controller to prevent excessive cooling for already too cold temperature CVs due to the rest of the system at that time increasing its cooling capacity. Valves 3, 4, 5, 6, 7 and 9 are not moved much by the controller since the control action on the pumps and cooling towers are sufficient to bring the associated temperatures back to setpoint during the load disturbance injection and removal. Most valve openings converge back to values close to their original starting values when the load disturbance is

removed, but not exactly the same values since the objective function hyper space is slightly different at the end of the experiment. This is due to the pump speeds in particular running slightly lower than at the beginning of the experiment due to the power minimisation objective (see Figure 4.17).

The control valve openings for the hybrid control case for the plant load disturbance scenario, are trended in Figure 4.37.

4.4.3 Ambient temperature disturbance scenario

Changes in ambient conditions can have a large effect on the cooling tower performance. The ambient temperature changes in this scenario are made more rapidly than what would occur naturally in order to clearly demonstrate their effect. The performance of the controllers is compared to that of Base Case B as described in Section 4.2.2.2.

The NMPC and HNMPC controllers are required to maintain the hydrocarbon heat exchanger outlet temperatures at setpoint, and in addition to consume as little power as possible. The ambient temperature was first stepped down from a starting value of 25 °C to 5 °C and kept there for 4 hours to allow the controller to act and the variables to settle, then the temperature was stepped back to 25 °C for another 4 hour period (see Figure 4.19).

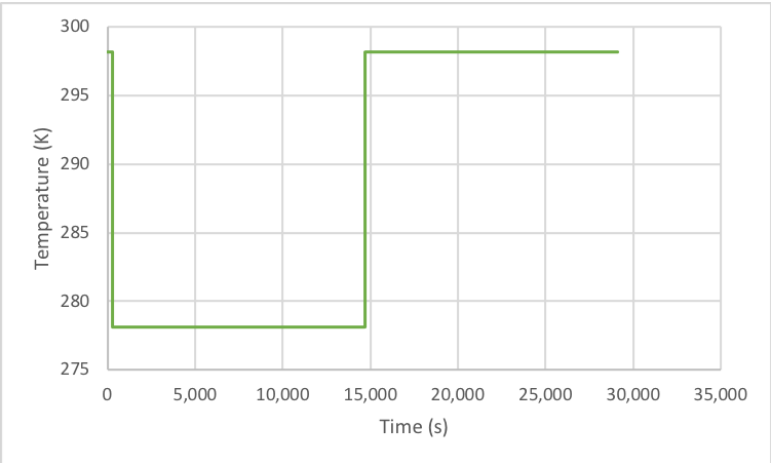

Figure 4.19. Ambient temperature during ambient temperature disturbance scenario.

Both the NMPC and HNMPC restrict the drop in hydrocarbon outlet temperature and limit the overshoot above setpoint when the ambient temperature is restored (see Figure 4.20). The NMPC performs well, but the HNMPC performs significantly better in this scenario.

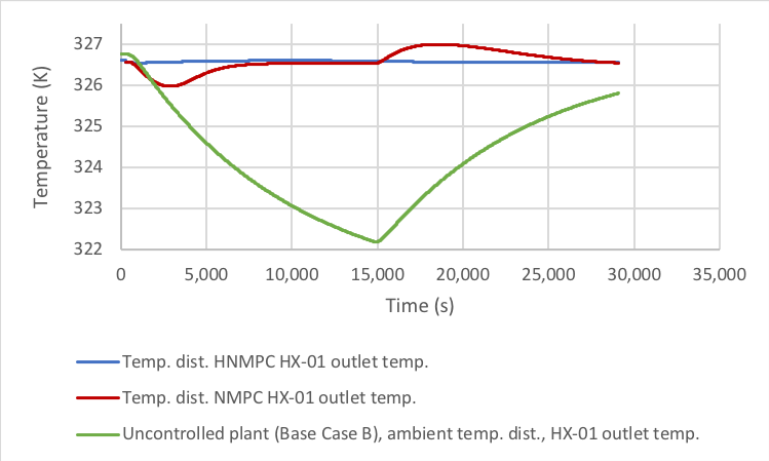

Figure 4.20. HX-01 outlet temperature during ambient temperature disturbance scenario.

All the heat exchanger hydrocarbon outlet temperature trends for the continuous control case are shown in Figure 4.38, and for the hybrid control case in Figure 4.39 for the temperature disturbance scenario.

The NMPC controller keeps reducing the CT fan speeds (Figure 4.21) in order to minimise cooling when the ambient temperature drops. The HNMPC controller does not change the fan speeds for CTs 2 and 3. It does however shut down the CT 1 fan when the ambient temperature drops, and starts it up again when the ambient temperature increases as shown in Figure 4.22. In this way the HNMPC controller balances the demand for power minimisation and temperature control.

The CT fans have more of an influence on the hydrocarbon temperature than the speed of the pumps (Figure 4.23). The HNMPC controller therefore uses the pump speed MVs to mainly reduce the power consumption (Figure 4.24) in the circuit. It can afford to only make small changes to the pump speeds as it has shut down a CT fan (Figure 4.21) in order to reject the ambient temperature disturbance. The NMPC controller does not have this luxury, and therefore its pump speed movements are more pronounced.

The control valve movements have less of an influence on the hydrocarbon temperature than the CT fans, and also a smaller influence on power consumption than the CW pumps. The HNMPC controller

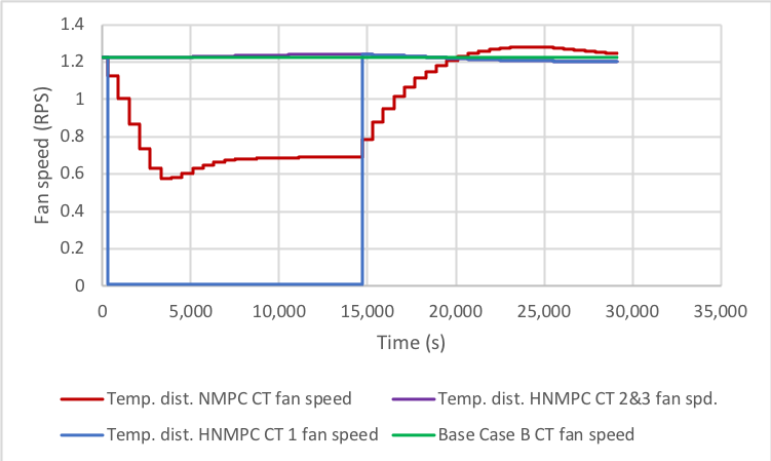

Figure 4.21. CT fan speed during ambient temperature disturbance scenario.

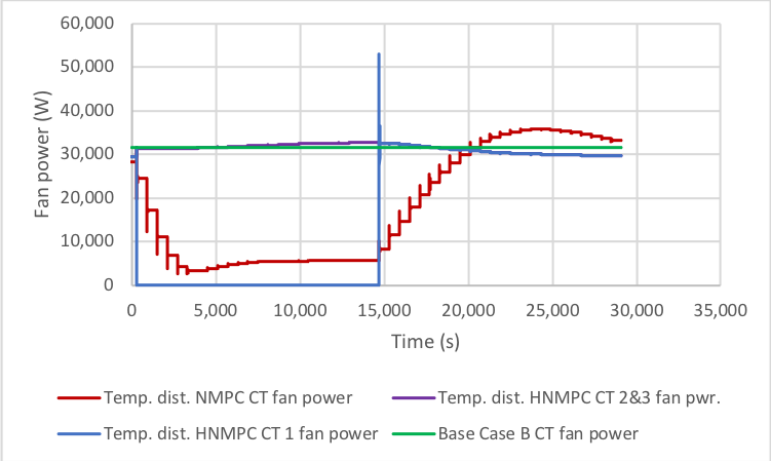

Figure 4.22. CT fan power during ambient temperature disturbance scenario.

does not need to change the valve opening much as it has shut down a CT fan (Figure 4.21) in order to reject the ambient temperature disturbance. The valve opening movements resulting from the NMPC controller are more pronounced as for the pump speeds (Figure 4.25).

All the control valve openings are trended for the continuous control case in Figure 4.40, and for the hybrid control case in Figure 4.41 for the ambient temperature disturbance scenario.

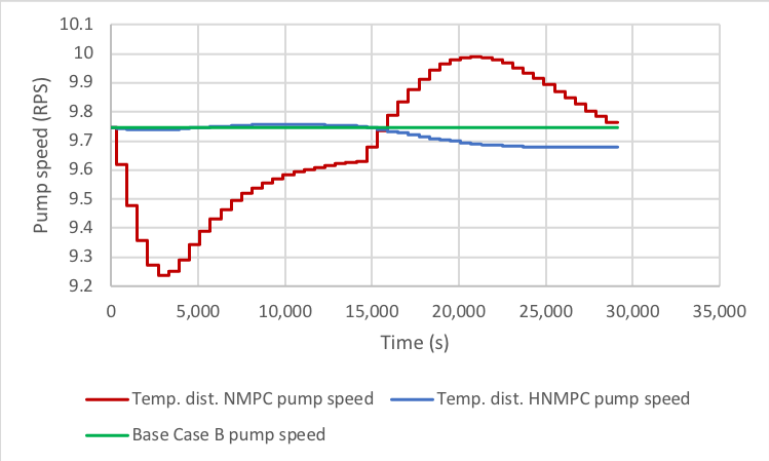

Figure 4.23. CT pumps speeds during ambient temperature disturbance scenario.

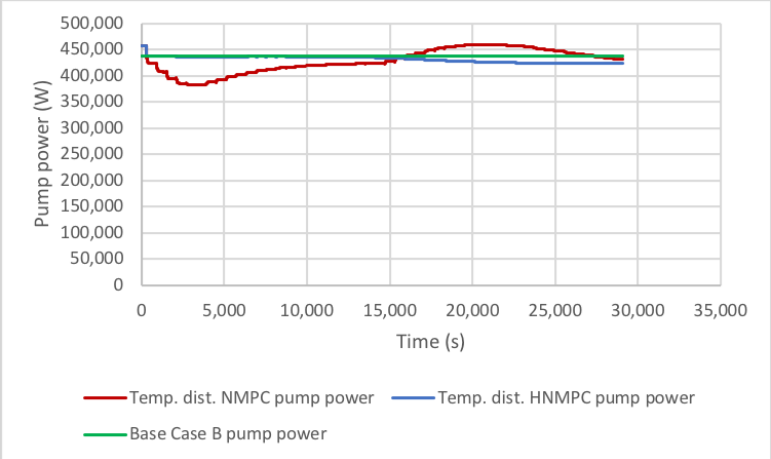

Figure 4.24. CT pumps power during ambient temperature disturbance scenario.

4.4.4 Ambient humidity disturbance scenario

The ambient humidity changes are made more rapidly than what would occur naturally. The performance of the controllers is compared to that of Base Case B as described in Section 4.2.2. The relative humidity was first stepped down from a starting value of 50% to 0% and kept there for 4 hours to allow the controller to act and the variables to settle, then the humidity was stepped back to 50% for another 4 hours (see Figure 4.26).

Both controllers contain the hydrocarbon outlet temperature well compared to the uncontrolled plant Base Case B response (Figure 4.27). The hybrid controller does slightly better since it has more

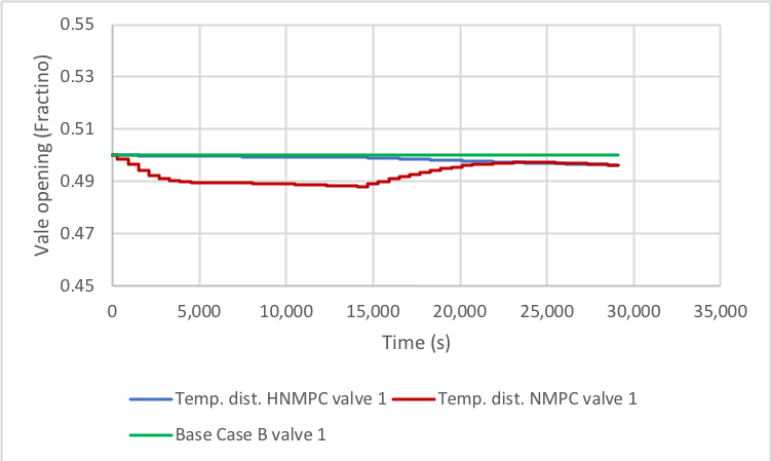

Figure 4.25. CW valve 1 movement during ambient temperature disturbance scenario.

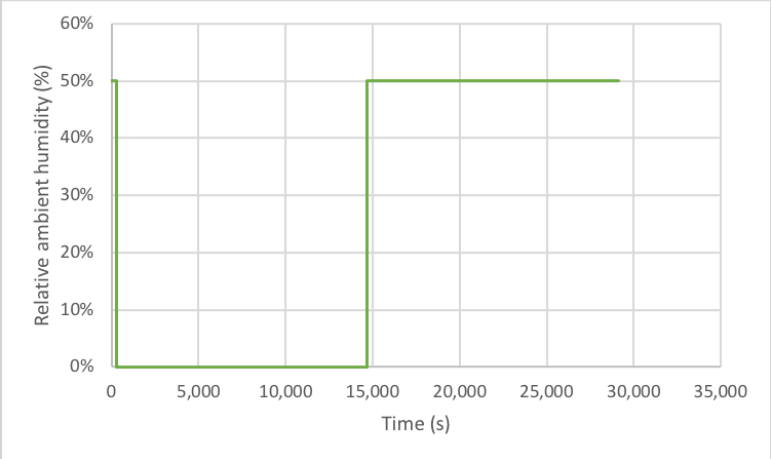

Figure 4.26. Ambient humidity steps to test humidity disturbance in the NMPC and HNMPC controller cases.

degrees of freedom with the Boolean MVs also at its disposal (see Table 4.5).

All the heat exchanger outlet temperatures for the continuous control case are shown in Figure 4.42, and for the hybrid control case in Figure 4.43 for the ambient humidity disturbance scenario.

Both controllers ramp down the CT fan speeds (Figure 4.28) in order to contain the drop in hydrocarbon temperature when the ambient humidity is stepped down. The HNMPC controller shuts down fan 1 for some time in order to further minimise power (Figure 4.29) and assist in containing the temperature drop. The opposite occurs when the humidity is stepped back up.

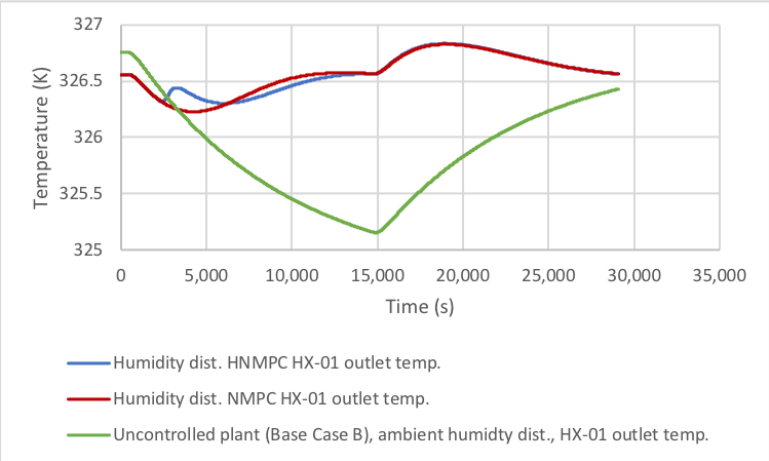

Figure 4.27. HX-01 outlet temperature during ambient humidity disturbance scenario.

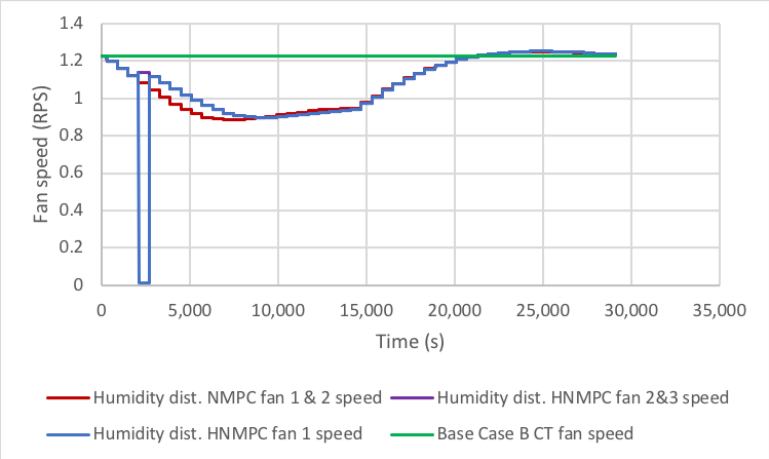

Figure 4.28. Cooling tower fan speed during ambient humidity disturbance scenario.

The cooling water pumps are used mainly by the controllers to minimise power, and less to control temperature. That is the reason that the pump speeds are not moved much over the simulation time (Figure 4.30).

Pump power consumption is further minimised over the life of the scenario (Figure 4.31), and when the temperature is under control again at the end of the scenario, the pump power consumption is slightly less than at the start of the scenario.

The cooling water valves are also moved to a much smaller degree by the controller during the ambient humidity steps. See Figure 4.32 for the trajectory of valve 1.

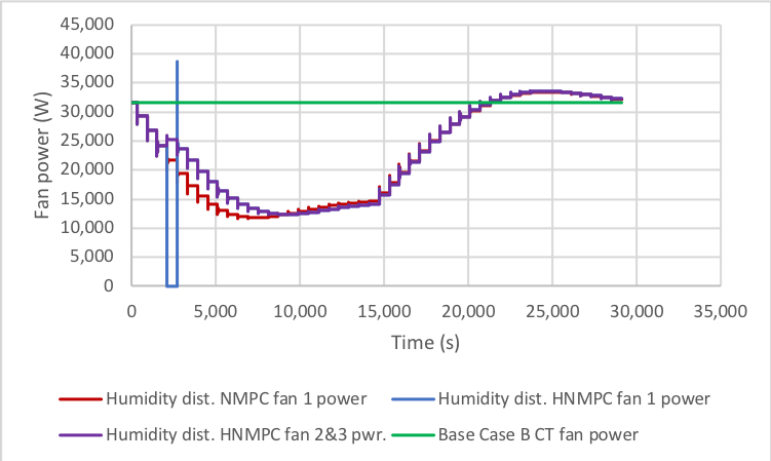

Figure 4.29. Cooling tower fan power during ambient humidity disturbance scenario.

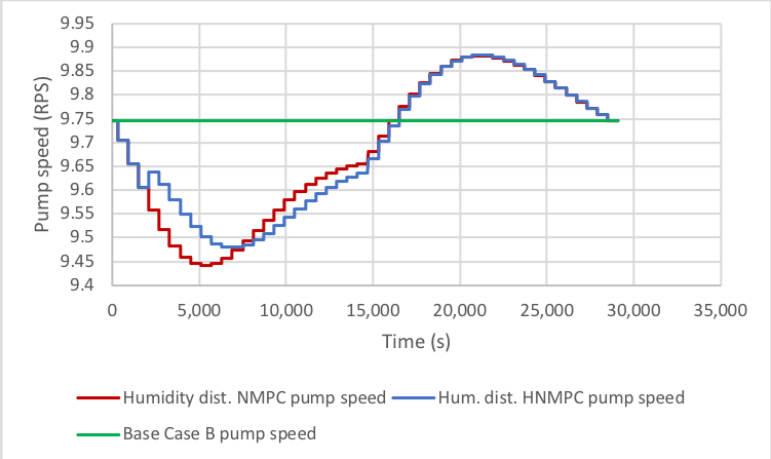

Figure 4.30. Cooling water pump speeds during ambient humidity disturbance scenario.

All the control valve openings trends for the continuous control case are shown in Figure 4.44, and for the hybrid control case in Figure 4.45 for the ambient humidity disturbance scenario.

4.4.5 Quantitative comparison and discussion

4.4.5.1 Base Case A

In Table 4.4, the power consumption and resulting electricity cost of the control strategies in the power consumption scenario are shown. Since no fast discrete action is required due to the slowly varying disturbances prevalent in this scenario, the NMPC and HNMPC controllers take the same action. The

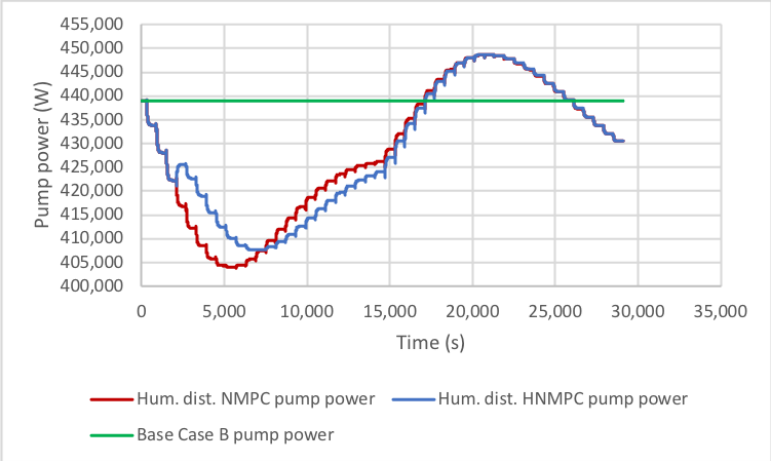

Figure 4.31. Cooling water pump power during ambient humidity disturbance scenario.

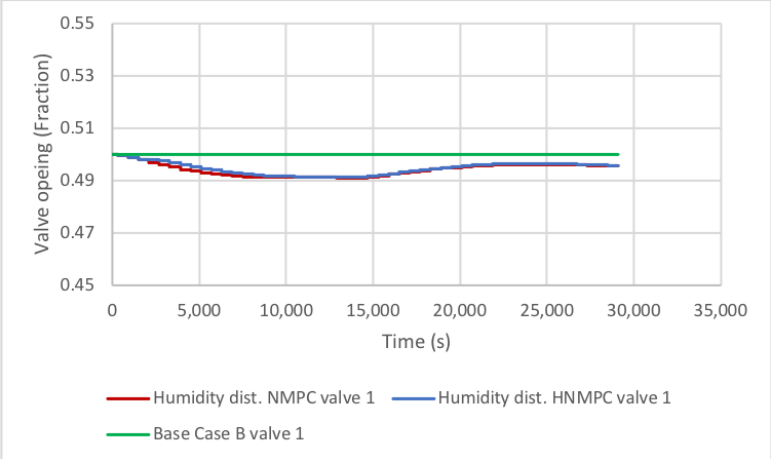

Figure 4.32. Cooling water valve 1 movement during ambient humidity disturbance scenario.

controllers are able to reduce the total power consumption and cost by 44.1% compared with Base Case A over the 24 hour simulation period (the electricity pricing is the same over the 24 hour period).

From Figure 4.9 and Figure 4.11 it can be seen that the controllers reduce the fan speeds by a greater degree than the pump speeds. The controllers find an optimal ratio between fan and pump speed for the cooling and temperature control requirement, given the power minimisation objective.

Table 4.4. Comparison of 24 hour power minimisation scenario.

Cntrl. strategy	**Power (kWh)**	**Cost (USD)**	**Power reduction vs. Base Case A**
Base Case A	52,826	2,536	0%
NMPC & HNMPC	29,543	1,418	44.1%

4.4.5.2 Base Case B

In Table 4.5 the Integral Square Error (ISE) sums and the total power consumption are shown for each scenario. The ISE is calculated as the difference between the normalised controlled variable (actual divided by engineering unit range) and the normalised setpoint, squared and totalled for the whole simulation period. The ISE figure tabled is the sum of the 11 temperature CV ISE figures.

Table 4.5. Comparison of 8 hour disturbance rejection scenarios and control strategy runs.

Scenario & control strategy	**Temp. ISE**	**Temp. ISE reduction vs. NMPC**	**Power (kWh)**	**Power reduction vs. Base Case B**
Load dist., NMPC	3.10E-04	0%	4,051	48.5%
Load dist., HNMPC	2.58E-04	16.6%	4,255	45.9%
Amb. temp. dist., NMPC	1.06E-05	0%	7,289	7.3%
Amb. temp. dist., HNMPC	2.33E-06	78.0%	7,537	4.1%
Amb. hum. dist., NMPC	5.99E-06	0%	7,395	5.9%
Amb. hum. dist., HNMPC	5.56E-06	7.3%	7,410	5.7%
Base Case B power			7,860	0%

Both the NMPC and HNMPC controllers succeeded in controlling the temperature well in the face of significant disturbances (see Figures 4.13, 4.20 and 4.27). The HNMPC controller performed better than the NMPC controller, particularly when rejecting the ambient temperature disturbance for which its temperature ISE score was 78% smaller. The reduction in power consumption shown in Table 4.5 is due mainly to the controllers turning down the rotating equipment in order to control temperature.

The power consumed by the NMPC and HNMPC controllers is less than that of Base Case B, with the biggest reduction occurring in the load disturbance scenario.

Since the objective function (4.1) is tuned to focus the controllers on temperature control (Table 4.3), the HNMPC controller uses the extra degrees of freedom it has compared to the continuous controller to further improve temperature control as can be seen in the second and third columns of Table 4.5. This improved temperature control comes at the cost of slightly higher power consumption compared to the NMPC controller (see last column of Table 4.5).

4.4.6 Monetary benefit

Monetary benefit estimation of advanced control and optimisation for the cooling water network is focused on electricity consumption minimisation in this work. Advanced process temperature control also has significant monetary benefits, but quantification of those benefits is different for each sub-plant serviced by the different cooling water heat exchangers, and is beyond the scope of this work. The estimated savings achieved by the NMPC/HNMPC control strategies is therefore derived from the power minimisation scenario and compared with Base Case A.

The NMPC and HNMPC controllers achieved a significant reduction in power consumption over Base Case A while maintaining the temperature at setpoint (Figure 4.8). These gains come from the reduction in fan and pump speeds as shown in Figures 4.10 and 4.12. The daily power cost savings is estimated to be USD 1,118 (2,536 - 1,418) as shown in Table 4.4. Assuming that the full design cooling capacity of the cooling water network is effectively not needed for 70% of the year and that these benefits can be achieved for that period of time, the annual total saving is USD 285,540.

The capital cost of making the cooling water network amenable to advanced control (Section 4.2.3) is estimated at USD 243,000. Assuming the implementation of HNMPC as per the design in this paper is done by the plant owner's control engineers, a cost of capital discount rate of 8%, and annual inflation of 2%, then the business case is a 10 year NPV of USD 1,829,251, an IRR of 120% and a 0.95 year payback period.

The electricity rates in the Arabian Gulf are typically lower than most other economies. The benefits of

the control strategies developed in this work could be increased in other markets because of potentially higher pricing of electricity. In addition, if improved temperature control is taken into account as well, the benefits of advanced control will be even larger.

4.4.7 Computation platform

All simulations were conducted on a 2.5 GHz Intel Core i7 computer with 16 GB memory. The code was compiled in C++ 11 and ran on the Mac OS 10.13 operating system. This platform takes 33.1 seconds to run the plant model [1] in open-loop for 8 hours of simulation time. Running the HNMPC controller in closed-loop requires an average of 66.99 seconds per iteration, which is well within the 10-minute running interval indicated in Table 4.3. It takes 54.1 minutes for the closed-loop simulation to complete an 8-hour simulation time period.

In Addendum A, an overview of the software design of the simulation environment and controller is documented.

4.5 CHAPTER SUMMARY

Various continuous controller designs and techniques were considered. These included Particle Swarm Optimisation, first and second order gradient based numerical optimisation techniques, and Reinforcement Learning methods. For the discrete side of the hybrid controller, PSO and tree based MINLP techniques were experimented with and considered. A tree based MINLP approach was designed and included in the HNMPC controller. Reasons for the design choices between alternatives were provided. Further details on the detail design, configuration and tuning of the chosen and developed methodologies were provided.

In this chapter, detailed results of all simulations performed with the designed plant model and NMPC and HNMPC controllers in closed-loop were provided, and discussed.

In Chapters 3 and 4 the methods applied as well as the results of the developed solutions to the research problem, were presented. This included discussion and analysis of results per open-loop step test and per control case and scenario.

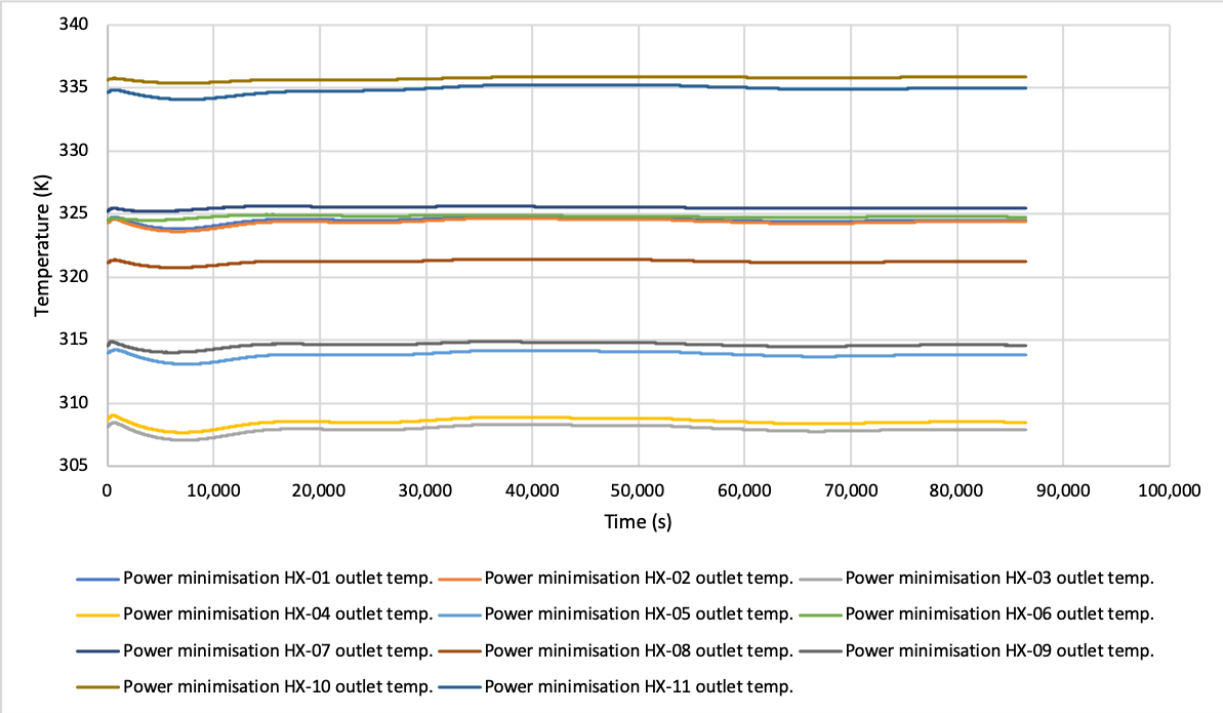

Figure 4.33. All heat exchanger outlet hydrocarbon temperatures, power minimisation scenario under continuous and hybrid control.

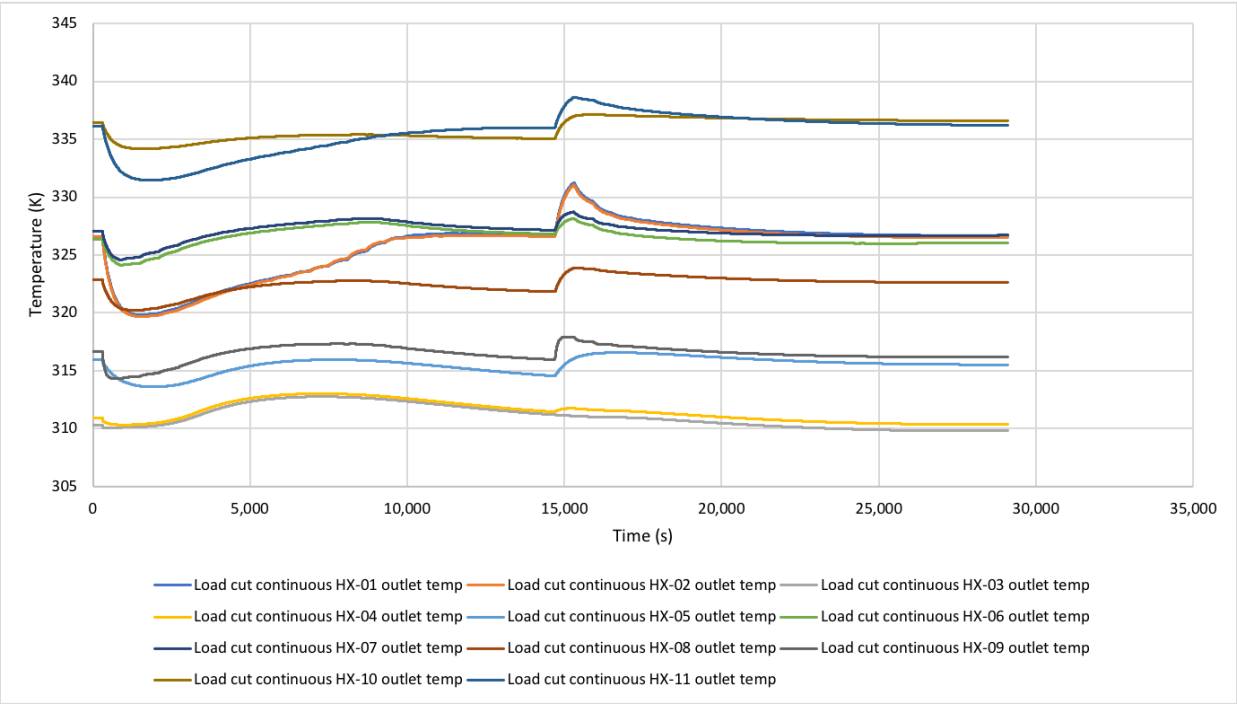

Figure 4.34. All heat exchanger outlet hydrocarbon temperatures, with plant load disturbance under continuous control.

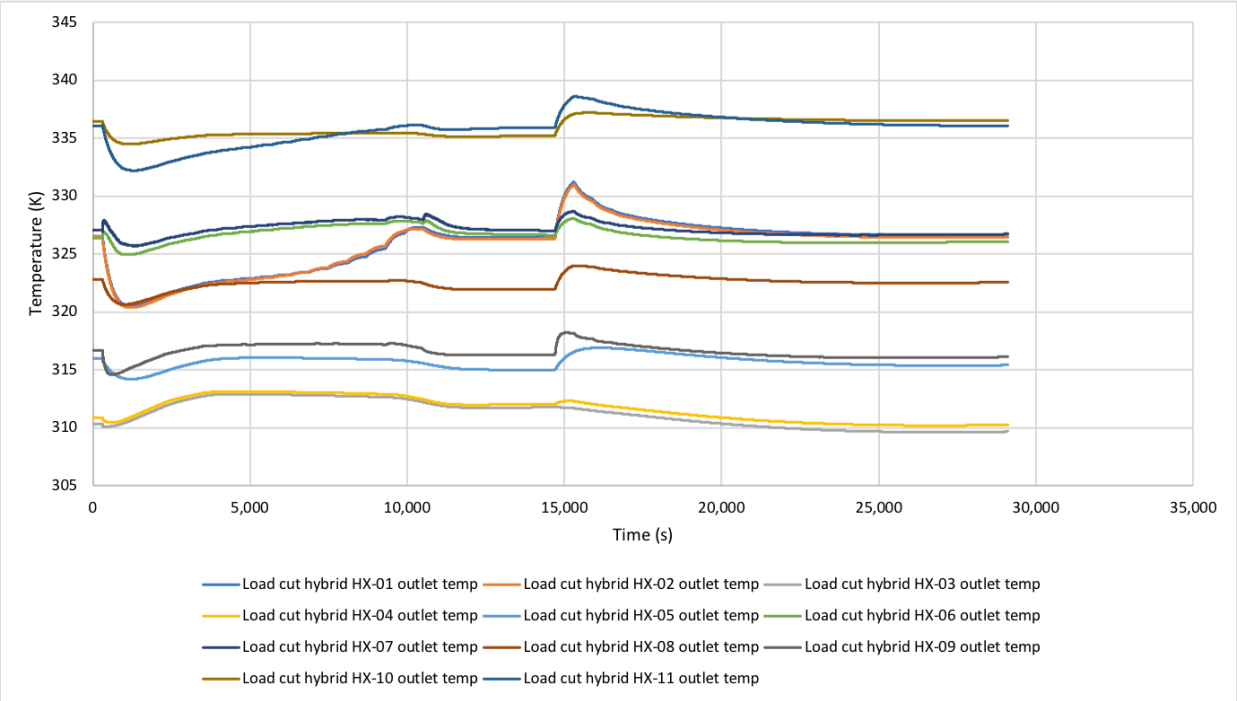

Figure 4.35. All heat exchanger outlet hydrocarbon temperatures, with plant load disturbance under hybrid control.

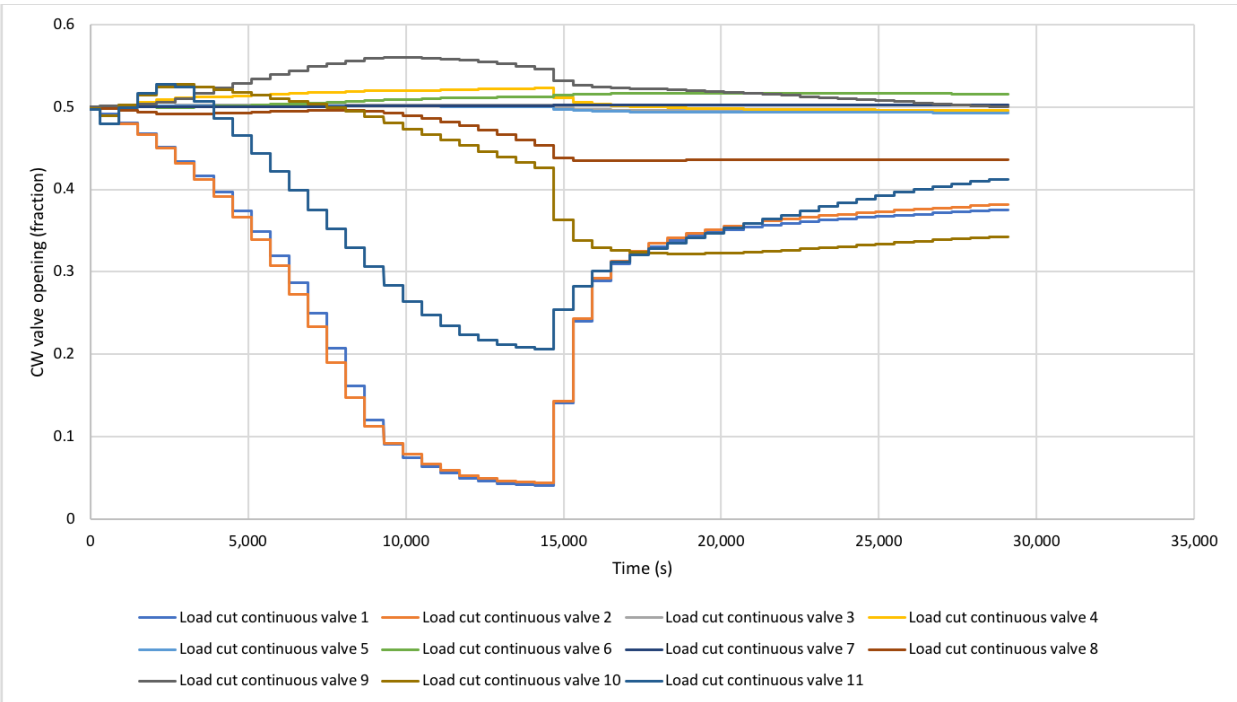

Figure 4.36. All cooling water valve movements under continuous control with plant load disturbance.

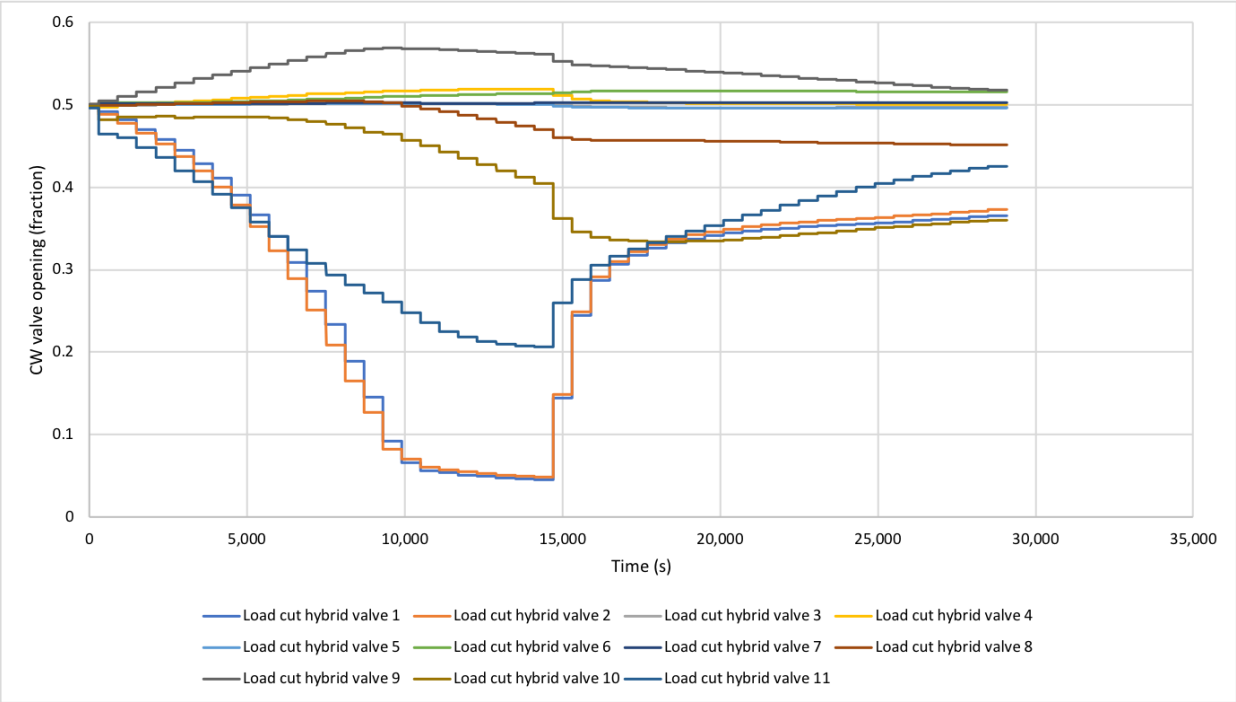

Figure 4.37. All heat exchanger CW control valve openings, with plant load disturbance under hybrid control.

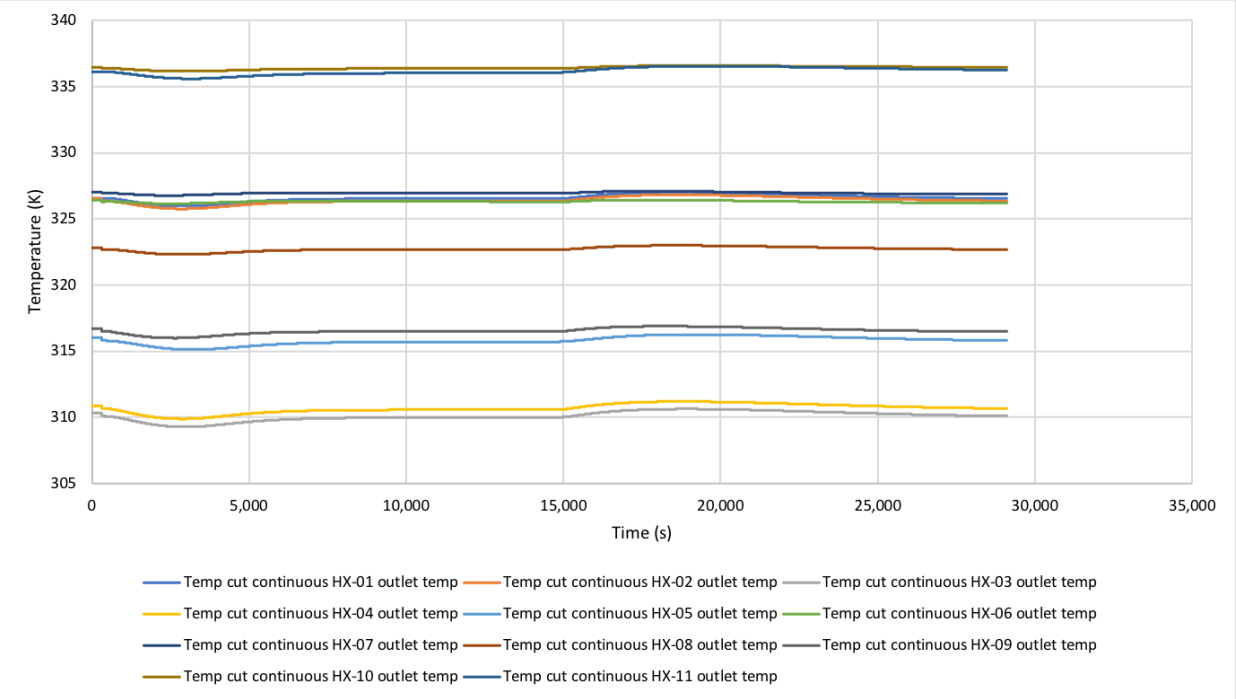

Figure 4.38. All heat exchanger outlet hydrocarbon temperatures, with ambient temperature disturbance under continuous control.

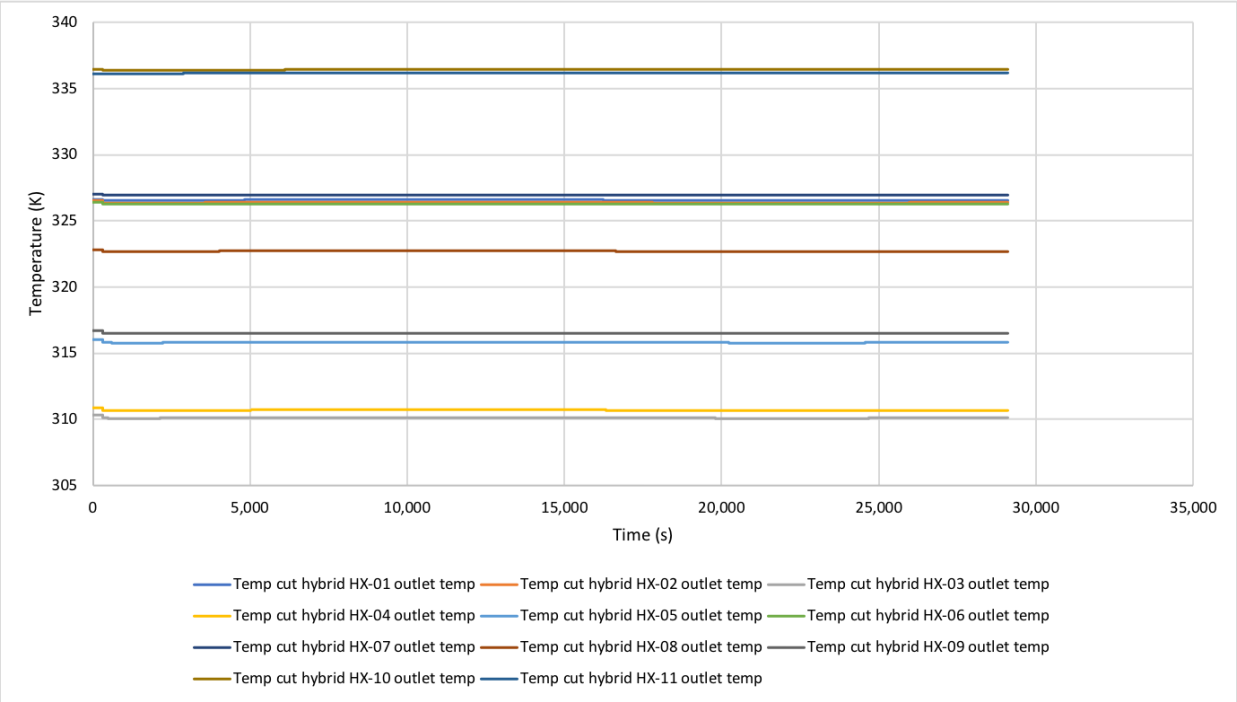

Figure 4.39. All heat exchanger outlet hydrocarbon temperatures, with ambient temperature disturbance under hybrid control.

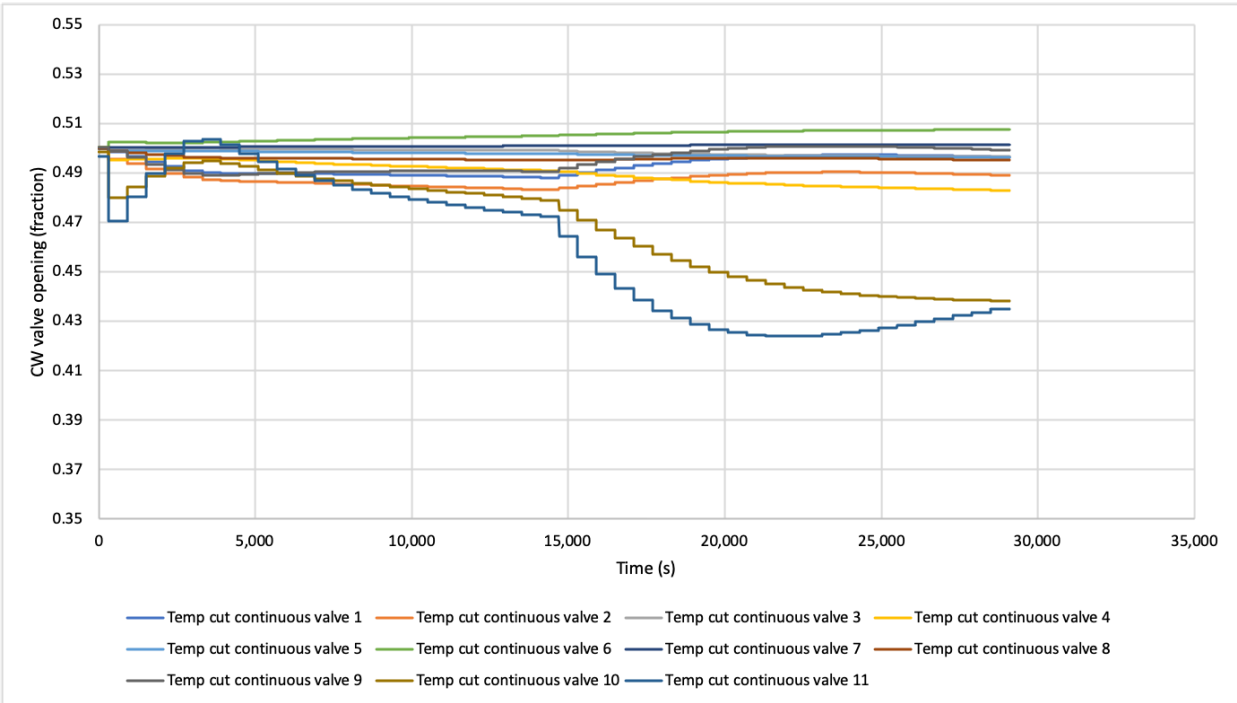

Figure 4.40. All heat exchanger CW control valve openings, with ambient temperature disturbance under continuous control.

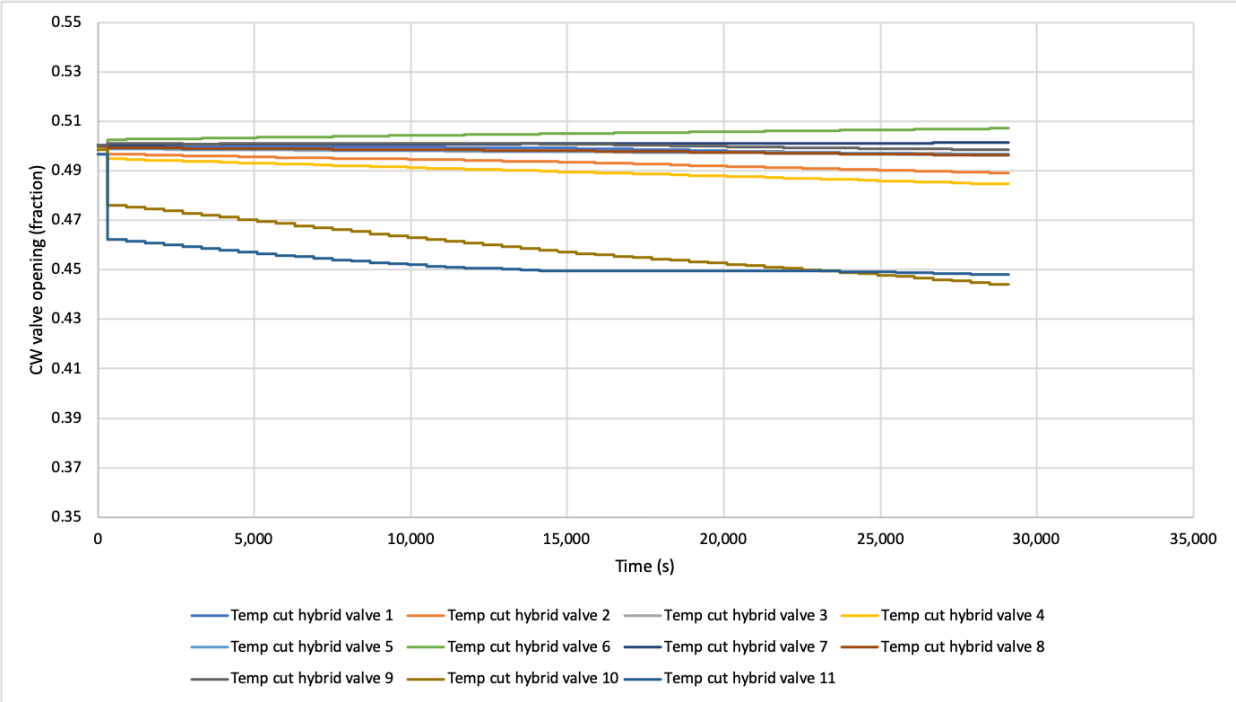

Figure 4.41. All heat exchanger CW control valve openings, with ambient temperature disturbance under hybrid control.

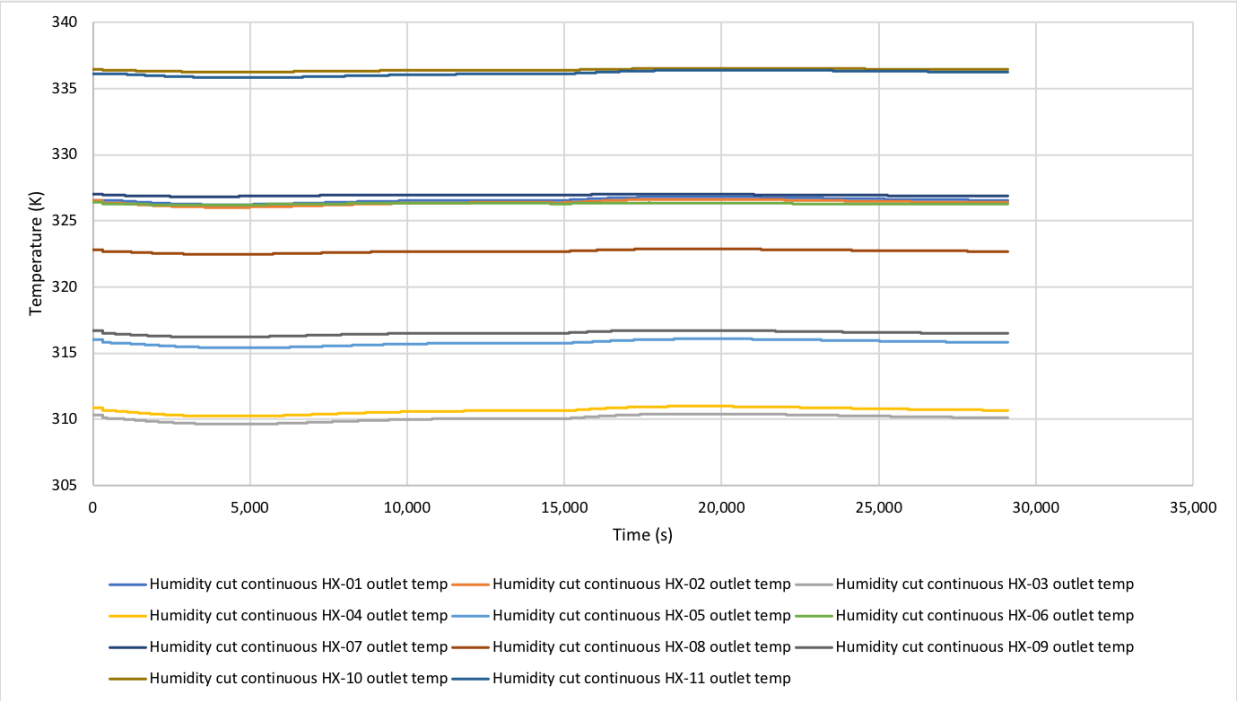

Figure 4.42. All heat exchanger outlet hydrocarbon temperatures, with ambient humidity disturbance under continuous control.

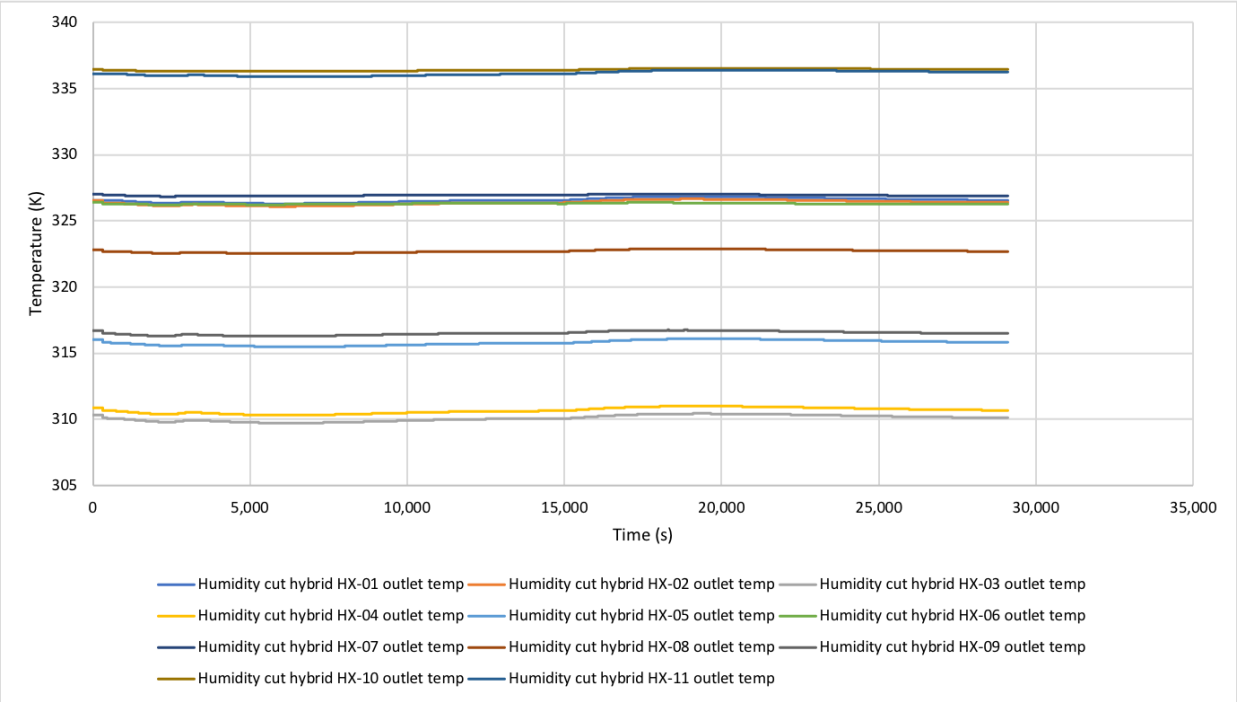

Figure 4.43. All heat exchanger outlet hydrocarbon temperatures, with ambient humidity disturbance under hybrid control.

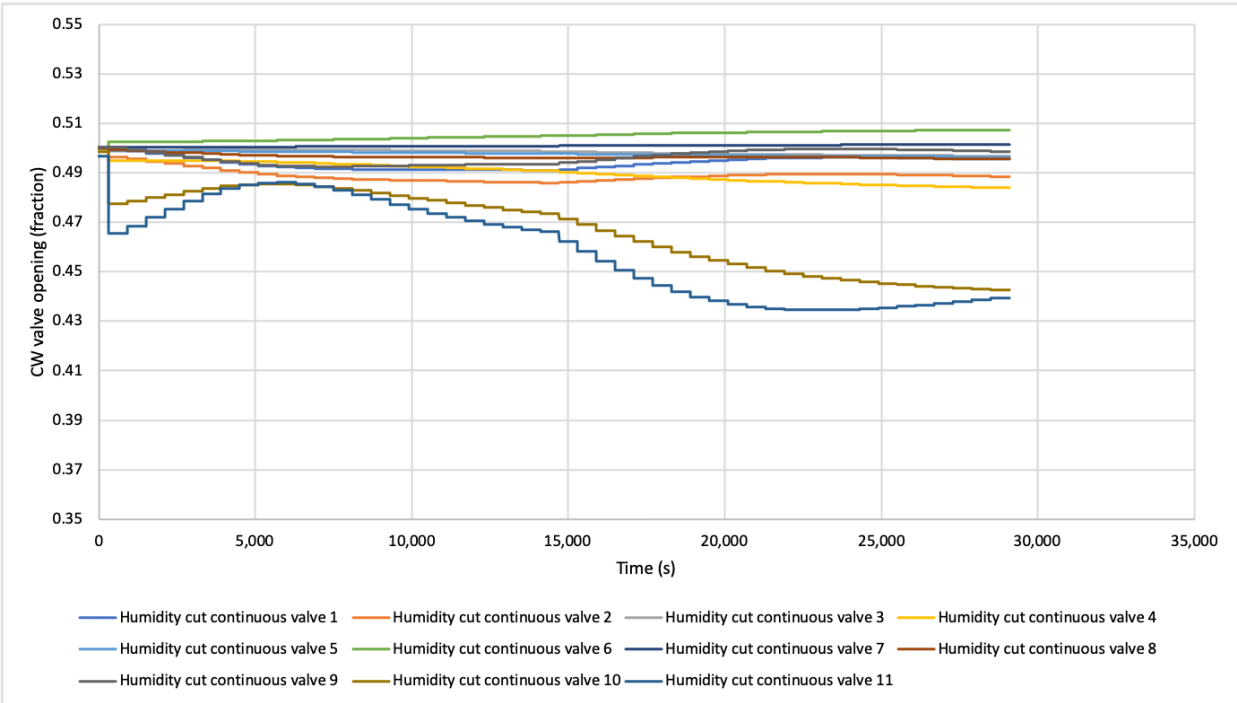

Figure 4.44. All heat exchanger CW control valve openings, with ambient humidity disturbance under continuous control.

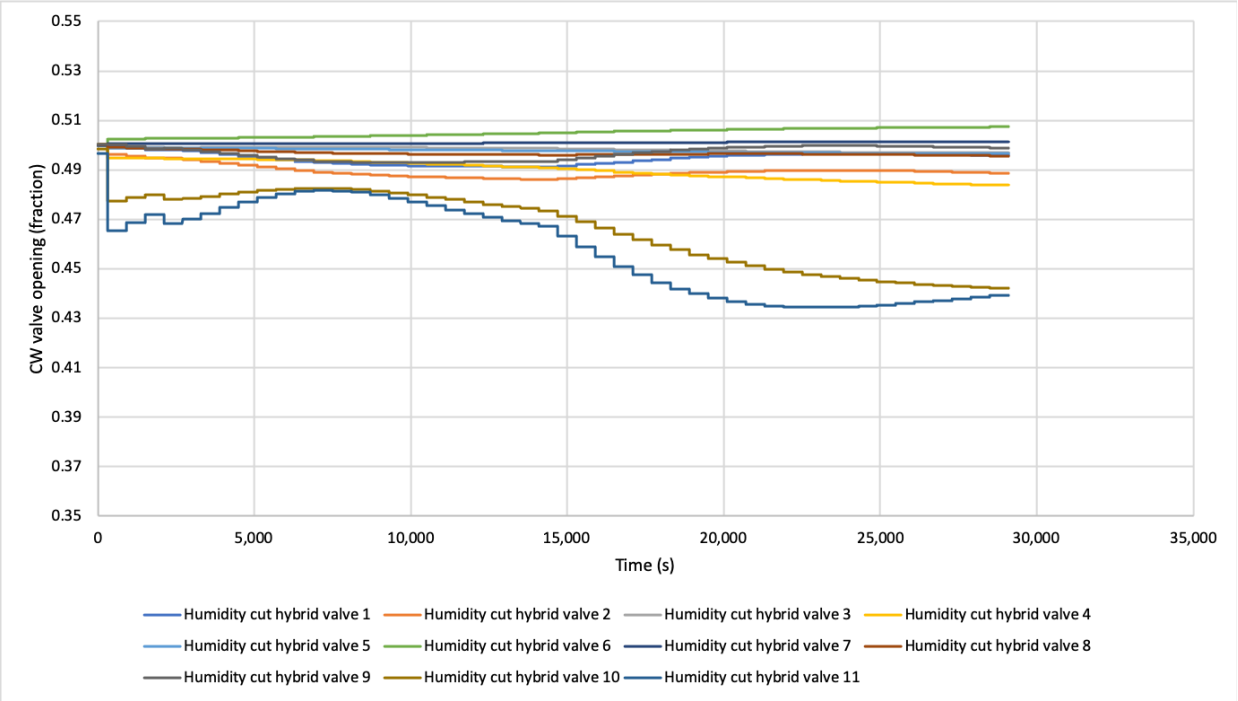

Figure 4.45. All heat exchanger CW control valve openings, with ambient humidity disturbance under hybrid control.

CHAPTER 5 DISCUSSION & CONCLUSION

5.1 CHAPTER OVERVIEW

In this chapter in Section 5.2 the results from Chapters 3 and 4 are summarised, consolidated, discussed and analysed.

At the end of the chapter, an overall conclusion for the work is derived in Section 5.3. A proposed list of future research directions to build on this work, is included in Section 5.4.

5.2 DISCUSSION

5.2.1 Plant modelling discussion

The plant exhibits strong non-linear and integrated hydraulic and thermodynamic behaviour. This is true of both steady-state process gains, and times to steady-state.

There are significant differences in time to steady-state between some of the MVs and the CVs. In general the closer the MVs are to the hydrocarbon streams, the faster the process dynamics. Changes in the water side of the cooling tower (e.g. pump speed changes), affect the process temperatures significantly faster (1 hour time to steady-state) than air side changes (e.g. the process takes 4 hours or more to reach steady-state after changes in the DVs ambient humidity and ambient temperature, and the MV fan speed).

The integrated hydraulics of the plant is seen in the dynamic transients when the mass flow of the circuit is changed (e.g. Figure 3.23). The non-linear nature of the installed hydraulic characteristic of the cooling water flow valve to each exchanger is seen in Figure 3.22. The power consumed by the rotating equipment in the circuit exhibits a non-linear relationship with respect to plant state variables, especially fan power with respect to speed (Figure 3.11). The plant temperatures also exhibit a non-linear relationship with the cooling tower on-off status as shown in Figure 3.14.

This non-linear behaviour further justifies the use of a non-linear control solution to achieve the control objectives of the research. Using NMPC and HNMPC thus enables the controllers to control the plant over a wide operating region, including different discrete states for the fans and pumps.

5.2.2 Hybrid Non-Linear Model Predictive Controller discussion

From Figure 4.34 and Figure 4.36 it can be seen that the gradient descent control action results in more aggressive control action for CVs that are further away from set point, which is desired behaviour. For CVs closer to set point (within 2 K in the Figure 4.34) the control action takes longer to convergence to the optimum point. This is an expected result that is reflected by the literature on gradient descent optimisation (see Section 2.3.5). Tuning the objective function could change behaviour a little, as could using a second order optimisation algorithm. Second order optimisation has other problems associated with it though, and this slower convergence closer to set point behaviour would be desired for many practical process scenarios.

5.3 CONCLUDING REMARKS

A first principles dynamic model for the plant was developed in this research. The plant model was fitted and validated using real process data. The model was shown to perform as expected in open-loop simulation, and is well suited to be used as the internal model of the HNMPC controller developed to solve the control problem of the cooling water network.

The overall model developed and simulated in this work is aimed at a specific process layout (Figure 3.1). However, the developed models are for unit operations, which have been linked in a modular fashion through the hydraulic modelling performed. The developed models can also be used for a

different process layout, when fitted to relevant data and linked through the appropriately configured hydraulic models.

A first principles dynamic model, augmented with Hammerstein models for some of the unit operations, as constructed in this work is also very useful in order to analyse the performance of the actual plant, identify steady-state deviations from design early, online dynamic early event detection, as well as in operator training.

In this work, NMPC techniques are applied to control and optimise the cooling water network. MVs that are both continuous and discrete are applied and compared with each other. The non-linear optimisation technique applied as part of the optimisation and control algorithm is gradient descent for the continous MVs [61, 92], and graph traversal for the Boolean MVs [83].

The control strategies are simulated for disturbance rejection scenarios typically experienced by a real plant. Three cases are introduced - a base case with constant MVs, a controller with continuous MVs, and a hybrid controller with continuous and discrete MVs. The performance achieved with the three scenarios are compared and discussed.

The work described in this dissertation shows that HNMPC, combined with a verified fundamental process model, can successfully control hydrocarbon temperatures, as well as minimise power consumption for the cooling water network studied. Continuous MVs can run on their own in a continuous NMPC controller, and can then also be combined with Boolean MVs in a HNMPC controller where the rotating equipment are switched on and off as needed. The introduction of Boolean MVs allows for improved performance as compared to when only continuous MVs are used as in the case of the NMPC.

The NMPC and HNMPC controllers are able to exploit the power minimisation opportunity whenever the cooling requirement and ambient conditions allow it by optimally decreasing fan and pump speeds from their original design settings while controlling temperature. This results in significant annual savings in electricity costs for the plant owner, as estimated in Section 4.4.6.

The HNMPC controller is able to perform well for a wide range of plant states and conditions. Large plant disturbances such as reduction of the hydrocarbon mass flow to the cooling water exchangers,

and large changes in ambient temperature and humidity conditions, can be controlled, contained and managed by the HNMPC control system.

The HNMPC controller described in this work performs better than the NMPC controller due to it having access to Boolean MVs and hybrid plant states in addition to the continuous MVs. This greater flexibility enables the HNMPC controller to better minimise the objective function and drive the plant towards an optimum state.

Overall hybrid NMPC has great potential for online optimisation of the cooling water networks of processing facilities. Normally plants of this kind do not have variable speed drives for the fans and pumps. This work has shown that these MVs are very effective in enabling a NMPC to reject disturbances that typically occur on an industrial plant. A capital investment roadmap could take a phased approach to first install Variable Speed Drives and continuous NMPC, followed by a second phase of Hybrid NMPC. Power consumption cost savings [30, 113] should be able to justify a business case for actuation and HNMPC control of such plants.

A powerful graphical design environment was designed and built utilising polymorphism and Object Oriented Programming to construct the models and controllers in order to execute the open-loop and closed-loop simulation and experiments.

5.4 FUTURE WORK

Possible future research directions to build on this work are:

1. The addition of setpoint change simulation scenarios to complement the disturbance rejection control focused on in this work.
2. The use of gradient descent techniques for the parameter fitting task for fitting the developed plant model to the plant data.
3. The application of different gradient descent non-linear optimisation algorithms (e.g. Nesterov, ADAM [86] etc.) in addition to the momentum algorithm applied in this work.
4. The design and simulation based on a suitable optimal filtering technique of a non-linear robust observer for the system assuming the entire system state is not measured during online control.

5. The addition of noise and/or model error between the plant model, and the controller internal model to test the robustness of the controller and improve its design towards this end.
6. The use of deep Reinforcement Learning [84, 108, 109] to implement an adaptive control strategy that will learn an optimal policy for the cooling water network by interacting with the plant model. The initial policy could be learned through simulation in closed-loop with the plant model, and then once it has sufficiently converged, it could be applied to the real plant where it will keep learning if the true plant state could be observed. One potentially promising aspect of this research direction, is that the policy that will be learned could potentially handle discrete as well as continuous MV outputs if the neural network architecture and training algorithm is correctly designed. This has been shown to work in other problem domains [84].

REFERENCES

[1] J. H. Viljoen, C. J. Muller, and I. K. Craig, "Dynamic modelling of induced draft cooling towers with parallel heat exchangers, pumps and cooling water network," *Journal of Process Control*, vol. 68, pp. 34–51, 2018.

[2] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. London, UK: Springer-Verlag, 2007.

[3] C. J. Muller and I. K. Craig, "Modelling of a dual circuit induced draft cooling water system for control and optimisation purposes," *Journal of Process Control*, vol. 25, pp. 105–114, 2015.

[4] M. M. Castro, T. W. Song, and J. M. Pinto, "Minimization of operational costs in cooling water systems," *Transactions of the Institution of Chemical Engineers*, vol. 78, pp. 192–201, 2000.

[5] G. F. Cortinovis, M. T. Ribeiro, J. L. Paiva, T. W. Song, and J. M. Pinto, "Integrated analysis of cooling water systems: Modeling and experimental validation," *Applied Thermal Engineering*, vol. 29, pp. 3124–3131, 2009.

[6] P. Malinowski, M. Sulowicz, and J. Bujak, "Neural model for forecasting temperature in a distribution network of cooling water supplied to systems producing petroleum products," *International Journal of Refrigeration*, vol. 34, pp. 968–979, 2011.

[7] J. R. Eliot and C. T. Lira, *Introductory Chemical Engineering Thermodynamics*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice-Hall, 2012.

[8] D. Green and R. Perry, *Perry's Chemical Engineers' Handbook*, 8th ed. New York, USA: McGraw-Hill, 2007.

[9] P. Ahuja, *Numerical Methods in Chemical Engineering*. New Delhi, India: PHI Learning Private Limited, 2010.

[10] S. Skogestad, *Chemical and Energy Process Engineering*, 1st ed. Boca Raton, FL, USA: CRC Press, 2009.

[11] J. C. Kloppers and D. G. Kröger, "A critical investigation into the heat and mass transfer analysis of wet counter flow cooling towers," *International Journal of Heat & Mass Transfer*, vol. 48, pp. 765–777, 2005.

[12] X. Li and Y. Li, "Dynamic modeling of mechanical draft counter-flow wet cooling tower with modelica," in *Proceedings of the International Refridgeration and Air Conditioning Conference*. West Lafayette, USA: Purdue University, July 2010, pp. 2322–2330.

[13] A. Löfgren, "Modeling and simulation of a cooling tower with extended uses," Master's thesis, Department of Chemical Engineering, Lund University, Lund, Sweden, 2015.

[14] W. L. McCabe, J. C. Smith, and P. Harriott, *Unit Operations of Chemical Engineering*, 7th ed. New York, USA: McGraw-Hill, 2005.

[15] R. Stull, "Wet-bulb temperature from relative humidity and air temperature," *Journal of Applied Meteorology and Climatology*, vol. 50, pp. 2267–2269, 2011.

[16] S. P. Fisenko, A. A. Brin, and A. I. Petruchik, "Evaporative cooling of water in a mechanical draft cooling tower," *International Journal of Heat and Mass Transfer*, vol. 47, pp. 165–177, 2004.

[17] E. Rubio-Castro, M. Serna-Gonzalez, J. M. Ponce-Ortega, and M. M. El-Halwagi, "Synthesis of cooling water systems with multiple cooling towers," *Applied Thermal Engineering*, vol. 50, pp. 957–974, 2013.

[18] M. Picon-Nunez, G. T. Polley, L. Canizalez-Davalos, and E. K. Tamakloe, "Design of coolers for use in an existing cooling water network," *Applied Thermal Engineering*, vol. 43, pp. 51–59, 2012.

[19] J. M. Ponce-Ortega, M. Serna-Gonzalez, and A. J. Gutierrez, "Optimization model for re-circulating cooling water systems," *Computers and Chemical Engineering*, vol. 34, pp. 177–195, 2010.

[20] M. A. Al-Nimr, "Dynamic thermal behaviour of cooling towers," *Energy Conversion Management*, vol. 39, pp. 631–636, 1998.

[21] Z. Ma, S. Wang, and F. Xiao, "Online performance evaluation of alternative control strategies for building cooling water systems prior to in situ implementation," *Applied Energy*, vol. 86, pp. 712–721, 2008.

[22] F. Alnaimat, J. F. Klausner, and R. Mei, "Transient analysis of direct contact evaporation and condensation within packed beds," *International Journal of Heat and Mass Transfer*, vol. 54, pp. 3381–3393, 2011.

[23] M. Nasrabadi and D. P. Finn, "Mathematical modeling of a low temperature low approach direct cooling tower," *Applied Thermal Engineering*, vol. 64, pp. 273–282, 2014.

[24] S. Middleman, *An Introduction to Mass and Heat Transfer - Principles of Analysis and Design*. New York, USA: John Wiley & Sons, 1998.

[25] C. A. X. Marques, C. H. Fontes, M. EmbiruÃğu, and R. A. Kalid, "Efficiency control in a commercial counter flow wet cooling tower," *Energy Conversion and Management*, vol. 50, pp. 2843–2855, 2009.

[26] J. C. Kloppers and D. G. Kröger, "The Lewis factor and its influence on the performance prediction of wet-cooling towers," *International Journal of Thermal Sciences*, vol. 44, pp. 879–884, 2005.

[27] Y. Huang, K. Zhang, S. Yang, and Y. Jin, "A method to measure humidity based on dry-bulb and wet bulb temperatures," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 16, pp. 2984–2987, 2013.

[28] A. L. Buck, "New equations for computing vapour pressure pressure and enhancement factor," *Journal of applied meteorology*, pp. 1527–1532, 1981.

[29] W. K. Lewis, "The evaporation of a liquid into a gas," *Trans. ASME*, vol. 44, pp. 325–340, 1922.

[30] C. J. Muller and I. K. Craig, "Energy reduction for a dual circuit cooling water system using advanced regulatory control," *Applied Energy*, vol. 171, pp. 287–295, 2016.

[31] M. Vaccarini, A. Carbonari, and M. Casals, "Development and calibration of a model for the dynamic simulation of fans with induction motors," *Applied Thermal Engineering*, vol. 111, pp. 647–659, 2017.

[32] E. Al-Bassam and R. Alasseri, "Measurable energy savings of installing variable frequency drives for cooling towers fans, compared to dual speed motors," *Energy and Buildings*, vol. 67, pp. 261–266, 2013.

[33] R. Saidur, S. Mekhilef, M. B. Ali, A. Safari, and H. A. Mohammed, "Applications of variable speed drive (VSD) in electrical motors energy savings," *Renewable and Sustainable Energy Reviews*, vol. 16, pp. 543–550, 2012.

[34] P. C. Sen, *Principles of Electric Machines and Power Electronics*. USA: John Wiley and Sons, 1997.

[35] J. Szymczyk and K. Karaskiewicz, "The modeling of centrifugal pump transients," *Transaction of the Institute of Fluid Flow Machinery - Polish Academy of Sciences*, vol. 130, pp. 83–92, 2015.

[36] Y. Wang and Q. Chen, "A direct optimal control strategy of variable speed pumps in heat exchanger networks and experimental validations," *Energy*, vol. 85, pp. 609–619, 2015.

[37] J. Sun, X. Feng, and Y. Wang, "Cooling-water system optimisation with a novel two-step sequential method," *Applied Thermal Engineering*, pp. 1–8, 2015.

[38] W. Ramirez, *Computational Methods for Process Simulation*, 2nd ed. Oxford, UK: Butterworth-Heineman, 1997.

[39] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, *Process Dynamics and Control*, 4th ed. Danvers, MA, USA: John Wiley and Sons, 2016.

[40] D. K. Rollins, N. Bhandari, A. M. Bassily, G. M. Colver, and S. T. Chin, "A continuous-time nonlinear dynamic predictive modeling method for hammerstein processes," *Ind. Eng. Chem. Res.*, vol. 42, pp. 860–872, 2003.

[41] E. F. Camacho, D. R. Ramirez, D. Limon, D. M. de la Pena, and T. Alamo, "Model predictive control techniques for hybrid systems," *Annual Reviews in Control*, vol. 34, pp. 21–31, 2010.

[42] S. A. Billings, "Identification of nonlinear systems: A survey," *IEE Proceedings Part D*, vol. 127, no. 6, pp. 272–285, 1980.

[43] T. Gao, B. G. Sammakia, B. T. Murray, A. Ortega, and R. Schmidt, "Cross flow heat exchanger modeling of transient temperature input conditions," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 11, pp. 1796–1807, 2014.

[44] M. Bakosova and J. Oravec, "Robust model predictive control for heat exchanger network," *Applied Thermal Engineering*, vol. 73, pp. 924–930, 2014.

[45] C. H. Edwards and D. E. Penney, *Elementary Differential Equations with Boundary Value Problems*. New Jersey, USA: Prentice-Hall, 1993.

[46] S. M. Safdarnejad, J. R. Gallacher, and J. D. Hedengren, "Dynamic parameter estimation and optimization for batch distillation," *Computers & Chemical Engineering*, vol. 86, pp. 18–32, 2016.

[47] K. Dai and N. Wang, "A hybrid DNA based genetic algorithm for parameter estimation of dynamic systems," *Chemical Engineering Research and Design*, vol. 90, pp. 2235–2246, 2012.

[48] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*. Perth, Australia: IEEE, December 1995, pp. 1942–1948.

[49] ——, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Computational Cybernetics and Simulation*. Orlando, USA: IEEE, October 1997, pp. 4104–4108.

[50] M. Clerc, *Particle Swarm Optimisation*. London, UK: ISTE Ltd, 2006.

[51] Q. B. Jin, Z. J. Cheng, J. Dou, L. T. Cao, and K. W. Wang, "A novel closed loop identification method and its application of multivariable system," *Journal of Process Control*, vol. 22, pp. 132–144, 2012.

[52] K. Deng, Y. Sun, S. Li, Y. Lu, J. Brouwer, P. G. Mehta, M. Zhou, and A. Chakraborty, "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming," *IEEE Trans. on Automation Science and Engineering*, vol. 12, no. 2, pp. 565–579, 2015.

[53] X. H. Yin and S. Li, "Energy efficient predictive control for vapor compression refrigeration cycle systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, 2018.

[54] C. J. Muller, I. K. Craig, and N. L. Ricker, "Modelling, validation, and control of an industrial fuel gas blending system," *Journal of Process Control*, vol. 21, no. 6, 2011.

[55] C. J. Muller and I. K. Craig, "Economic hybrid non-linear model predictive control of a dual circuit induced draft cooling water system," *Journal of Process Control*, vol. 53, pp. 37–45, 2017.

[56] N. L. Ricker, C. J. Muller, and I. K. Craig, "Fuel-gas blending benchmark for economic performance evaluation of advanced control and state estimation," *Journal of Process Control*, vol. 22, no. 6, pp. 968–974, 2012.

[57] R. Dzedzemane, J. D. le Roux, C. J. Muller, and I. K. Craig, "Steam header state-space model development and validation," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 207–212, 2018.

[58] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, 1st ed. London, UK: Springer-Verlag, 2009.

[59] M. Bauer and I. K. Craig, "Economic assessment of advanced process control - a survey and framework," *Journal of Process Control*, vol. 18, pp. 2–18, 2008.

[60] X. Li, Y. Li, J. E. Seem, and P. Li, "Extremum seeking control of cooling tower for self-optimizing efficient operation of chilled water systems," in *Proceedings of the 2012 American Control Conference*. Montréal, Canada: IFAC, June 2012, pp. 2322–2330.

[61] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control, Theory and Algorithms*, second edition ed. Springer, Switzerland: Oxford University Press, 2017.

[62] F. Allgower and A. Zheng, *Nonlinear Model Predictive Control*. Basel, Switzerland: Springer Basel, 2000.

[63] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, pp. 2967–2986, 2014.

[64] R. Findeisen, F. Allgower, and L. T. Biegler, *Assessment and Future Directions of Nonlinear Model Predictive Control*. Berlin, Germany: Springer-Verlag, 2007.

[65] L. T. Biegler, X. Yanga, and G. A. G. Fischer, "Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization," *Energy*, vol. 30, pp. 104âĂŞ–116, 2015.

[66] R. L. Negrete and F. J. D-Amato, "Fast nonlinear model predictive control - formulation and industrial process applications," *Computers and Chemical Engineering*, vol. 51, pp. 55–64, 2013.

[67] U. M. Sridhar, A. Govindarajan, and R. R. Rhinehar, "Demonstration of leapfrogging for implementing nonlinear model predictive control on a heat exchanger," *ISA Transactions*, vol. 60, pp. 218–227, 2016.

[68] L. Grüne, J. Pannek, M. Seehafer, and K. Worthmann, "Analysis of unconstrained nonlinear model predictive control schemes with time varying control horizon," in *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, Hawaii, USA: IEEE, December 2012, pp. 2605–2610.

[69] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. New Jersey, USA: Basic Books, 2015.

[70] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Florida, USA: Chapman and Hall/CRC, 2009.

[71] P. J. Fleming and C. M. Fonseca, "Genetic algorithms in control systems engineering: A brief introduction," in *IEEE colloquium on Genetic Algorithms for Control Systems Engineering*. Perth, Australia: IEEE, May 1993, pp. 1–5.

[72] K. Feng, J. Lu, and J. Chen, "Nonlinear model predictive control based on support vector machine and genetic algorithm," *Chinese Journal of Chemical Engineering*, vol. 23, pp. 2048–2052, 2015.

[73] M. Abbaszadeh and M. Solgi, "Constrained nonlinear model predictive control of a polymerization process via evolutionary optimization," *Journal of Intelligent Learning Systems and*

Applications, vol. 6, pp. 35–44, 2014.

[74] N. Patel and N. Padhiyar, "Modified genetic algorithm using box complex method application to optimal control problems," *Journal of Process Control*, vol. 26, pp. 35–50, 2015.

[75] S. Botha, D. Le Roux, and I. K. Craig, "Hybrid non-linear model predictive control of a run-of-mine ore grinding mill circuit," *Minerals Engineering*, vol. 123, pp. 49–62, 2018.

[76] A. Alfi and H. Modares, "System identification and control using adaptive particle swarm optimization," *Applied Mathematical Modelling*, vol. 35, pp. 1210–1221, 2011.

[77] J. P. Coelho, P. B. de Moura Oliveira, and J. B. Cunha, "Greenhouse air temperature predictive control using the particle swarm optimisation algorithm," *Computers and Electronics in Agriculture*, vol. 49, pp. 330–344, 2005.

[78] C. Guo, Y. Zhang, X. You, X. Chen, and Y. Zhang, "Optimal control of continuous annealing process using pso," in *Proceedings of the IEEE International Conference on Automation and Logistics*. Shenyang, China: IEEE, August 2009, pp. 602–606.

[79] J. S. Heo, K. Y. Lee, and R. Garduno-Ramirez, "Multiobjective control of power plants using particle swarm optimization techniques," *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 552–561, 2006.

[80] A. Tarique and H. A. Gabbar, "Particle swarm optimization based turbine control," *Intelligent Control and Automation*, vol. 4, pp. 126–137, 2013.

[81] J. Nocedal and S. J. Wright, *Numerical Optimization*, second edition ed. New York, USA: Springer, 2006.

[82] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, USA: Cambridge University Press, 2004.

[83] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization*. New York, USA: Oxford University Press, 1995.

[84] R. S. Sutton and A. G. Barto, *Reinforcement Learning - An Introduction*. Cambridge, Massachusetts, USA: The MIT press, 2018.

[85] M. Vidyasagar, *Non-linear Systems Analysis*, 2nd ed. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2002.

[86] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. Massachusetts, USA: The MIT Press, 2016.

[87] R. Fletcher, *Practical Methods of Optimization*. West Sussex, England: John Wiley and Sons, 1987.

[88] F. Potra and S. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 281–302, 2000.

[89] M. H. Wright, "The interior-point revolution in optimization: History," *Recent Developments, and Lasting Consequences, American Mathematical Society*, vol. 42, no. 2, 2004.

[90] D. DeHaan and M. Guay, "A real-time framework for model-predictive control of continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 11, pp. 2047–2057, 2007.

[91] A. N. Iusem, "On the convergence properties of the projected gradient method for convex optimization," *Computational and Applied Mathematics*, vol. 22, no. 1, pp. 37–52, 2003.

[92] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, pp. 145–151, 1999.

[93] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30 th International Conference on Machine*

Learning. Atlanta, USA: ICML, June 2013, pp. 1139–1147.

[94] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson, "Global convergence of the heavy-ball method for convex optimization," in *European Control Conference*. Linz, Austria: IFAC, July 2015, pp. 310–315.

[95] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, 1964.

[96] A. E. Bryson, *Applied optimal control - optimization, estimation, and control*. New York, USA: Taylor and Francis Group, 1975.

[97] P. H. Calamai and J. J. More, "Projected gradient methods for linearly constrained problems," *Mathematical Programming*, vol. 39, pp. 93–116, 1987.

[98] A. S. Conceicao, A. P. Moreira, and P. J. Costa, "A nonlinear model predictive control strategy for trajectorytracking of a four-wheeled omnidirectional mobile robot," *Optimal Control Applications and Methods*, vol. 29, pp. 335–252, 2008.

[99] H. Lim, Y. Kang, C. Kim, and J. Kim, "Experimental verification of nonlinear model predictive tracking control for six-wheeled unmanned ground vehicles," *International Journal of Precision Engineering and Manufacturing*, vol. 15, no. 5, pp. 831–840, 2014.

[100] M. A. Abbas, R. Milman, and J. M. Eklund, "Obstacle avoidance in real time with non-linear model predictive control of autonomous vehicles," *Canadian Journal of Electrical and Computer Engineering*, vol. 40, pp. 12–22, 2017.

[101] B. J. Guerreiro, C. Silvestre, R. Cunha, and A. Pascoal, "Trajectory tracking nonlinear model predictive control for autonomous surface craft," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2160–2175, 2014.

[102] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[103] B. Mayer, M. Killian, and M. Kozek, "A branch and bound approach for building cooling supply control with hybrid model predictive control," *Energy and Buildings*, vol. 128, pp. 553–566, 2016.

[104] V. Cacchiani and C. C. DoAmbrosio, "A branch-and-bound based heuristic algorithm for convex multi-objective minlps," *European Journal of Operational Research*, pp. 2–38, 2016.

[105] C. E. Long, P. K. Polisetty, and E. P. Gatzke, "Deterministic global optimization for nonlinear model predictive control of hybrid dynamic systems," *International Journal of Robust and Nonlinear Control*, vol. 17, pp. 1232–1250, 2007.

[106] O. Yechiel and H. Guterman, "A survey of adaptive control," *International Robotics and Automation Journal*, vol. 3, no. 2, 2017.

[107] B. Recht, "A tour of reinforcement learning - the view from continuous control," *arXiv preprint*, vol. 1806.09460, 2018.

[108] L. Busoniu, R. Babuska, B. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, Florida, USA: CRC Press, 2010.

[109] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proceedings of the 2016 International Conference on Learning Representations*. San Juan, Puerto Rico: ICLR, May 2016, pp. 1–14.

[110] S. V. Bedekar, P. Nithiarasu, and K. N. Seetharamu, "Experimental investigation of the performance of a counter-flow packed-bed mechanical cooling tower," *Energy*, vol. 23, no. 11, pp. 943–947, 1998.

[111] Saudi Electricity Company, "Saudi Arabia Electricity consumption tariffs," https://www.se.com.sa/en-us/customers/Pages/TariffRates.aspx, 2019, accessed: 2019-01-31.

[112] B. Stroustrup, *The C++ Programming Language*, 4th ed. Upper Saddle River, NJ, USA: Pearson Education Inc., 2013.

[113] B. Matthews and I. K. Craig, "Demand side management of a run-of-mine ore milling circuit," *Control Engineering Practice*, vol. 21, no. 6, pp. 759–768, 2013.

[114] C. Nagel, B. Evjen, J. Glynn, K. Watson, and M. Skinner, *C# 2012 and .Net 4.5*, 1st ed. Indianapolis, Indiana, USA: John Wiley and Sons, 2013.

[115] W. J. Chun, *Core Python Programming*, 2nd ed. Boston, MA, USA: Pearson Education Inc., 2006.

ADDENDUM A COMPUTATIONAL PLATFORM DEVELOPMENT

A complex plant model as per Section 3.4 can be more easily built for simulation purposes, if each unit operation can be "dragged and dropped" on a graphical user interface where the operations can be linked with each other by the model designer as needed. If each operation is implemented as a class (object) that knows how to behave according to each dynamic and static process model, the plant model can then be constructed quicker for simulation purposes. Each unit operation in the plant model can then be iterated over by the simulator. If each class inherits from a base class, Object Oriented Programming (OOP) by polymorphism will then ensure through late binding at run time that the correct sub-class' code is called during each iteration for each unit operation. A programming language that fully supports polymorphism for class hierarchies was needed to implement this design.

The designed class hierarchy is shown in Figure A.1. The base class is called "BaseClass" from which "BaseProcessClass" is derived. All streams between equipment in the plant model is represented by the "Stream" class derived form "BaseProcessClass". "UnitOp" inherits from "BaseProcessClass", and its children are the classes that consitute all the unit operations needed in the total plant model as per Section 3.4. Each class is instantiated into objects as many times and linked as needed to form the model as shown at a high level in Figure 3.1. Each class has all the code describing its process models. Various support classes were developed to support the core process model classes, including:

1. A class to house all the code for the HNMPC controller (see Section 4.3.3). This class will have an internal pointer to the object housing the total plant model in its current state, in order to use the model as its internal model. Every time the controller is run, the current plant model state will be cloned, which will be used internally in this class to execute the numerical integration

needed for the Jacobian, MINLP per node, and objective function calculation. The class also contains a data structure housing all the CVs and MVs.

2. Various classes to support the numerical methods are needed: Complex arithmetic for solutions of the Peng Robinson equations, matrix mathematical calculations, etc.
3. A class to perform all the Peng Robinson state equation calculations with associated molecule properties.
4. A class to support the control signals being split from the HNMPC controller to the fans and pumps to ensure the continuous controller writes the same set point to each motor as described in Section 4.3.3.
5. Various classes to support user interaction to enable properties of equipment to be specified.
6. Classes to support simulation results to be serialised and saved to disk, and model files to be serialised and saved to disk.
7. Classes to support user graphical manipulation of the unit operations and linking with streams to build the plant model.

The initial model and controller code was developed in C# [114] running in a Windows Environment. C# was chosen as it is a fully Object Oriented Programming language enabling deep class library development with full support for polymorphism. C# code is compiled and results in fast execution to numerically integrate the stiff ODEs of the cooling tower models. In addition the tree graph that is traversed by the hybrid controller during each execution iteration requires fast computation resources [105]. In Matlab full polymorphism is not supported, and program execution is slower.

In order to experiment with some of the powerful optimisation libraries available in Python [115], an Object Oriented language, the code was also implemented in Python 3.5 on a Macintosh platform. However, execution of the iterative algorithms of solving the non-linear ODEs was found to be quite slow in Python. Python is ideally suited for the Deep Reinforcement Learning experiments, and the Tensorflow library was used to test a closed loop system with Actor Critic policy gradient optimisation using (2.87). This could also be done in C++ though, since various Deep Learning libraries are available for C++ as well (e.g. Tensorflow and PyTorch). Since RL was chosen to not be used to design the non-linear controller, this benefit of Python was not taken into account in order to select an implementation platform.

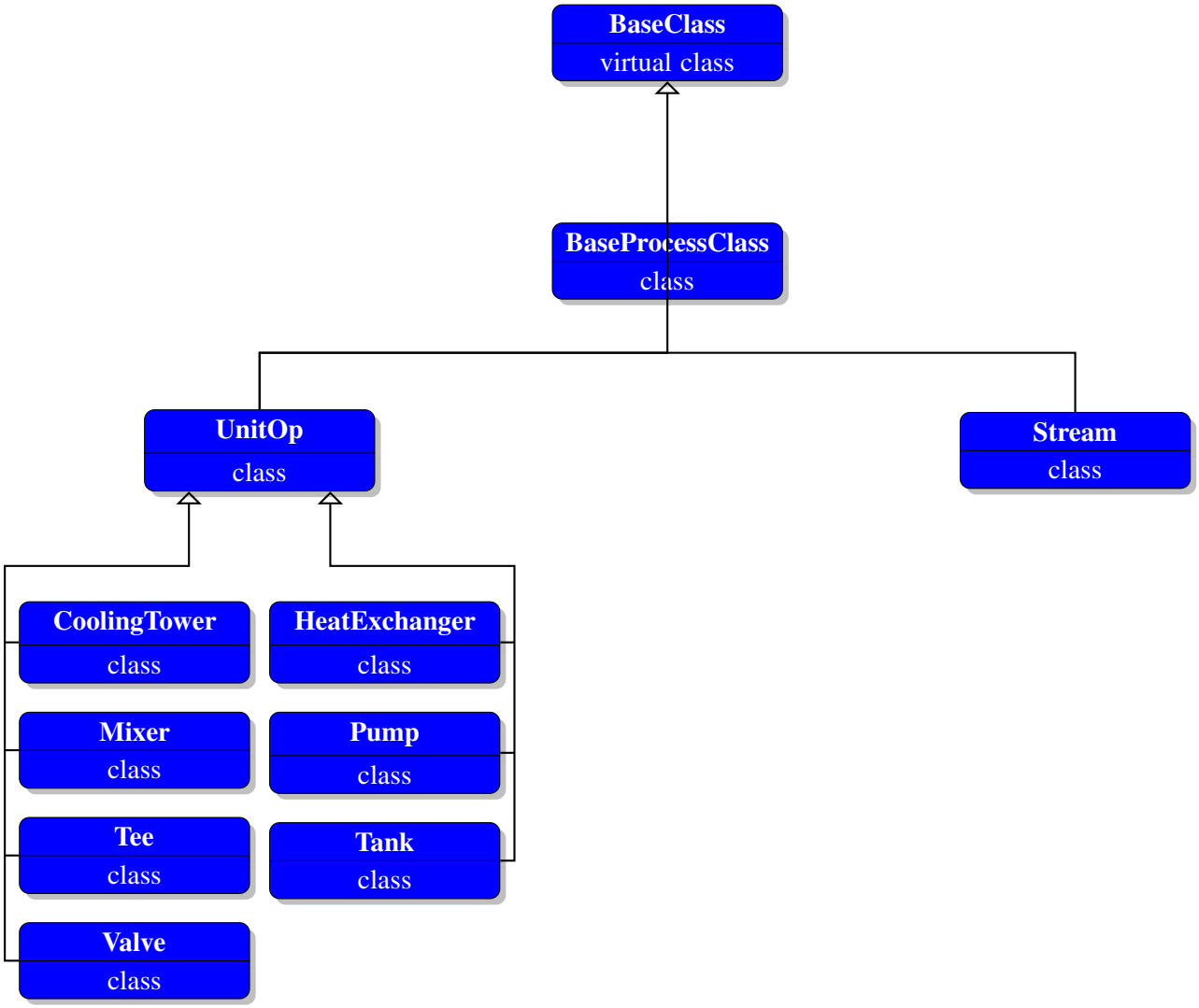

Figure A.1. Class hierarchy of the code of the plant model as implemented in C++.

The code was also implemented in C++ [112] on a Macintosh platform. Implementation in C++ achieves an increase in speed of execution over that of C#, includes the full polymorphism class library development support, and C++ enables a code base that can be compiled with open source software on a Windows, Macintosh, or Linux platform. Implementation in C# results in more of a restriction to the Windows platform.

C++ was found to be the preferred platform given the above requirements and benefits. The final model and HNMPC controller was implemented in this environment.