The Modelling of Organic Rankine Cycle Systems Intended for Industrial Optimization

Vathna Am

School of Engineering
May 2020

Supervised by:
Assoc. Prof David I Wilson
Dr. Jonathan Currie

A thesis submitted to Auckland University of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

The Organic Rankine Cycle (ORC) is one of the most efficient heat recovery technologies for low-temperature heat resources that are used in the geothermal power plant industry and waste heat recovery systems. As such, ORC systems are proving to be a sustainable technology that can help address some of the current concerns surrounding global warming and environmental pollution from using non-renewable resources. However, optimizing large and advanced systems can be a complex task that involves various fields of many variables from the thermodynamics to the plant's topologies to the environmental regulations. Therefore, this thesis will focus on providing a systematic modelling framework that can be used to optimize ORC systems efficiently.

Commonly in the literature, ORC systems are modelled using the sequential-modular (SM) approach where the unit operation modules are connected in the order of the plant's process. This forms the flowsheet of the ORC system, which is then solved using a nonlinear equation solver to converge to a feasible operating point. Generally, the SM model is optimized by manually varying the plant parameters or by using advanced optimization algorithms to maximize/minimize the objective function. This is not an efficient approach and can lead to various optimization and numerical issues, such as failure to converge to a solution and long execution times.

A more efficient method, but often more difficult to construct and troubleshoot, is to model the ORC system using the equation-oriented (EO) approach where the system is expressed as a set of equations. Provided the equations are algebraic and twice differentiable, the optimization solvers can exploit the model structure, the underlying equations and the relationship between the decision variables to effectively optimize the ORC system. Generally, the equations are not algebraic and consist of thermodynamic routines or external functions that are not differentiable and incompatible with white-box solvers that can deterministically guarantee global optimality. Therefore, this thesis will propose a modelling approach and provide a set of tools to model an ORC system in order to tailor the model for derivative-based and white-box solvers. This involves deriving the set of equations that describe the ORC system and approximating the nonlinear terms that are not differentiable using regression tools. As a result, the optimization performance of the algebraic EO model can be more than 29000x faster than the SM model.

The problem with any approximated model is to ensure that the model can accurately represent the original system. While the accuracy of the model can sometimes be improved by using highly nonlinear model fits, such as higher order polynomial functions, they can contribute to the nonlinearity of the model and degrade the optimization performance. Therefore, this thesis will introduce a piecewise regression approach to improve the accuracy of the approximated model and decrease the nonlinearity of the optimization problem. As a result, the performance of some solvers can be significantly improved and, in some cases, more than 6x faster than using the single fit approximations.

Building on the piecewise fit approximation work, the final study that was carried out in this research focused on a mixed-integer linear programming (MILP) formulation of the ORC model. This is to address the gap in the literature where the MILP formulations of ORC models have not been extensively investigated, despite the general view that nonlinear problems are harder to solve than linear problems. This involves utilizing existing integer and linear programming techniques and piecewise linear approximations. Consequently, the size of the optimization problem increases considerably due to the auxiliary variables and constraints from the linearization procedure, which can degrade the optimization performance.

This research provides an alternative approach to the SM model for modelling large and complex ORC systems that are robust and efficient for optimization. The proposed modelling framework will be implemented on three real-world geothermal power plants that vary in size and topology. In addition, this research shows that while decreasing the nonlinearity of the optimization problem can improve the performance, it is not advisable to completely linearize the model as it can have an adverse effect due to the large number of auxiliary variables and constraints that are generated.

Acknowledgements

I firstly wish to thank my academic supervisors, Associates Professor David I Wilson and Dr Jonathan Currie, for their technical and emotional support with this research. Their patience and guidance helped me through many difficult times during my PhD study, especially within the fields of optimization and chemical engineering. I would like to also pay my special regards to Dr Michael Gschwendtner for his help and guidance in thermodynamics.

I want to give the greatest gratitude to my family, especially my parents, Lim Chhorn Am and Voy Lee Ung, for their emotional and financial support throughout my life and especially in my academic endeavours. I want to thank my brothers, Vathnack Am and Sok Leng Am, for the countless times they have supported me emotionally in every aspect of my life.

To my colleagues that I have met during my PhD study, I want to say thank you for encouraging to maintain a positive outlook and keep working on my research even when there were times it seemed impossible. Lastly, I want to thank the Auckland University of Technology for the financial support in both my undergraduate and postgraduate studies, which have given me the opportunity to pursue my university education without any financial burden.

List of Publications

V. Am, J. Currie, and D. I. Wilson, “A systematic approach to modeling organic Rankine cycle systems for global optimization,” in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, May 2017, pp. 487-492.

V. Am, J. Currie, and D. I. Wilson, “A Comparison between NLP and MILP Formulations of Organic Rankine Cycle Systems for Optimization,” in *13th International Symposium on Process Systems Engineering (PSE 2018)*, ser. Computer Aided Chemical Engineering, M. R. Eden, M. G. Ierapetritou, and G. P. Towler, Eds. Elsevier, 2018, vol. 44, pp. 991 - 996.

Contents

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Vathna Am ______________________________

List of Figures

List of Tables

Nomenclature

The thesis will use the following notations in Table 1 for mathematical equations.

Table 1: Mathematical notations.

Notation	Description
$\mathbf{X}$	Matrix (bold and capitalized)
$\mathbf{x}$	Column vector (bold and lower case)
x	Scalar (italic and lower case)
$\hat{x}$	Approximation (hat operator)

The following list of nomenclatures in Table 2 will be used within this work.

Table 2: Thesis nomenclature.

Symbol	Description	Unit
$\dot{m}$	Mass flow	kg/s
h	Enthalpy	kJ/kg
s	Entropy	kJ/(kg K)
T	Temperature	K
P	Pressure	bar
$\dot{W}$	Mechanical work	kW
$\dot{Q}$	Heat duty	kW
C	Cost	k$
η	Efficiency	%
$\mathcal{J}$	Objective function	-
z	Mass flow fraction	-

Table 3 lists the abbreviations and acronyms that are frequently used in the written texts or as subscripts in mathematical equations throughout this thesis.

Table 3: Thesis subscripts and acronyms.

Symbol	Description
pump	Pump
turb	Turbine
cond	Condenser
desh	Desuperheater
preh	Preheater
evap	Evaporator
suph	Superheater
recu	Recuperator
pp	Pinch-point
WF	Working Fluid
CW	Cooling Water
BR	Brine or Geofluid
CS	Cold Side
HS	Hot Side
DOE	Department of Energy
ORC	Organic Rankine Cycle
BCPP	Binary Cycle Power Plant
BORC	Basaran Organic Rankine Cycle
DOEP	Department of Energy Pilot Plant
USGP	USGeo Plant
MBPP	Magmamax Binary Power Plant
LT	Low Temperature
HP	High Pressure
LP	Low Pressure or Linear Program
NLP	Nonlinear Program
MINLP	Mixed-Integer Nonlinear Program
MIQCLP	Mixed-Integer Quadratically Constrained Linear Program
MIQCQP	Mixed-Integer Quadratically Constrained Quadratic Program
MILP	Mixed-Integer Linear Program
EO	Equation-oriented
SM	Sequential-modular
is	Isentropic Process
crit	Critical
hx	Heat Exchanger
bk	Breakpoint
net	Net Power
f	Saturated liquid
g	Saturated vapour

Chapter 1

Introduction

1.1 Organic Rankine Cycle

With the increasing concerns surrounding the negative environmental impacts associated with non-renewable resources and the growing pressure for the power generation sector to reduce greenhouse gases emission and pollutants, there is now much more recognition and interest in the global benefits of renewable power generation technologies. Through technological advances in geothermal exploration and well drilling technology, geothermal energy is proving to be a reliable source of sustainable and renewable energy. According to the World Energy Council, there are 24 countries that have used geothermal energy for electricity generation as of 2014 [8]. In New Zealand, almost 20% of the electricity supply is produced from geothermal energy [9]. Most of the geothermal resources in New Zealand are located in Taupo Volcanic Zone and Ngawha (Northland) with temperatures varying from 70 °C to >220 °C in some areas [10].

One of the most efficient ways to exploit the abundant low-temperature geothermal resources is via the Organic Rankine Cycle (ORC) systems. The energy conversion process of an ORC system is very similar to a conventional steam cycle, where the thermal energy is converted to kinetic energy that can subsequently be used to generate electricity, as shown in Figure 1.1. In addition, the ORC systems are not limited to geothermal energy but can be implemented using other thermal resources, such as solar energy, thermal waste, or biomass energy, thus broadening their applications in other industries. This can also help to reduce the amount of waste heat that is lost to the environment and contributing to global warming.

For a basic ORC system, as shown in the middle of Figure 1.1, a heat source is used to vaporize a working fluid in an evaporator before expanding through a

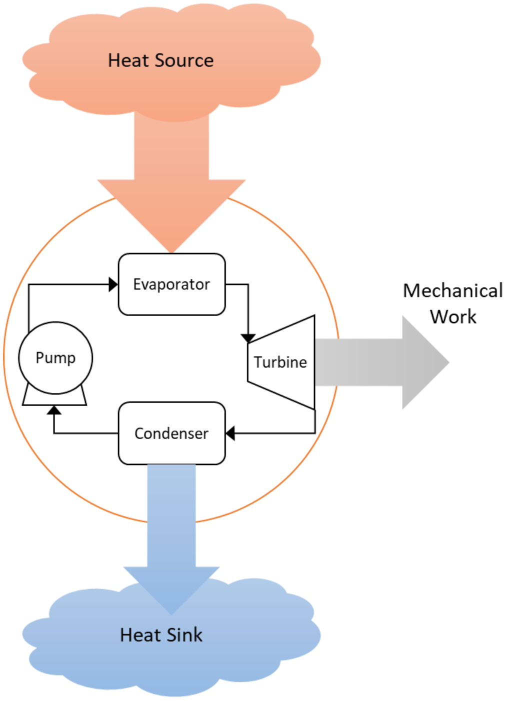


Figure 1.1: The energy conversion process of an Organic Rankine Cycle system.

turbine. The vapour exits the turbine at a lower pressure and condenses back to its liquid state before it is fed back to the evaporator via a pump. This basic design forms the basis for many complex and innovative ORC systems that are designed to improve the performance and reduce thermodynamic losses. These advanced systems can consist of two or more ORC systems that operate at different pressure levels with various inclusion of additional unit operations, such as a recuperative heat exchanger, to help maximize the utilization of the heat source. A good overall summary of different advanced ORC systems can be found in [3, 11].

Organic Rankine Cycle systems are predominantly used in the geothermal power plant industry as Binary Cycle Power Plants (BCPP). Of all the geothermal power plants (GPP), BCPPs are the most common type of GPP in the world with around 279 units according to [12], constituting to over 46% of all geothermal units. Given that there are a large number of ORC systems ranging in various topologies around the world and the potential for more low-temperature thermal resources to be exploited, there is a need for an efficient and robust approach to modelling and optimizing these systems. In addition, ORC systems inherently have a low thermal efficiency due to the small temperature difference between the heat source and the heat sink [3]. Therefore, even for an ORC with a 10% thermal efficiency, a 1% efficiency increase would translate to a 10% improvement that can mean the difference between an economically viable plant and one that is not.

Often these ORC plants are simulated using a modular-based approach in a modelling software, such as Aspen Plus [13] or GateCycle [14]. The models then are subsequently optimized using heuristic methods or optimization algorithms, as shown in [15, 16]. However, this approach is not the most robust or efficient for optimization and can lead to various convergence and numerical issues, as detailed below.

1.2 Modular-based Optimization Models

Organic Rankine Cycle systems are generally modelled as modular-based models using the sequential-modular (SM) method, where the unit operation modules are connected in the order of the plant's process. Each module is solved in a sequential order where the output of one module is computed from the output of the previous module in the flowsheet. This makes constructing the SM model of an ORC system very institutive and simple to troubleshoot when the input stream at the beginning of the plant's process is known. However, there are some key limitations associated with the SM approach, especially when dealing with large and complex ORC optimization problems.

One of the major disadvantages with the SM model is often the restriction of accurate derivative information due to the use of complex thermodynamics, internal/external unit operation modules, and a nonlinear equation solver. This can degrade the performance of derivative-based optimization solvers, such as `fmincon` (MATLAB's solver) [17] or IPOPT [18], that require the first and second derivatives of the objective function and constraints in order to find a solution. If the derivative information is not provided to the solver, the derivatives are approximated using the finite difference method that can result in various optimization issues, such as premature termination and long computation times [19]. Therefore, one should always try to find a method that provides accurate Jacobian and Hessian information to the solver in order to obtain efficient optimization performance. This will be further discussed in Chapter 2. In addition, given the nonlinear nature and the rigidity structure of the SM model, it is limited to black-box solvers that cannot guarantee global optimality, which can be a disadvantage in today's competitive market.

1.3 Equation-based Optimization Models

Alternatively, Organic Rankine Cycle systems can be modelled as equation-based models using the equation-oriented (EO) approach, where systems are represented as a set of equations that are solved simultaneously. The structure of the EO approach allows standard optimization problems to be formulated efficiently, and the mass and energy balance equations are solved simultaneously with the optimization problem [20]. However, unlike the SM approach where the model can typically be constructed by connecting pre-built unit operation modules, the underlying equations of the EO model generally must be manually derived. This can be very time consuming and difficult to troubleshoot the model, especially for advanced ORC systems.

In addition, if all the equations in the EO model are algebraic and analytically differentiable, this algebraic structure can be exploited to analytically calculate the required derivatives and provide the matrix sparsity information to the solvers. This means that the optimization issues associated with the SM approach mentioned above can be bypassed or at least reduced. Furthermore, provided the equations are limited to a set of certain functions that are supported by white-box solvers, the model can be optimized using global solvers, such as SCIP [21] and BARON [22], to deterministically find the global optimum of the plant.

1.4 Optimization Formulation

Optimizing an ORC model using a state-of-the-art optimization solver does not always guarantee an efficient and robust optimization. The formulation and structure of the optimization problem also play a significant role in the overall performance of the optimization. In operations research, there are many different chemical processes that can be formulated into different optimization problems from linear programming (LP) to mixed-integer nonlinear programming (MINLP) problems, which can vary in complexity and size. In terms of the ORC systems, they are inherently nonlinear problems due to the thermodynamic processes of the unit operations and the thermodynamic properties of the working fluid. Consequently, the steady-state models of ORC systems consist of nonlinear equations that generally give rise to nonconvex problems with multiple local optimums [23] and are very hard to solve, especially with large and complex models.

Often optimization problems can be reformulated using various integer and linear programming techniques, such as the Glover's linearization scheme [24] and the Charnes-Cooper transformation [25], which can help address some financial restrictions, time constraints, compatibility and availability issues with modelling software and optimization solvers, and/or restrictions of establishing global optimality. Historically, nonlinear problems are reformulated to linear problems because LP solvers, or even mixed-integer linear programming (MILP) solvers, were more computationally efficient and readily available than the nonlinear programming (NLP) solvers. However, it can be very tedious and time consuming for practitioners to reformulate optimization problems. Furthermore, with the recent advancements in nonlinear solvers and improvements in optimization modelling platforms and techniques, as well as computer hardware, it might now be redundant for practitioners to reformulate certain real-world NLP problems.

1.5 Research Questions and The Thesis Contributions

In order to fully utilize the full potential of the ORC system and maximize the usage of the available renewable resources efficiently, it is important that a reliable and efficient modelling and optimization approach is available. However, while it is widely acknowledged in the literature that an EO model is more efficient and robust for optimization than an SM model, it is not extensively investigated or proposed in the literature for large advanced ORC systems. Therefore, this research aims to provide a modelling framework for an ORC system to reduce the issues associated with the conventional SM optimization approach and to provide an efficient model that is tailored for optimization. This will be significantly beneficial for certain optimization problems that are computationally expensive, such as optimizing the plant with respect to different working fluids and mixtures. In addition, given that ORC systems typically have low thermal efficiencies and can vary in plant design and operation conditions, there would generally be many possible local optimums that can limit them from fully exploiting the thermal resources.

In order to tailor the ORC model to the derivative-based and white-box solvers, the model can be approximated using regression analysis to reformulate the model to an algebraic structure that can provide analytical derivative and matrix sparsity information. However, it can be very difficult to accurately approximate the ORC system given the complicated thermodynamic terms and processes in the steady-state model. It is possible to use highly nonlinear model fits to increase the accuracy, but this can also increase the nonlinearity of the model and degrade the optimization performance. Therefore, this research investigates a more constructive method of approximating the ORC model using piecewise fits, which can minimize the deleterious effect of the nonlinearity of the approximated model but not at the expense of its accuracy.

Generally, nonlinear problems are much harder to solve than linear problems, but the linearization of an ORC model has not been widely investigated. This could be due to the strenuous and tedious linearization procedure that involves introducing piecewise linear approximations and other linearization techniques. Consequently, the advantages and disadvantages of a linear ORC model would be hard to investigate without first providing an efficient method to linearize a nonlinear ORC model. As such, this research aims to explore a systematic approach to convert a nonlinear model to an equivalent mixed-integer linear programming model and compare the optimization performance of both formulations.

The overall objective of this thesis is to provide a framework that can be used to model large and complex ORC systems that are intended for industrial optimization and to address the shortcomings highlighted above. To achieve this aim, the following research questions are posed:

1. *Can we formulate equation-orientated models of large and complex ORC systems that are algebraic in structure and tailored for efficient and robust optimization?*
2. *Can the performance of some optimization solvers be improved by reducing the deleterious effect of the nonlinearity of the optimization ORC model without compromising, if not improving, on the accuracy of the approximated model?*
3. *Is it possible to reformulate the nonlinear algebraic EO model of an ORC system into an equivalent mixed-integer linear programming model using linear/integer programming techniques? In addition, given the improvement in nonlinear solver algorithms and the advancement in computer hardware, can the mixed-integer linear model improve or match the performance of the nonlinear counterpart as traditionally concluded in the literature?*

To undertake these research questions, this thesis will detail an algebraic EO modelling approach on three real-world binary cycle power plants. Leveraging of the algebraic structure of the EO model, three different formulations will be investigated, namely, NLP, MINLP and MILP. A set of regression and modelling tools will be discussed and used to reformulate the ORC model from one formulation to another. The research questions will be critiqued later in Chapter 8.

1.6 Thesis Outline

In order to coherently address the research questions stated in the previous section, this thesis is structured as follows:

Chapter 2 reviews existing studies of ORC modelling and optimization. This chapter will highlight and review the advantages and disadvantages of the common methodologies and tools. The shortcomings in the literature will be discussed in more detail.

Chapter 3 introduces the algebraic EO modelling approach. A detailed description of the modelling procedure is carried on a basic ORC system.

Chapter 4 applies the algebraic EO modelling approach to three large and complex real-world ORC systems, namely, binary cycle power plants. In addition, the equivalent SM model of each plant will also be presented and validated.

Chapter 5 optimizes the ORC models in Chapter 4 using both black-box solvers and white-box solvers. Each algebraic EO model will be optimized with respect to different working fluids and the optimization performance will be analysed.

Chapter 6 details the implementation of a piecewise fit algorithm that can help to improve the performance of some optimization solvers without compromising on the accuracy of the model.

Chapter 7 details the reformulation of the nonlinear optimization problem into an equivalent mixed-integer linear programming problem. The MILP model will be optimized with state-of-the-art MILP solvers and compared to the nonlinear counterparts.

Chapter 8 concludes the thesis and highlights the key contributions of this work. This is then followed by an assessment of the proposed research questions and recommendations for future developments and investigations.

Chapter 2

Literature Review

This chapter will discuss the energy conversion process of an Organic Rankine Cycle (ORC) system and look at the previous studies surrounding the modelling and optimization of this system. The chapter will review the different classes of mathematical programming and the available modelling and optimization tools that are used by practitioners on ORC systems. This is then followed by a discussion on the advantages and disadvantages of the different optimization techniques and highlighting the need for a robust and efficient optimization approach. Finally, the chapter ends with a discussion on the shortcomings in the field of ORC optimization and highlight the areas that will be addressed in this research.

2.1 Introduction

The world's dependence on non-renewable energy has led to various environmental issues that have a detrimental impact on the planet. More than 66% of the world's electricity is generated from fossil fuels according to [26]. Consequently, there has been hard evidence to show the negative implications of utilizing fossil fuel, from the rise in global temperature to environmental pollution. In response to the concerns surrounding global warming and ambitious efforts to address climate change through the Paris Agreement [27], there has been a great interest in renewable and sustainable energy resources in recent years. One of which is in the field of low-temperature thermal resources.

There are a vast amount of low-temperature thermal resources, such as geothermal energy, industrial waste thermal energy, and solar energy, that are available to be exploited and have the potential to help address some of the concerns surrounding the use of non-renewable resources. One of the most efficient and sustainable ways to utilize low-temperature thermal resources is through an Organic Rankine Cycle (ORC) system. The energy conversion process of an ORC system is thermodynamically very similar to that of other conventional thermal power plants, such as a fossil fuel power plant and a nuclear power plant.

In order to effectively utilize the available low-temperature thermal resources, ORC systems need to be modelled and optimized. Often they are simulated using the sequential-modular approach, via modelling software such as Aspen Plus [13] or GateCycle [14], and then are subsequently optimized using heuristic methods or optimization algorithms [15, 28]. However, one major concern with using this approach lies in the robustness and efficiency of the model when it is used in optimization problems. Therefore, this chapter will review the limitations of common practices of optimizing ORC systems and propose a modelling approach that is tailored for efficient optimization.

2.2 The Organic Rankine Cycle Process

One main difference between the ORC system and the Rankine cycle is the type of working fluid that is used. Instead of utilizing water as the working fluid, ORC systems generally use a hydrocarbon or a refrigerant that has a lower boiling point than that of water. This is due to the low-temperature heat source to the system. Figure 2.1 shows the process flow diagram of a basic ORC system, where the working fluid is heated up and vaporized by the heat source (in this case a geofluid) inside the preheater and evaporator, and then expands through the turbine to produce mechanical work. Next, the exhaust vapour exits the turbine and is condensed to a saturated liquid (or subcooled liquid) before it is fed back to the preheater and evaporator via a feed pump.

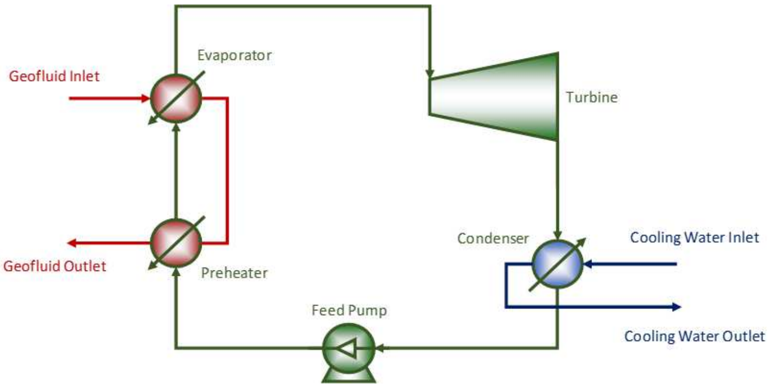


Figure 2.1: The process flow diagram of a basic Organic Rankine Cycle.

The ORC systems are largely employed in the geothermal power industry due to the vast amount of low-temperature thermal resources. A binary cycle power plant

(BCPP) is a geothermal power plant that utilizes the ORC technology and makes up the largest number of units out of all the geothermal power plants in the world. In addition to high power geothermal ORC power plants, there are also micro ORC systems that generate low power around 20-50 kW from low-temperature the heat sources, such as solar energy, biomass and waste heat from various thermal utility systems [29].

Although the ORC systems are proving to be one of the leading energy conversion technologies for a low-temperature heat source, they generally have low thermal efficiencies due to the small temperature difference between the heat sink and the heat source [3]. Therefore, this can be a deterrent for many companies to invest in the development of these plants if they are not optimized efficiently to fully utilize the available heat source. In addition, given that every BCPP is unique and ranges from various geothermal resources to the topology to the working fluid of the plant, it is very unlikely that the optimized conditions of one plant can be applied to another plant in a different location or with operating constraints. Therefore, there is a need for an efficient and robust off-line optimization approach that is applicable to a wide range of ORC systems and tailored for various optimization solvers.

While there are studies in the literature that address the performance and/or economic optimization, such as described in [30, 31, 32, 33, 15, 34], they do not normally focus on the modelling and optimization methodologies or formulations that are specifically for the ORC systems. Generally, the study is focused on one particular plant design where one or more different aspects of the plant are optimized.

For instance, Astolfi et al. [30] optimized the thermal efficiency of an ORC system with respect to the inlet pressure and temperature of the turbine and different working fluids. The ORC model was constructed in MATLAB and optimized using the `fmincon` optimization solver (Active-set algorithm), where the thermodynamic properties of the working fluids were calculated using REFPROP [35] thermodynamic package. Given that the model uses REFPROP explicitly, it can be difficult to obtain accurate derivative information of the model, which can lead to long computational time or failure to converge to a solution (see Section 2.9).

A study carried out by Ghasemi et al.[15] constructed a model of an existing ORC system in Aspen Plus. The net output power of the plant was optimized with respect to the ambient temperature using the built-in SQP optimizer in Aspen Plus. Their results showed that the net output power is restricted by the capacity of the air-cooled condensing system at high ambient temperatures. The disadvantage with this approach is that it utilizes the sequential-modular (SM) method (see Section 2.6.1) where the model was constructed using the built-in unit operation modules. The rigorous structure of this model makes obtaining accurate derivative information very hard or impossible and, therefore, can result in the same optimization problems as highlighted earlier with [30]. Furthermore, this modelling structure is generally not compatible with white-box solvers (see Section 2.7.2) that can guarantee the "best" optimal solution.

Conversely, a paper by Huster et al. [23] addresses the issues found in the previous two papers by constructing two ORC models using two distinct methods. The first model was constructed in the SM format but can provide accurate function evaluations, gradients, and relaxations for their proposed global optimization framework using customized functions. The second model was formulated in the GAMS optimization environment where it is algebraic, differentiable and compatible with a state-of-art white-box solver, namely BARON [22]. The aim of the paper was to compare the performance between their proposed optimization framework and BARON by optimizing the net output power and the levelized cost of electricity of the system. While the results showed that their global optimization framework was faster than BARON, the disadvantage with this approach is that the first model is not compatible with other white-box solvers due to its SM structure. Also, the thermodynamic functions that were used in the second model are highly nonlinear and this can have a significant impact on the optimization performance, which might explain the long computational time of BARON as shown in the paper.

Therefore, this research will provide a modelling framework and a set of tools to address the shortcomings stated about the papers discussed above and aim to illustrate the contributions using three real-world binary cycle power plants.

2.3 Organic Rankine Cycle Optimization

The term *optimization* can be defined from a practical standpoint as to obtain the "best solution" to a given process within constraints [19]. This is a key aspect of chemical engineering and one of the major quantitative tools in decision making. The competitive advantage of being able to operate at the optimal performance for process systems is pivotal to the economic viability of many businesses in various industries, and thus drives the need for optimization strategies to be developed and implemented.

Process optimization involves the task of deriving a quantitative scalar performance index, or referred to as the *objective function*, to quantify the "best solution". This can be, but not limited to, the operational costs, the production yield, or the system efficiency. The objective function is maximized/minimized by manipulating the values of the *decision variables*, which can represent the physical size of the equipment, the plant parameters or the working fluid thermodynamic properties, e.g., temperature, pressure, and enthalpy. Commonly, these decision variables must be adjusted as so to comply with the system and process constraints, which are typically translated to the physical limitations of the system, environmental regulations, financial constraints and so forth. The following subsections will discuss some of the most common areas of ORC system optimization in the literature.

2.3.1 Steady-State Optimization

A system or a process is said to be in a steady-state condition if the state variables, i.e., temperature, pressure, volume, enthalpy, entropy, etc., that describe the behaviour or process of the system are unchanged with respect to time. The state variables can change from point to point in the system, but they are stay fixed at any one point during the entire process. Consequently, the term *steady-state optimization* relates to obtaining the optimal steady-state operating point of the system, such as maximizing net output power and thermal efficiency, minimizing specific heat exchanger area, fuel consumption, gas production, etc.

Most of the studies in the literature on ORC system optimization are focused on steady-state optimization, typically in process synthesis, retrofitting and operational optimization, and economic analysis. Generally, an ORC system will be simulated in a process simulator (see Section 2.6), such as Aspen Plus[13] or VMGSim[36], that are subjected to a number of constraints set by the environmental regulations, financial restrictions and equipment limitations. The simulated model will then be optimized using a heuristic method or a mathematical algorithm that will attempt to search for a better operating point within the allowable search space.

Process synthesis optimization is focused on a new plant flowsheet or design to meet the specified engineering and financial goals. This is the early stages of the plant development process where the decision on the plant's topology and parameters are determined. This can be a challenging task given the number of factors, such as different BCPP designs, availability of working fluids, and different operating conditions, that can profoundly affect the costs, efficiency, future maintenance and scheduling operations of the plant. Examples of process synthesis optimization are presented by [33, 37, 30, 34, 38, 39, 40].

In contrast, retrofitting and operational optimization deals with the inclusion of new features/technologies or varying the operating conditions (i.e, temperature, pressure, mass flow, etc.) of an existing plant or flowsheet in order to improve the proposed objective function, such as efficiency, increase output, and reduce costs.

An example can be observed in a study carried out by [15], where they optimized the net output power of an existing BCPP by suggesting different optimal operation strategies with respect to the ambient temperature of the plant. In contrast, research by [41, 42, 43, 44] investigates the feasibility of introducing an ORC system to an existing process system or flowsheet, such as a flash steam power plant and a waste heat recovery system, in order to optimize the overall system.

Economic optimization investigates the costs associated with the construction, maintenance, or the profits that can be generated from the plant. The economic analysis is usually associated with the size of the main unit operations, i.e., heat exchangers and turbine, as they are generally assumed to be a significant factor in the cost of the plant. Consequently, the total area of the heat exchangers to the net output is a common objective function, as proposed in [33, 45, 46]. The investment cost of the plant can also be used as the objective function, i.e., the ratio of the total cost of the plant (plus the maintenance and labour cost) to the net output power, as is discussed in [47, 31]. The cost of each unit operation can be calculated/estimated using fixed correlations that are a function of the plant's output power or surface area of the plant component. These unit operation costs can then be used to calculate the payback period and the net present value that can give more insight into the plant's economics.

This research will focus on steady-state optimization because of the different areas that can be investigated in this field as demonstrated in the literature. Therefore, we believe it would be significantly beneficial to investigate and provide a modelling approach that is efficient and robust for off-line steady-state optimization.

2.3.2 Dynamic Optimization

The term *dynamic optimization* focuses the transient response and how the plant behaves with respect to time in areas such as process stability, emergency scenarios, the plant responsiveness to load change, shutdown and startup, and changes in the heat source. Similar to steady-state models, the ORC systems are generally constructed in a process flowsheet simulator, such as VMGSim [36], Aspen Plus Dynamics [48], Modelica [49]. Generally, these modelling environments allow practitioners to construct their dynamic models using the built-in unit operation modules/components that have been tested and validated, or from their own pre-built or interfaced component modules. Examples of dynamic ORC models include work by [50] that investigated a large commercial ORC power plant, and [51, 52] that focused on small-scale ORC systems.

Given that the interest in steady-state behaviour is more dominant in the literature, and also because BCPPs are base-load power plants that generally operate in a state-state condition continuously for long periods of time, this research will not focus on the dynamic behaviour of the plant. However, readers can refer to references given above if they are keen on learning more about dynamic models and optimization.

2.3.3 Working Fluid Selection

The performance, environmental and cost implications of an ORC system are heavily influenced by the selection of the working fluid. While there are more than 50 working fluids that have been considered in the literature [53], there are some restrictions that are associated with certain fluids due to their thermodynamic properties, environmental impact, health, safety, cost and availability. The selection of the working fluid cannot just meet the thermodynamic properties but should also be stable within the desired operating conditions and non-corrosive with materials in contact. There is an extensive amount of studies in the literature on working fluid selection and comparison among different operating conditions and ORC configurations, such as described in [1, 46, 54, 53, 55], and more recently in [56, 57, 58].

Although there is a substantial amount of research on working fluid selection in the literature, there is no one best working fluid that can meet all the criteria of every ORC system and operating condition [53]. Given that each ORC system is unique to where it is located and to how it is built and function, it would be problematic to attempt to obtain an optimal solution of an ORC plant by extrapolating the optimal results of another plant from a different country and operating condition. Thus, it would be more beneficial is to provide a modelling approach to efficiently optimize the plant with respect to all the possible pure working fluids and mixtures within a reasonable time frame, especially for large and advanced ORC systems. This will be addressed in Chapters 5 and 6.

2.3.3.1 Types of Working Fluids

The working fluids can be divided into three groups, namely, dry, wet and isentropic, by examining the slope of the saturated vapour curve (dT/ds) on the T-s diagram, as shown in Figure 2.2. Since the slope of the isentropic fluids is a vertical line (or close to it), dT/ds will equal to infinity (or a very large number). Therefore, it is more useful to use the inverse of the slope (ds/dT), as discussed in [53, 59], to classify the working fluids. This implies that for dry fluids $ds/dT > 0$, for isentropic fluids $ds/dT \approx 0$, and for wet fluid $ds/dT < 0$. Generally, dry and isentropic fluids are more appropriate for ORC systems because they are usually superheated after an isentropic expansion, thus there will be no liquid droplets at the turbine output that might cause blade corrosion [54, 58]. However, if $ds/dT \gg 0$, this adds more load to the condenser or the recovery system and can waste valuable energy [53]. Conversely, a wet fluid will put more burden on the superheating process to avoid going into the two-phase region, but desuperheating the fluid after the expansion will require less work [53].

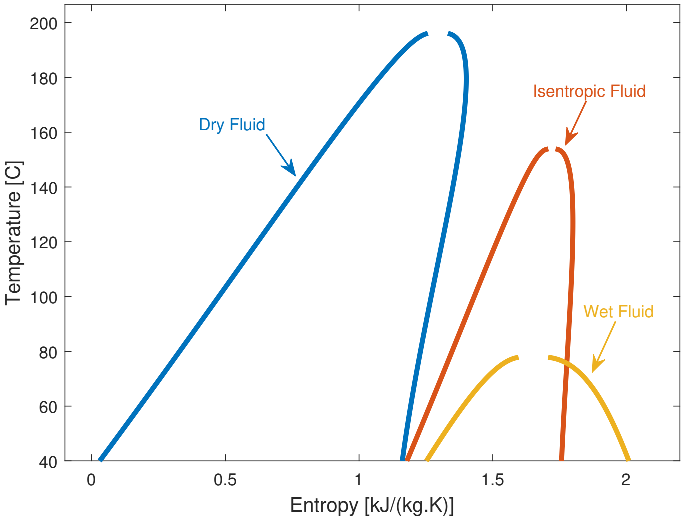


Figure 2.2: Three different categories of working fluids: dry fluid (Pentane), wet fluid (R32), and isentropic fluid (R245fa). Note the discontinuities are due to JSteam's limitations in calculating the saturated entropy values at temperatures close to the critical point.

In addition to pure working fluids, mixtures are also considered for the ORC system, as described in [56, 60, 61]. There are potential advantages that can be achieved by using mixtures. Unlike pure fluids that change phase at a constant temperature, mixtures evaporate and condense at variable temperature [3]. Therefore, they are sometimes a better match to the brine curves and, hence, can reduce the thermodynamic losses in the heat exchangers [3].

2.3.3.2 Safety and Environmental Impact

The safety concerns need to be taken into account when selecting the type of working fluid. Generally, fluids that are non-corrosive, non-flammable, non-toxic, and have low global warming potential (GWP) and low ozone depletion potential (ODP) are usually favoured, but these properties are typically mutually exclusive. Table 2.1 compares these properties for some of the working fluids and water for reference.

The risks associated with the flammability and toxicity of the working fluid can be reduced or managed by introducing safety measures, such as eliminating ignition

Table 2.1: The safety and environmental properties of different working fluids (adapted from [3]). The ozone depletion potential (ODP) and the global warming potential (GWP) are normalized at 1 for R12 and carbon dioxide, respectively.

Fluid	**Toxicity**	**Flammability**	**ODP**	**GWP**
R12	Nontoxic	Nonflammable	1	4500
R114	Nontoxic	Nonflammable	0.7	5850
Propane	Low	Very high	0	3
i-Butane	Low	Very high	0	3
n-Butane	Low	Very high	0	3
i-Pentane	Low	Very high	0	3
n-Pentane	Low	Very high	0	3
R32	Low	Low flammable	0	675
R134a	Very low	Nonflammable	0	1300
R245fa	Very low	Nonflammable	0	1020
Carbon dioxide	Nontoxic	Nonflammable	0	1
Ammonia	Toxic	Low	0	0
Water	Nontoxic	Nonflammable	0	-

sources or using air-tight equipment, in order to permit their use. In contrast, there are a number of the refrigerants have been banned or phased out over the years due to their negative impact on the environment based on their GWP, ODP and atmospheric lifetime. Some of the fluids that have already been banned include R11, R12, R113, R114, and R115, while other fluids such as R21, R22, R123, R124, R142b are expected to be phased out in 2020 to 2030 [53].

2.3.3.3 Cost and Availability

The cost and availability of the working fluid can be a factor in the design of the ORC system. Often some countries might ban or have strict regulations on certain fluids and, thus, make them too expensive to be utilized. Generally, the cost of an ORC system can be reduced by using low-cost hydrocarbons or fluids that are mass produced. In addition, the selection of working fluid can directly affect the size of the unit operations, which can contribute to the overall capital and maintenance cost of the plant. As shown in [3, 62], the size of the turbine can be approximated from the working fluid properties, which can be a factor in selecting the appropriate working fluid.

2.4 Classes of Optimization Problems

One of the oldest methods in process optimization is the *trial and error* approach. This involves trialling different solutions to the optimization problem until the best solution is achieved. Consequently, this approach can be very expensive and tedious to execute and does not always guarantee an optimal result or meet all the required constraints. However, due to the intuitive simplicity of this method, it is still implemented in many applications, such as in [39, 38], or especially for processes that cannot easily be modelled.

Alternatively, with the introduction of digital computers, industrial optimization problems can be optimized more efficiently and systematically using mathematical optimization algorithms and techniques. This is referred to as *operations research*. The optimization problems discussed in the previous section can be formulated into different classes of optimization problems based on the type of variables and constraints. Figure 2.3 shows the general overall view of the common classes of optimization problem formulations that are found in chemical engineering.

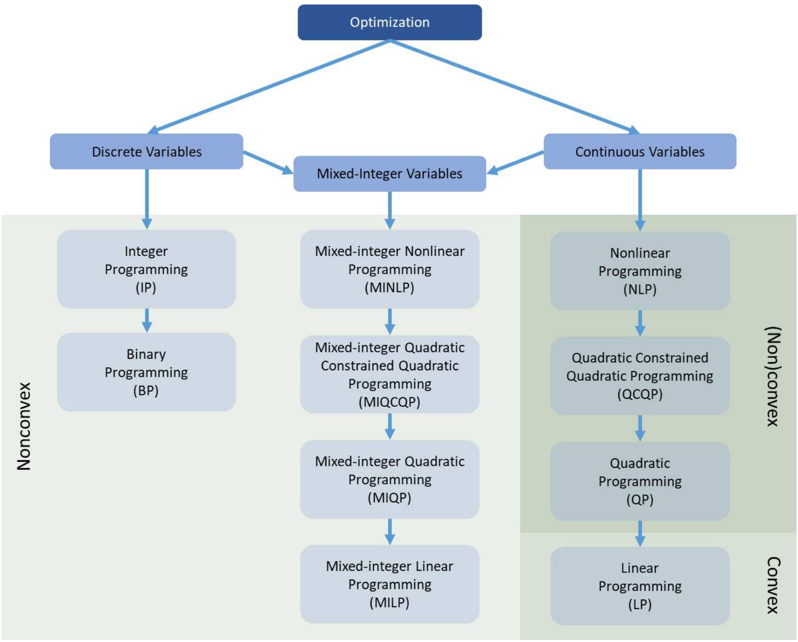

Optimization
Discrete Variables
Mixed-Integer Variables
Continuous Variables
Integer Programming (IP)
Binary Programming (BP)
Mixed-integer Nonlinear Programming (MINLP)
Mixed-integer Quadratic Constrained Quadratic Programming (MIQCQP)
Mixed-integer Quadratic Programming (MIQP)
Mixed-integer Linear Programming (MILP)
Nonlinear Programming (NLP)
Quadratic Constrained Quadratic Programming (QCQP)
Quadratic Programming (QP)
Linear Programming (LP)
Nonconvex
(Non)convex
Convex

Figure 2.3: Classes of optimization formulations.

The most complex optimization problem formulation is the mixed-integer nonlinear program of the form

$$
\begin{aligned}
\min \quad & f(x, y) \\
\text{s.t.} \quad & h_i(x, y) = 0, \\
& g_j(x, y) \leq 0, \\
& x \in \mathbb{R}, \\
& y \in \mathbb{Z}, \\
& i = \{1, 2, 3, ..., n\}, \\
& j = \{1, 2, 3, ..., m\},
\end{aligned} \tag{2.1}
$$

where $f(x, y)$ is the objective function that is subjected to equality constraint(s), $h_i(x, y)$, and/or inequality constraint(s), $g_j(x, y)$. The continuous and discrete variables are represented as x and y, respectively. In some cases, the discrete variables y in (2.1) can take on binary values to represent logical or discrete decisions, which will be used in this research. The optimization problems presented in this thesis will follow the minimization standard form specified in this section. Therefore, the objective function will be negated for a maximization problem.

Provided that $f(x, y)$, $h_i(x, y)$ and $g_j(x, y)$ are linear functions, this reduces (2.1) to a mixed-integer linear program (MILP) of the form

$$
\begin{aligned}
\min \quad & \mathbf{a}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} + c \\
\text{s.t.} \quad & \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{y} = \mathbf{k}, \\
& \mathbf{G}_{\text{eq}}\mathbf{x} + \mathbf{H}_{\text{eq}}\mathbf{y} \leq \mathbf{k}_{\text{eq}}, \\
& \mathbf{x} \in \mathbb{R}, \\
& \mathbf{y} \in \mathbb{Z}.
\end{aligned} \tag{2.2}
$$

Moreover, an MILP is reduced to an integer program (IP) if there are only discrete variables. Conversely, if (2.1) consists of only continuous variables, then it is reduced to a nonlinear program (NLP) of the form

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & h_i(x) = 0, \\
& g_j(x) \leq 0, \\
& x \in \mathbb{R}, \\
& i = \{1, 2, 3, ..., n\}, \\
& j = \{1, 2, 3, ..., m\}.
\end{aligned} \tag{2.3}
$$

The NLP problem can be reduced to a quadratic program (QP) if $f(x, y)$ is a quadratic function and both $h_i(x, y)$ and $g_j(x, y)$ are all linear functions. Furthermore, if all the functions in (2.3) are linear functions, then the NLP problem is reduced to linear program (LP).

As shown in Table 2.2, different mathematical programming problems are encountered in various areas of process systems engineering, from process synthesis and design to scheduling and planning to unit operation models. Generally, most optimization problems in Table 2.2 are non-trivial and difficult to find the global optimum solution as there can be many alternative solutions that can be directly linked to complex economics and performance of the system. Furthermore, often the optimal solution translates to large economic savings that can become a major disadvantage if a suboptimal solution is selected. Therefore, optimization has become and continues to be a powerful tool for various industries and companies to remain competitive.

Table 2.2: Overview of the different optimization problems that are encountered in process systems engineering (adapted from [4]).

	LP	**MILP**	**QP**	**NLP**	**MINLP**
Process design and synthesis					
Heat exchangers	✓	✓		✓	✓
Mass exchangers	✓	✓		✓	✓
Separations		✓		✓	✓
Reactors	✓			✓	✓
Flow sheeting				✓	✓
Process operations					
Scheduling	✓	✓			✓
Supply chain	✓	✓			✓
Real-time optimization	✓		✓	✓	✓

While the optimization formulations that will be discussed in this thesis are NLP (Chapter 3 and 4), MINLP (Chapter 6) and MILP (Chapter 7), the thesis is focused on the integration between the formulation and the optimization solvers, and not necessarily targeting the formulation to the optimization areas in Table 2.2.

There are many integer and linear techniques that can be implemented to reformulate a nonlinear problem to other optimization formulations in order to achieve various advantages, such as better performance, compatibility with certain optimization solvers, and cope with financial limitations. In regards to ORC optimization, they are naturally nonlinear problems, and fall either in the NLP or MINLP category due to the direct use of thermodynamic and performance equations. The ORC models are generally optimized as a nonlinear problem, especially for large systems, and do not undergo any type of reformulation or linearization. This is a gap in the literature that will be investigated in Chapters 6 and 7. As will be shown in those chapters, it is possible to convert from an NLP problem to an MINLP or an MILP problem using integer/linear programming techniques in order to achieve a higher model accuracy, a better optimization performance, and/or to tailor the model for a specific group of optimization solvers.

There are a wide range of different optimization algorithms and techniques that can be used to solve the optimization problems in Table 2.2. Some optimization strategies are more suited for problems with a larger number of inequality constraints, such as the barrier (or interior point) method [63]. While other optimization methods are more suitable for problems that are unconstrained and have non-differentiable functions, such as derivative-free global algorithms [64]. Therefore, in addition to the type of variables and constraints, as classified above, the optimization algorithm is also selected based other characteristics of the optimization problem, such as the number of inequality and/or equality constraints, convexity, matrix sparsity information, differentiability of functions, and model structure. It is not enough to only have state-of-the-art optimization algorithms for an efficient and successful optimization, but it is also as important that the optimization model is tightly integrated with the algorithms to work together in synergy. With the advent of advanced optimization modelling environments, more attention can be focused on formulating ORC optimization problems that are tailored for specific a group of solvers. Some of these environments provide high-fidelity unit operation models, regression tools and inbuilt optimization solvers that allow users to build a model that can be solved within platform without the need for external interfaces to other engineering models or optimization toolboxes.

2.5 Optimization Algorithms

With a better understanding of the advantages and disadvantages of different optimization algorithms, practitioners can now focus more on formulating well-posed models that are tailored to specific solvers and improve the overall performance. Depending on the type of problem and the underlying equations and variables, different algorithms are used to solve the optimization problem. The following subsections will briefly describe some of the common algorithms implemented in different optimization solvers.

2.5.1 Nonlinear Programming Algorithms

One of the main difficulties in solving NLP problems is dealing with inequality constraints because they can either act as equality constraints or just strictly satisfies the inequality condition at the optimal solution. Most derivative-based NLP solvers, like `fmincon` and IPOPT, are Newton-type algorithms and address this issue using the active-set sequential quadratic programming (SQP) strategy or the interior point (IP) method. In the active-set SQP approach [19, 64], the inequality constraints are addressed by sequentially solving an approximated quadratic programming problem of the original NLP problem with the Karush-Kuhn-Tucker (KKT) conditions using the Newton steps. In contrast, the IP method [19, 64] attempts to solve NLP problems by adding a penalty term to the objective function in order to address the inequality constraints. The penalty term ensures that the solution at each iteration is within the feasible region and thus will satisfy the constraints at the optimal solution. Similar to the active-set SQP method, the penalized NLP problem with the KKT conditions are solved using the Newton steps.

Alternatively, there are also a set derivative-free solvers do not require any derivative information and rely on the objective and constraint evaluations to solve the optimization problem. These solvers are mostly based on the simplex and pattern search methods, such as classical direct search algorithms [65], or based on the physical analogies of heuristics, such as genetic algorithms [66] that were derived from the evolution of natural selection, and simulated annealing [67] that was inspired by the process of annealing in metallurgy. As such, they can be applied to a variety of optimization models and problems where the derivative information of the problem is not available, inaccurate, or impossible to obtain. While it can be easy to apply these solvers to a wide range of optimization models and problems, derivative-free solvers do not scale well with the number of decision variables [63] and have difficulties with complex constrained problems [68]. Therefore, they are more suitable for optimization problems that are unconstrained or bound constrained and consist no more than a few dozen variables [63].

Unless optimizing convex problems, these NLP algorithms can only ensure local optimality and are dependent on a good starting point and accurate derivatives (for derivative-based solvers). Modern NLP solvers are now significantly more reliable and are widely adopted due to the development of the quasi-Newton method and globalization strategies, and the more recent filter method [69], that helped to overcome the difficulties in dealing with indefinite Hessians and poor starting points [64, 19].

2.5.2 Mixed-Integer Linear Programming Algorithms

Most of the mixed-integer linear programming methods are derived from the branch and bound algorithm, where the MILP problems are "branched" into subproblems, called nodes, and solved as relaxed LP problems at imposed bounds. The MILP problem will continue to recursively split into nodes until the difference between the upper bound (integer solution) and lower bound (LP solution) are within an acceptable tolerance. Another well-known approach for solving MILP problems is the cutting plane method. This approach works by relaxing the MILP problem to a complementary LP problem and cuts parts of the relaxed continuous feasible region until the solution search space only contains feasible solutions [64]. Therefore, this will result in a convex hull reformulation of the MILP problem where an integer point is at every extreme point of the feasible region. Once the convex hull formation of the MILP is achieved, the optimal solution of the original problem can be found by solving the LP relaxation.

Since the 1990s, there have been significant advancements in MILP methods, such as better pre-processing techniques, more efficient LP codes, different heuristics, better branching rules, and incorporating cutting planes to the branch and bound method, which have greatly improve the performance of MILP solvers [70]. For some perspective, based on a comparison test of the performance of the CPLEX solver on 1,852 real-world examples of MILP problems, the 2007 version (11.0) of CPLEX was on average 30,000x faster than the 1991 version (1.2) [70]. In other words, the speed of the solver increased roughly twofold every year. In addition, out of the all the problems that were tested, the 1991 version only managed to solve 15% of all the problems, while the 2007 version was able to solve 69% of all the problems. Similarly, a performance test carried out internally by the commercial solver GUROBI developers showed that the 2018 version (8.0) was about 53x faster than the 2009 version (1.1)[71].

2.5.3 Mixed-Integer Nonlinear Programming Algorithms

The MINLP algorithms are often built by amalgamating other algorithms from nonlinear programming, linear programming, and integer programming, and thus MINLP solvers are generally not completely developed from the ground up [72]. In most cases, the basis of the MINLP algorithms is built on other LP, MIP and NLP solvers, e.g., IPOPT, FilterSQP, CBC, and CPLEX. Therefore, some MINLP solver will revert to the MIP and NLP solvers that are provided in the optimization environment to solve the necessary subproblems.

One of the most common MINLP algorithms is the branch and bound method, which is a direct extension of the MILP approach where NLP relaxation subproblems are solved at each node, instead of LP relaxation subproblems [64]. The resulting NLP relaxation is then solved by an NLP solver. Another common MINLP algorithm is the outer-approximation method [73, 74, 75] where instead of relaxing the integrality restrictions and retaining the nonlinear constraints, the solver retains the integrality constraints and relax the nonlinear functions [72]. As a result, the algorithm uses gradient-based linearizations of the nonlinear functions to obtain a relaxation at solution points of NLP subproblems. Subsequently, an MIP solver is then used to solve the resulting MIP relaxation. Generally, MINLP problems are much harder to solve than NLP and MILP problems, as they involve solving both kinds of problems together and are solved iteratively multiple times.

2.6 Optimization Modelling Environments

Generally, large and complex chemical process systems are modelled and optimized in advanced optimization modelling environments. Depending on the type of the models and the capability of the modelling environments, the models can be simulated in a dynamic or steady-state condition. These models should ideally provide an accurate representation of the original process over a large range of operating conditions to allow for optimization, and thus potentially give the systems a competitive advantage over their current state.

During 1955-1959, simulations were limited to design calculations for single units due to the slow speed and size of computers, and the limited availability of high-level computer languages [76]. In 1958, the M.W. Kelloggs Corp introduced Flexible Flowsheet that implemented the sequential-modular approach that solved each unit module separately in a sequential order [76]. By the early 1960s, the development of process modelling environments based on the sequential-modular approach was growing in popularity. In addition to the sequential-modular approach, another method called the equation-oriented approach was also developed during this period, where models are represented as a set of mathematical equations and are solved simultaneously. With the advent of powerful large-scale computers between 1965 and 1969, more flowsheet programs were becoming a reliable and cost-effective simulation tool. Nowadays, most process simulators are usually written in C++ or Java using an object-oriented approach [5] that have more capabilities to handle large and complex system with more than 100,000 equations [77, 4] by providing built-in tools to exploit the model structure and derivative information for an efficient modelling and optimization. Table 2.3 shows some of the early and current process system modelling/optimization environment.

The following subsections will discuss the two common modelling approaches, namely, sequential-modular and equation-oriented approach, and highlight some of the advantages and disadvantages of each method.

Table 2.3: Process system modelling/optimization environments (adapted from [5, 6]).

Name	**Year**	**Developer**	**Type**	**Application(s)**
PACER	1963	Paul T. Shannon	Sequential-modular	Steady-state flowsheet [78]
PROCESS	1966	Simulation Sciences	Sequential-modular	Steady-state simulation
DESIGN	1969	ChemShare	Sequential-modular	For gas and oil applications
CHESS	1969	Univ.of Houston	Sequential-modular	Flowsheeting/sizing/costing
FLOWTRAN	1970	Monsanto, United States	Sequential-modular	Flowsheeting/sizing/costing
FLOWPACK II	1972	ICI	Sequential-modular	Flowsheeting
ASCEND	1970-1980	Carnegie Mellon Univ.	Equation-oriented	Dynamic simulation
Aspen Plus	1976	U.S. Dept. of Energy, MIT	Sequential-modular	All-purpose flowsheeting system
TISFLO	1970-1980	DSM, the Netherlands	Equation-oriented	Flowsheet simulation [79]
FLOWSIM	Early 1980s	Univ. of Connecticut	Equation-oriented	Chemical process flowsheet simulator [80]
SPEEDUP	1986	Imperial College	Equation-oriented	Dynamic simulation
ProSimPlus	1989	ProSim	Sequential-modular	Steady-state simulation
Quasilin	Late 1980s	CADCentre	Equation-oriented	Simulation/optimization
gPROMS	1997 [81]	Imperial College; PSE Ltd.	Equation-oriented	Steady-state/dynamic modelling
COCO Simulator		AmsterCHEM	Sequential-modular	Flowsheeting

2.6.1 Sequential-Modular Modelling

One of the oldest approaches to flowsheeting and still the dominant method for steady-state simulation is the sequential-modular (SM) approach [82]. This approach solves each unit module/unit operation in the order of the plant's process, thus the output stream of each module (plant equipment) is calculated from the input stream with respect to the module's parameters [83]. In many cases, the environments allow the user to "drag & drop" high-fidelity unit operations into the flowsheet and connect them together. Therefore, the stream information, i.e., mass flow, temperature, pressure, etc., will propagate from the beginning of the plant process through to the end of the flowsheet.

Solving SM models can be as simple as a "once-through calculation" for processes without recycles of material and/or energy, where each unit operation is solved separately in a sequential order when all the input streams and unit operation parameters are given or can be calculated. For recycle processes, like the ORC system, they are solved using a nonlinear equation solver, such as MATLAB's `fsolve` or Excel's built-in nonlinear solver, where a particular stream in the recycle is first estimated and then the unit operations are solved iteratively in the loop until the stream converges to an acceptable tolerance, e.g., the change in plant parameters between iterations must be < 0.0001. Consequently, this method is not efficient for optimization because the flowsheet needs to convergence to a new physically realisable point at every optimization iteration, which can be expensive to execute, especially with a large and complex system with multiple recycles.

Due to the intuitive and logical procedure of the sequential-modular approach, it is generally preferred and implemented by commercial flowsheet tools and process engineers [84, 83]. In addition, given that the flowsheet is a series of unit operations connected together in the order of plant process, it relatively easy to troubleshoot and debug the model, especially for small and simple systems. Generally, the SM approach can produce high-fidelity models due to the use of rigorous unit operation modules and thermodynamic routines, which are sometimes externally interfaced to the modelling environment, that have been pre-built and validated extensively by first or third-party users and developers. This means that the model is typically a good representation of the original system and can easily be modified to accommodate various level of accuracy.

Some common SM-based modelling software include Aspen Plus [13], JSteam Excel Add-In [2], and COCO [85].

2.6.2 Equation-Oriented Modelling

The equation-oriented (EO) model represents a system as a set of equations that are solved simultaneously; thus, this means that recycles can be treated as another set of equations. Therefore, it is generally much more efficient and faster to solve EO models than the SM-based counterparts by eliminating or reducing the need to iterate and converge the system recycle(s). The EO approach can be dated back to the 1960s [4] and was predominantly limited to the academic domain at this time [86], hence why most of the early software packages were developed by universities and not industrial companies.

Some of the early EO-based modelling platforms include SPEEDUP developed by the Imperial College London in 1986 [6], Quasilin developed Cambridge University in 1988 and ASCEND (now known as Aspen Custom Modeler) developed by Carnegie Mellon University in 1996 [87]. These modelling environments allow practitioners to construct models by defining variables and writing equations to represent unit operations. Model flowsheets could then be built from these predefined unit operations. Due to the limited memory and the slow speed of early generation computers, these packages were not adequate to solve large and complex chemical engineering problems. Therefore, in some cases, the complexity of the problems was oversimplified and typically led to a misleading representation of the system behaviour [6].

With the improvement of modern computer technology, software engineering, and advanced optimization algorithms, we see the advent of powerful commercial modelling environments that are capable of processing complex chemical processing system. Some current EO-based modelling environments include AMPL [88] and GAMS [89]. These environments take advantage of the EO model structure and some offer tools that can supply first and second derivatives and matrix sparsity information that can be beneficial to gradient-based solvers.

One major advantage of the EO approach is that it allows standard optimization problems to be formulated relatively efficiently from the model structure, thus the system of equations, i.e., mass and energy balance equations, are solved simultaneously with the optimization problem [20]. This bypasses the need to converge the flowsheet at every optimization iteration. However, given that the system is expressed as a set of mathematical equations, it can be harder to troubleshoot and construct than the SM-based model, especially for large and complex systems that consist of multiple recycles.

2.7 Optimization Solvers

There are hundreds of optimization solvers that are available to academic researchers and commercial users [90]. The selection of what type of solver to use depends on the structure/format of the simulation model, as well as the optimization problem, which can involve the type and number of variables and constraints, and the convexity and the nonlinearity of the optimization problem. While it is possible to simply wrap a solver around a simulation model, such as with the SM model, and allow the solver to search for a better simulation solution, this can be very inefficient and lead to a long execution time. Therefore, the difficulty that many researchers face is in structuring the optimization problem in order for it to be suitable and tailored to certain optimization solvers. More than often, this is not a trivial task as most problems are very complex and there are certain limitations and requirements when using some of the optimization solvers.

Due to the nonlinear mass and energy balance equations and the thermodynamic correlations that are used in the steady-state model of an ORC system, the optimization problems that are formulated from the model are generally nonconvex and have multiple local optima. While it is important to achieve robust and efficient optimization, we believe that it is as important to obtain the global optimal solution over the suboptimal solution because the difference could translate to a large economic advantage. Therefore, the following subsections will discuss the advantages and disadvantages of two types of optimization solvers, namely, black-box and white-box solvers, and highlight certain types of global solvers that can deterministically guarantee global optimality.

2.7.1 Black-box Solvers

When an optimizer cannot exploit the internals of the objective function and constraints of the optimization problem as they are generally supplied as function callbacks, it is referred to as a *black-box solver*. This is common with the SM models where the unit operation modules and external function calls make it difficult to obtain accurate derivatives, the underlying equations, and the relationship between the decision variables and the constraints or objective function. As a result, these models are usually optimized using black-box solvers where the problem is entirely or partially supplied as functions that accept in a decision variable vector and output an objective value or a vector of constraint evaluations.

Black-box solvers can be classified into two main categories, namely, derivative-based solvers and derivative-free solvers. The former category of solvers require the derivatives of the constraints and objective function in order to solve the problem. Examples of these solvers are IPOPT [18], `fmincon` [17], FILTERSD [91], KNITRO [92] and SNOPT [93]. The derivative information is obtained from a number of methods, such as finite difference, automatic differentiation, or symbolic differentiation, which are discussed in Section 2.9. Sometimes, obtaining these derivatives can be troublesome and computationally expensive. For simple problems that can be expressed using basic algebraic equations, such as linear and quadratic problems, the derivatives can be calculated analytically (to numerical precision) and supplied to the optimizers. However, for more complex nonlinear problems that require calling external functions, such as REFPROP function routines, it can be computationally expensive and difficult to obtain accurate derivative information.

As mention in Section 2.5.1, derivative-free solvers do not rely on the derivative information of the problem, but instead the function evaluations to find the optimal solution. Many of derivative-free solvers are derived from heuristic strategies, such as genetic algorithms, simulated annealing, and particle swarm algorithm, and perform random search methods to find the solution. Therefore, derivative-free solvers are suitable for dealing with black-box models/functions where the underlying equations are basically unknown to the solvers. Examples of derivative-free solvers include `ga` [94], `patternsearch` [94], NOMAD [95], and PSWARM [96]. While some derivative-free solvers, such as `patternsearch`, genetic algorithm and particle swarm, are often referred to as *global optimization methods* [68], it is important to note that they cannot guarantee the global solution. This can be misleading and different from certain white-box solvers in the next subsection that can deterministically converge to the global solution.

2.7.2 White-box Solvers

When an optimizer can exploit the internals of the objective function and constraints, and has access to the structure of the underlying equations, the relationship between the decision variables and the objective function or constraints, and accurate derivative information, it is referred to as a *white-box solver*. Therefore, this includes all (MI)LP/QP/QCQPs solvers because the entire optimization problem can be supplied as a series of vectors and matrices. In order to achieve the same level of accessibility with nonlinear optimization problems and present the underlying equations and correlations between decision variables and constraints, an algebraic model needs to be provided to the solvers. This means that the optimization model must meet certain requirements and cannot consist of stochastic terms or conditional statements, and only consists of certain operations and functions, such as `log`, `power`, and `exp`, that are permitted by the solvers. As a result, this allows the solvers to exploit the model structure of the problem and carry out preprocess methods to reduce the search space, generate the derivative internally, and identify the underlying relationships between the variables and constraints to implement relaxation methods and branching algorithms to efficiently solve the nonlinear problems.

In addition, unlike the black-box solvers discussed in the previous subsection, certain white-box solvers can leverage of the algebraic structure to deterministically guarantee the *global solution* for nonconvex optimization problems. This is different from the stochastic methods, such as genetic algorithms, particle swarm, pattern search, and simulation annealing, that are sometimes classified as global optimization methods but require infinite runtime to guarantee a global solution [23]. These white-box solvers are based on the spatial branch and bound method that partitions the feasible region into subregions. Subsequently, at each subregion, the lower bound is computed from the convex relaxation of the objective function and constraints, and the upper bound is computed from the objective function of the original nonconvex problem [63]. The algorithm compares and eliminates all subregions that have infeasible constraint relaxations or lower bounds that are higher than the lowest upper bound. This process repeats until the upper and lower bound converge and is within an acceptable tolerance. Examples of white-box solvers include ANTIGONE [97], BARON [22], SCIP [98], and COUENNE [99].

In contrast, some (MI)LP/QP/QCQPs solvers, such as CPLEX [100], can also find the global solution for nonconvex problems by using the Boolean Quadric Polytope (BQP) cuts that exploit a cutting plane technique to efficiently solve QP and MIQP models to global optimality [101]. This feature can easily be invoked by setting the solver's optimality target parameter and solving the optimization problem as usual. Unfortunately, the BQP cuts in CPLEX does not apply to mixed-integer linear programs (MILP).

Given that most ORC models are constructed using rigorous and detailed unit operations that are developed from complex semi-empirical correlations, experimental data, the equation of state, etc., they are usually restricted to black-box solvers, as observed in studies by [30, 31, 102, 103, 104]. This can be a significant disadvantage for low thermal efficiency ORC systems where a small difference in the capital cost, the performance and/or the investment returns can mean the difference between a viable plant and one that is not. In addition, since ORC systems can typically operate up to 25+ years, these small improvements can add up and significantly benefit the overall performance and longevity of the plant. Therefore, it is important to achieve global optimality in order to ensure that the optimized ORC system has a competitive advantage over other plants in the industry.

2.8 Optimization Toolboxes

Developing a fully functioning state-of-the-art optimization solver and managing its code can be a strenuous task and takes many years to build a reliable and robust solver. Therefore, process modelling environments, such as MATLAB, GAMS, AMPL and AIMMS, typically will have some built-in optimization solvers and/or provide interfaces to third-party solvers that are managed separately from the integrated development environment (IDE). This allows a range of optimization solvers to be available to the end users, albeit there might be some license terms and conditions associated with each individual solver that can restrict their use. Generally, commercial solvers will perform better than open-source solvers because they are more frequently updated and typically better funded. In addition to the optimization solvers, other functionalities and packages are sometimes included with the toolbox/IDE to help build and formulate the optimization problem, such as thermodynamic property packages, symbolic toolboxes, and intuitive APIs, which can significantly improve the overall optimization study. The following subsections will detail the optimization toolbox that was used in this research and also highlight other alternatives toolboxes.

2.8.1 OPTI Toolbox

For this research, the OPTimization Interface (OPTI) Toolbox [105] was used as the main optimization solver supplier/interface. This optimization toolbox provides an extensive set of commercial and open-source solvers for linear, nonlinear, continuous and discrete optimization problems, including powerful solvers such as BARON [22], CPLEX [100], IPOPT [18] and NOMAD [95], see [105] for the full list of solvers.

One of the main advantages of OPTI Toolbox is its compatibility with MATLAB, which offers a huge standard library for scientific computing and a high-level programming language that allows for object-oriented programming to make the development and implementation of the proposed modelling framework relatively simple. In addition, the figure plotting capability of MATLAB is significantly efficient, simple to utilize and is compatible with other MATLAB toolboxes without worrying about incompatible library packages. This is a valuable feature in this research due to the need to visually analyse the regression fits in order to help improve the ORC model approximations. In addition, OPTI Toolbox provides the SymBuilder Framework that builds on MATLAB's Symbolic Math Toolbox that will automatically generate the required Jacobian and Hessian matrices from the proposed algebraic optimization problem, thus allowing users to concentrate on the model formulation.

2.8.2 Alternative Optimization Toolboxes

Every optimization toolbox has their advantages and disadvantages, and the selection of which one to use can be based on many factors, such as the availability of solvers, IDE compatibility, cost, usability and technical support. The MATLAB Optimization Toolbox [94] provides a wide range of in-built optimization solvers for LP, MILP, QP, and NLP problems, that handle most of the process engineering problems in Table 2.2. However, the toolbox is not free with MATLAB and needs to be purchased separately from the IDE. In addition, there are no solvers that can handle MINLP problems and they are all restricted to only black-box optimization, which is a limitation for some areas of this research.

The closest optimization toolbox to OPTI Toolbox is YALMIP [106], which consists of 6 internal solvers and supports up to 59 external solvers. While YALMIP contains white-box global solvers, they are generally intended for bilinear problems where polynomial programs with high order polynomial functions are automatically converted to bilinear programs by the interface [107]. This is not suitable for this work since it centres around the formulations of the ORC models and their contributions.

Another alternative toolbox is Google Optimization Tools (OR-Tools) [108], which is an open-source software suite that is mainly aimed for linear programming and mixed-integer programming problems. While the software is actively maintained, it is quite limited in the number of solvers available. Currently, there only seven solvers, namely, CBC, CLP, GLOP, GLPK, GUROBI, CPLEX and SCIP, which are mostly LP, MILP and MIQP solvers.

Newer frameworks such as PyOMO [109] and JuMP [110] are available for Python and Julia, respectively, and they provide a wide range of solvers that can solve LP to MINLP problems. However, given that MATLAB is the main IDE for this research, both PyOMO and JuMP are currently not suitable for this research.

2.9 The Importance of Accurate Derivatives

It is widely acknowledged that providing an inaccurate gradient or Jacobian and Hessian information can significantly degrade the performance of the optimizer [19, 84]. Failure to obtain accurate gradients can lead to three issues. First, inaccurate derivatives can lead to premature termination of the algorithm due to the calculations of search directions that may not lead to descent directions. Second, inaccurate gradients can lead to ill-conditioned Hessians (or approximations of the

Hessians) that can result in poor search directions and more iterations. Third, the optimization algorithm will not be able to confirm that the solution is an optimum if the derivative information is not accurate, even if it has found the solution. Generally, derivative-based solvers will allow the user to enter the Jacobian and Hessian information that can help reduce or bypass the issues mentioned above. If the derivative information is not supplied to the solver manually, there are a number of methods that derivative-based optimizers can implement in order to acquire the necessary derivative information. The following subsections discuss the advantages and disadvantages of the common methods, namely, the finite difference method, the automatic differentiation and the analytical differentiation, that are used by derivative-based solvers.

2.9.1 Finite Difference Method

By default, most algorithms will use the finite difference method (FDM) to obtain the required derivatives if the gradient information is not provided. One major advantage of the FDM is that it will work on any smooth function, even if it is an external function (such as thermodynamic property routines from REFPROP [35]) or a MATLAB based function. The general principle of the FDM is very simple, the derivative is approximated by dividing the difference between two close points on the function by the distance between the two points. Mathematically, this can be expressed as

$$\overset{\text{forward difference}}{f'(x) \approx \frac{f(x+h)-f(x)}{h}} \quad \text{or } \overset{\text{backward difference}}{f'(x) \approx \frac{f(x)-f(x-h)}{h}} \tag{2.4}$$

for the forward difference and the backward difference formula for a univariate function, where $f'(x)$ is the approximate gradient at x, $f(x)$ is the function evaluation at x, $f(x+h)$ is the function evaluation at $x+h$, and h is a finite small number but not necessarily infinitesimally small. It is also possible to approximate by using the centred difference that can yield a more accurate approximation, which is defined as

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}. \tag{2.5}$$

The FDM can be extended to find the higher-order derivatives and also for multivariable functions that are common in nonlinear optimization problems. However, one major downside to the FDM is that it can only provide an estimate of the derivatives and can lead to optimization problems mentioned in the Section 2.9. Furthermore, the accuracy of the approximation can significantly degrade for higher-order derivatives, especially if the forward or backward difference is used.

2.9.2 Automatic Differentiation

Automatic differentiation (AD), also known as computational differentiation or algorithmic differentiation, is a standard method for most optimizing environments and is an efficient way to compute derivatives to numerical precision. The benefit from obtaining accurate first and second derivative information using the AD approach allows the implementation of very efficient large-scale NLP algorithms that are able to handle up to 100,000 variables and constraints [4], which is a significant advantage over the FDM. The AD method builds on the idea that a computer consists of primitive operations (such as addition, multiplication, and division) and functions that can be overloaded with a derivative rule in conjunction with the function value. Subsequently, the derivative of the complex function is then just a repeated implementation of the chain-rule to the composition of the primitive function sequence. For further information on the AD method, refer to [111] for an excellent approach on how to implement the AD method in MATLAB by utilizing the object-oriented concept and overloading the original arithmetic operators and elementary functions with a derivative rule alongside the function value. This allows the derivatives to be calculated automatically when the user evaluates a mathematical expression. The drawback of the AD method is that it will only work with the operators and functions that have been overloaded, hence the derivatives of external functions cannot be computed.

2.9.3 Symbolic Differentiation

Symbolic differentiation (SD), or analytical differentiation, calculates the derivatives by implementing the differentiation rules, such as the product rule, the chain rule, the inverse function rule, etc., to a symbolic mathematical equation. Generally, this method is implemented using a symbolic mathematics toolbox, such as SymPy in Python [112] and Symbolic Math Toolbox in MATLAB [113], that helps define the variables and mathematical equations in order to symbolically differentiate and generate the first and second order derivative expression. This means that the method limited to the available differentiation rules and requires the functions to be expressed symbolically. In addition, generating the analytical expressions can be computationally expensive and take longer to execute, especially with second derivatives and when there are a lot of variables and equations to differentiate. However, the benefit of this approach is that the derivative expressions only need to be derived once prior to the optimization process, and then they can be reused for any number of optimization studies. Furthermore, similar to the AD approach, the accuracy of the computed derivatives is to numerical precision and thus can significantly improve the performance and the computational time of the optimizer. However, the SD method can be slower than the AD approach if the derivative matrices are very dense.

2.10 A Review of Shortcomings in the Literature

This chapter focused on the literature review of the common practices and tools that are associated with ORC system optimization. The review began with a brief description of the ORC process as a binary cycle power plant and highlight the significant potential of exploiting the abundant low-temperature resources around the world and consequently reducing greenhouse gas emissions and pollutants. Given the different range of topologies and operating conditions, the ORC system needs to be modelled and optimized to ensure their economic viability and optimal operation.

Generally, in the literature, studies on ORC system optimization are focused on the results of the optimization problem and not the efficiency or robustness of the modelling and optimization procedure. Typically, an SM model of ORC system will be constructed in a process modelling environment, such as Aspen Plus, with an optimization solver wrapped around the model that aims to search for a better simulation solution. While this can sometimes be acceptable for small "toy" problems, it is not the most efficient way for optimization as it can restrict the derivative information and require a nonlinear equation solver for converging the system flowsheet and recycle(s). In addition, this approach limits the model to black-box optimization and thus generally cannot ensure global optimality.

While it is acknowledged that the EO approach is a better alternative to the SM approach, there is a lack of research on the EO formulation for the ORC systems, especially regarding large real-world ORC systems, that can provide accurate derivative and matrix sparsity information to the solvers. Furthermore, solving optimization problems with fast and reliable optimization solvers is only one aspect of achieving efficient optimization capabilities. There is a gap in the literature regarding the different optimization formulations of an ORC system, such as NLP, MINLP and MILP, that can be reformulated using various integer and linear programming techniques. There are advantages and disadvantages associated with each formulation and investigating these attributes can help lead to better optimization performance and/or model accuracy.

In order to address these shortcomings, this thesis will focus on the integration of the off-line steady state optimization models with state-of-the-art optimization algorithms to achieve an optimization strategy that is robust and efficient. This is addressed in three distinct parts. First, by providing a systematic modelling approach for ORC systems that are robust and efficient for optimization and also compatible with certain white-box solvers. Second, provide a regression tool to improve the performance of the optimization by without compromising on the accuracy of the model. Third, utilize the available linear/integer programming techniques to formulate an equivalent MILP model to take advantage of the available MILP solvers or when the NLP or MINLP models fail to converge. Figure 2.4 shows the general flow diagram of the research structure and the proposed modelling framework.

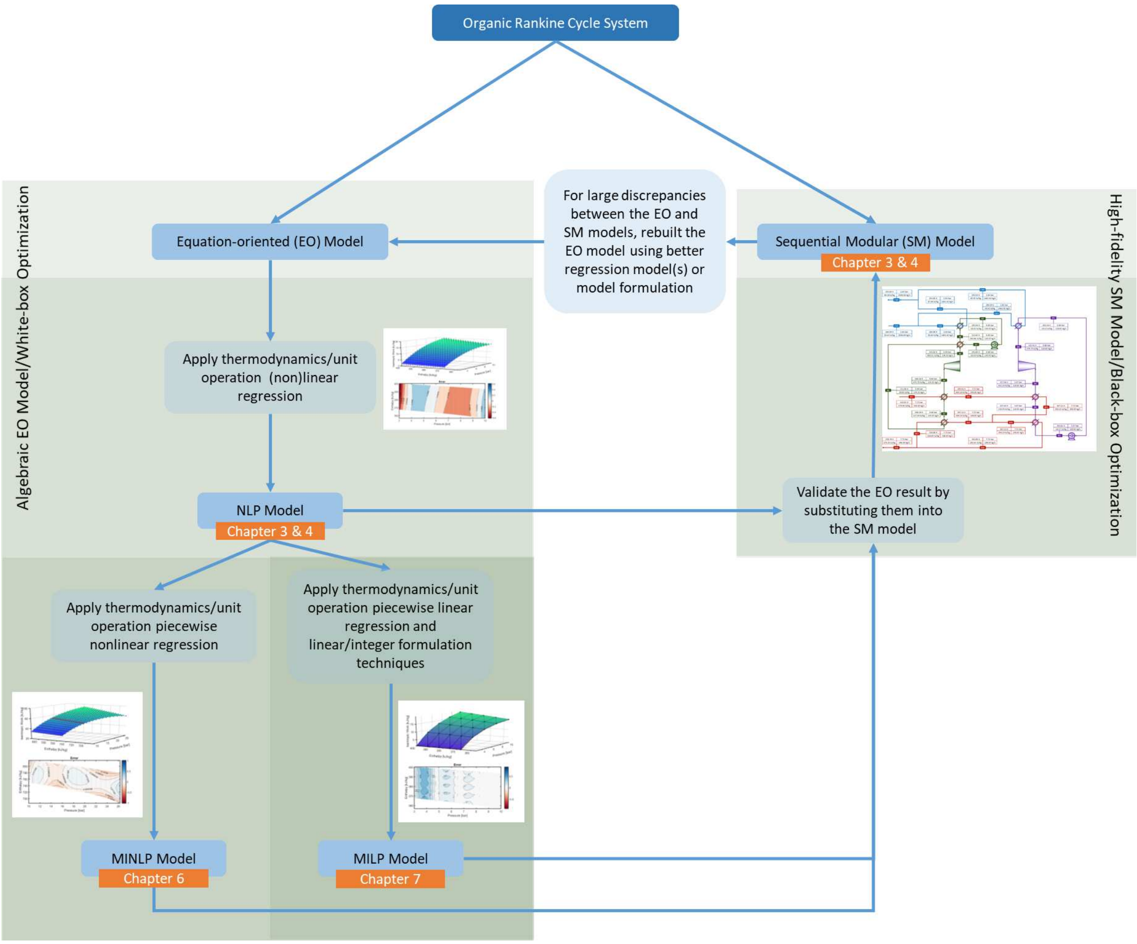

Organic Rankine Cycle System
Algebraic EO Model/White-box Optimization
High-fidelity SM Model/Black-box Optimization
Equation-oriented (EO) Model
For large discrepancies between the EO and SM models, rebuilt the EO model using better regression model(s) or model formulation
Sequential Modular (SM) Model
Chapter 3 & 4
Apply thermodynamics/unit operation (non)linear regression
NLP Model
Chapter 3 & 4
Validate the EO result by substituting them into the SM model
Apply thermodynamics/unit operation piecewise nonlinear regression
Apply thermodynamics/unit operation piecewise linear regression and linear/integer formulation techniques
MINLP Model
Chapter 6
MILP Model
Chapter 7

Figure 2.4: The structure flow diagram of this research that illustrates the relationship between the different optimization formulations and the modelling approaches.

The proposed model framework aims to address the optimization field concerning the steady-state operational conditions and the working fluid selection, which are common in the literature as presented in [1, 30, 23, 58]. Thus, this assumed that there is an existing plant with sufficient plant data or a detailed schematic plant diagram with a heat and mass balance analysis that can be used to construct a model. Therefore, this research does not try to address the configuration, sizing and design of the underlying equipment, as this is outside the scope of this study and other methodologies might be more suitable, such as the Pinch Technology [114].

This thesis will analyse three different advanced real-world ORC systems that vary in size and configuration in order to illustrate the optimization flexibility and efficiency of the proposed modelling framework. Two model types will be presented for each ORC system, namely, the EO and SM model, as shown in Figure 2.4. The purpose of the EO model is for optimization, whereas the SM model is intended for validating the optimized results of the EO models. There will be three different optimization formulations, i.e., NLP, MINLP, and MILP, that are derived from applying regression analysis and integer/linear programming techniques to the EO model, and they will be presented in the chapters highlighted in Figure 2.4. The advantages and disadvantages of each optimization formulation, as well as their research contributions, will be discussed in relation to the shortcomings mentioned above.

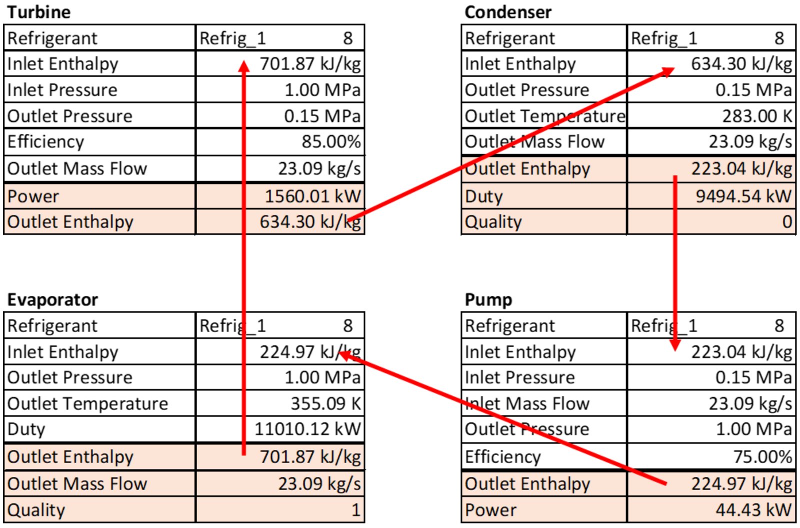
Chapter 3

The Algebraic Equation-Oriented Formulation

This chapter will investigate an algebraic equation-oriented (EO) modelling approach that is robust and efficient for optimization and is also compatible with white-box solvers. The modelling approach will be first implemented on a simple ORC system to concisely provide an overview of the proposed modelling procedure. Later, the same modelling approach will be applied to three real-world ORC systems in the next chapter. This chapter will begin with a discussion of the common sequential-modular (SM) optimization methods and highlight the advantages and disadvantages associated specifically with the ORC systems. This is then followed by a full description of the algebraic EO model development and optimization using both black-box and white-box solvers. Finally, an optimization performance of the algebraic EO model will be presented, as well as a discussion on how to leverage the high-fidelity feature of the SM model to validate the algebraic EO model.

3.1 Introduction

The formulation of the optimization model is one of the most important factors for a successful optimization study [63]. Often an ORC system is simulated using the SM approach, via modelling software such as Aspen Plus [13] or GateCycle [14], where the unit operation modules are linked together and solved sequentially in the direction of the plant's process, see Figure 3.1. This makes the flowsheet relatively straightforward and easy to solve and troubleshoot when the input stream at the beginning of the plant's process is known. In addition, the SM approach can produce high fidelity models due to the rigorous unit operation modules that are used to construct the models. Many SM tools offer a wide range of unit operation modules or allow external modules to be added to their extensive libraries that are a close representation of the original unit operations. This is an important feature that will be exploited later in this work to validate our algebraic EO models.



Turbine

Refrigerant	Refrig_1 8
Inlet Enthalpy	701.87 kJ/kg
Inlet Pressure	1.00 MPa
Outlet Pressure	0.15 MPa
Efficiency	85.00%
Outlet Mass Flow	23.09 kg/s
Power	1560.01 kW
Outlet Enthalpy	634.30 kJ/kg

Condenser

Refrigerant	Refrig_1 8
Inlet Enthalpy	634.30 kJ/kg
Outlet Pressure	0.15 MPa
Outlet Temperature	283.00 K
Outlet Mass Flow	23.09 kg/s
Outlet Enthalpy	223.04 kJ/kg
Duty	9494.54 kW
Quality	0

Evaporator

Refrigerant	Refrig_1 8
Inlet Enthalpy	224.97 kJ/kg
Outlet Pressure	1.00 MPa
Outlet Temperature	355.09 K
Duty	11010.12 kW
Outlet Enthalpy	701.87 kJ/kg
Outlet Mass Flow	23.09 kg/s
Quality	1

Pump

Refrigerant	Refrig_1 8
Inlet Enthalpy	223.04 kJ/kg
Inlet Pressure	0.15 MPa
Inlet Mass Flow	23.09 kg/s
Outlet Pressure	1.00 MPa
Efficiency	75.00%
Outlet Enthalpy	224.97 kJ/kg
Power	44.43 kW

Figure 3.1: A sequential-modular model of a basic ORC system using the JSteam Exel Add-In. Non-shaded cells are inputs and shaded cells are outputs.

Generally, the SM models are optimized using heuristic methods or optimization algorithms as demonstrated in the work carried out by [15, 28]. For small optimization problems with only a few variables and constraints, this approach can work reasonably well; however, for large ORC systems or large optimization problems, it can become very inefficient and can lead to various optimization issues. First, some of the plant variables are not easily accessible for optimization due to the rigidity of some of the plant's component modules in the modelling simulator. Therefore, either the practitioners develop a new module or find an alternative external module that can be interfaced into the environment. Second, it is very difficult to obtain accurate derivative information of the optimization problem, which is an important input argument for the derivative-based solvers to efficiently optimize the problem. In most cases, the solvers use the finite difference method to estimate the required derivatives, which can lead to various convergence and termination issues. Third, the optimization may be very slow or fail to converge due to difficulty in solving the internal recycle processes at every optimization iteration. Fourth, the optimization solvers that are compatible with the SM models generally cannot guarantee the global solution.

An alternative method is to use the equation-oriented approach, where systems are represented as a set of equations that are solved simultaneously. This opens up the model and allows the practitioners access to more decision variables that can be optimized. In contrast to the SM approach, EO approach allows standard optimization problems to be formulated relatively efficiently from the model structure, and the mass and energy balance equations are solved concurrently with the optimization problem [20]. This means that recycles in the system are treated as another set of equations and thus makes it significantly more efficient than the SM optimization approach. However, similar to the SM approach, one major problem about the EO approach is the difficulty of obtaining accurate derivatives from thermodynamic packages or complicated thermodynamic correlations that are used in the underlying EO model equations [115]. Consequently, the optimization solver will resort to using the finite difference method to estimate the derivatives, and this can result in various convergence and termination issues. While it is possible to use derivative-free solvers to avoid using or obtaining the derivative information, these solvers are generally more suitable for unconstrained or bound optimization problems, and they scale poorly with the number of variables [63]. In this research, this problem will be addressed by using regression analysis to approximate the thermodynamic correlations using differentiable nonlinear/piecewise functions over an acceptable range of temperatures and pressures that are applicable to the plant. The contribution of this chapter is the application of the proposed modelling approach to an ORC system to address the issues associated with the common optimization methods as mentioned above.

3.2 Algebraic Equation-Oriented Modelling Procedure

The task of constructing an algebraic EO model amounts to deriving a set of deterministic algebraic equations that describe the process of the system and approximating the output characteristic of the unit operations using regression and thermodynamic analysis. It will become apparent later in Chapter 4 that this modelling approach is non-trivial and requires a good understanding of the simulation problem so that the model accuracy is not over compromised in order to reduce the computational complexity of the optimization problem, especially with larger system models. This section will fully discuss the optimization procedure of a basic ORC system, as shown in Figure 3.2, to clearly present the proposed modelling approach. Later, the modelling approach will be applied to three different real-world ORC systems to highlight the contribution of the modelling approach.

3.2.1 Basic ORC System Description

The topology of a basic ORC system consists of a cycle that typically include a turbine, a condenser, a feed pump, and an evaporator. This type of ORC system has been analysed extensively in many academic papers [116, 38, 1] and in many standard thermodynamic textbooks such as [3, 117], thus it makes an excellent simple model for this demonstration. Therefore, the ORC system described in [1] was used for this research, which will be referred to as Basaran ORC (BORC) and corresponds to the process flow diagram in Figure 3.2. In order to coherently and clearly present the ORC system diagrams, the unit operations that are discussed in the following subsections and the process flow diagrams that are presented in this thesis will follow the colour scheme as described in Table 3.1. Furthermore, the equations of the unit operations in the following subsections do not only apply to Figure 3.2 but will also apply other ORC systems that will be discussed in this thesis, hence the equation indices will be generalised and do not specifically correlate to Figure 3.2.

Table 3.1: The colour scheme of the Organic Rankine Cycle unit operations and process flow diagrams that are presented in this thesis.

Red	Heat Source/Brine/Geofluid
Blue	Heat Sink/Cooling Water
Green	Working Fluid/ORC-A
Purple	Working Fluid/ORC-B

The constant parameters that were used for the basic ORC system are shown in Table 3.2, which were adapted from [1] to correspond with the original ORC system. Please note that P_{A5} was calculated using REFPROP [35] to ensure that the quality of the working fluid was at the saturated liquid point. Furthermore, the outlet temperature of the cooling water was decreased from 288 K to 282 K to ensure that the laws of thermodynamics are upheld, see Section 3.2.2.1.

Table 3.2: The constant parameters of the Basaran ORC system (adapted from [1]).

Plant Constant	**Value**	**Unit**
$\dot{m}_{\text{BR}}$	75	kg/s
T_{BR1}	369	K
T_{BR3}	334	K
P_{BR}	5	bar
T_{CW1}	280	K
T_{CW2}	282	K
P_{CW}	5	bar
η_{turb}	85	%
η_{pump}	75	%
T_{A5}	283	K
P_{A5}	$P_{f@T_{\text{A5}}}$	bar

For the purpose of this research, the ORC system will follow the same assumptions as the journal paper. This means the system is a steady-state and steady-flow

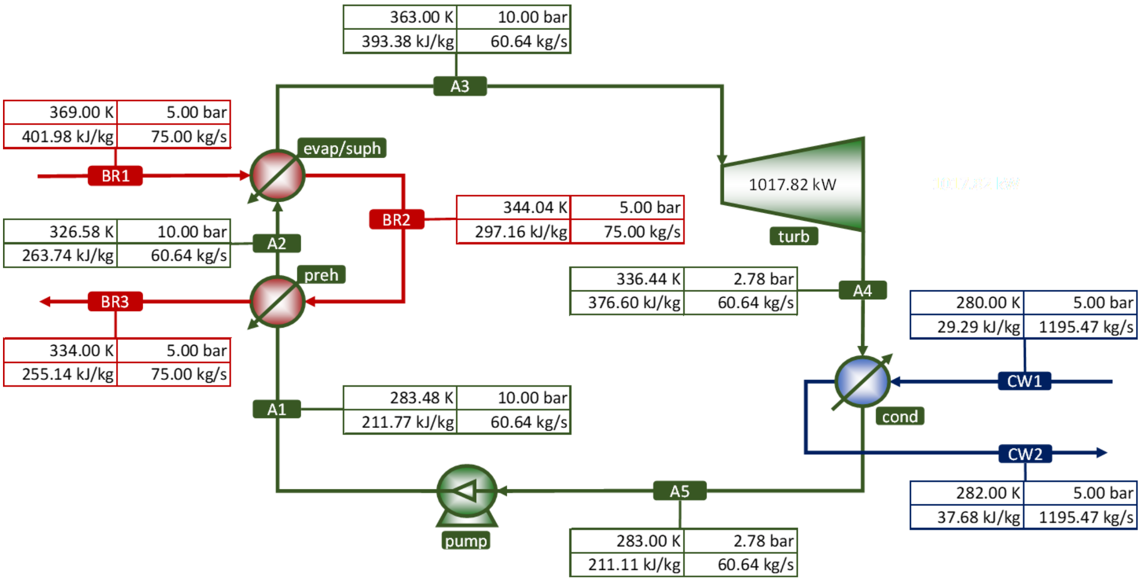


Figure 3.2: The process flow diagram of a basic Organic Rankine Cycle system as per [1] where the working fluid is R227ea. The plant was constructed in JSteam Excel Add-In v3.20 software [2].

.

process; whereby changes in kinetic and potential energy are neglected, and losses induced by friction are neglected. The thermodynamic and transport properties of the brine are treated as pure water, and any dissolved salts and non-condensible gases are neglected. Furthermore, the pressure drops across heat exchangers and pipelines are neglected. The following subsections will analyse the common unit operations that are found in the ORC systems, and they will be used for the rest of the thesis unless state otherwise. The simplifications and assumptions of the unit operation analysis are based on how the ORC models are presented and analysed in their respective references. Therefore, for a fair comparison and validation of the models, the analysis of the unit operations are assumed to be under ideal operation. However, it is possible to extend this modelling approach to account for a more realistic model by using regression methods to approximate the output characteristics of the unit operation model.

3.2.1.1 Turbine Analysis

The overall goal of the ORC system is to convert thermal energy into mechanical energy that can be used to generate electricity. This mechanical energy is produced when a high-pressure vapour expands through a turbine and causes a mechanical shaft to rotate. The thermodynamic analysis of the ORC turbine, see Figure 3.3, is familiar to that of a steam turbine.

Under the common assumption that the potential and kinetic energy are negli-

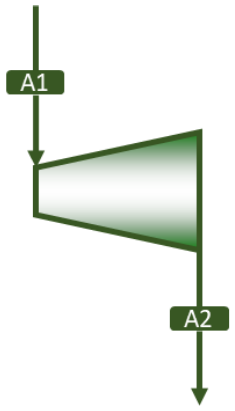


Figure 3.3: Turbine for organic Rankine cycle.

gible with adiabatic and steady-state operation, the output work is expressed as

$$\dot{W}_{\mathrm{turb}} = \dot{m}_{\mathrm{WF}}(h_{\mathrm{A1}} - h_{\mathrm{A2}}) = \eta_{\mathrm{turb}}\dot{m}_{\mathrm{WF}}(h_{\mathrm{A1}} - h_{\mathrm{is,A2}}), \tag{3.1}$$

where $\dot{m}_{\mathrm{WF}}$ is the mass flow rate of the working fluid, h is the enthalpy (where subscript is stands for the isentropic process), and η_{turb} is the isentropic turbine efficiency. Due to the $h_{\mathrm{is,A2}}$ term in (3.1), the output turbine work cannot be explicitly be stated in the algebraic EO model because of the thermodynamic correlation of the isentropic process, i.e., the input entropy must equal the output entropy. Therefore, the output work will need to be approximated as an algebraic function, which will be discussed in Section 3.2.2.4.

The turbine isentropic efficiency, η_{turb}, (as well as the feed pump in the next subsection) will be considered as a constant parameter in this research to remain consistent with what is stated in the original plant's reference and to simplify the ORC model. It is very common in the literature and in most textbooks that the isentropic efficiency is assumed as a fixed parameter, as shown in [29, 30, 31, 39, 118, 34, 3]. Therefore, the turbine models that are presented in this research will also use a constant isentropic efficiency.

However, it is possible to use a variable isentropic efficiency and incorporate it into the algebraic EO models that are presented in this research, but the isentropic efficiency must be expressed as an algebraic function that is compatible with the white-box solvers. For example, a study carried out by [119] considers the turbine isentropic efficiency as a function of the enthalpy drop, $(h_{\mathrm{A1}} - h_{\mathrm{A2}})$, and the volumetric flow rate at the turbine outlet. Therefore, the isentropic efficiency of the turbine can be treated as a set of constraints and expressions in the algebraic EO model.

3.2.1.2 Feed Pump Analysis

In an ORC process, the feed pump (see Figure 3.4) is required to increase the pressure of the working fluid and move the fluid to the heat exchanger(s) to be heated up by the heat source.

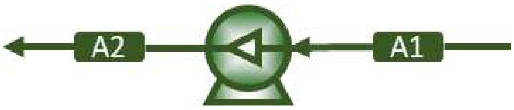


Figure 3.4: Pump for organic Rankine cycle.

Using the same assumptions as the turbine and the same variable nomenclature, the power imparted by the feed pump to the working fluid is

$$\dot{W}_{\text{pump}} = \dot{m}_{\text{WF}}(h_{\text{A2}} - h_{\text{A1}}) = \dot{m}_{\text{WF}}(h_{\text{is,A2}} - h_{\text{A1}})/\eta_{\text{pump}}, \tag{3.2}$$

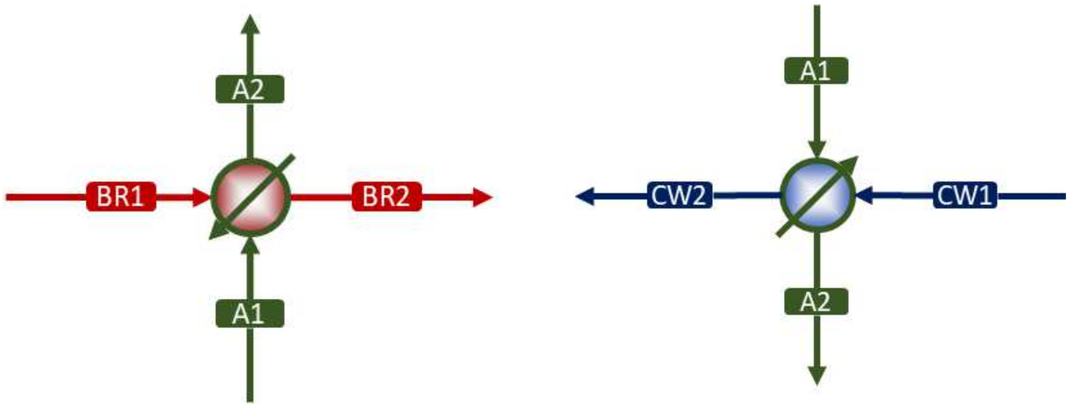
where η_{pump} is the isentropic pump efficiency. Similar to the turbine analysis, the feed pump work need to be approximated due to the $h_{\text{is,A2}}$ term in (3.2).

3.2.1.3 Heat Exchangers Analysis

The heat exchanger (see Figure 3.5) is the general term given to a unit operation that enables the thermal energy from the hot medium to be transferred to the cold medium. It is typical for the ORC system flowsheet to have a separate heat exchanger for the different stages of the heating or cooling process to provide accessibility to the pressures and temperatures at each point. Therefore, for the heating process, this might involve a preheater (or economizer) that heats the fluid to the saturated liquid point, an evaporator that vaporizes the fluid to the saturated vapour curve, and a superheater that heats the fluid to the superheated region. Conversely, for the cooling process, this might include a desuperheater that cools the fluid to the saturated vapour curve, and a condenser that condenses the fluid to the saturated liquid curve (or the subcooled region).

The thermodynamic analysis is the same for both the heater and cooler with the assumption that they are well-insulated and that all the heat transfer is only between the hot fluid and the cold fluid. It is also assumed that the heat exchanger operates in a steady-flow process and that the changes in the kinetic and potential energy are negligible. Under these assumptions, the governing equation for the heater as a thermodynamic system is

$$\dot{m}_{\text{WF}}(h_{\text{A2}} - h_{\text{A1}}) = \dot{m}_{\text{BR}}(h_{\text{BR1}} - h_{\text{BR2}}), \tag{3.3}$$



(a) A heater with geofluid/brine (BR) (b) A cooler with cooling water (CW)

Figure 3.5: Heat exchangers for organic Rankine cycle.

and for the cooler is

$$\dot{m}_{\mathrm{WF}}(h_{\mathrm{A1}} - h_{\mathrm{A2}}) = \dot{m}_{\mathrm{CW}}(h_{\mathrm{CW2}} - h_{\mathrm{CW1}}), \tag{3.4}$$

where both equations follow the same variable nomenclature as the previous unit operation analysis.

For the sake of completeness, the heat transfer surface area of the heat exchanger between the two fluids, A, can be calculated using the general heat transfer relationship:

$$\dot{Q}_{\mathrm{hx}} = UA\Delta T_{\mathrm{LMTD}}, \tag{3.5}$$

where the $\dot{Q}_{\mathrm{hx}}$ is the heat exchanger duty, U is the overall heat transfer coefficient, and ΔT_{LMTD} is the log-mean-temperature difference, which corresponds to

$$\Delta T_{\mathrm{LMTD}} = \frac{(T_{\mathrm{BR1}} - T_{\mathrm{A2}}) - (T_{\mathrm{BR2}} - T_{\mathrm{A1}})}{\ln\left[\frac{(T_{\mathrm{BR1}} - T_{\mathrm{A2}})}{(T_{\mathrm{BR2}} - T_{\mathrm{A1}})}\right]} \tag{3.6}$$

for the heater in Figure 3.5a. The overall heat transfer coefficient U is a measurement that quantifies the heat exchanger's overall ability to transfer heat. A rough approximated value of U for different scenarios can be found in [3]. For practitioners who are concerned with capital cost optimization of the plant, it might be cost-effective to minimise the area of the heat exchangers as they are a major factor in the overall cost of the plant.

3.2.1.4 Throttle Valve Analysis

The valve unit operation (as shown in Figure 3.6) is a flow-restriction component that decreases the pressure of the fluid. This is different from the turbine where there is no output work produced due to the pressure drop but can result in a large drop in temperature.

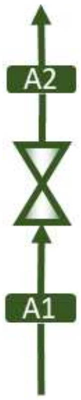


Figure 3.6: Valve for organic Rankine cycle.

Since the valves are usually a small component, there not enough time or surface area for the heat transfer to occur effectively, thus the process is usually assumed to be adiabatic. In addition, any changes in kinetic and potential energy that may occur are usually very small and thus are neglected. Consequently, for a steady-flow single stream valve, the energy equation is reduced to

$$h_{\mathrm{A1}} = h_{\mathrm{A2}}, \tag{3.7}$$

where the inlet enthalpy is equal to the outlet enthalpy.

3.2.2 The Algebraic Equation-Oriented Model

The construction of an algebraic EO model begins by deriving the mass and energy balance equations of the system. Some of the nonlinear terms that are present in the mass and energy balance equations can be approximated using regression analysis, namely, the input/output power of the pump and the turbine, to allow for the analytical derivatives to be calculated. Subsequently, a set of operational constraints of the ORC system needs to be derived to ensure a practical operation, such as the turbine inlet temperature must not exceed the lower and upper limits, and the turbine exhaust gas must be in the vapour region. Lastly, all the decision variables will need to be bounded to reduce the size of the search region and improve computational time. The following subsections will discuss the algebraic EO model in more detail, and highlight the compromise between the accuracy of the model and the computational complexity of the optimization problem.

3.2.2.1 Temperature Profile of Heat Exchangers

At first glance, the model data presented in [1] seems to be accurate and thermodynamically correct. However, after further investigation, it was found that the data provided in [1] was violating the second law of thermodynamics. Figure 3.7 shows the state properties that were given about the condenser in Table 7 of [1], where the working fluid was R227ea. This does not give enough information about the temperature profile of the condenser and its pinch-point. Therefore, let's break the condenser unit operation into two components, i.e., the condenser (cond) and desuperheater (desh) as shown in Figure 3.8, and carry out some thermodynamic calculations to work out the temperature at every stage of the cooling process.

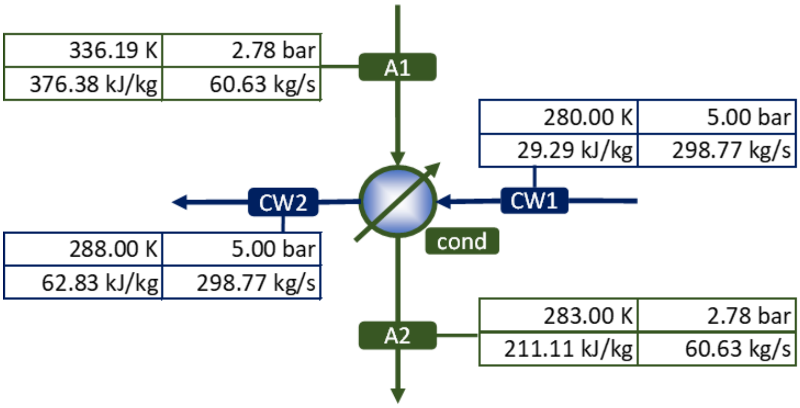


Figure 3.7: The condenser of the ORC system presented in [1] with the state properties obtained directly from Table 7 in [1], where the hot medium is R227ea and the cold medium is water.

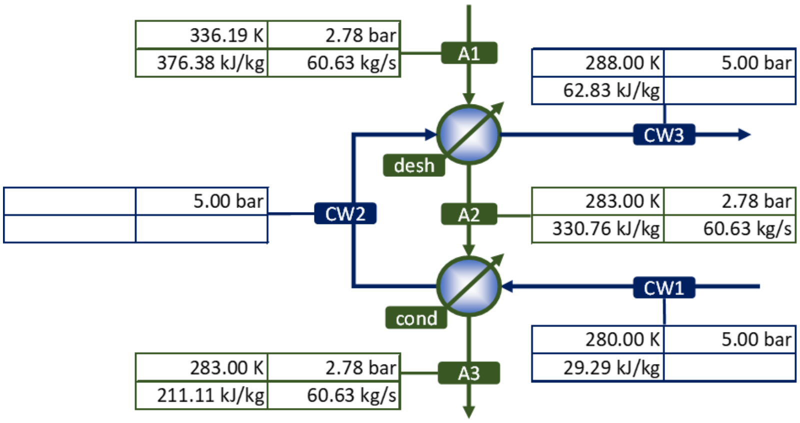


Figure 3.8: The condenser unit operation of [1] separated into two parts consisting of a condenser and a desuperheater. The state properties were obtained from Figure 3.7, where the hot medium is R227ea and the cold medium is water. Note that the temperature at A2 is the saturation vapour temperature at 2.78 bar, which is the same as the saturation liquid temperature at A3.

The desuperheater cools the working fluid to a saturation vapour, and the condenser cools the working fluid to a saturation liquid. If we calculate the duties of the condenser and the desuperheater, we get

$$\begin{aligned}\dot{Q}_{\mathrm{cond}} &= \dot{m}_{\mathrm{WF}}(h_{\mathrm{A2}} - h_{\mathrm{A3}}) = 7.2546 \times 10^{3}\,\mathrm{kW},\\ \dot{Q}_{\mathrm{desh}} &= \dot{m}_{\mathrm{WF}}(h_{\mathrm{A1}} - h_{\mathrm{A2}}) = 2.7658 \times 10^{3}\,\mathrm{kW}.\end{aligned} \tag{3.8}$$

Assuming that the condenser is perfectly insulted, we get

$$\begin{aligned}\dot{Q}_{\mathrm{cond}} &= \dot{m}_{\mathrm{CW}} c_{\mathrm{CW}}(T_{\mathrm{CW2}} - T_{\mathrm{CW1}}) = \dot{m}_{\mathrm{CW}}(4.1993)(T_{\mathrm{CW2}} - 280),\\ \dot{Q}_{\mathrm{desh}} &= \dot{m}_{\mathrm{CW}} c_{\mathrm{CW}}(T_{\mathrm{CW3}} - T_{\mathrm{CW2}}) = \dot{m}_{\mathrm{CW}}(4.1993)(288 - T_{\mathrm{CW2}}),\end{aligned} \tag{3.9}$$

where c_{CW} is the constant specific heat of the cooling water that was calculated using the inlet pressure and temperature value. Consequently, there are two unknowns, namely, $\dot{m}_{\mathrm{CW}}$ and T_{CW2}, and two equations; therefore, it possible to solve for the unknown variables as follows:

$$\begin{bmatrix}\dot{Q}_{\mathrm{cond}}\\ \dot{Q}_{\mathrm{desh}}\end{bmatrix} = \begin{bmatrix} c_{\mathrm{CW}} & -c_{\mathrm{CW}}T_{\mathrm{CW1}}\\ -c_{\mathrm{CW}} & c_{\mathrm{CW}}T_{\mathrm{CW3}}\end{bmatrix}\begin{bmatrix}\dot{m}_{\mathrm{CW}}T_{\mathrm{CW2}}\\ \dot{m}_{\mathrm{CW}}\end{bmatrix} \tag{3.10}$$

This results in $\dot{m}_{\mathrm{CW}} = 298.3\,\mathrm{kg/s}$ and $T_{\mathrm{CW2}} = 285.8\,\mathrm{K}$. As shown in Figure 3.9, the temperature at state CW2 is higher than A2, which violates the second law of thermodynamics. The temperature of the cooling water should never be higher than the working fluid temperature at any point in the heat exchanger because this means that heat transfer is reversed. This is more evident in Figure 3.10, which shows the heat exchange process between the cooling water and the working fluid. According to the second law of thermodynamics, heat transfer can only occur from the hot medium to the cold medium; therefore, this implies that the data provided in [1] is thermodynamically incorrect.

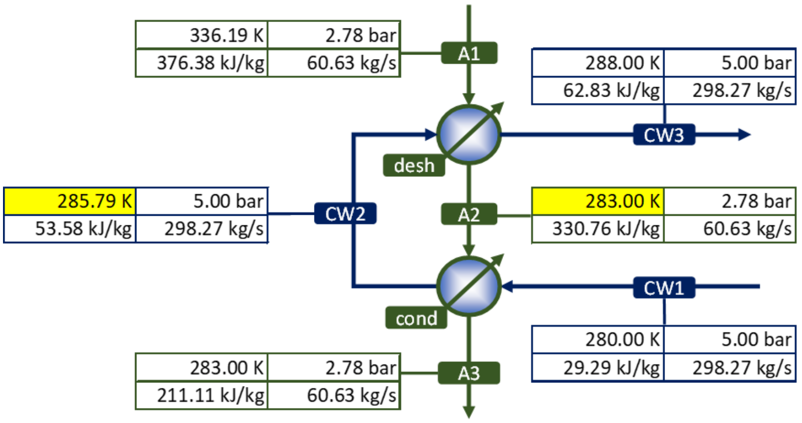


Figure 3.9: The calculated state properties of the condenser and desuperheater of Figure 3.8.

This problem can be solved if the outlet temperature of the cooling water (CW3)

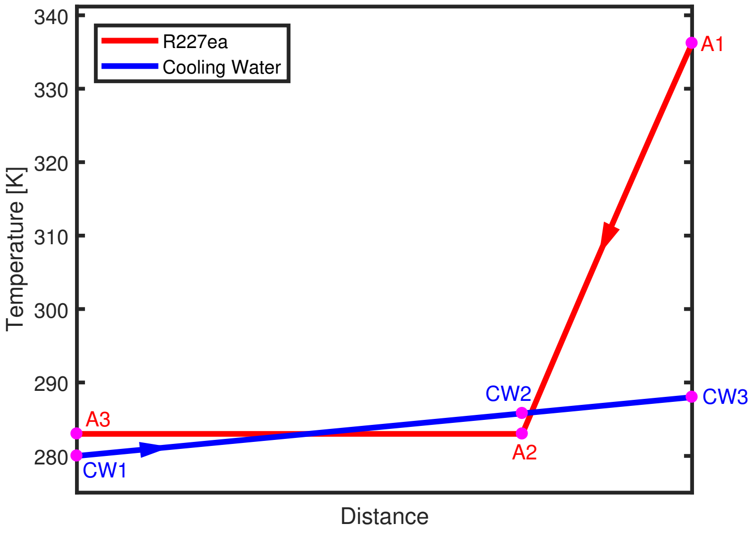


Figure 3.10: Diagram showing the heat exchange process between the cooling water and the working fluid R227ea. The states refer to Figure 3.9. The x-axis represents the path of the fluid flow in the heat exchanger.

is less than the outlet temperature of the working fluid (A3). To illustrate this solution let's make the outlet temperature of the cooling water 1 K less than the outlet working fluid and put no restriction on the mass flow rate of the cooling water, and carrying out the calculation again. As shown Figure 3.11 and Figure 3.12, the temperature at A2 is now higher than the temperature at CW2. However, this resulted in a significant increase in the cooling water mass flow rate to compensate for the decrease in the outlet temperature of the cooling water.

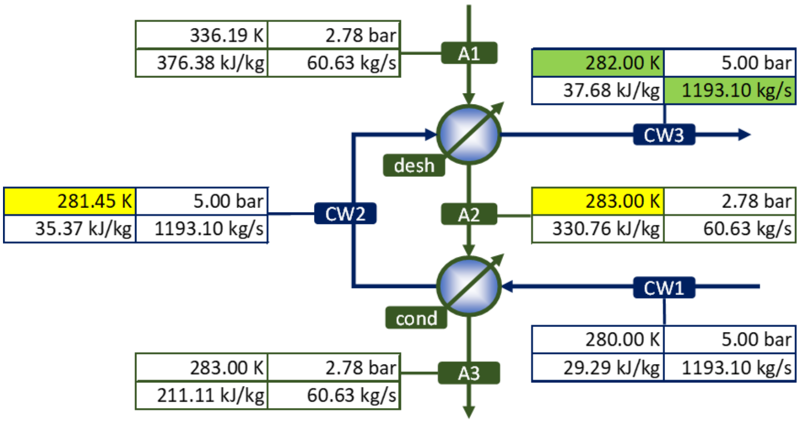


Figure 3.11: The calculated state properties of the condenser and desuperheater of Figure 3.8 when the outlet temperature of the cooling water was decreased from 288 K to 282 K.

The same problem can also occur at the heater side of the ORC system, i.e., the evaporator, as shown in Figure 3.13. In order to analyse the temperature profile, the evaporator need to be separated into two parts, namely, the preheater and

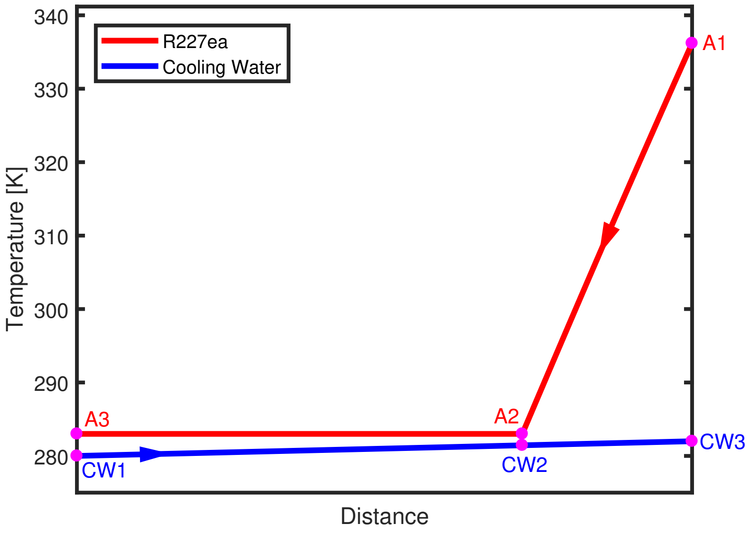


Figure 3.12: Diagram showing the heat exchange process between the cooling water and the working fluid R227ea when the temperature at CW3 was decreased from 288 K to 282 K. The states refer to Figure 3.11. The x-axis represents the path of the fluid flow in the heat exchanger.

evaporator/superheater, as shown in Figure 3.14. Assuming that the heat exchanger is perfectly insulated, the heat transfer between the brine and the working fluid can be written as

$$\begin{aligned} \dot{Q}_{\text{evap/suph}} &= \dot{m}_{\text{BR}} c_{\text{BR}} (T_{\text{BR1}} - T_{\text{BR2}}) = \dot{m}_{\text{WF}} (h_{\text{A3}} - h_{\text{A2}}), \\ \dot{Q}_{\text{preh}} &= \dot{m}_{\text{BR}} c_{\text{BR}} (T_{\text{BR2}} - T_{\text{BR3}}) = \dot{m}_{\text{WF}} (h_{\text{A2}} - h_{\text{A1}}), \end{aligned} \tag{3.11}$$

where $c_{\text{BR}} = 4.2102\,\text{kJ/kgK}$ (calculated using the inlet pressure and temperature value). By substituting the known terms into (3.11), we get

$$\begin{aligned} (75)(4.2102)(369 - T_{\text{BR2}}) &= \dot{m}_{\text{WF}}(719.83 - 402.55), \\ (75)(4.2102)(T_{\text{BR2}} - 334) &= \dot{m}_{\text{WF}}(402.55 - 224.97). \end{aligned} \tag{3.12}$$

With two equations and two unknown variables, namely, $\dot{m}_{\text{WF}}$ and T_{BR2}, the equations can be solved simultaneously.

$$\begin{bmatrix} \dot{m}_{\text{BR}} c_{\text{BR}} T_{\text{BR1}} \\ -\dot{m}_{\text{BR}} c_{\text{BR}} T_{\text{BR3}} \end{bmatrix} = \begin{bmatrix} h_{\text{A3}} - h_{\text{A2}} & \dot{m}_{\text{BR}} c_{\text{BR}} \\ h_{\text{A2}} - h_{\text{A1}} & -\dot{m}_{\text{BR}} c_{\text{BR}} \end{bmatrix} \begin{bmatrix} \dot{m}_{\text{WF}} \\ T_{\text{BR2}} \end{bmatrix} \tag{3.13}$$

This results in $\dot{m}_{\text{WF}} = 22.3\,\text{kg/s}$ and $T_{\text{BR2}} = 346.6\,\text{K}$. As shown in Figure 3.15 and Figure 3.16, the temperature at BR2 is less than the saturated liquid temperature at A2, which again violates the second law of thermodynamics. One way to solve this problem is to decrease the pressure of the working fluid and carry out the calculation again. For this example, let decrease the pressure of the working fluid from 10 bar to 6.359 bar so that the saturated temperature at A2 is 1 K less than the outlet

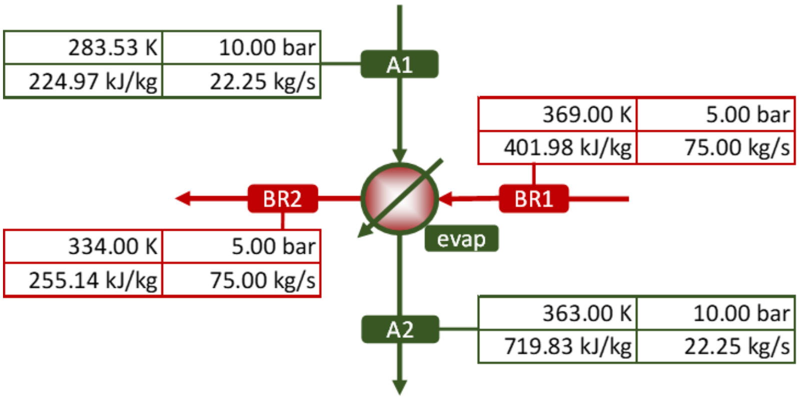


Figure 3.13: The evaporator of the ORC system presented in [1] with the state properties obtained directly from Table 10 in [1], where the hot medium is geothermal brine (water) and the cold medium is R600.

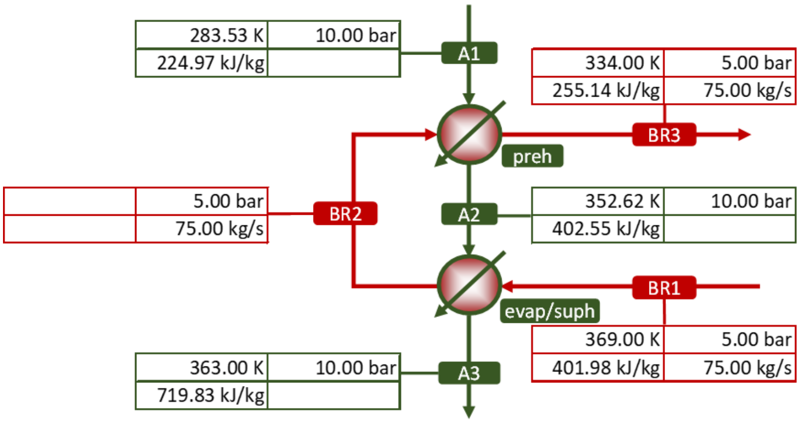


Figure 3.14: The evaporator unit operation of [1] separated into two parts consisting of a preheater and evaporator/superheater. The state properties were obtained from Figure 3.13., where the hot medium is geothermal brine (water) and the cold medium is R600. Note that the temperature at state A2 is the saturation liquid temperature at 10 bar.

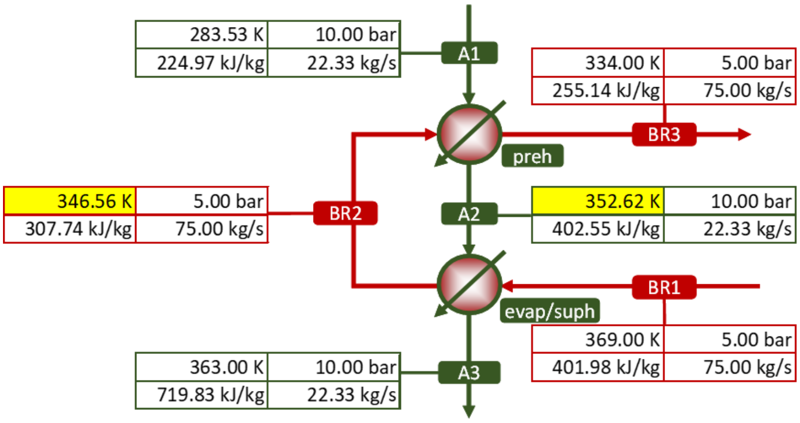


Figure 3.15: The calculated state properties of the preheater and evaporator/superheater of Figure 3.14.

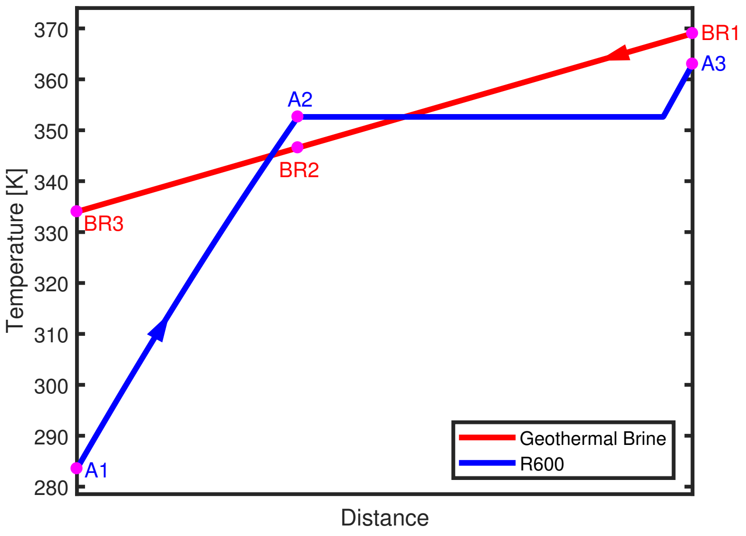


Figure 3.16: Diagram showing the heat exchange process between the geothermal brine and the working fluid R600. The states refer to Figure 3.15. The x-axis represents the path of the fluid flow in the heat exchanger.

temperature of the brine. As shown in Figure 3.17 and Figure 3.18, this results in the temperature at BR2 being higher than A2, but with a decrease in the working fluid mass flow rate.

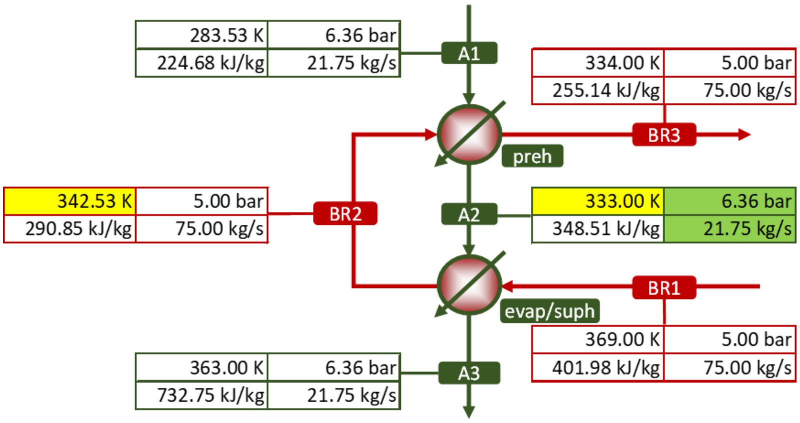


Figure 3.17: The calculated state properties of the preheater and evaporator/superheater of Figure 3.14 when the pressure of the working fluid (R600) was decreased from 10 bar to 6.359 bar.

Therefore, to ensure that these anomalies do not occur in the proposed ORC model, some temperature constraints need to be added to the optimization problem as shown in Figure 3.19. Consequently, the temperature of the outlet cooling water will be decreased below or equal to the temperature at the outlet of the condenser, and allow the mass rate of the cooling water to increase respectively. In addition, if the original system does not have a preheater unit operation, the heater will need to be separated into two separate components, namely, into a preheater and an

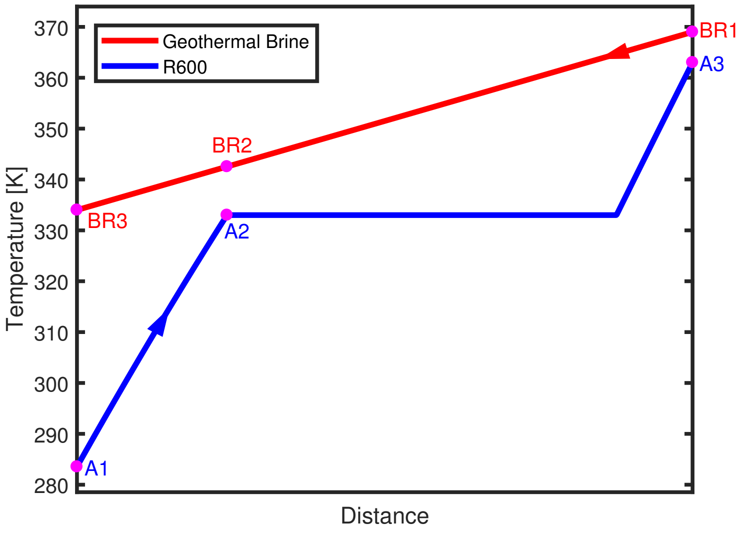


Figure 3.18: Diagram showing the heat exchange process between the geothermal brine and the working fluid R600 when the pressure of the working fluid was decreased from 10 bar to 6.359 bar. The states refer to Figure 3.17. The x-axis represents the path of the fluid flow in the heat exchanger.

evaporator/superheater, to access the temperature at saturated liquid. This will allow the temperature constraint to be added to the optimization model as shown in Figure 3.19. Since the temperatures at the inlet of the cooling water and pump

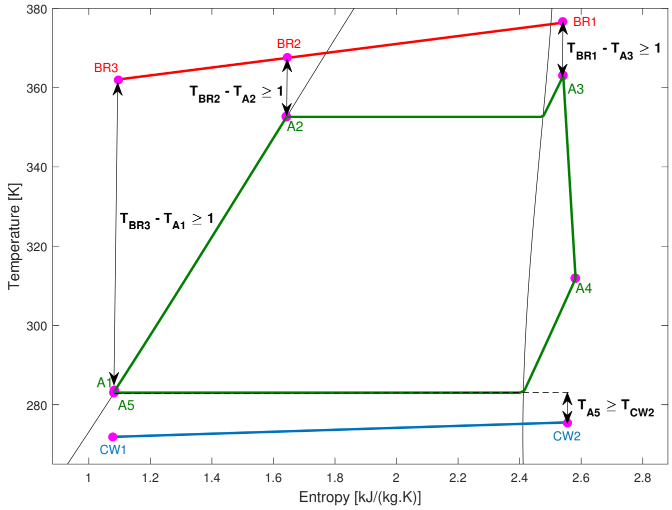


Figure 3.19: A diagram showing the temperature constraints to ensure that the heat transfer only flow from the hot medium to the cold medium. The indices correlate to the Figure 3.2. Note that this diagram is for illustration purposes only, the entropy values for the brine (red) and the cooling water (blue) would generally not be the same as the working fluid.

are fixed, there is no need for a temperature constraint as these temperatures will not change during the optimization procedure. These constraints will be explicitly written out for the Basaran ORC system in Section 3.2.2.5.

3.2.2.2 Optimization Parameters

In order to formulate the optimization problem for the ORC system, there are some optimization parameters that need to be stated. Since the pressure and temperature at the turbine inlet are given as 10 bar and 363 K in the reference paper [1], these values were set as the upper bounds for these two variables. To avoid damaging and degrading the performance of the turbine, the lower bound for the inlet turbine temperature was set to the saturated vapour temperature at 10 bar.

To optimize the model with respect to different working fluids as per the reference, namely, R134a, R143a, R152a, R227ea, R236ea, R290, R600, and R600a, and to show the flexibility of the proposed optimization approach, the lower bound of inlet turbine pressure was set to 1 bar above the saturated liquid pressure at the condenser outlet. Finally, the quality of the working fluid at the condenser outlet was set to saturated liquid to comply with the general operation of an ORC system. Note that the optimization problem described above proposed a system where the pump outlet pressure can vary; however, this modelling approach does not restrict to only this degree of freedom. As shown in [120], it is possible to formulate an optimization problem where the pump inlet pressure is a decision variable, thus highlighting the flexibility of this approach.

Two objective functions will be investigated for this optimization problem, namely, maximizing the net output power and minimizing the specific rotating machinery cost. The net output power is defined as

$$\min \qquad \mathcal{J} = -\dot{W}_{\text{net}} = -\left(\sum_{i=1}^{n} \dot{W}_{\text{turb},i} - \sum_{j=1}^{m} \dot{W}_{\text{pump},j} \right), \tag{3.14}$$

where $\dot{W}_{\text{turb}}$ is the turbine work, n is the number of turbines, $\dot{W}_{\text{pump}}$ is the pump work, and m is the number of pump. While the specific rotating machinery cost is defined as

$$\min \qquad \mathcal{J} = \frac{\sum_{i=1}^{n} C_{\text{turb},i} + \sum_{j=1}^{m} C_{\text{pump},j}}{\dot{W}_{\text{net}}}, \tag{3.15}$$

where C denotes the cost of the rotating machinery. Note that the objective function follows the general standard of optimization format in Section 2.4, thus the objective function is negated for a maximization problem. The cost C can be calculated using the following cost correlation:

$$C = C_0 \left(\frac{W}{W_0} \right)^{a}, \tag{3.16}$$

where W is the work of the unit operation, while C_0, W_0 and a are the constant parameters that were derived from [121, 122, 31] for the turbine and pump, as shown in Table 3.3.

Table 3.3: Constant parameters for calculating the cost of the turbine and pump.

	C_0 [k\$]	W_0 [kW]	a
Turbine	950	3678	0.70
Pump	14	200	0.67

The algebraic EO models were constructed in MATLAB R2017a using OPTI Toolbox v2.27 (utilizing the built-in SymBuilder framework) to efficiently use a range of different solvers without the need to reformulate the problem for the different solver interfaces. The optimization problem will be optimized using various NLP solvers, namely, IPOPT, `fmincon`, FILTERSD, BARON, SCIP, and `patternsearch`, which were selected based on their availability for academic use and their compatibility with OPTI and MATLAB.

3.2.2.3 Mass and Energy Balance

The mass and energy balance equations can be derived based on the first law of thermodynamics. Referring to Figure 3.2 and the associated labels on the diagram, the energy balance equations of the system can be expressed as follows:

$$\begin{aligned} \dot{m}_{\mathrm{WF}} h_{\mathrm{A1}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR2}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A2}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR3}} &= 0, \\ \dot{m}_{\mathrm{WF}} h_{\mathrm{A2}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR1}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A3}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR2}} &= 0, \\ \dot{m}_{\mathrm{WF}} h_{\mathrm{A3}} - \dot{W}_{\mathrm{turb}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A4}} &= 0, \\ \dot{m}_{\mathrm{WF}} h_{\mathrm{A4}} + \dot{m}_{\mathrm{CW}} h_{\mathrm{CW1}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A5}} - \dot{m}_{\mathrm{CW}} h_{\mathrm{CW2}} &= 0, \\ \dot{m}_{\mathrm{WF}} h_{\mathrm{A5}} + \dot{W}_{\mathrm{pump}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A1}} &= 0, \end{aligned} \tag{3.17}$$

where $\dot{m}_{\mathrm{WF}}$ is the mass flow rate, h is the enthalpy value, $\dot{W}_{\mathrm{pump}}$ is the pump input power, $\dot{W}_{\mathrm{turb}}$ is the turbine output power, BR is the brine and CW is the cooling water. From the fixed parameters mentioned in Table 3.2, some of the terms in (3.17), namely, $\dot{m}_{\mathrm{BR}} h_{\mathrm{BR1}}$, $\dot{m}_{\mathrm{BR}} h_{\mathrm{BR3}}$, h_{CW1}, h_{CW2}, and h_{A5}, can already be calculated and substituted into the model. Note that the mass balance equations are not needed for this cycle because the working fluid mass flow is constant throughout the cycle and there are no losses or gains to the mass flow.

3.2.2.4 Unit Operation Approximations

In order to achieve an algebraic model structure, rigorous unit operation functions cannot be used in the model due to the isentropic terms in (3.1) and (3.2). Therefore, the two work expressions in (3.17), namely, $\dot{W}_{\text{turb}}$ and $\dot{W}_{\text{pump}}$, need to be approximated as a function of enthalpy and/or pressure. This can be carried out using a regression toolbox, such as `optifit` [105], MATLAB Curve Fitting Toolbox or `pwfit` (see Section 6.2.1).

Based on the assumptions made in Section 3.2.2.2 that the state of the working fluid is a saturated liquid at the pump inlet (see Table 3.2) and that there are no pressure drops across the heat exchangers, i.e., $P_{\text{A1}} = P_{\text{A3}} \in [P_{\text{A5}}{+}1,\, 10]$ bar (where 10 bar is the turbine inlet pressure value from [1]), a set of isentropic pump work values can be calculated at various input pressures. The correlation between isentropic pump work and the outlet pressure can then be approximated via regression analysis, as shown in Figure 3.20. The criteria to select the order of the regression polynomial closely resemble that of the Akaike and Bayesian information criteria as presented in [123], where there is a defined metric that penalises the order of the polynomial (parameter) with respect to the SSE value (the goodness of the fit). This concept was adopted in this thesis but in a more general sense by increasing the polynomial order only if the regression error plots and the discrepancies between the EO model and the SM model (where the optimized results are substituted into the SM model) are too large. This approach seems to be sufficient for this research and allows the user the flexibility to make their own compromise between the complexity and accuracy of the polynomial regression. As a result, the pump input power now becomes

$$\dot{W}_{\text{pump}} = \frac{\Delta\hat{h}_{\text{is,pump}}\dot{m}_{\text{WF}}}{\eta_{\text{pump}}}, \tag{3.18}$$

where

$$\Delta\hat{h}_{\text{is,pump}} = -1.02 \times^{-5} P_{\text{A1}}^2 + 0.0691 P_{\text{A1}} - 0.192 \tag{3.19}$$

is the approximated isentropic work. Note that the indices in (3.19) refer to Figure 3.2.

The turbine output power can be approximated using a similar approach but with two independent variables, namely, the inlet pressure and enthalpy. Based on the assumptions made in Section 3.2.2.2 that the inlet temperature can vary between $T_{\text{A4}} \in [T_{g@10\,\text{bar}},\, 363]$ K, the inlet pressure can vary between $P_{\text{A3}} = P_{\text{A1}} \in [P_{\text{A5}}{+}1,\, 10]$ bar, and the outlet pressure is equal to P_{A5}, a set of isentropic turbine work values can be calculated, as shown by the black dots (top plot) in Figure 3.21. Since there are no explicit turbine temperature terms in (3.17), the respective inlet enthalpy values were calculated for the inlet temperatures. This results in the following turbine work expression:

$$\dot{W}_{\text{turb}} = \Delta\hat{h}_{\text{is,turb}}\eta_{\text{turb}}\dot{m}_{\text{WF}}, \tag{3.20}$$

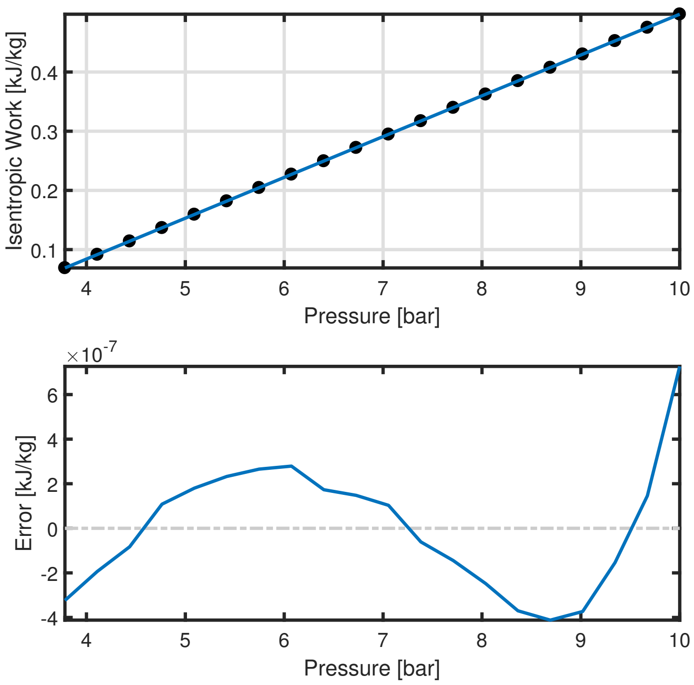


Figure 3.20: The regression fit and error plot of the pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

where

$$\begin{aligned}\Delta\hat{h}_{\text{is,turb}} =& 0.0206P_{\text{A1}}^3 - 0.41P_{\text{A1}}^2 - 0.000552P_{\text{A1}}^2 h_{\text{A4}} + \\ & 0.0186P_{\text{A1}}h_{\text{A4}} + 0.556P_{\text{A1}} - 0.0424h_{\text{A4}} - 0.174\end{aligned} \tag{3.21}$$

is the approximated isentropic work of the turbine as a function of the inlet pressure and enthalpy.

Note that the proposed modelling approach does not restrict the operating range of the pump and turbine to what is stated above. They can be changed to suit other optimization problems if the ORC system is required to operate under a different operating condition.

3.2.2.5 Operational Constraints

For the ORC system to operate within practical limits, there are some operational constraints that need to be considered. These are superheating/subcooling requirements, the pressure drop across heat exchangers, heat loss, etc. In compliance with [1] specifications and assumptions, the turbine inlet conditions need to be constrained. Based on the optimization specifications in Section 3.2.2.2, the turbine inlet temperature needs to be constrained between $T_{f@10\,\text{bar}}$ and 363 K, see Fig-

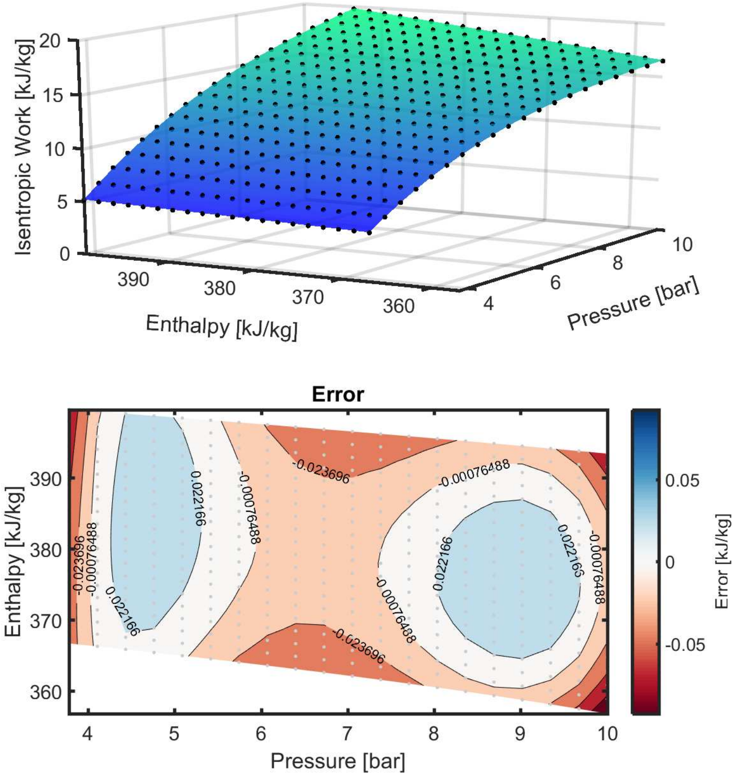


Figure 3.21: The regression fit and error plot of the turbine isentropic work as a function of the inlet enthalpy and pressure. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

ure 3.22. Since there no explicit turbine temperature terms in the algebraic EO model, the corresponding enthalpy values at both temperature limits were calculated using REFPROP and regressed. Therefore, the lower- and upper-temperature limit can be constrained as

$$\begin{aligned} h_{\mathrm{A3}} &\leq \hat{h}_{@369\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}}, \\ h_{\mathrm{A3}} &\geq \hat{h}_{@T_{g@10\,\mathrm{bar}},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}}, \end{aligned} \tag{3.22}$$

where

$$\begin{aligned} \hat{h}_{@369\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}} &= -0.000526P_{\mathrm{A1}}^3 - 0.00574P_{\mathrm{A1}}^2 - 0.839P_{\mathrm{A1}} + 403, \\ \hat{h}_{@T_{g@10\,\mathrm{bar}},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}} &= -0.00432P_{\mathrm{A1}}^3 + 0.0289P_{\mathrm{A1}}^2 - 1.33P_{\mathrm{A1}} + 372. \end{aligned} \tag{3.23}$$

The regression plots of the lower- and upper-temperature limits are shown in Figure 3.23.

In addition, to ensure that state A2 operates at the saturated liquid point, the

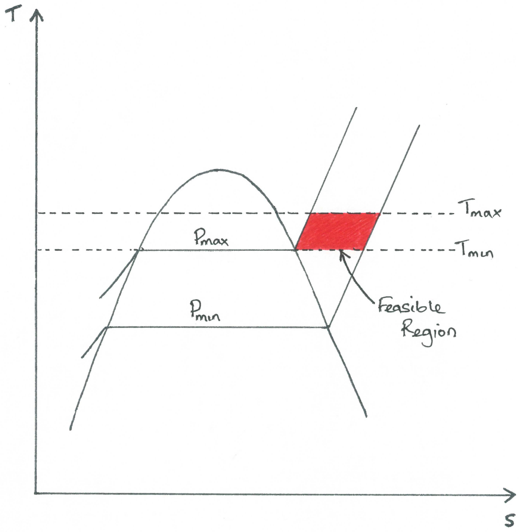
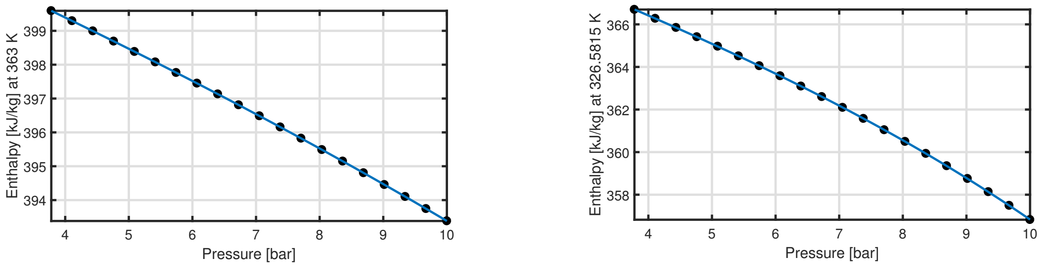


Figure 3.22: The feasible region of the turbine inlet condition.



(a) $\hat{h}_{@369\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}}$

(b) $\hat{h}_{@T_{g@10\,\mathrm{bar}},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}}$

Figure 3.23: The regression curve regressions the turbine inlet enthalpy at the lower- and upper-temperature limit as a function of the inlet pressure. The black dots are the REFPROP thermodynamic calculations.

following constraint was added

$$h_{\mathrm{A2}} = \hat{h}_{f@P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]}, \tag{3.24}$$

where

$$\hat{h}_{f@P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]} = 0.0297P_{\mathrm{A1}}^3 - 0.94P_{\mathrm{A1}}^2 + 15.2P_{\mathrm{A1}} + 176. \tag{3.25}$$

The approximation curve of (3.25) is shown in Figure 3.24.

Lastly, in order to ensure that the second law of thermodynamics is not violated (as discussed in Section 3.2.2.1), the following temperature constraints were

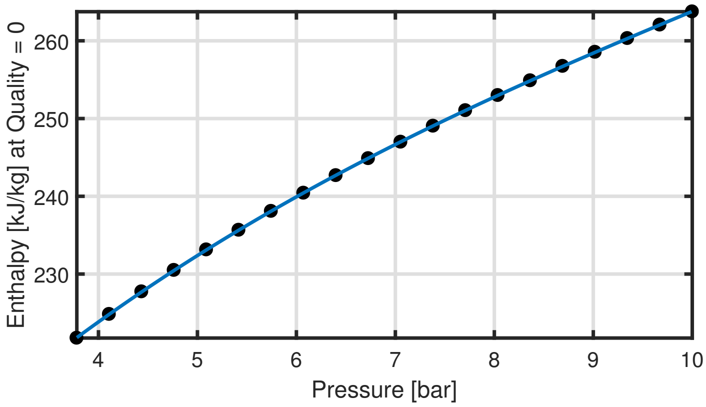


Figure 3.24: The saturation liquid curve regressions for the BORC algebraic EO model. The black dots are the REFPROP thermodynamic calculations.

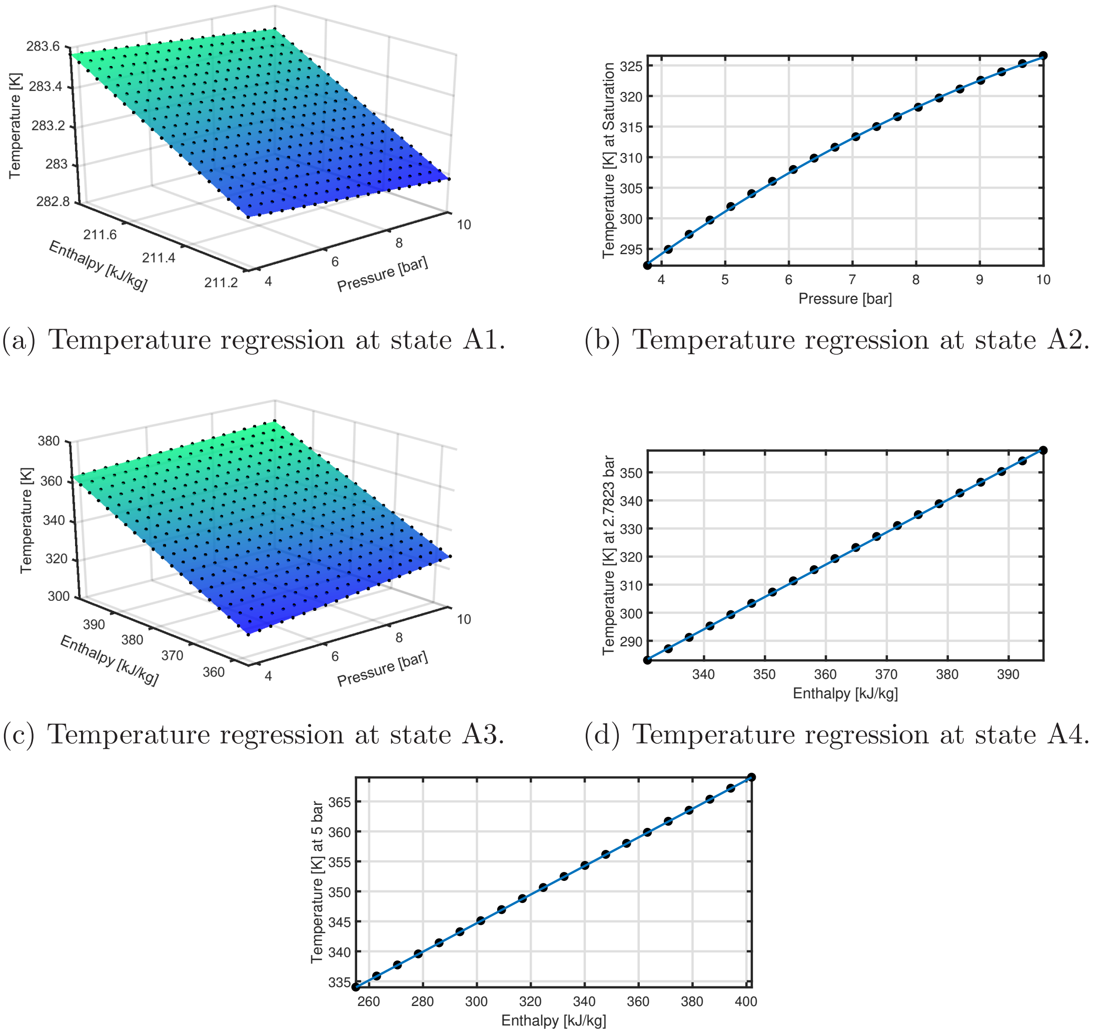
introduced in the optimization problem:

$$\begin{aligned} T_{\text{BR1}} - \hat{T}_{\text{A3}} &\geq 1, \\ \hat{T}_{\text{BR2}} - \hat{T}_{\text{A2}} &\geq 1, \\ T_{\text{BR3}} - \hat{T}_{\text{A1}} &\geq 1, \\ \hat{T}_{\text{A4}} - T_{\text{CW2}} &\geq 1, \end{aligned} \tag{3.26}$$

where

$$\begin{aligned} \hat{T}_{\text{A1}} &= -0.0146P_{\text{A1}} + 0.88h_{\text{A1}} + 97.3, \\ \hat{T}_{\text{A2}} &= -0.312P_{\text{A1}}^2 + 9.72P_{\text{A1}} + 260, \\ \hat{T}_{\text{A3}} &= -3.77 \times 10^{-5}h_{\text{A3}}^2 - 0.0179P_{\text{A1}}h_{\text{A3}} + 1.21h_{\text{A3}} + 8.12P_{\text{A1}} - 119, \\ \hat{T}_{\text{A4}} &= -0.000753h_{\text{A4}}^2 + 1.7h_{\text{A4}} - 197, \\ \hat{T}_{\text{BR2}} &= 0.238h_{\text{BR2}} + 273. \end{aligned} \tag{3.27}$$

While the last equation in (3.26) might seem excessive because the temperature of the cooling water was set lower than the temperature at the outlet of the condenser, this constraint was added to help troubleshoot the model if there is an error or if the solver converge to a infeasible point, and to reinforce that temperature difference at any point between the hot and cold medium is always positive. Figure 3.27 shows the temperature regressions in (3.27). In some cases, in order to approximate all the possible temperature values for some states, the temperatures were calculated using the 2D grid coordinates of the pressure and enthalpy values that were generated using MATLAB's `meshgrid` function. Consequently, in some cases, a few temperatures are replaced with the mean temperature of the corresponding row when they are in the two-phase region due to the range of the pressure and enthalpy bounds. While the constraints and bounds will restrict the ORC system from operating in the two-phase region, the modification to the temperature matrix was made to achieve a more accurate fit.



(a) Temperature regression at state A1.

(b) Temperature regression at state A2.

(c) Temperature regression at state A3.

(d) Temperature regression at state A4.

(e) Temperature regression for the brine.

Figure 3.25: The temperature regressions of the Basaran ORC system as shown in Figure 3.2. The black dots are the REFPROP thermodynamic calculations.

3.2.2.6 Bounds

In order to reduce the search region and decrease the execution time, all the decision variables need to be bounded within a sensible range. For this optimization problem, the bounds were calculated based on the thermodynamic approximations and assumptions that were made about the ORC system. Therefore, the bounds for the BORC system using R227ea as the working fluid were as follows for the enthalpy values:

$$\begin{aligned}
211.20 &\leq h_{\mathrm{A1}} \leq 211.77, \\
221.77 &\leq h_{\mathrm{A2}} \leq 263.74, \\
356.82 &\leq h_{\mathrm{A3}} \leq 399.59, \\
330.77 &\leq h_{\mathrm{A4}} \leq 395.72, \\
255.14 &\leq h_{\mathrm{BR}} \leq 401.98,
\end{aligned} \tag{3.28}$$

for the mass flow rates and pressure:

$$\begin{aligned} 10.00 \leq \dot{m}_{\mathrm{WF}} &\leq 100.0, \\ 1100.0 \leq \dot{m}_{\mathrm{CW}} &\leq 1300, \\ 3.78 \leq P_{\mathrm{A1}} &\leq 10.00, \end{aligned} \tag{3.29}$$

and for the rotating machinery works:

$$\begin{aligned} 733.20 \leq \dot{W}_{\mathrm{net}} &\leq 1222.00, \\ 30.17 \leq \dot{W}_{\mathrm{pump}} &\leq 50.28, \\ 763.37 \leq \dot{W}_{\mathrm{turb}} &\leq 1272.28. \end{aligned} \tag{3.30}$$

Note that since the ORC system will be subjected to various working fluids, the mass flow bounds were set to accommodate for all the different fluids, whereas the pressure and enthalpy bounds will change depending on the working fluid due to its thermodynamic properties.

3.2.3 Optimization Parameters and Settings

Once all the mass and energy balance equations, constraints, and bounds have been derived, the optimization problem can be formulated and should result in the general nonlinear programming format that is shown in (2.3). Please refer to Appendix A for the full explicit optimization problem of the Basaran ORC system. With the optimization problem constructed using the SymBuilder framework, the following model statistics were obtained:

```
SymBuilder Object
BUILT in 0.515s with:
-  11 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
-  15 constraint(s)
        -  2 linear
        -  8 quadratic
        -  5 nonlinear
- 22 bound(s)
-  0 integer variable(s)
```

Utilizing the OPTI Toolbox, the optimization problem was optimized using NLP solvers. Unless stated otherwise, all the optimization procedures were performed on a 64bit Windows 3.1GHz Intel Core i5. The default OPTI settings for the all the solvers were used, except for the maximum number of nodes (5×10^6), maximum number of iterations (5×10^6), maximum number of function evaluations (5×10^6) and maximum execution time (3.6×10^3).

3.2.4 Basaran ORC Optimized Results

The algebraic EO model was optimized with six NLP solvers, and the optimized net output power results are shown in Table 3.4. Two interesting points can be made from the optimized results. First, not every solver converged to the same solution and, therefore, this indicates that a local optimum was found. Second, the white-box global solvers and the gradient-free solvers were slower than the derivative-based solvers. This is expected as both the white-box global solver and the gradient-free solver algorithms generally require more function evaluations, and hence will contribute to longer execution times. In addition, the proposed formulation allows the derivative and matrix sparsity information to be provided to the solvers, which would have contributed to a more efficient performance and better computational times for the derivative-based solvers.

Table 3.4: The optimized net output powers and the solve times of the BORC algebraic EO model.

Solver	$\mathcal{J}$ **[kW]**	**Time [s]**
IPOPT	1018.13	0.018
`fmincon`	1018.13	0.045
FILTERSD	1018.13	0.004
BARON	1018.13	0.513
SCIP	1018.13	0.488
`patternsearch`	980.34	525.7

The highest net output power was 1018.13 kW, and the lowest was 980.34 kW. Both white-box solvers (SCIP and BARON) and all of the derivative-based solvers (IPOPT, `fmincon`, and FILTERSD) found the global optimum. Whereas, `patternsearch` converged to a suboptimal solution of 980.34 kW. This confirms that even though some gradient-free solvers are classified as “global” solvers, they cannot guarantee the global optimality, unlike the white-box global solvers. Unless the optimization problem is convex, the black-box solvers cannot ensure that the solution is the global optimum.

Comparing this proposed optimization formulation to a similar study by Huster et al. [23] that utilised the EO approach to optimize the net output power of a similar basic ORC system, the average solve time of BARON presented in this thesis is significantly faster by around 26x than the results shown in Table 1 of [23]. This could be due to the use of simple regression fits, i.e., polynomial functions, compared to the nonlinear thermodynamic equations proposed in [23]. While the journal paper focused on highlighting the performance of their global optimization framework exceeding that of BARON, the results did give a rough indication of the performance of optimizing an ORC system using BARON and the algebraic EO approach outside this research, and also highlight the interest in the literature surrounding global optimality. Therefore, this illustrates the need for this research and the proposed modelling formulation.

Taking the optimization problem further and optimizing the specific rotating machinery cost, the optimized results in Table 3.5 were obtained. Again, `patternsearch` did not find the same solution as the white-box solvers and converged to a local optimum of 399.87 k$/MW. Whereas all the derivative-based solvers and the white-box solvers converged to a global optimum of 398.19 k$/MW.

Table 3.5: The optimized specific rotating machinery costs and the solve times of the BORC algebraic EO model.

Solver	$\mathcal{J}$ [k$/MW]	**Time** [s]
IPOPT	398.19	0.016
`fmincon`	398.19	0.089
FILTERSD	398.19	0.006
BARON	398.19	0.741
SCIP	398.19	0.340
`patternsearch`	399.87	527.0

Similar to the previous objective function, all the derivative-based solvers converged to a solution significantly faster than both the white-box solvers and `patternsearch`. In addition, all the solvers converged to the same operating point as the previous objective function, which indicates the global optimum for the specific rotating machinery cost is at the same operating point as the global optimum for the net output power.

For future optimization problems, `patternsearch` will no longer be investigated as they are not suitable for large-scale and constrained problem [63]. In addition, `patternsearch` cannot deterministically guarantee the global solution and cannot fully take advantage of the algebraic structure of the proposed model formation, namely, the derivative information, given that it is a derivative-free solver. Furthermore, given the poor performance of `patternsearch` presented by this simple optimization problem compared to the other solvers, it would be more sensible to use derivative-based solvers that can achieve a much faster execution time.

3.2.5 Algebraic Equation-Oriented Model Validation

As previously mentioned in Section 3.1, the SM models are not tailored for optimization compared to the proposed EO model, especially for large ORC systems (see Section 5.2.2). However, the SM model can be used to validate the algebraic EO model since it provides a high fidelity model of the ORC system due to the use of rigorous unit operations and thermodynamic packages. This is an important part of the proposed modelling framework, as it will show the reliability and accuracy of the approximated EO model to the original system.

The high fidelity SM model in this research was constructed using JSteam MATLAB Interface v1.72 [124], which was considered as the "gold standard" for the purpose of this study as it is the closest representation of the original system. This is because the unit operation modules that are provided in JSteam are modelled using explicit thermodynamic calculations and REFPROP routines, not via approximations, thus gives a very accurate model of the original plant. While it is possible to use another simulation platform, such as Aspen Plus [13] and VMGSim [36], the built-in MATLAB interface and its compatibility with OPTI Toolbox made JSteam the most efficient choice for this research.

In addition to validating the algebraic model, once the SM model is constructed and solved using the nominal plant data with a nonlinear system solver (such as `fsolve` in MATLAB), it can be used as the initial guess for the optimization problem for the algebraic EO model. Since different initialization values can result in a different optimization time and solution, it is important that a sensible initial guess is used for the optimization problem. Therefore, for this research, the initialization values were taken from the SM model that was solved using the original plant information to minimize the issues associated with a poor initial guess.

Table 3.6 shows the discrepancies between the original BORC system in [1] and the JSteam SM model. The nominal design values were taken directly from [1] and were substituted into the SM model, namely, P_{A1}, T_{A2}, and T_{A3}, that corresponded to 10 bar, 326.58 K and 363 K, respectively.

Due to the anomalies found in [1] regarding the heat exchangers as discussed in Section 3.2.2.1, there were some modifications to the original ORC system. This included decreasing the outlet temperature of the cooling water from 288 K to 282 K, and adding a preheater unit operation to the ORC system to ensure the laws of thermodynamics are being upheld. This consequently led to some large discrepancies between the journal paper and the JSteam SM model, noticeably the flow rate of the cooling water in Table 3.6. Furthermore, the pressure at the condenser outlet was recalculated to ensure that the working fluid is at the right saturated point due to rounding errors. This means that the pressure values changed slightly from 2.7800 bar to 2.7823 bar. Consequently, this propagated along the cycle and affects the enthalpy value at other stages of the system. As a result, this can also contribute to the discrepancies between the journal paper and the JSteam SM model.

Given the numerous unforeseen errors found in [1], including the false calculations of the saturation temperatures of some of the working fluids (see Fig. 11 and Fig. 12 in [1], where the saturation temperature at 10 bar for R290 and R600a should be 26.9°C and 79.5°C, respectively, not 47°C and 87°C) and the thermodynamic violation in the heat exchangers (see Section 3.2.2.1), the optimized results will not be compared to the journal paper as their results are thermodynamically invalid and thus will not give a fair and reliable comparison. However, this paper offers a great opportunity to illustrate some of the important checks and errors that can go unnoticed when modelling ORC systems, and they need to be addressed to uphold the laws of thermodynamics and conventions.

Table 3.6: The comparison between the original nominal design properties [1] and the JSteam SM model.

Plant **Parameter**	**Nominal** **Design**	**SM Model** **(Base Case)**	**Error** **[%]**
Total Heat Duty Input, [kW]	11013	11013	0.00
Condenser Duty, [kW]	10021	10036	0.15
Turbine Power, [kW]	1030.7	1017.8	1.25
Pump Power, [kW]	38.8	40.22	3.66
Net Power, [kW]	991.91	977.6	1.44
Working Fluid, [kg/s]	60.63	60.64	0.02
Cooling Water, [kg/s]	298.77	1195.5	300.13
Thermal Efficiency, [%]	9.01	8.88	1.44
Specific Machinery Cost, [k$/MW]	397.82	400.26	0.61

In order to ensure that the thermodynamic approximations have not violated any thermodynamic laws or give an inaccurate representation of the ORC system, the optimized results will be substituted to the high fidelity SM model. Ideally, the discrepancies between the two models should be as small as possible, which will indicate that the approximated EO model is an accurate representation of the original ORC system. As shown in Table 3.7, the errors between the two models are less 1%, which is very favourable. For this ORC system, the solvers did converge to the same operating point for both objective functions because there is little room for improvement given the size/complexity of the basic ORC system. However, for more complex ORC systems, such as the ones that will be presented in Chapter 4, the solvers are less likely to converge to the same operating point.

Note that due to the thermodynamic approximations and the imposed constraints on the optimization problem, as well as the tolerances and stopping criteria

Table 3.7: The validated algebraic EO model for both objective functions. The values in the square brackets show the relative error [%] between the SM model and the algebraic EO model.

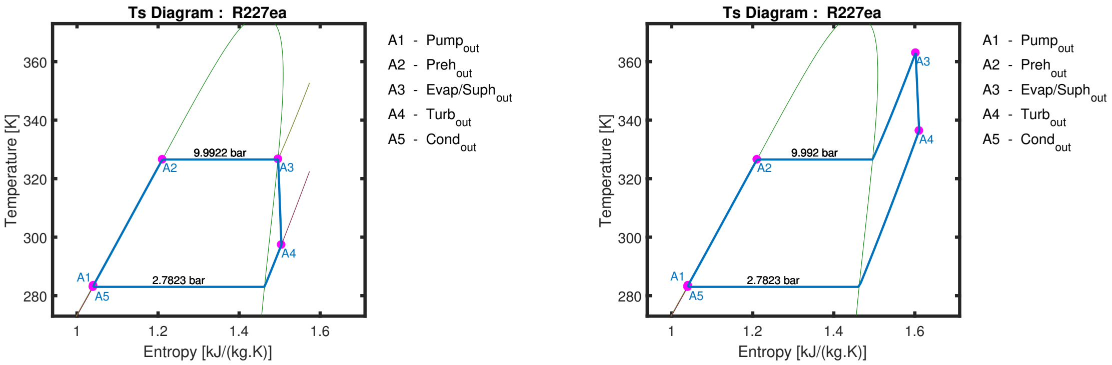
Plant Parameter	**IPOPT/**`fmincon`**/** **BARON/SCIP/** **FILTERSD/**	`patternsearch`
Total Heat Duty Input, [kW]	11013 [0.00%]	11013 [0.00%]
Condenser Duty, [kW]	10001 [0.06%]	10036 [0.03%]
Turbine Power, [kW]	1062.6 [0.55%]	1017.3 [0.32%]
Pump Power, [kW]	50.26 [0.04%]	40.18 [0.02%]
Net Power, [kW]	1012.4 [0.57%]	977.07 [0.33%]
Working Fluid, [kg/s]	75.85 [0.00%]	60.64 [0.00%]
Cooling Water, [kg/s]	1191.3 [0.06%]	1195.5 [0.03%]
Thermal Efficiency, [%]	9.19 [0.54%]	8.87 [0.34%]
Specific Machinery Cost, [k$/MW]	398.96 [0.19%]	400.32 [0.11%]

of the solvers, the optimal solution can render different from the “true” global optimum. Thus, there will generally be some discrepancies between the SM model and the approximated EO model. It is possible to reduce this error by using a higher order fit or a piecewise fit, which will be addressed in Chapter 6. However, with any approximation models, not just within this research, there will be some dependencies between the original system and the model that can deviate from the optimum solution. This is not a new problem in operations research, and practitioners need to take into account the accuracy of the model when any approximations are carried out and decide if the small difference in the optimal solution is acceptable for their purpose.

Given that the ORC process can be analysed on a Ts diagram, an alternative way to validate the algebraic EO model is to superimpose the different stages of the ORC process onto the Ts diagram, as shown in Figure 3.26. The Ts diagram will give a visual analysis of the optimized system and indicate if it is operating as an ORC system and has not violated any thermodynamic constraints.

To an engineer, the general ORC process and configuration are generally known prior to the optimization process, thus they can reasonably speculate where a feasible operating point can be in order to comply with the thermodynamic laws and mechanical constraints from the Ts diagram. This includes ensuring that the state properties at the inlet and outlet of the turbine (A3 and A4) is not in the two-phase region, the temperature of the condenser outlet (A5) is at the specified value and at the saturated liquid or subcooled region, and the output pressure of the pump (A1) is within the specified bounds.

Furthermore, to ensure that the thermodynamic violation does not occur in the heat exchangers, the temperature profile of the preheater, evaporator/superheater,



(a) BARON (b) `patternsearch`

Figure 3.26: The Ts diagram of the optimized algebraic EO model using BARON and `patternsearch`, where the each solver converged to the same operating point for both objective functions.

and condenser can be plotted and analysed, as shown in Figure 3.27 and Figure 3.28.

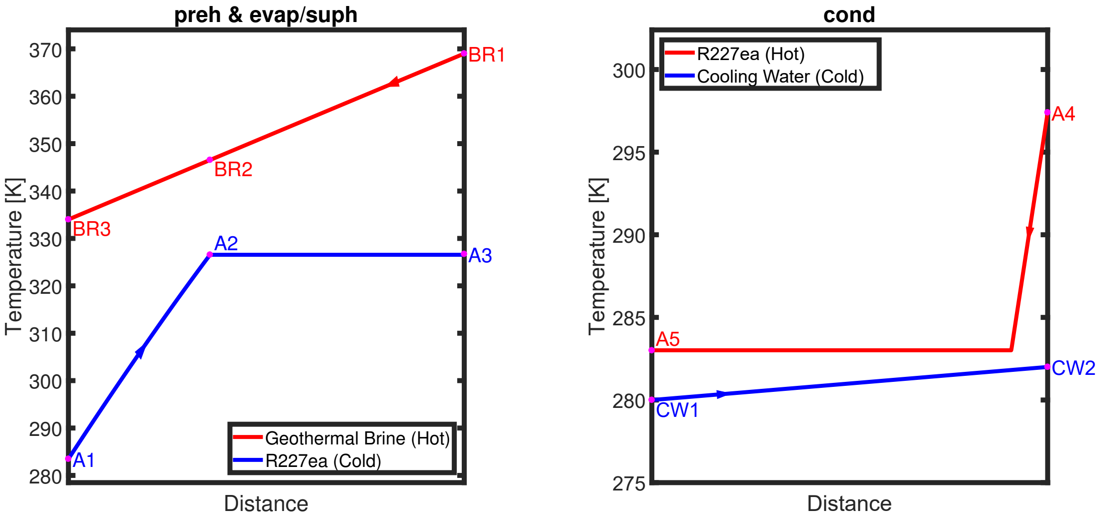


Figure 3.27: Diagrams showing the heat exchange process between the geothermal brine/cooling water and the working fluid, where the net power of the plant was optimized using BARON. The states refer to Figure 3.2. The x-axis represents the path of the fluid flow in the heat exchanger.

With the temperature constraints in Figure 3.19 incorporated into the optimization problem, the anomalies discussed in Section 3.2.2.1 did not occur in both of the optimized scenarios and did not violate the second law of thermodynamics.

3.2.6 Working Fluid Selection for the BORC System

In addition to finding the optimal operating point for the ORC system, the optimization model can be used to investigate various aspects of the plant, such as the working fluid selection. For the BORC system, the eight pure working fluids that were investigated in the original reference paper [1] were selected for this optimiza-

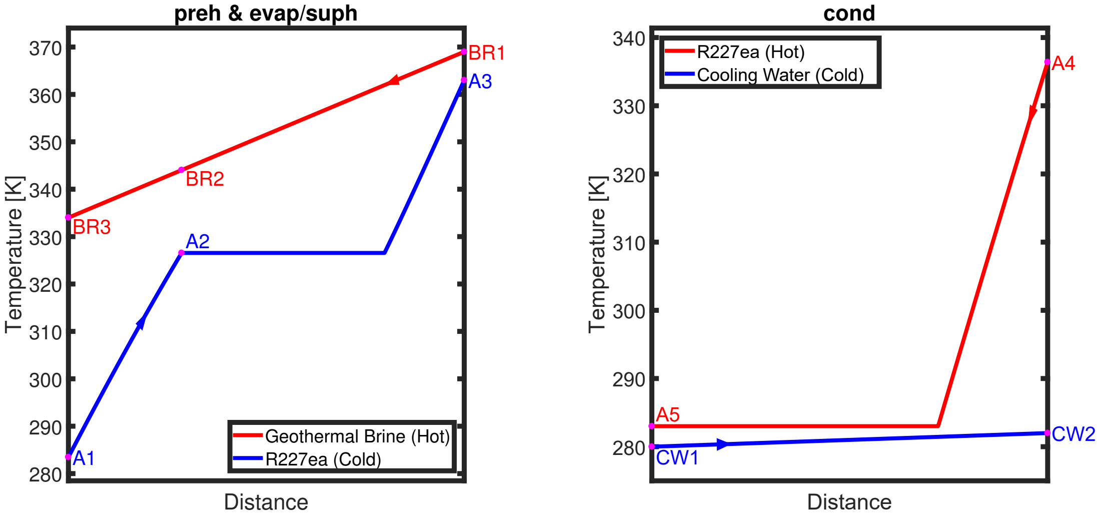


Figure 3.28: Diagrams showing the heat exchange process between the geothermal brine/cooling water and the working fluid, where the net power of the plant was optimized using `patternsearch`. The states refer to Figure 3.2. The x-axis represents the path of the fluid flow in the heat exchanger.

tion problem. It is important to mention that optimizing the ORC system in regards to the working fluid selection is an exhaustive optimization procedure that involves having an outer iterative loop that goes through all the potential working fluids. Trying to carry out this optimization procedure using the traditional SM approach would be very inefficient and can lead to various optimization issues, as opposed to the proposed algebraic EO model.

Optimizing the net output power of the ORC model using BARON with respect to different working fluids yielded the optimized results in Figure 3.29.

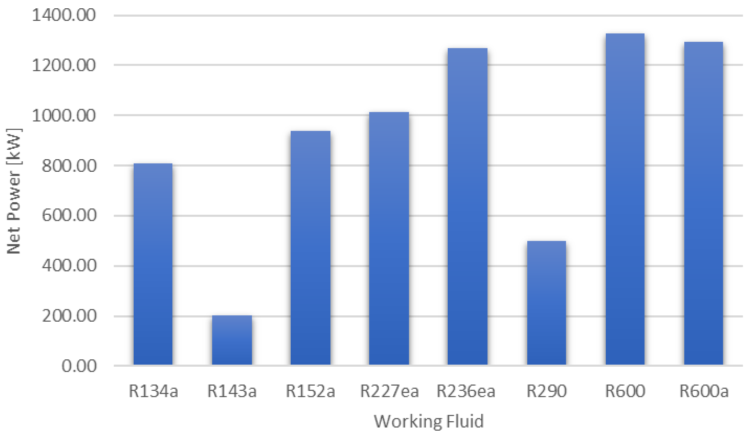


Figure 3.29: The validated optimal net output powers of the BORC system with respect to different working fluids.

Two interesting results can be observed from Table 3.8. First, R600 obtained the highest net output power of 1325.4 kW and R143a obtained the lowest value

Table 3.8: The validated optimal net output powers [kW] that were optimized using BARON.

	BARON	**Solve Time [s]**
R134a	807.38	0.643
R143a	202.03	1.855
R152a	940.07	0.567
R227ea	1012.4	0.513
R236ea	1266.4	0.541
R290	499.64	0.643
R600	1325.4	0.654
R600a	1293.9	0.510

of 202.03 kW, which shows that the output net power of the plant can drastically be influenced by the working fluid selection. Second, the average solve time of BARON is less than 2 s for all of the working fluids. This is expected given the small size of the optimization problem presented in this chapter. This highlights one of the contributions for this optimization framework, which is that it can efficiently optimize the plant with respect to different working fluids within a reasonable time frame and can achieve global optimality for each working fluid using white-box solvers.

Similarly, the same working fluids were optimized with respect to the specific rotating machinery cost objective function using BARON, as shown in Table 3.9. Figure 3.30 compares the optimized results of the different working fluids against each other, which shows that R600 has the lowest specific rotating machinery cost of 359.28 k$/MW and R143a has the highest value of 645.63 k$/MW.

Table 3.9: The validated optimal specific rotating machinery costs [k$/MW] that were optimized using BARON.

	BARON	**Solve Time [s]**
R134a	420.81	0.547
R143a	645.63	0.509
R152a	400.23	0.501
R227ea	398.96	0.741
R236ea	363.62	0.432
R290	489.76	0.508
R600	359.28	0.479
R600a	365.75	0.601

Note that the optimization problem present in this research only focused on two objective functions, but there are other aspects of the ORC system that can be optimized, such as the specific heat exchanger area, the utilization efficiency, mass flow

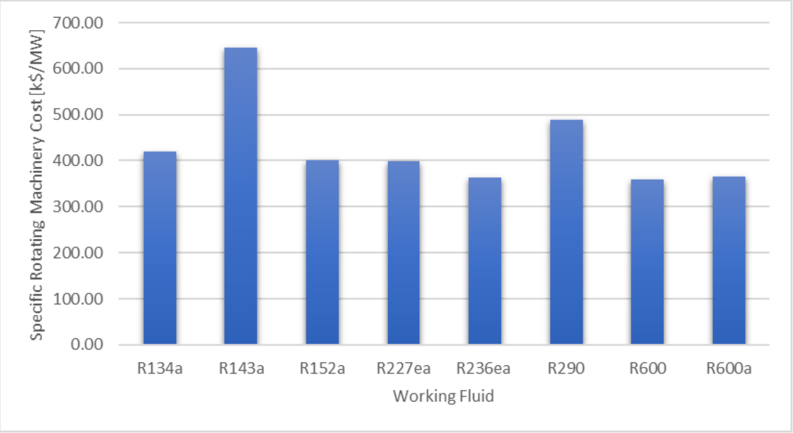


Figure 3.30: The validated optimized specific rotating machinery costs of the BORC system with respect to different working fluids.

rates, and different operating conditions, that might be viewed more significant to the viability of the plant. Therefore, although the optimized results presented in this chapter was limited to two aspects of an ORC system, this should not undermine the contributions of the proposed algebraic EO modelling approach. Compared to most ORC models that are constructed using the SM approach, the proposed algebraic model allows the analytical derivative information to be provided to the solver and, thus, bypassing the issues associated with using the finite difference method as discussed in Section 2.9. In addition, the algebraic structure also allows global white-box solvers that can deterministically guarantee the global solution to be used, which can be a major advantage in a competitive market. Furthermore, the equation-oriented structure of the model allows the system recycle(s) to be efficiently optimized and solved simultaneously without the need of solving an internal flowsheet using a nonlinear equation solver. These attributes make the proposed algebraic EO model efficient and robust for optimization and, hence, addressing the main focus of this research.

3.3 Summary

This chapter has introduced an algebraic equation-oriented modelling approach for an ORC system that is tailored for efficient optimization. This involves deriving a set of algebraic equations that describes the ORC system using thermodynamic and regression analysis. As such, a detailed description of the model formulation and optimization procedure was given, which included deriving the energy balance equations, approximating the unit operation processes and thermodynamic terms, and validating the optimized results. Compared to the traditional SM model, this proposed algebraic model structure can be exploited by the derivative-based solvers, such as IPOPT, `fmincon`, FILTERSD, etc., and allows analytical derivatives (accurate to numerical precision) to be supplied and improve the overall optimization performance of the solvers. In addition, this modelling approach also allows the implementation of certain advanced white-box global solvers, namely, SCIP and BARON, that ensure the global optimality of the optimization problem.

Both the white-box and black-box solvers were used to optimized the algebraic EO model and were compared against each other. As expected, both white-box solvers managed to find the global solution for the EO model, whereas `patternsearch` converged to a local solution. Interestingly, some black-box solvers, namely, IPOPT, `fmincon`, and FILTERSD, also managed to converge to the same solution as the white-box solvers, but this is not always guaranteed. Furthermore, the ORC model was optimized with respect to different working fluids in order to demonstrate the proposed modelling approach in an exhaustive optimization search application.

The following chapter will implement the same modelling method on three real-world ORC systems to highlight the flexibility and the scalability of the proposed modelling approach.

Chapter 4

Algebraic Equation-Oriented and Sequential-Modular Models: Case Studies

This chapter details the construction of the algebraic equation-oriented (EO) models of three real-world Organic Rankine Cycle systems, namely, industrial binary cycle power plants. This builds on the modelling approach discussed in the previous chapter and applying it to a larger and more complex system. These models will form the basis of the other formulations that will be investigated in Chapter 6 and Chapter 7. In addition, the validation of the sequential-modular (SM) models will also be presented. The chapter is divided into three sections where each section is dedicated to one of the three plants. Each section will start with a description of the plant and provide a process flow diagram of the system. This is then followed by a detailed description of the algebraic EO model and the thermodynamic approximations that were carried out. Finally, the equivalent SM model is presented and validated against the original nominal design.

4.1 Introduction

The three ORC plants that will be presented as the case studies are the DOE Pilot Plant [3], the USGeo Plant [3], and the Magmamax Binary Power Plant [3, 7]. These three plants were selected based on their advanced configurations that represent a wide range of ORC systems, as well as their freely available design and parameter information that is sufficient enough to model the plant. In order to simplify the modelling procedure and for a fair comparison between the original plant information and the models, the following simplifications and assumptions were made: the plants are assumed to be a steady-state and a steady-flow process, changes in kinetic and potential energy are negligible, losses induced by friction are negligible, and the thermodynamic and transport properties of the brine and cooling water were treated as pure water.

The three algebraic EO models were constructed in MATLAB using the SymBuilder framework via the OPTI Toolbox. This is to allow the Jacobian and Hessian matrices, including the matrix sparsity information, to be automatically generated and provided to the optimization solvers. These algebraic EO models are accompanied by an equivalent SM model that is constructed using the JSteam MATLAB Interface for each of the plant, which will be used to validate the optimized results of the algebraic EO model. In addition, both the algebraic EO models and the SM models were constructed with flexibility in mind, thus the models can be subjected to different working fluids that comply with the upper and lower limits of the operating conditions.

4.2 DOE Pilot Plant

One way to minimize thermodynamic losses in the heat exchangers is to reduce the temperature difference between the hot and cold fluid, e.g., the geofluid and the working fluid. This can be achieved via a dual-pressure ORC system, where there is a two-stage heating-boiling process that achieves a closer temperature match between the working fluid and the geofluid than the conventional basic ORC system. The United States Department of Energy Pilot Plant (DOEP) is a dual-pressure binary cycle power plant, which consists of a high-pressure (HP) and a low-pressure (LP) ORC element. The plant only has one circulating pump that is used to generate the high-pressure fluid for the system. After exiting from the pump, the working fluid is heated inside the low-temperature (LT) preheater before being separated into the LP and HP elements. A portion of the working fluid enters the control valve and the LP evaporator/superheater before expanding through the LP turbine. Whereas the other portion of the working fluid continues to be heated by the HP preheater and HP evaporator/superheater before expanding through the HP turbine. The exhaust vapours from both turbines are mixed together before entering the water-cooled condenser and the pump. Figure 4.1 shows the process flow diagram of the DOEP that was modelled for this research. The following subsections will detail the algebraic EO model of the DOEP and the validation of the SM model.

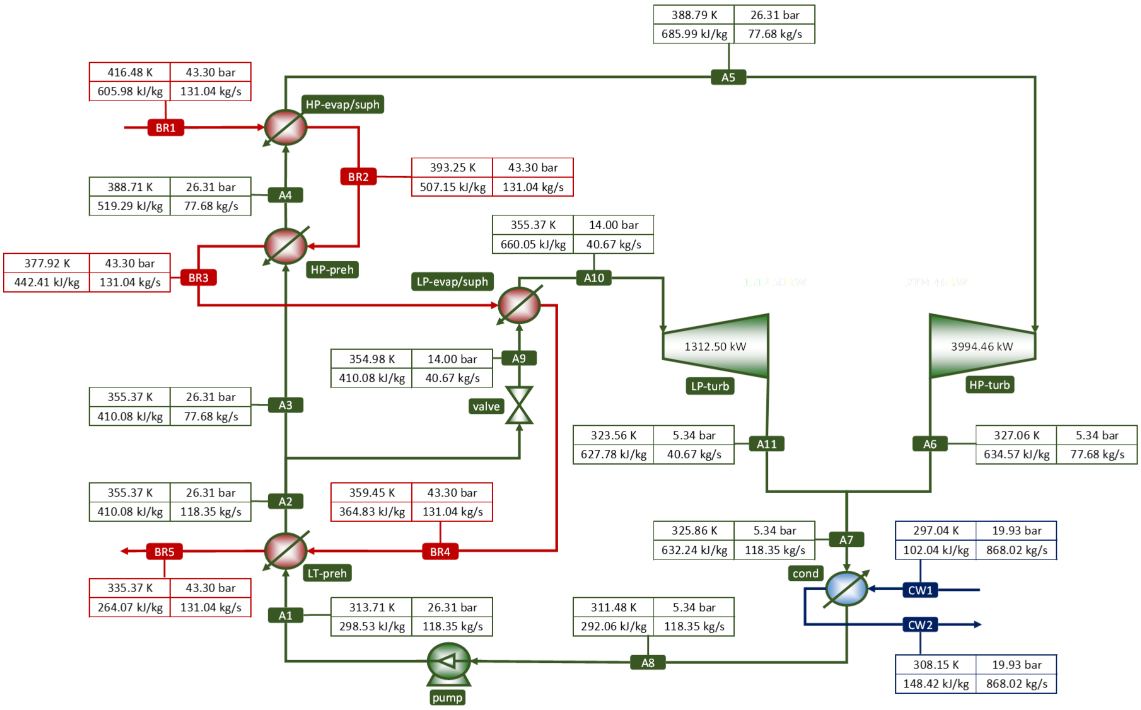


Figure 4.1: The process flow diagram of the DOE Pilot Plant with R600a as the working fluid. The state-point properties were obtained from the solved SM model, see Section 4.2.2.

4.2.1 DOEP Algebraic Equation-Oriented Model

The algebraic EO model of the DOEP was based on the nominal design state-point properties that were adapted from [3], as shown in Table 4.1. In an attempt to maintain the design specifications of the plant when the model is optimized under different operating conditions, the following constants listed in Table 4.2 were used in the model. These constants were adapted and derived from the nominal design state-point properties in Table 4.1 and from the reference text [3], such as the mass flow fraction and the isentropic efficiencies of the pump. Note that in order to optimize the model with respect to different working fluids, the pressure at state A8 was recalculated at the saturation point using the condenser outlet temperature and offset by 0.26 bar to account for the subcooling condition as per [3].

Using the process flow digram in Figure 4.1, the energy balance equations of the

Table 4.1: The DOEP nominal design state-point properties that were obtained from [3]. Refer to Figure 4.1 for the state labels.

State label	Temperature [K]	Pressure [bar]	Entropy [kJ/kgK]	Enthalpy [kJ/kg]	Mass Flow [kg/s]
BR1	416.48	43.30	1.76	603.83	131.04
BR2	394.26	-	-	508.93	131.04
BR3	378.71	-	-	442.87	131.04
BR4	360.93	-	-	367.97	131.04
BR5	335.37	-	-	260.74	131.04
A1	313.71	26.31	1.32	298.70	117.68
A2	355.37	26.31	1.66	410.38	117.68
A3	355.37	26.31	1.66	410.38	77.24
A4	388.71	26.31	1.95	519.98	77.24
A5	388.71	26.31	2.38	686.45	77.24
A6	-	-	-	-	-
A7	326.48	5.34	2.38	625.72	117.68
A8	311.48	5.34	1.31	292.26	117.68
A9	355.37	14.00	1.66	410.84	40.45
A10	355.37	14.00	2.37	659.93	40.45
A11	-	-	-	-	-
CW1	297.04	19.93	0.35	100.25	948.76
CW2	308.15	-	-	146.77	948.76

algebraic EO model can be expressed as follows:

$$\begin{aligned}
\dot{m}_{\mathrm{WF}} h_{\mathrm{A8}} + \dot{W}_{\mathrm{pump}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A1}} &= 0,\\
\dot{m}_{\mathrm{WF}} h_{\mathrm{A1}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A2}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR4}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR5}} &= 0,\\
z\dot{m}_{\mathrm{WF}} h_{\mathrm{A3}} - z\dot{m}_{\mathrm{WF}} h_{\mathrm{A4}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR2}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR3}} &= 0,\\
youtuz\dot{m}_{\mathrm{WF}} h_{\mathrm{A4}} - z\dot{m}_{\mathrm{WF}} h_{\mathrm{A5}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR1}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR2}} &= 0,\\
z\dot{m}_{\mathrm{WF}} h_{\mathrm{A5}} - \dot{W}_{\mathrm{turbHP}} - z\dot{m}_{\mathrm{WF}} h_{\mathrm{A6}} &= 0,\\
\dot{m}_{\mathrm{WF}} h_{\mathrm{A7}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A8}} + \dot{m}_{\mathrm{CW}} h_{\mathrm{CW1}} - \dot{m}_{\mathrm{CW}} h_{\mathrm{CW2}} &= 0,\\
(1-z)\dot{m}_{\mathrm{WF}} h_{\mathrm{A9}} - (1-z)\dot{m}_{\mathrm{WF}} h_{\mathrm{A10}} + \dot{m}_{\mathrm{BR}} h_{\mathrm{BR3}} - \dot{m}_{\mathrm{BR}} h_{\mathrm{BR4}} &= 0,\\
(1-z)\dot{m}_{\mathrm{WF}} h_{\mathrm{A10}} - \dot{W}_{\mathrm{turbLP}} - (1-z)\dot{m}_{\mathrm{WF}} h_{\mathrm{A11}} &= 0,\\
z\dot{m}_{\mathrm{WF}} h_{\mathrm{A6}} + (1-z)\dot{m}_{\mathrm{WF}} h_{\mathrm{A11}} - \dot{m}_{\mathrm{WF}} h_{\mathrm{A7}} &= 0,
\end{aligned} \tag{4.1}$$

where $\dot{m}$ is the mass flow rate, z is the mass flow fraction, h is the enthalpy, and $\dot{W}$ is the mechanical work. In order to simplify the algebraic EO model and allow the model to be optimized with respect to different working fluids, it was assumed that there were no changes in the pressure across the heat exchangers, the HP turbine inlet temperature and pressure were set to $T_{\mathrm{A5}} \in [T_{g@26.31\,\mathrm{bar}}, 388.79]\,\mathrm{K}$ and $P_{\mathrm{A5}} = P_{\mathrm{A1}} \in [P_{\mathrm{A8}} + 5, 26.31]\,\mathrm{bar}$, and the LP turbine inlet temperature and pressure were set to $T_{\mathrm{A10}} \in [T_{g@14.00\,\mathrm{bar}}, 355.37]\,\mathrm{K}$ and $P_{\mathrm{A10}} \in [P_{\mathrm{A8}} + 1, 14.00]\,\mathrm{bar}$. Note the upper and lower limits of the variables were selected to closely match the operating

Table 4.2: The DOE Pilot Plant constant parameters.

Plant Constants	**Value**	**Unit**
$\dot{m}_{\text{BR}}$	131.00	kg/s
T_{BR1}	416.48	K
T_{BR5}	335.37	K
P_{BR}	43.30	bar
T_{CW1}	297.04	K
T_{CW2}	308.15	K
P_{CW}	19.90	bar
η_{turb}	85.00	%
η_{pump}	60.54	%
T_{A8}	311.48	K
P_{A8}	$P_{f@T_{\text{A8}}} + 0.26$	bar
z_{WF}	77.24/117.68	-

conditions of the plant according to its reference for a fair comparison, and to allow the plant to be optimized over a reasonable range of temperatures and pressures without damaging/degrading the operation of plant's components.

Since the original working fluid of the plant is isobutane (R600a), the upper and lower limits of the HP and LP turbine inlet temperatures are approximately equal to each other based on the assumption made above. This can cause some difficulties when approximating the turbine isentropic work across the inlet temperature values. Therefore, the upper-temperatures were offset by 1 K to allow for a more appropriate range between the upper and lower limits. To clearly present the regression fits over a range of turbine temperature values, the following thermodynamic approximations will be carried out using R134a as the working fluid of the plant.

Based on the assumptions stated above, the isentropic work of the HP turbine can be calculated and approximated as a function of the inlet pressure and enthalpy, as shown a Figure 4.2. Consequently, the HP turbine work can be expressed as

$$\begin{aligned}\Delta\hat{h}_{\text{is,turbHP}} =& 0.00128P_{\text{A5}}^3 - 0.088P_{\text{A5}}^2 - 5.52\times10^{-5}P_{\text{A5}}^2h_{\text{A5}} + \\ & 0.00876P_{\text{A5}}h_{\text{A5}} + 0.0156P_{\text{A5}} - 0.0755h_{\text{A5}} + 5.07, \\ \dot{W}_{\text{turbHP}} =& \Delta\hat{h}_{\text{is,turbHP}}\dot{m}_{\text{WF}}z_{\text{WF}}\eta_{\text{turb}},\end{aligned} \tag{4.2}$$

where the isentropic work $\Delta\hat{h}_{\text{is,turbHP}}$ was fitted using a cubic polynomial surface. For the LP turbine, the approximated expression of the output mechanical work can be expressed as

$$\begin{aligned}\Delta\hat{h}_{\text{is,turbLP}} =& 0.00524P_{\text{A10}}^3 - 0.19P_{\text{A10}}^2 - 0.000218P_{\text{A10}}^2h_{\text{A10}} + \\ & 0.0153P_{\text{A10}}h_{\text{A10}} - 0.312P_{\text{A10}} - 0.131h_{\text{A10}} + 16.7, \\ \dot{W}_{\text{turbLP}} =& \Delta\hat{h}_{\text{is,turbLP}}\dot{m}_{\text{A}}(1 - z_{\text{WF}})\eta_{\text{turb}},\end{aligned} \tag{4.3}$$

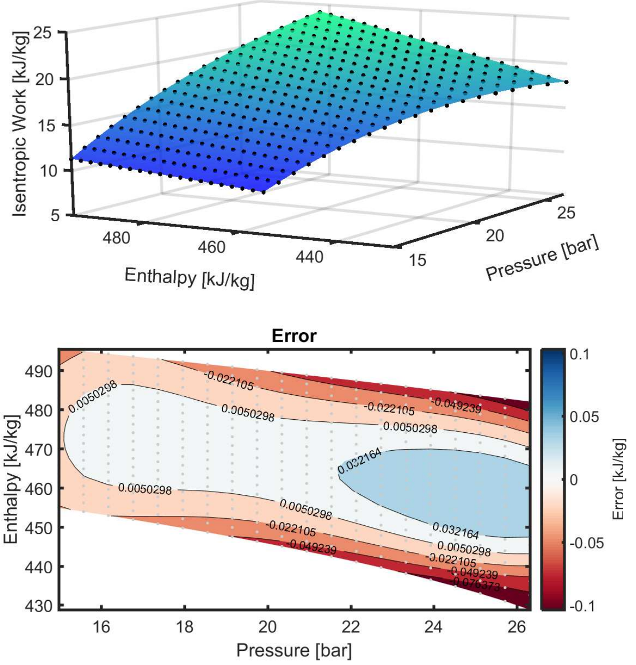


Figure 4.2: The regression fit and error plot of the DOEP HP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

where the isentropic work $\Delta\hat{h}_{\text{is,turbLP}}$ was fitted using a cubic polynomial surface, as shown in Figure 4.3.

Similarly, the isentropic pump work can be approximated as a function of the outlet pressure using the same approach, as shown in Figure 4.4. This resulted in the following expressions:

$$\begin{aligned} \Delta\hat{h}_{\text{is,pump}} =& 0.0866 P_{\text{A5}} - 0.862, \\ \dot{W}_{\text{pump}} =& \frac{\Delta\hat{h}_{\text{is,pump}} \dot{m}_{\text{WF}}}{\eta_{\text{pump}}}, \end{aligned} \tag{4.4}$$

where the isentropic work $\Delta\hat{h}_{\text{is,pump}}$ was fitted using a linear curve.

In order to ensure that the optimized model operates as an ORC system and retains the original design specifications of the plant, some operational constraints

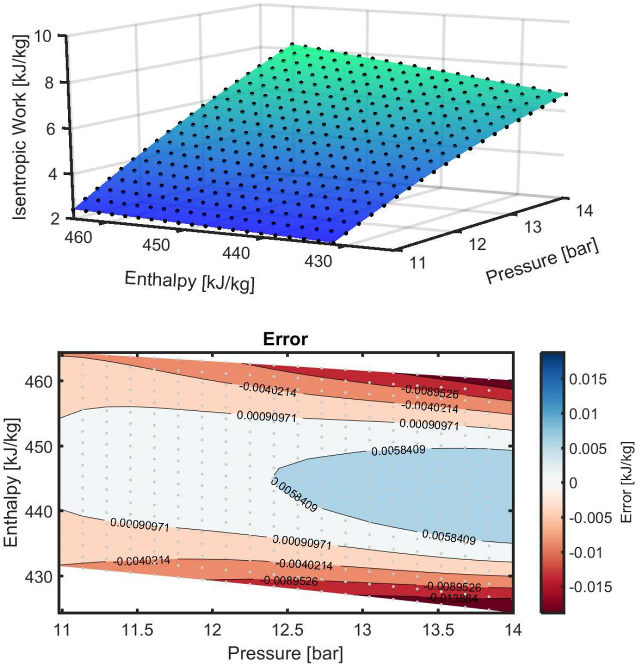


Figure 4.3: The regression fit and error plot of the DOEP LP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

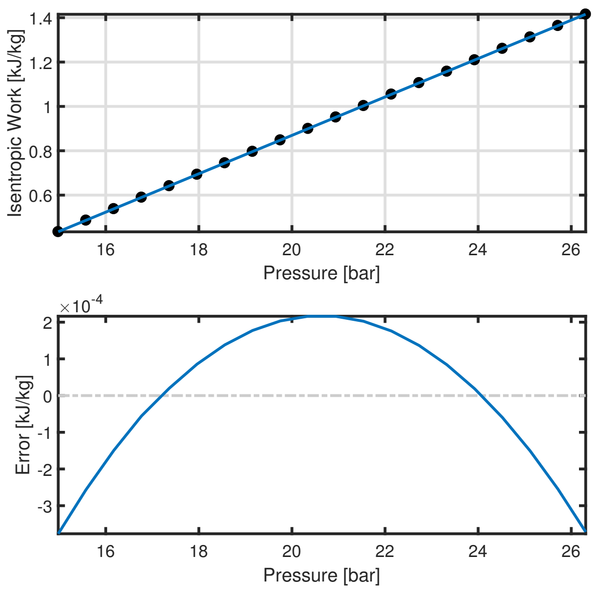


Figure 4.4: The regression fit and error plot of the DOEP pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

were added to the optimization problem. This results in

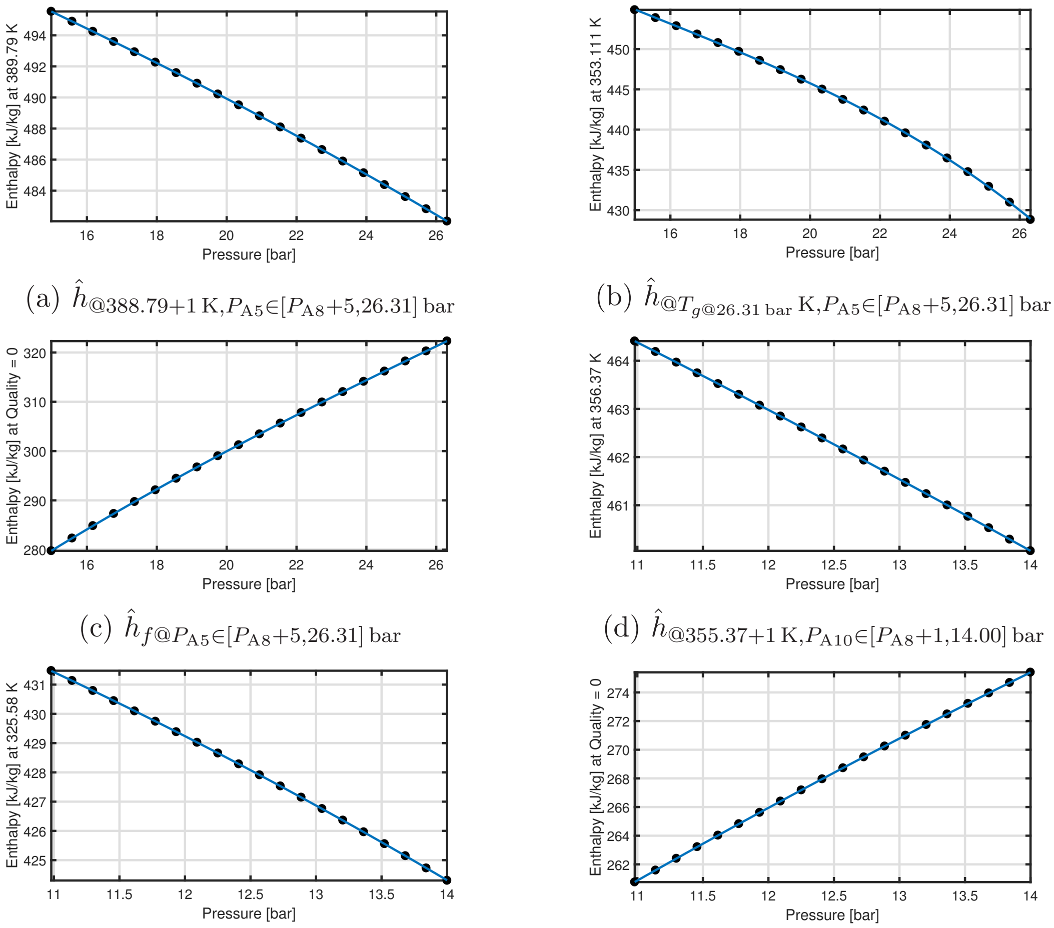
$$
\begin{aligned}
h_{\mathrm{A5}} &\leq \hat{h}_{@388.79+1\,\mathrm{K}, P_{\mathrm{A5}} \in [P_{\mathrm{A8}}+5, 26.31]\,\mathrm{bar}}, \\
h_{\mathrm{A5}} &\geq \hat{h}_{@T_{g@26.31\,\mathrm{bar}}\,\mathrm{K}, P_{\mathrm{A5}} \in [P_{\mathrm{A8}}+5, 26.31]\,\mathrm{bar}}, \\
h_{\mathrm{A4}} &= \hat{h}_{f@P_{\mathrm{A5}} \in [P_{\mathrm{A8}}+5, 26.31]\,\mathrm{bar}}, \\
h_{\mathrm{A10}} &\leq \hat{h}_{@355.37+1\,\mathrm{K}, P_{\mathrm{A10}} \in [P_{\mathrm{A8}}+1, 14.00]\,\mathrm{bar}}, \\
h_{\mathrm{A10}} &\geq \hat{h}_{@T_{g@14.00\,\mathrm{bar}}\,\mathrm{K}, P_{\mathrm{A10}} \in [P_{\mathrm{A8}}+1, 14.00]\,\mathrm{bar}}, \\
h_{\mathrm{A9}} &= \hat{h}_{f@P_{\mathrm{A10}} \in [P_{\mathrm{A8}}+1, 14.00]\,\mathrm{bar}}, \\
h_{\mathrm{A3}} &= h_{\mathrm{A9}}, \\
h_{\mathrm{A2}} &= h_{\mathrm{A9}} \\
P_{\mathrm{A5}} &\geq P_{\mathrm{A10}},
\end{aligned}
\tag{4.5}
$$

for the HP and LP ORC elements, where the first two equations ensure that the inlet state of the HP turbine does not operate outside the specified temperature and pressure range, the third equation ensures that state A4 stays at the saturated liquid point, the fourth and fifth equation ensure the inlet state of the LP turbine does not operate outside the specified temperature and pressure range, the sixth equation ensures that state A9 is at the saturated liquid point, the seventh and eighth equation ensure the enthalpy values at state A3, A2, and A9 are equal, and the ninth equation ensures that the inlet pressure of the HP turbine is higher than (or equal to) the inlet pressure of the LP turbine. Furthermore, the following constraints are also needed for the geofluid system:

$$
\begin{aligned}
h_{\mathrm{BR3}} &\leq h_{\mathrm{BR2}}, \\
h_{\mathrm{BR4}} &\leq h_{\mathrm{BR3}}.
\end{aligned}
\tag{4.6}
$$

The right-hand side thermodynamic terms of the first 6 constraints in (4.5) were regressed as a function of pressure in order to be compatible with the white-box solvers, as shown in Figure 4.5.

Furthermore, in order to ensure that temperature difference at the inlet and outlet of each heat exchanger is greater than (or equal to) 1 K to offset for the thermodynamic approximations, and also to ensure that the heat is only transferred from the hot fluid to the cold fluid, the inlet and outlet temperatures of each heat



(a) $\hat{h}_{@388.79+1\,\mathrm{K},P_{\mathrm{A5}}\in[P_{\mathrm{A8}}+5,26.31]\,\mathrm{bar}}$

(b) $\hat{h}_{@T_{g@26.31\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{A5}}\in[P_{\mathrm{A8}}+5,26.31]\,\mathrm{bar}}$

(c) $\hat{h}_{f@P_{\mathrm{A5}}\in[P_{\mathrm{A8}}+5,26.31]\,\mathrm{bar}}$

(d) $\hat{h}_{@355.37+1\,\mathrm{K},P_{\mathrm{A10}}\in[P_{\mathrm{A8}}+1,14.00]\,\mathrm{bar}}$

(e) $\hat{h}_{@T_{g@14.00\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{A10}}\in[P_{\mathrm{A8}}+1,14.00]\,\mathrm{bar}}$

(f) $\hat{h}_{f@P_{\mathrm{A10}}\in[P_{\mathrm{A8}}+1,14.00]\,\mathrm{bar}}$

Figure 4.5: The thermodynamic regressions for the operational constraints in (4.5). The black dots are the REFPROP thermodynamic calculations.

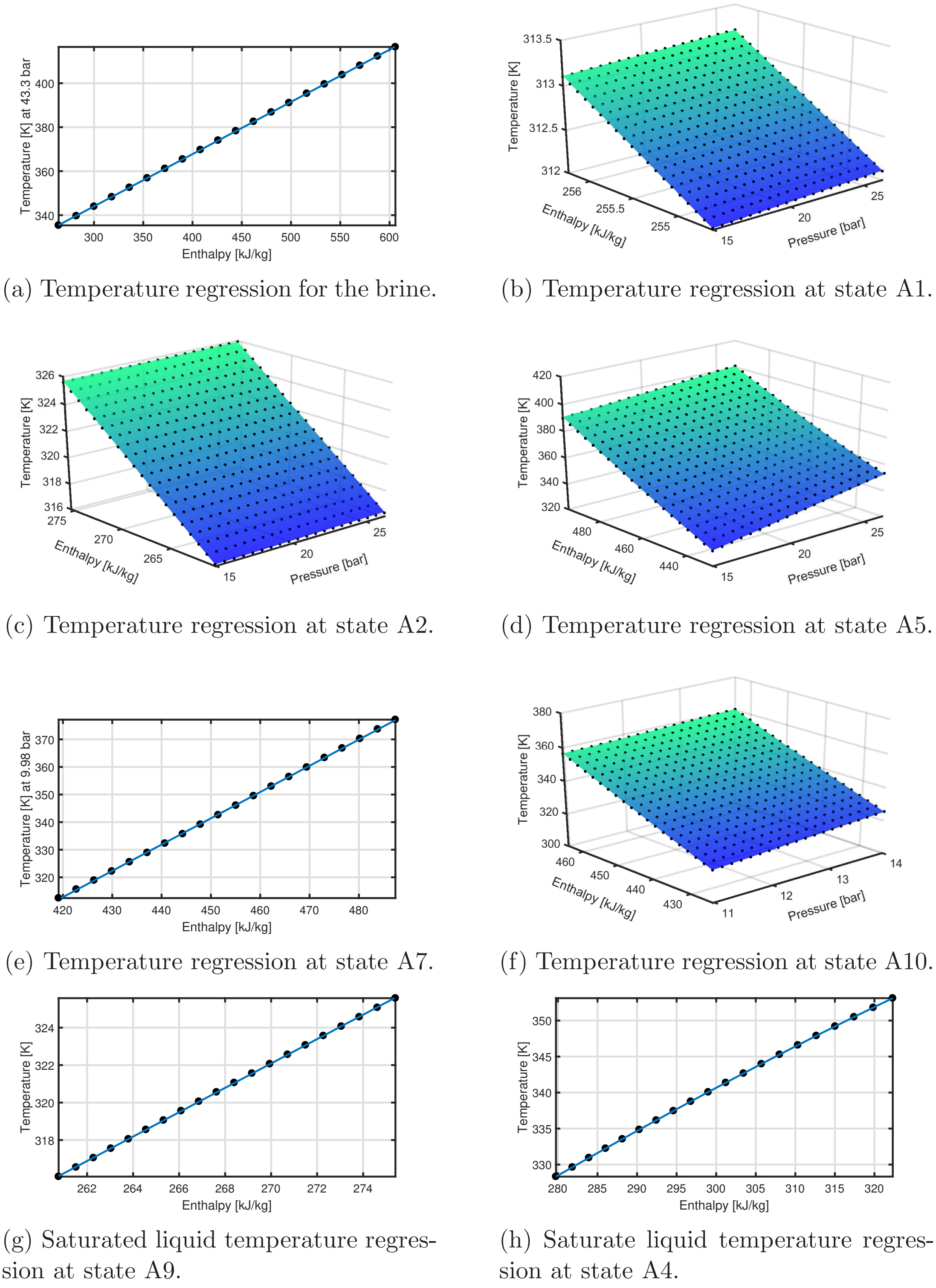
exchanger were constrained as follows:

$$\begin{aligned}
\hat{T}_{\mathrm{A7}} - T_{\mathrm{CW2}} &\geq 1,\\
\hat{T}_{\mathrm{BR4}} - \hat{T}_{\mathrm{A2}} &\geq 1,\\
T_{\mathrm{BR5}} - \hat{T}_{\mathrm{A1}} &\geq 1,\\
\hat{T}_{\mathrm{BR3}} - \hat{T}_{\mathrm{A10}} &\geq 1,\\
\hat{T}_{\mathrm{BR4}} - \hat{T}_{\mathrm{A9}} &\geq 1,\\
\hat{T}_{\mathrm{BR2}} - \hat{T}_{\mathrm{A4}} &\geq 1,\\
\hat{T}_{\mathrm{BR3}} - \hat{T}_{\mathrm{A2}} &\geq 1,\\
T_{\mathrm{BR1}} - \hat{T}_{\mathrm{A5}} &\geq 1.
\end{aligned} \tag{4.7}$$

The temperature terms in (4.7) can be approximated as a function of enthalpy and/or pressure, as shown in Figure 4.6, where the approximated range (i.e., the enthalpy and pressure range) can be obtained from the previous unit operation or thermodynamic approximations in the operational constraints.

Lastly, in order to reduce the search space and improve on the computational time, the decision variables were bounded within a suitable range based on the assumptions and simplifications made at the beginning of this section, such as the HP turbine inlet temperature must be $T_{\mathrm{A5}} \in [T_{g@26.31\,\mathrm{bar}}, 388.79]\,\mathrm{K}$ and the pump outlet pressure must be $P_{\mathrm{A5}} = P_{\mathrm{A1}} \in [P_{\mathrm{A8}} + 5, 26.31]\,\mathrm{bar}$. For the DOE Pilot Plant described above, the bounds were

$$\begin{aligned}
14.98 \leq P_{\mathrm{A5}} &\leq 26.31,\\
254.64 \leq h_{\mathrm{A1}} &\leq 256.26,\\
260.77 \leq h_{\mathrm{A2}} &\leq 275.40,\\
260.77 \leq h_{\mathrm{A3}} &\leq 275.40,\\
279.76 \leq h_{\mathrm{A4}} &\leq 322.32,\\
428.82 \leq h_{\mathrm{A5}} &\leq 495.53,\\
419.13 \leq h_{\mathrm{A6}} &\leq 487.40,\\
419.13 \leq h_{\mathrm{A7}} &\leq 487.40,\\
1930.86 \leq \dot{W}_{\mathrm{turbHP}} &\leq 3585.88
\end{aligned} \tag{4.8}$$



(a) Temperature regression for the brine.

(b) Temperature regression at state A1.

(c) Temperature regression at state A2.

(d) Temperature regression at state A5.

(e) Temperature regression at state A7.

(f) Temperature regression at state A10.

(g) Saturated liquid temperature regression at state A9.

(h) Saturate liquid temperature regression at state A4.

Figure 4.6: The temperature regressions at various stages of the DOE Pilot Plant. The black dots are the REFPROP thermodynamic calculations.

for the HP elements;

$$
\begin{aligned}
10.98 &\leq P_{\mathrm{A10}} \leq 14.00, \\
260.77 &\leq h_{\mathrm{A9}} \leq 275.40, \\
424.31 &\leq h_{\mathrm{A10}} \leq 464.41, \\
419.13 &\leq h_{\mathrm{A11}} \leq 462.65, \\
344.76 &\leq \dot{W}_{\mathrm{turbLP}} \leq 640.26
\end{aligned}
\tag{4.9}
$$

for the LP elements; and

$$
\begin{aligned}
264.07 &\leq h_{\mathrm{BR}} \leq 605.98, \\
100.00 &\leq \dot{m}_{\mathrm{CW}} \leq 1200.0, \\
50.00 &\leq \dot{m}_{\mathrm{WF}} \leq 500.0, \\
334.54 &\leq \dot{W}_{\mathrm{pump}} \leq 621.28, \\
1941.1 &\leq \dot{W}_{\mathrm{net}} \leq 3604.9
\end{aligned}
\tag{4.10}
$$

for the brine, mass flow, pump work and the net power. Note that some of the bounds will change depending on the working fluid due to the differences in thermodynamic properties.

4.2.2 DOEP Sequential-Modular Model Validation

The DOEP SM model was constructed using the JSteam MATLAB Interface modelling framework and adhered to the same specifications as the algebraic EO model in Section 4.2.1. The SM model was built using the pre-built unit operation functions in JSteam that were arranged in a sequential order of the plant process, where the output of one function is the input argument of the proceeding function. The purpose of this SM model is to validate the optimized results of the EO model and also to check if the approximations made to the thermodynamic processes and properties are invalid or violate any thermodynamic laws. Therefore, if there is a large discrepancy between the SM model and the EO model, the approximations or the optimization constraints will need to be reviewed, as illustrated in Figure 2.4.

The nominal design state-point temperature and pressure values for DOEP from Table 20.4 in [3] (adapted in Table 4.1 for this research) were used to validate the SM model. Table 4.3 shows the comparison between the SM model and original published results, where the power analysis and cooling water mass flow were calculated using state-point properties of the working fluid in Table 4.1. Note that the LP and HP turbine power outputs of the nominal design were calculated using (3.1) and the isentropic efficiency in Table 4.2.

The SM model was solved by substituting the nominal state-point properties from [3] into the model, namely, T_{A2}, T_{A4}, T_{A5}, T_{A10}, P_5 and P_{10}, and using a nonlinear solver `fsolve` to converge to a feasible operating point. Due to rounding errors, the inlet temperatures of the HP turbine was recalculated at its corresponding pressure to ensure that the model complies with the normal operation of an ORC system, hence $T_{\text{A5}} = 388.79\,\text{K}$. When the model is subjected to different working fluids, the temperature at state A4 and A9 (thus, also A2) are recalculated to ensure that they are at the correct saturation point to adhere to the operating condition of the plant. Figure 4.1 and Figure 4.7 show the process flow diagram and the Ts diagram, respectively, of the solved SM model using the original isobutane (R600a) working fluid.

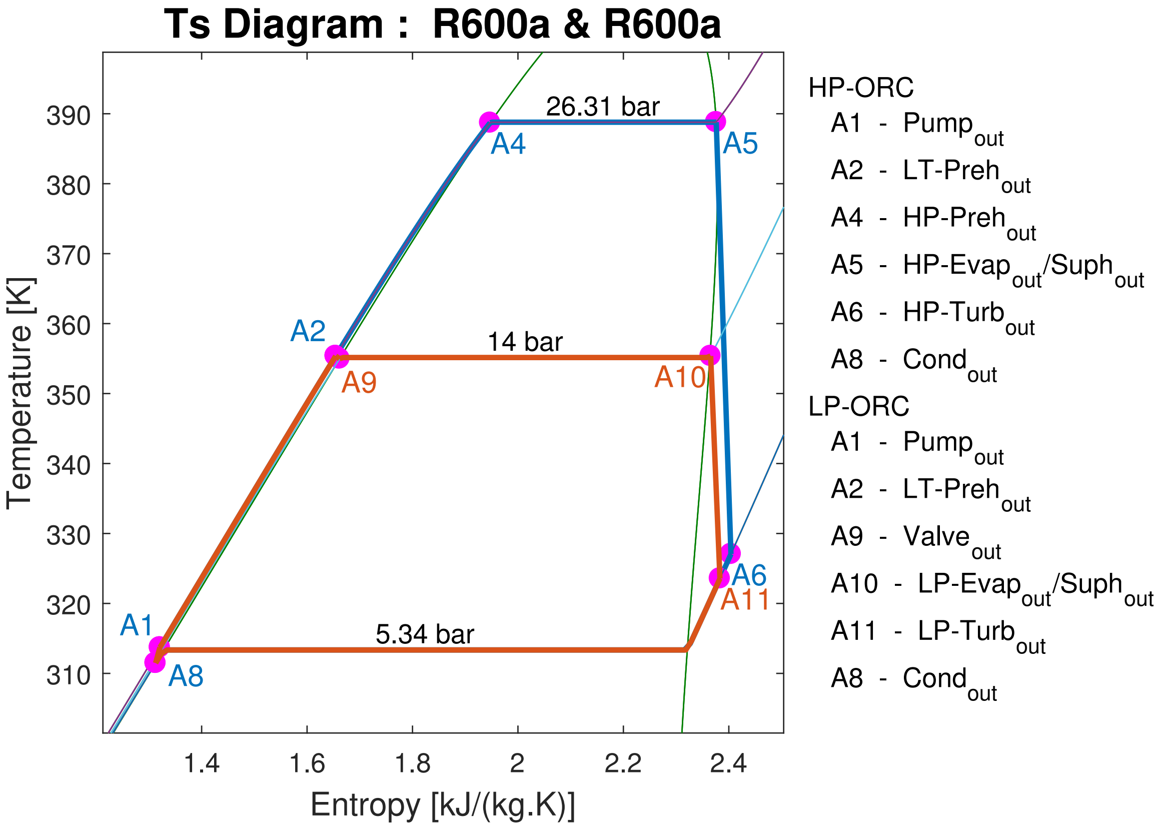


Figure 4.7: The Ts diagram of the DOE Pilot Plant's sequential-modular model.

The discrepancies between the original DOE nominal design state-point properties and the SM model is under 3%, which indicates that the SM model is an accurate presentation of the DOE Pilot Plant. The small discrepancies could be due to the rounding errors from the unit conversions and the difference in thermodynamic packages.

Since the SM model was solved using the original operating point of the plant, this solved SM model will be used as the base case for this research. The same will also apply when the model is subjected to different working fluids.

Table 4.3: A comparison between the DOEP nominal design [3] and the JSteam SM model.

Plant Parameter	**Nominal Design** [3]	**SM Model (Base case)**	**Error** [%]
LT Preheater Duty, [kW]	13142	13203	0.47
LP Evap/Suph Duty, [kW]	10073	10167	0.93
HP Preheater Duty, [kW]	8465.7	8483.7	0.21
HP Evap/Suph Duty, [kW]	12859	12950	0.71
Condenser Duty, [kW]	39242	40262	2.60
Total Input Duty, [kW]	44539	44803	0.59
HP Turbine Power, [kW]	3975.2	3994.5	0.49
LP Turbine Power, [kW]	1303.5	1312.5	0.69
Total Turbine Power, [kW]	5278.6	5307	0.54
Pump Power, [kW]	758.23	765.51	0.96
Net Power, [kW]	4520.4	4541.4	0.47
Working Fluid, [kg/s]	117.68	118.35	0.57
Cooling Water, [kg/s]	843.54	868.02	2.90
Thermal Efficiency, [%]	10.15	10.14	0.10
Specific Machinery Cost, [k$/MW]	331.14	330.89	0.08

For further validation, the heat exchange process between the geothermal brine/-cooling water and the working fluid can be plotted to check for any thermodynamic violations in the heat exchangers, as discussed in Section 3.2.2.1. Evidently, from analysing Figure 4.8, the temperature curves did not cross over each other during the heat exchange process, which indicates that the model does not violate the second law of thermodynamics.

4.3 USGeo Plant

The USGeo Plant (USGP) in southern Idaho, USA, consists of two separate ORC systems operating at different pressure levels. Therefore, unlike the DOEP, the working fluids of the HP and LP cycles are never mixed together. The configuration of the LP cycle is similar to the basic ORC system, where the working fluid is preheated and evaporated/superheated before expanding through the turbine, condensed and fed back to the heat exchangers. The HP cycle undergoes a similar process but employs a recuperator unit to improve the overall performance of the ORC system. This is achieved by exploiting the hot exhaust gas from the turbine to preheat the cold condensed working fluid before it enters the preheater to reduce the heat load on the condenser and increase the internal cycle efficiency. For the geofluid system, the geofluid first flows through the HP and LP evaporators/superheaters before splitting equally into two streams to the LP and HP preheaters. In contrast, the cooling water of the cooling system is divided before it goes through

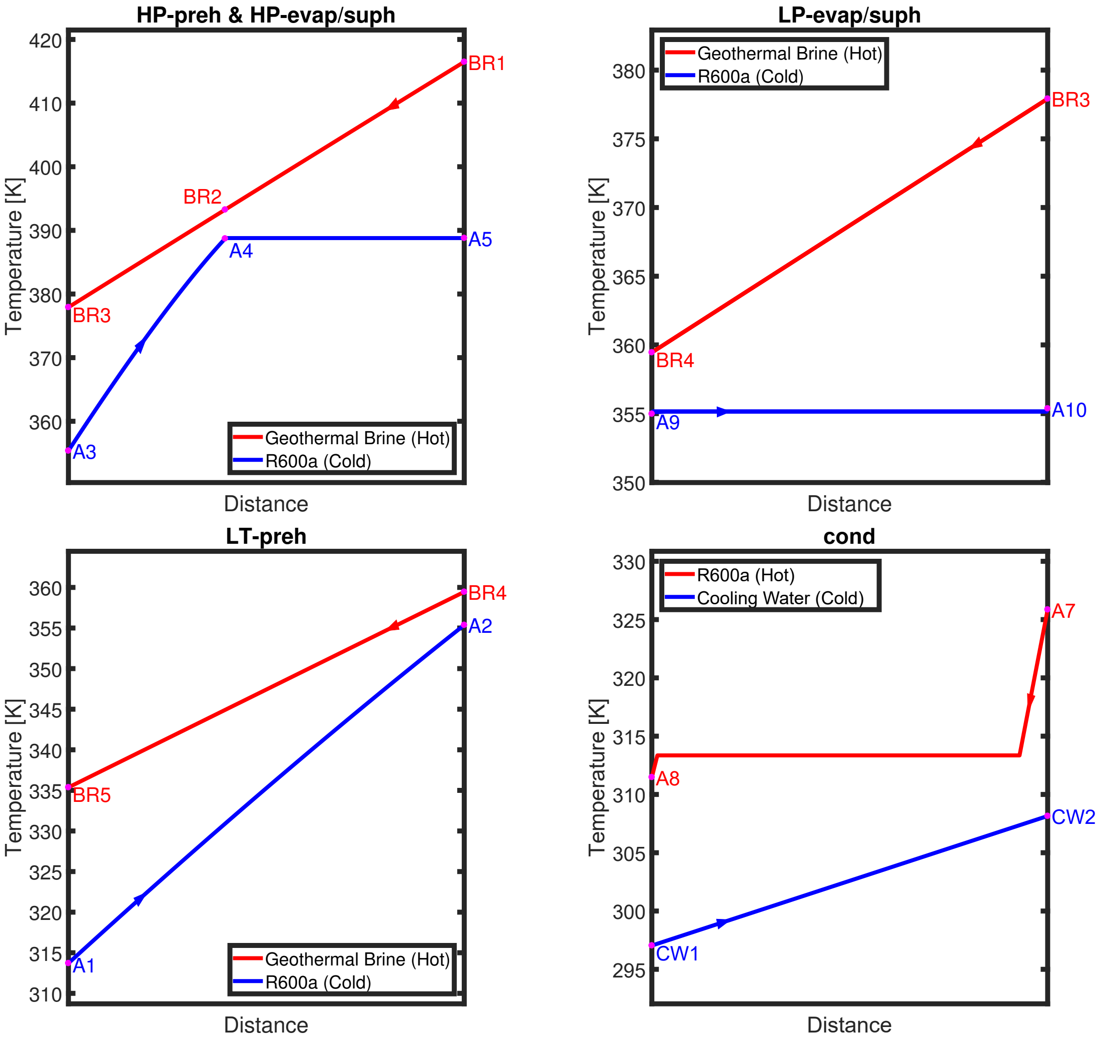


Figure 4.8: Diagrams showing the heat exchange between the geothermal brine/-cooling water and the working fluid. The states refer to Figure 4.1. The x-axis represents the path of the fluid flow in the heat exchanger.

the LP and HP condensers. For this research, the cooling system was simplified into two separate streams to better address the temperature constraints discussed in Section 3.2.2.1. Figure 4.9 shows the process flow diagram of the USGP that was modelled for this research. The following subsections will detail the algebraic EO model of the USGP and the validation of the SM model.

4.3.1 USGP Algebraic Equation-Oriented Model

The development of the algebraic EO model of the USGP was based on the nominal design state-point properties obtained from [3], as shown in Table 4.4. From analysing the state-point properties of the plant, it was noticed that the output temperature of the HP preheater, A3, is higher than the brine input, BR5. This should not happen as the temperature at A3 cannot exceed that of the heat source temperature because then the heat transfer will be reversed. Therefore, in order to

Table 4.4: The USGP nominal design state-point properties obtained from [3]. Refer to Figure 4.9 for the state labels.

State	Temperature	Pressure	Entropy	Enthalpy	Mass Flow
label	[K]	[bar]	[kJ/kgK]	[kJ/kg]	[kg/s]
BR1	410.93	7.72	1.72	580.31	396.89
BR2	383.15	7.24	1.42	462.15	396.89
BR3	360.59	6.83	1.16	367.02	396.89
BR4	360.59	6.83	1.16	367.02	198.45
BR5	360.59	6.83	1.16	367.02	198.45
BR6	336.15	6.34	0.87	264.42	198.45
BR7	341.37	6.34	0.93	286.31	198.45
BR8	338.76	6.34	0.90	275.35	396.89
A1	293.59	9.48	-0.06	-16.05	139.32
A2	311.26	9.48	0.08	24.48	139.32
A3	386.26	9.48	0.63	217.74	139.32
A4	386.26	9.48	1.30	476.02	139.32
A5	335.54	0.90	1.35	405.65	139.32
A6	313.43	0.90	1.23	365.74	139.32
A7	293.04	0.90	-0.06	-18.02	139.32
B1	292.82	5.07	-0.06	-18.18	128.85
B2	357.43	5.07	0.42	138.68	128.85
B3	357.43	5.07	1.24	431.43	128.85
B4	321.32	0.88	1.27	379.84	128.85
B5	292.54	0.88	-0.06	-19.15	128.85
CW1	285.09	2.83	0.18	50.48	1465.99
CW2	285.09	2.83	0.18	50.48	1465.99
CW3	293.82	1.65	0.31	86.92	1465.99
CW4	293.48	1.65	0.30	85.52	1465.99

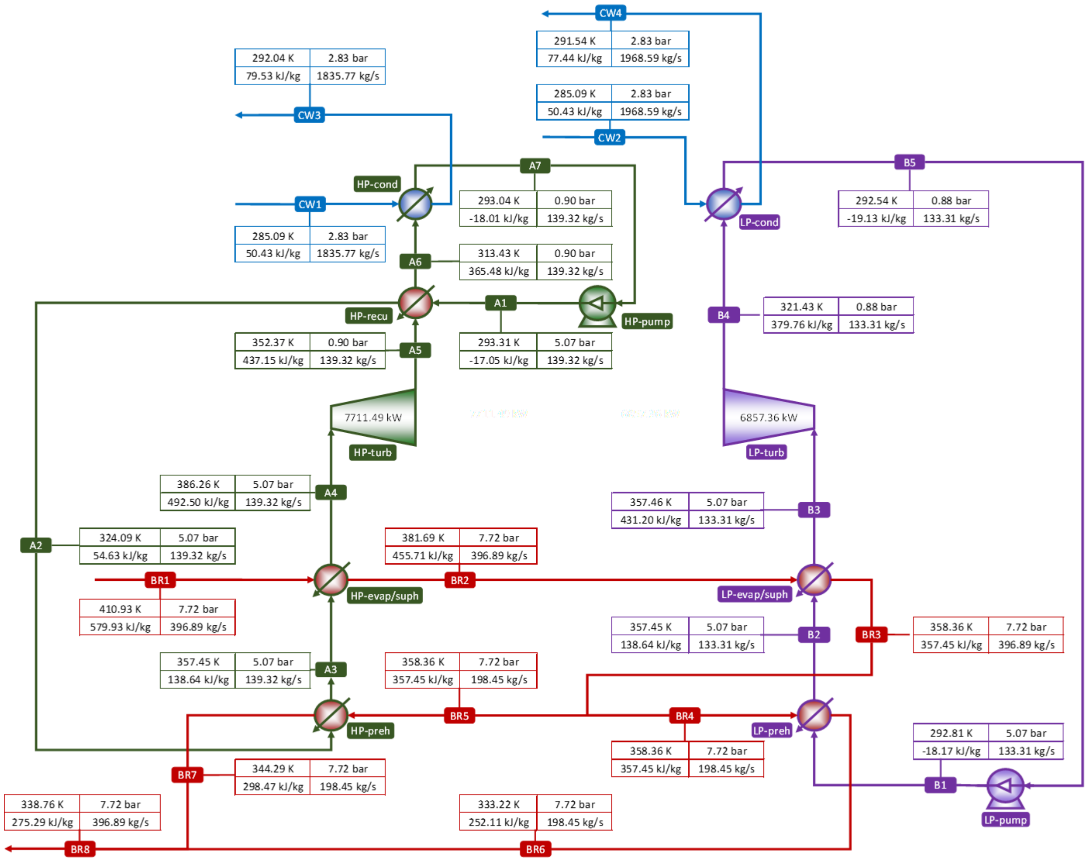


Figure 4.9: The process flow diagram of the USGeo Plant with the R601a as the working fluid. The state-point properties were obtained from the solved SM model, see Section 4.3.2.

address this issue, the pressure at state A3 was decreased to 5.07 bar from 9.48 bar and the saturated liquid temperature was recalculated at the new pressure level.

In order to retain some of the plant's design characteristics in Table 4.4, the plant's constants listed in Table 4.5 were used in the model. Note that both P_{A7} and P_{B5} were recalculated at the saturated liquid points using the condensers' outlet temperatures and offset by 0.14 and 0.13 bar, respectively. This is to ensure the subcooling conditions as per the plant's design state-point properties and for when the model is optimized with respect to different working fluids. The isentropic efficiencies of the turbines and pumps were calculated using state-point properties in Table 4.4. Furthermore, the outlet temperatures of the cooling water for the LP and HP ORC have been decreased to 1 K less than the outlet temperature of their respective condenser. This is to ensure that the thermodynamic violation discussed Section 3.2.2.1 does not occur.

Using the process flow diagram of the USGP in Figure 4.9, the energy balance

Table 4.5: The USGeo Plant constant parameters.

Plant Constants	Value	Unit
$\dot{m}_{\mathrm{BR}}$	396.89	kg/s
T_{BR1}	410.93	K
T_{BR8}	338.89	K
P_{BR}	7.72	bar
z_{BR}	0.50	-
η_{turbA}	80.38	%
η_{turbB}	81.88	%
η_{pumpA}	69.97	%
η_{pumpB}	70.02	%
T_{A7}	293.04	K
P_{A7}	$P_{f@T_{\mathrm{A7}}}$ + 0.14	bar
T_{B5}	292.54	K
P_{B5}	$P_{f@T_{\mathrm{B5}}}$ + 0.13	bar
T_{CW1}	285.09	K
T_{CW2}	285.09	K
T_{CW3}	T_{A7}-1	K
T_{CW4}	T_{B5}-1	K
P_{CW}	2.83	bar

equations can be derived as follows:

$$\begin{aligned}
h_{\mathrm{A1}}\dot{m}_{\mathrm{A}} + h_{\mathrm{A5}}\dot{m}_{\mathrm{A}} - h_{\mathrm{A2}}\dot{m}_{\mathrm{A}} - h_{\mathrm{A6}}\dot{m}_{\mathrm{A}} &= 0,\\
h_{\mathrm{A6}}\dot{m}_{\mathrm{A}} + h_{\mathrm{CW1}}\dot{m}_{\mathrm{CWA}} - h_{\mathrm{A7}}\dot{m}_{\mathrm{A}} - h_{\mathrm{CW3}}\dot{m}_{\mathrm{CWA}} &= 0,\\
h_{\mathrm{A7}}\dot{m}_{\mathrm{A}} + \dot{W}_{\mathrm{pumpA}} - h_{\mathrm{A1}}\dot{m}_{\mathrm{A}} &= 0,\\
h_{\mathrm{A2}}\dot{m}_{\mathrm{A}} + h_{\mathrm{BR5}}(1 - z_{\mathrm{BR}})\dot{m}_{\mathrm{BR}} - h_{\mathrm{A3}}\dot{m}_{\mathrm{A}} - h_{\mathrm{BR7}}(1 - z_{\mathrm{BR}})\dot{m}_{\mathrm{BR}} &= 0,\\
h_{\mathrm{A3}}\dot{m}_{\mathrm{A}} + h_{\mathrm{BR1}}\dot{m}_{\mathrm{BR}} - h_{\mathrm{A4}}\dot{m}_{\mathrm{A}} - h_{\mathrm{BR2}}\dot{m}_{\mathrm{BR}} &= 0,\\
h_{\mathrm{A4}}\dot{m}_{\mathrm{A}} - \dot{W}_{\mathrm{turbA}} - h_{\mathrm{A5}}\dot{m}_{\mathrm{A}} &= 0,\\
h_{\mathrm{B1}}\dot{m}_{\mathrm{B}} + h_{\mathrm{BR4}}z_{\mathrm{BR}}\dot{m}_{\mathrm{BR}} - h_{\mathrm{B2}}\dot{m}_{\mathrm{B}} - h_{\mathrm{BR6}}z_{\mathrm{BR}}\dot{m}_{\mathrm{BR}} &= 0,\\
h_{\mathrm{B2}}\dot{m}_{\mathrm{B}} + h_{\mathrm{BR2}}\dot{m}_{\mathrm{BR}} - h_{\mathrm{B3}}\dot{m}_{\mathrm{B}} - h_{\mathrm{BR3}}\dot{m}_{\mathrm{BR}} &= 0,\\
h_{\mathrm{B3}}\dot{m}_{\mathrm{B}} - \dot{W}_{\mathrm{turbB}} - h_{\mathrm{B4}}\dot{m}_{\mathrm{B}} &= 0,\\
h_{\mathrm{B4}}\dot{m}_{\mathrm{B}} + h_{\mathrm{CW2}}\dot{m}_{\mathrm{CWB}} - h_{\mathrm{B5}}\dot{m}_{\mathrm{B}} - h_{\mathrm{CW4}}\dot{m}_{\mathrm{CWB}} &= 0,\\
h_{\mathrm{B5}}\dot{m}_{\mathrm{B}} + \dot{W}_{\mathrm{pumpB}} - h_{\mathrm{B1}}\dot{m}_{\mathrm{B}} &= 0,
\end{aligned} \tag{4.11}$$

where $\dot{m}$ is the mass flow rate, h is the enthalpy, and $\dot{W}$ is the mechanical work. For the purpose of optimizing the model with respect to different working fluids, and in order to simplify the algebraic EO model, the following assumptions were made: There were no changes in the pressure across the heat exchangers, the HP turbine inlet temperature and pressure were set to $T_{\mathrm{A4}} \in [T_{g@5.07\,\mathrm{bar}}, 386.26]\,\mathrm{K}$ and $P_{\mathrm{A4}} = P_{\mathrm{A1}} \in [P_{\mathrm{A7}} + 1, 5.07]\,\mathrm{bar}$, the LP turbine inlet temperature and pressure were set to $T_{\mathrm{B3}} \in [T_{g@5.07\,\mathrm{bar}}, 357.43]\,\mathrm{K}$ and $P_{\mathrm{B3}} = P_{\mathrm{B1}} \in [P_{\mathrm{B5}} + 1, 5.07]\,\mathrm{bar}$, and the temperature difference $\Delta T_{\text{rec,A6-A7}}$ between T_{A6} and T_{A7} was set to 20.39 K as per [3]. Similar to the DOE Pilot Plant, the upper and lower limits of the variables were selected to closely match the operating conditions of the plant according to its reference for a fair comparison, and to allow the plant to be optimized over a reasonable range of temperatures and pressures without damaging/degrading the operation of plant's components.

Since the original working fluid of the plant is isopentane (R601a), the upper and lower limits of the LP turbine inlet temperature are approximately equal to each other based on the assumption made above. Therefore, the upper-temperature limit was increased by 1 K to offset for the isentropic work approximation. In order to clearly illustrate the development of the USGP model in this section and properly show the regression fits over a large temperature range, the thermodynamic approximations will be carried out using R245ca.

Based on the assumptions made above, the isentropic work of the HP turbine work was approximated as a function of the inlet pressures and enthalpy, as shown in Figure 4.10. Subsequently, the HP turbine work can be expressed as

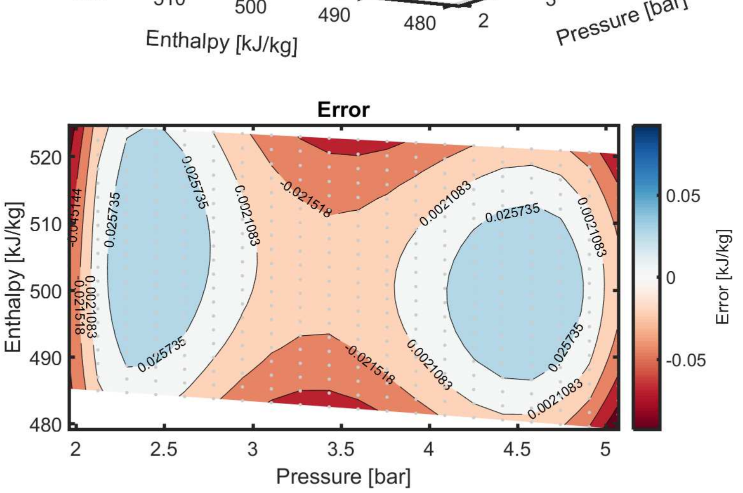


Figure 4.10: The regression fit and error plot of the USGP HP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

$$\begin{aligned}\Delta\hat{h}_{\text{is,turbA}} =& 0.205P_{\text{A1}}^3 - 1.87P_{\text{A1}}^2 - 0.00256P_{\text{A1}}^2h_{\text{A4}}+\\ & 0.0383P_{\text{A1}}h_{\text{A4}} + 0.845P_{\text{A1}} - 0.0194h_{\text{A4}} - 3.67,\\ \dot{W}_{\text{turbA}} =& \eta_{\text{turbA}}\dot{m}_{\text{A}}\Delta\hat{h}_{\text{is,turbA}},\end{aligned} \tag{4.12}$$

where the isentropic work $\Delta\hat{h}_{\text{is,turbA}}$ was fitted using a cubic polynomial surface. Similarly, the LP turbine's isentropic work was calculated and approximated as a function of the inlet pressure and enthalpy, as shown in Figure 4.11, which corresponded to the following output work:

$$\begin{aligned}\Delta\hat{h}_{\text{is,turbB}} =& 0.2P_{\text{B1}}^3 - 1.84P_{\text{B1}}^2 - 0.00251P_{\text{B1}}^2h_{\text{B3}}+\\ & 0.039P_{\text{B1}}h_{\text{B3}} + 0.323P_{\text{B1}} - 0.0173h_{\text{B3}} - 3.85,\\ \dot{W}_{\text{turbB}} =& \eta_{\text{turbB}}\dot{m}_{\text{B}}\Delta\hat{h}_{\text{is,turbB}},\end{aligned} \tag{4.13}$$

where the isentropic work $\Delta\hat{h}_{\text{is,turbB}}$ was approximated using a cubic polynomial surface.

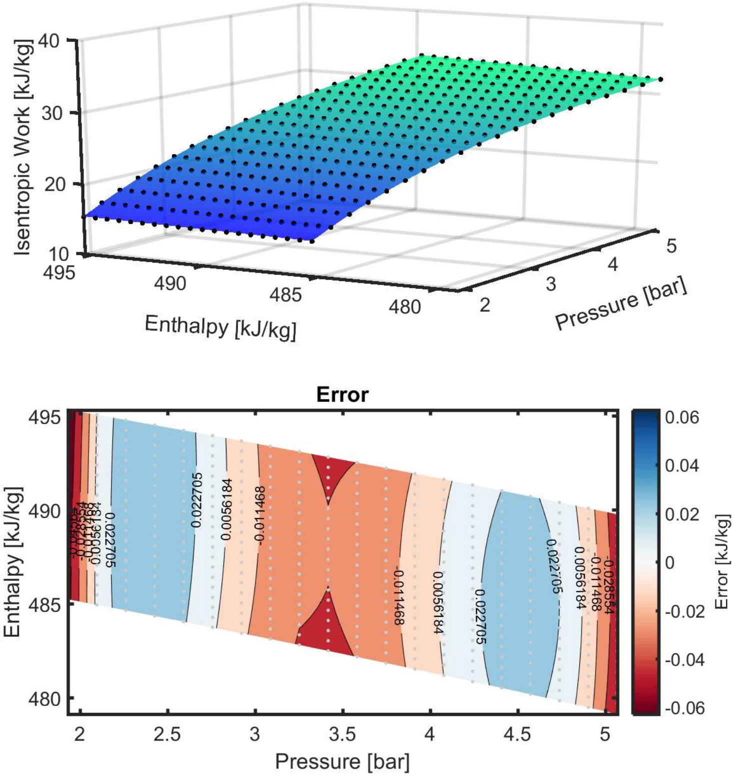


Figure 4.11: The regression fit and error plot of the USGP LP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

The next two terms in (4.11) to be approximated are the HP and LP pump work, which can be expressed as a function of the outlet pressure, as shown in Figure 4.12

and Figure 4.13. This resulted in the following expressions:

$$\begin{aligned} \Delta\hat{h}_{\text{is,pumpA}} &= 0.0714 P_{\text{A1}} - 0.0685, \\ \dot{W}_{\text{pumpA}} &= \dot{m}_{\text{A}} \Delta\hat{h}_{\text{is,pumpA}} / \eta_{\text{pumpA}} \end{aligned} \tag{4.14}$$

for the HP pump work, and

$$\begin{aligned} \Delta\hat{h}_{\text{is,pumpB}} &= 0.0713 P_{\text{B1}} - 0.0663, \\ \dot{W}_{\text{pumpB}} &= \dot{m}_{\text{B}} \Delta\hat{h}_{\text{is,pumpB}} / \eta_{\text{pumpB}} \end{aligned} \tag{4.15}$$

for the LP pump work, where both isentropic works Δh_{is} were fitted using a linear curve.

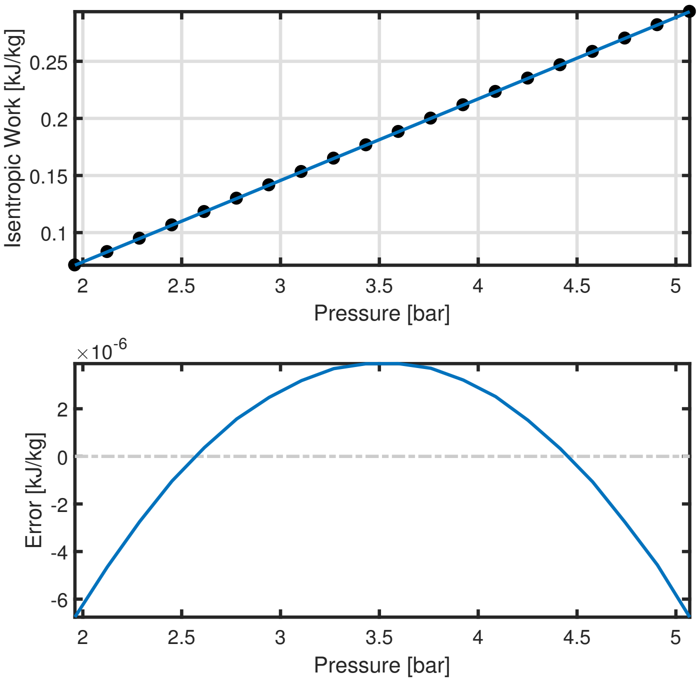


Figure 4.12: The regression fit and error plot of the USGP HP pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

To ensure the model is operating as a binary cycle power plant and adhere to the design characteristics of the original system, the operational constraints/limitations

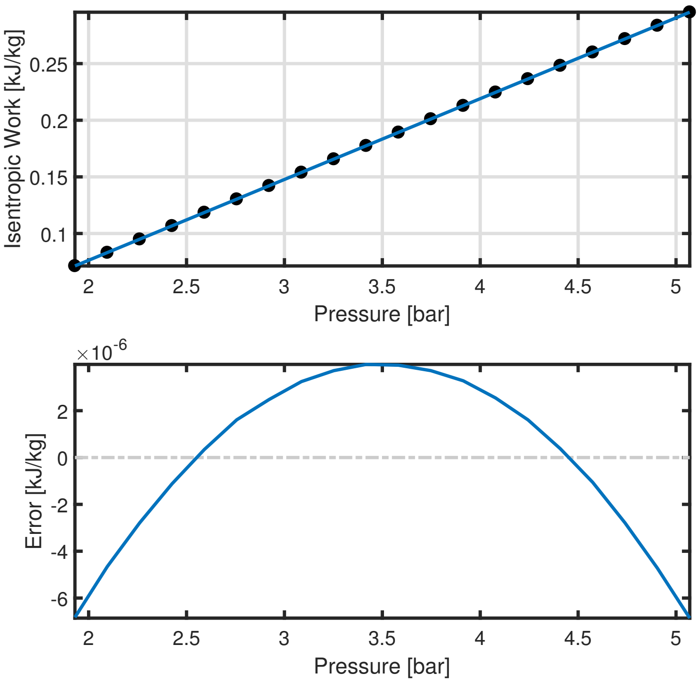
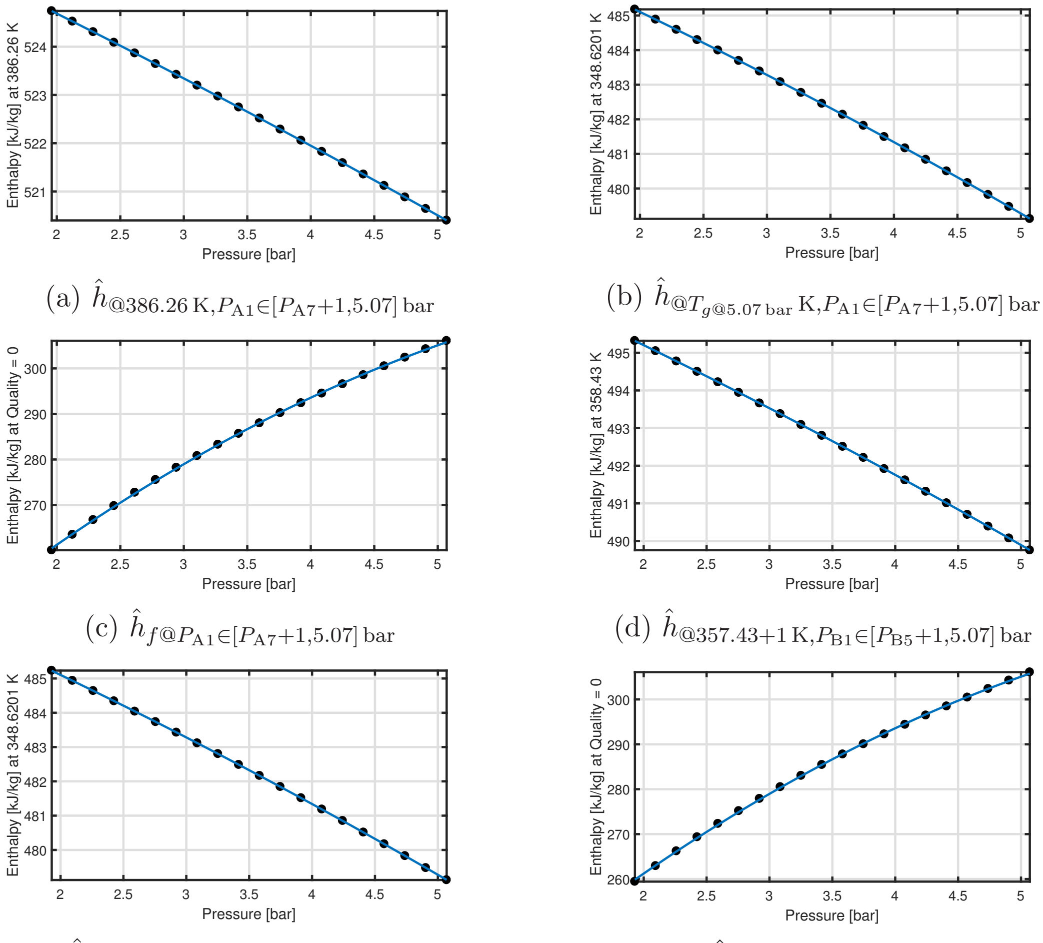


Figure 4.13: The regression fit and error plot of the USGP LP pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

of the plant need to be enforced as follows:

$$\begin{aligned}
h_{\mathrm{A4}} &\leq \hat{h}_{@386.26\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}},\\
h_{\mathrm{A4}} &\geq \hat{h}_{@T_{g@5.07\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}},\\
h_{\mathrm{A3}} &= \hat{h}_{f@P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}},\\
T_{\mathrm{A7}} &= \hat{T}_{\mathrm{A6}} - \Delta T_{\mathrm{recu,A6-A7}},\\
h_{\mathrm{B3}} &\leq \hat{h}_{@357.43+1\,\mathrm{K},P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}},\\
h_{\mathrm{B3}} &\geq \hat{h}_{@T_{g@5.07\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}},\\
h_{\mathrm{B2}} &= \hat{h}_{f@P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}},\\
h_{\mathrm{A1}} &\leq h_{\mathrm{A2}},\\
h_{\mathrm{A6}} &\leq h_{\mathrm{A5}},
\end{aligned} \tag{4.16}$$

where the terms on the right-hand side of (4.16), except the last two constraints, were approximated as they include thermodynamic routines that cannot be used with the white-box solvers, as shown in Figure 4.14 and Figure 4.15e.



(a) $\hat{h}_{@386.26\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}}$

(b) $\hat{h}_{@T_{g@5.07\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}}$

(c) $\hat{h}_{f@P_{\mathrm{A1}}\in[P_{\mathrm{A7}}+1,5.07]\,\mathrm{bar}}$

(d) $\hat{h}_{@357.43+1\,\mathrm{K},P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}}$

(e) $\hat{h}_{@T_{g@5.07\,\mathrm{bar}}\,\mathrm{K},P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}}$

(f) $\hat{h}_{f@P_{\mathrm{B1}}\in[P_{\mathrm{B5}}+1,5.07]\,\mathrm{bar}}$

Figure 4.14: The thermodynamic regressions for the operational constraints in (4.16). The black dots are the REFPROP thermodynamic calculations.

In addition, the following constraints were introduced for the brine system:

$$\begin{aligned}
&h_{\mathrm{BR6}} \leq h_{\mathrm{BR4}},\\
&h_{\mathrm{BR7}} \leq h_{\mathrm{BR5}},\\
&h_{\mathrm{BR5}} = h_{\mathrm{BR4}},\\
&h_{\mathrm{BR4}} = h_{\mathrm{BR3}},\\
&h_{\mathrm{BR3}} \leq h_{\mathrm{BR2}},\\
&h_{\mathrm{BR8}} = (h_{\mathrm{BR6}} + h_{\mathrm{BR7}})/2,
\end{aligned} \tag{4.17}$$

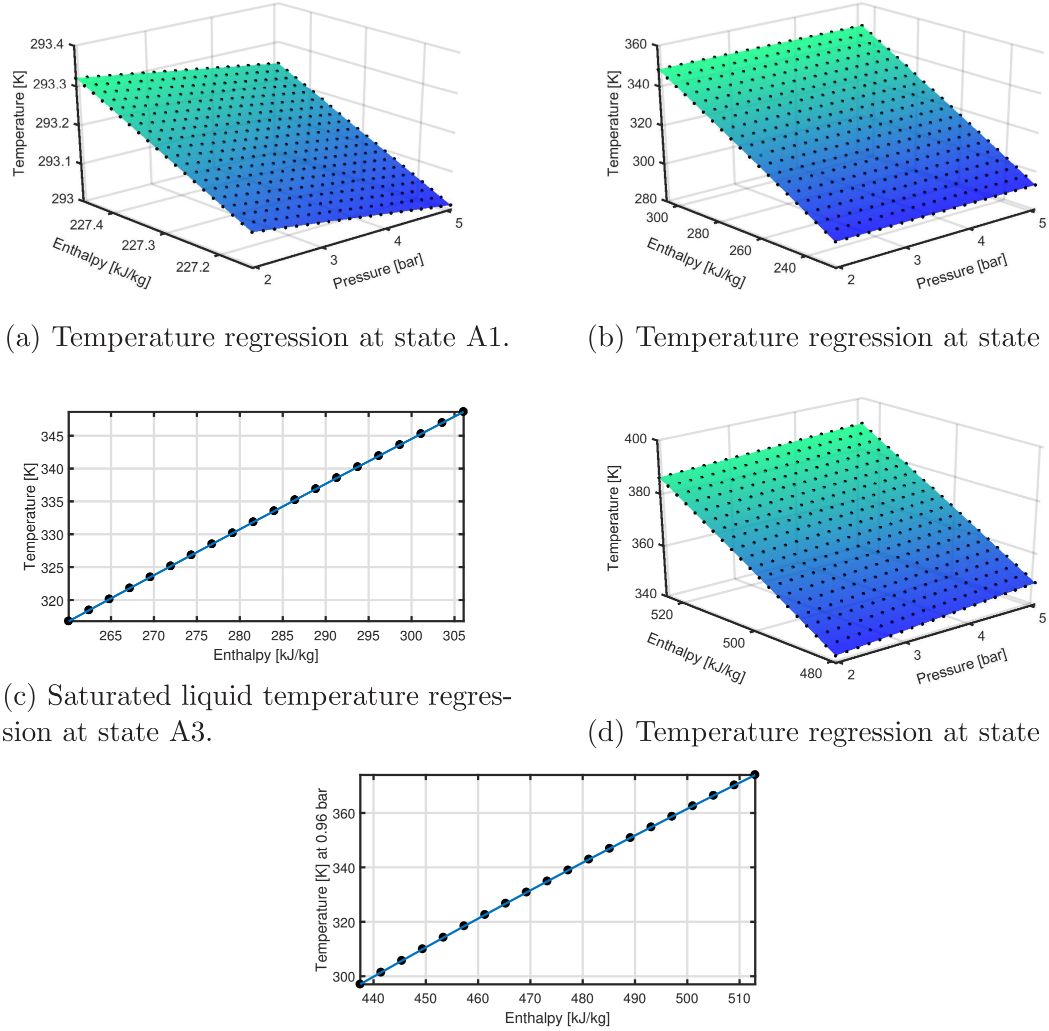
which did not require any thermodynamic regressions. Furthermore, in order to ensure that temperature difference at the inlet and outlet of each heat exchanger is greater than (or equal to) 1 K to offset for the thermodynamic approximations, and also to ensure that the heat is only transferred from hot fluid to the cold fluid, the temperatures of the HP heat exchangers were constrained as follows:

$$\begin{aligned}
\hat{T}_{\mathrm{A6}} - T_{\mathrm{CW3}} &\geq 1,\\
\hat{T}_{\mathrm{A5}} - \hat{T}_{\mathrm{A2}} &\geq 1,\\
\hat{T}_{\mathrm{A6}} - \hat{T}_{\mathrm{A1}} &\geq 1,\\
\hat{T}_{\mathrm{BR3}} - \hat{T}_{\mathrm{A3}} &\geq 1,\\
\hat{T}_{\mathrm{BR7}} - \hat{T}_{\mathrm{A2}} &\geq 1,\\
T_{\mathrm{BR1}} - \hat{T}_{\mathrm{A4}} &\geq 1,\\
\hat{T}_{\mathrm{BR2}} - \hat{T}_{\mathrm{A3}} &\geq 1,
\end{aligned} \tag{4.18}$$

where the HP cycle temperature regressions are shown in Figure 4.15 . Similarly, the temperatures at the inlet and outlet of the LP heat exchangers were constrained as follows:

$$\begin{aligned}
\hat{T}_{\mathrm{B4}} - T_{\mathrm{CW4}} &\geq 1,\\
\hat{T}_{\mathrm{BR3}} - \hat{T}_{\mathrm{B2}} &\geq 1,\\
\hat{T}_{\mathrm{BR6}} - \hat{T}_{\mathrm{B1}} &\geq 1,\\
\hat{T}_{\mathrm{BR2}} - \hat{T}_{\mathrm{B3}} &\geq 1,
\end{aligned} \tag{4.19}$$

where the LP cycle and geothermal brine temperature regressions are shown in Figure 4.16. Lastly, all the decision variables were bounded to reduce the search region, which were based on the assumptions stated above about the plant. As a



(a) Temperature regression at state A1.

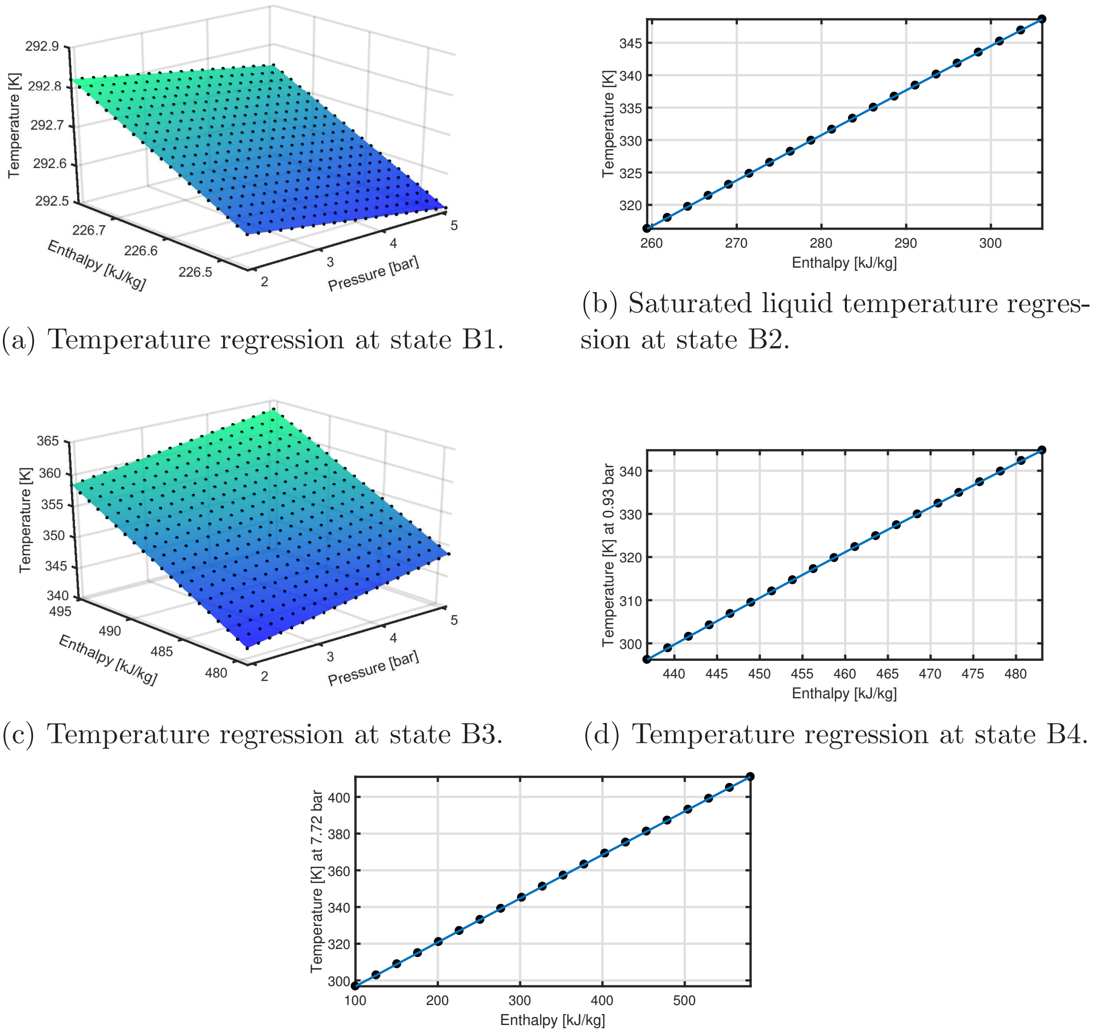
(b) Temperature regression at state A2.

(c) Saturated liquid temperature regression at state A3.

(d) Temperature regression at state A4.

(e) Temperature regression at state A5 and A6.

Figure 4.15: The temperature regressions for the USGP HP cycle. The black dots are the REFPROP thermodynamic calculations.



(a) Temperature regression at state B1.

(b) Saturated liquid temperature regression at state B2.

(c) Temperature regression at state B3.

(d) Temperature regression at state B4.

(e) Temperature regression for the brine.

Figure 4.16: The temperature regressions for the USGP LP cycle and geothermal brine. The black dots are the REFPROP thermodynamic calculations.

result, bounds were

$$\begin{aligned}
1.96 \leq P_{\mathrm{A1}} \leq 5.07,\\
227.14 \leq h_{\mathrm{A1}} \leq 227.46,\\
227.14 \leq h_{\mathrm{A2}} \leq 305.76,\\
260.49 \leq h_{\mathrm{A3}} \leq 305.76,\\
479.12 \leq h_{\mathrm{A4}} \leq 524.74,\\
437.47 \leq h_{\mathrm{A5}} \leq 512.99,\\
437.47 \leq h_{\mathrm{A6}} \leq 512.99,\\
10.00 \leq \dot{m}_{\mathrm{A}} \leq 200.00,\\
2825.1 \leq \dot{W}_{\mathrm{turbA}} \leq 5246.7,\\
40.78 \leq \dot{W}_{\mathrm{pumpA}} \leq 75.73
\end{aligned} \tag{4.20}$$

for the HP cycle;

$$\begin{aligned}
1.93 \leq P_{\mathrm{B1}} \leq 5.07,\\
226.46 \leq h_{\mathrm{B1}} \leq 226.78,\\
259.86 \leq h_{\mathrm{B2}} \leq 305.75,\\
479.12 \leq h_{\mathrm{B3}} \leq 495.32,\\
436.82 \leq h_{\mathrm{B4}} \leq 483.08,\\
100.00 \leq \dot{m}_{\mathrm{B}} \leq 800.00,\\
6212.1 \leq \dot{W}_{\mathrm{turbB}} \leq 11536.8,\\
95.69 \leq \dot{W}_{\mathrm{pumpB}} \leq 177.71
\end{aligned} \tag{4.21}$$

for the LP cycle; and

$$\begin{aligned}
100.00 \leq h_{\mathrm{BR}} \leq 579.93,\\
100.00 \leq \dot{m}_{\mathrm{CWA}} \leq 3000.0,\\
1000.0 \leq \dot{m}_{\mathrm{CWB}} \leq 4000.0,\\
8900.8 \leq \dot{W}_{\mathrm{net}} \leq 16530.0
\end{aligned} \tag{4.22}$$

for the brine, cooling water, and net power. Note that some of the bounds will change with respect to the working fluid due to the differences in their thermodynamic properties.

4.3.2 USGP Sequential-Modular Model Validation

Similar to the DOEP, the USGP SM model was constructed using the JSteam MATLAB Interface modelling framework and adhered to the same plant specifications and assumptions as the algebraic EO model in Section 4.3.1. The SM model was built using the pre-built unit operation functions in JSteam that were arranged in a sequential order of the plant process, where the output of one function is the input argument of the proceeding function.

The SM model was solved by substituting the nominal design state-point properties in Table 4.4, namely, T_{A3}, T_{A4}, T_{A6}, T_{B2}, T_{B3}, m_{A}, P_{B1}, and P_{A1} (where P_{A1} was changed to 5.07 bar, and the saturated temperature at state A3 was recalculated using the new pressure level), into the model and using MATLAB's nonlinear solver `fsolve` to converge to a feasible operating point. Due to rounding errors, some of the temperature values from Table 4.4 (i.e., the original reference) do not coincide with the right saturation point. Therefore, the temperature at state B3 and B2 were recalculated to ensure that the states were at the correct saturation point for all the simulations, which also applied to when the model is subjected to different working fluids.

Given that the value of P_{A1} was changed to 5.07 bar in order to address the issue that the working fluid temperature at the HP preheater outlet was higher than the input heat source, the USGP SM model cannot be validated with the original nominal design state-point properties. However, the Ts diagram of the plant can be plotted and analysed to check if the SM model violated any laws of thermodynamics. Figure 4.17 shows the Ts diagram of the solved SM model with the default isobutane working fluid. From the Ts diagram, the energy conversion process of the USGP SM model appears to follow the general operation of an ORC system and does not show any anomalies. Therefore, the plant can still be used for this research despite the modification made to the original nominal design state-point properties.

The solved SM model using the original working fluid, isopentane (R601a), was entered into the PFD in Figure 4.9, and the corresponding plant's power analysis, mass flows, and specific rotating machinery cost are shown in Table 4.6.

Given that the SM model was solved using the original operating point of the plant, this solved SM model will be used as the base case for this research. The same will also apply when the model is subjected to different working fluids.

For additional validation, the heat exchange process of between the geothermal brine/cooling water and the working fluid of the plant can be plotted to ensure that there are no thermodynamic violations. As shown in Figure 4.18, there are no

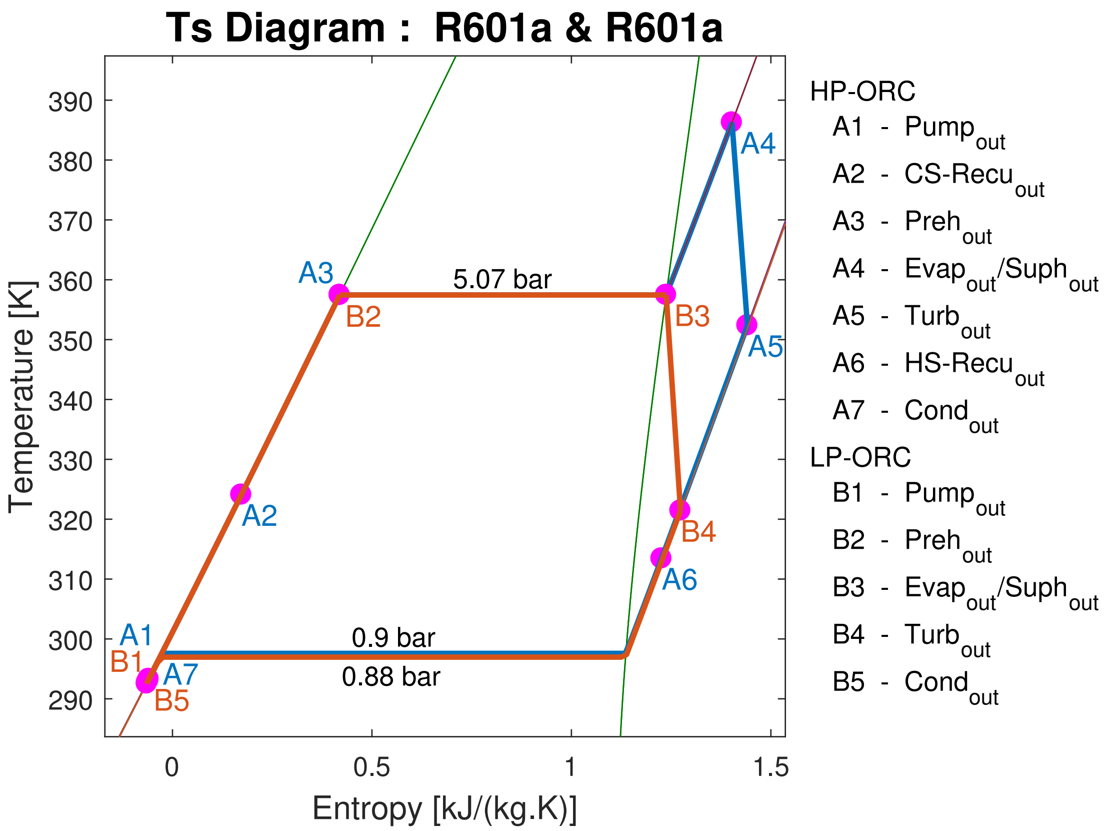


Figure 4.17: The Ts diagram of the USGeo Plant's sequential-modular model.

Table 4.6: The USGeo Plant's power analysis, mass flows, and specific rotating machinery cost of the JSteam SM Model.

Plant **Parameter**	**SM Model** **(Base case)**
LP Preheater Duty, [kW]	20904
LP Evap/Suph Duty, [kW]	38999
LP Turbine Power, [kW]	6857.4
LP Condenser Duty, [kW]	53174
LP Pump Power, [kW]	128.18
LP Working Fluid, [kg/s]	133.31
LP Cooling Water, [kg/s]	1968.6
HP Preheater Duty, [kW]	11705
HP Evap/Suph Duty, [kW]	49300
HP Turbine Power, [kW]	7711.5
HP Recuperator Power, [kW]	9986
HP Condenser Duty, [kW]	53427
HP Pump Power, [kW]	133.53
HP Working Fluid, [kg/s]	139.32
HP Cooling Water, [kg/s]	1835.8
Total Condenser Duty, [kW]	106601
Total Input Duty, [kW]	120909
Net Power, [kW]	14307
Thermal Efficiency, [%]	11.83
Specific Machinery Cost, [k$/MW]	215.66

anomalies that can be observed in the heat exchange process. Noticeably, some of

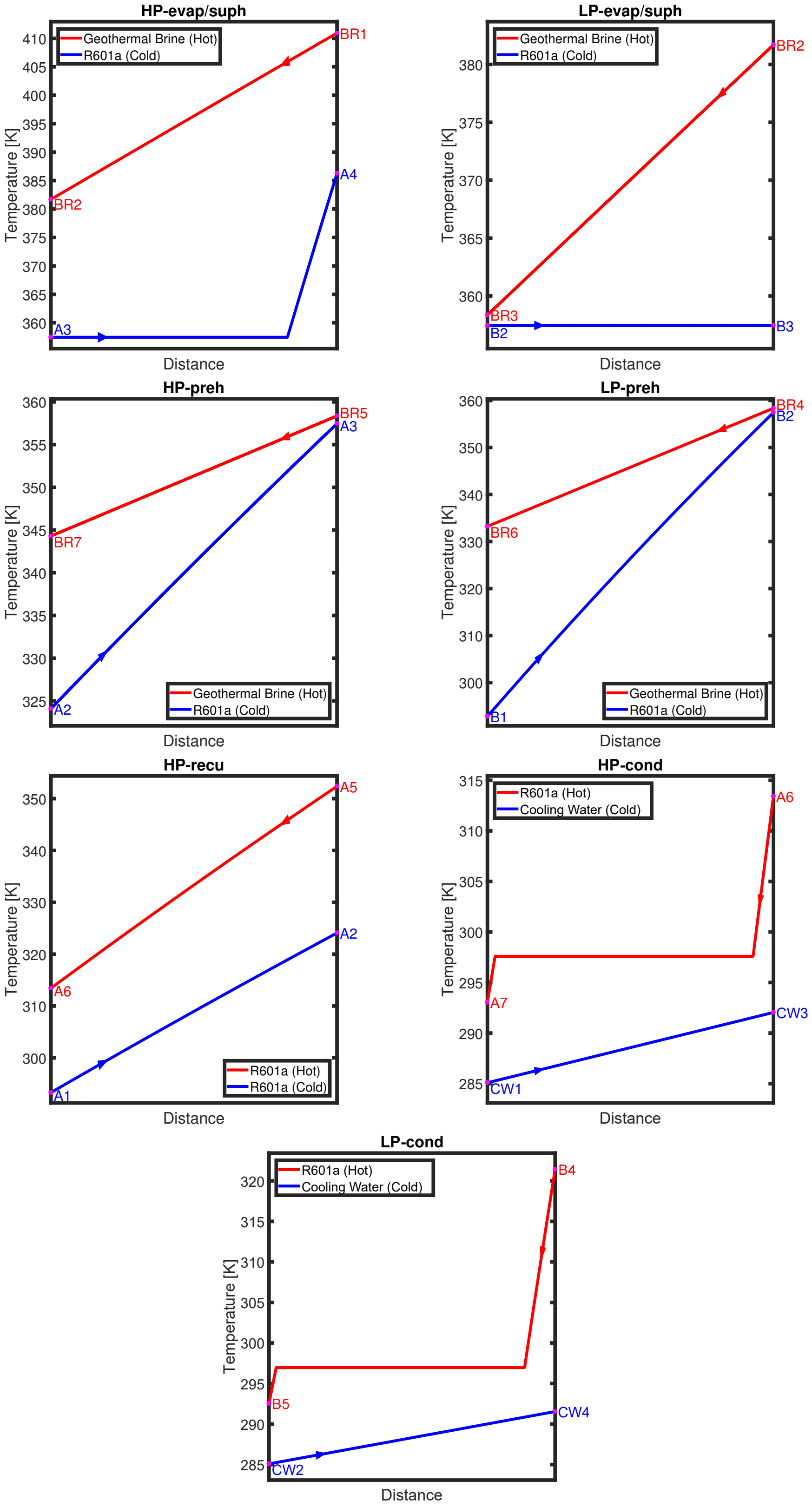


Figure 4.18: Diagrams showing the heat exchange between the hot medium and cold medium. The states refer to Figure 4.9. The x-axis represents the path of the fluid flow in the heat exchanger.

the pinch-point temperature differences shown in Figure 4.18 are very small, but theoretically they do not violate the laws of thermodynamics and are satisfactory for the purpose of this research to demonstrate the proposed modelling framework and the performance of the solvers.

4.4 Magmamax Binary Power Plant

Similar to the dual-pressure ORC system, it is also possible to achieve a good temperature match between the geofluid and the working fluid, and reduce the thermodynamic losses in the exchangers by using a dual-fluid binary cycle. One particular binary cycle power plant that employed this configuration was the Magmamax Binary Power Plant (MBPP), located at East Mesa, Imperial Valley, in California USA. The dual-fluid binary cycle is a complex system that utilizes two different working fluids in a coupled ORC system. The MBPP consisted of a high-pressure (HP) supercritical cycle and a low-pressure (LP) subcritical cycle. The two ORC systems were coupled by a heat exchanger that transfers the heat from the HP turbine exhaust gas to the LP cycle.

The MBPP consisted of three main turbines that were used to generate electricity and one auxiliary turbine for the feed pump. Given the limited information about the plant, the auxiliary feed-pump-turbine circuitry was omitted, and the two HP generator turbines were simplified into one turbine. Therefore, all the HP working fluid expands through only one primary HP turbine. Once the working fluid leaves the HP turbine, it goes through the recuperator system that was separated into two heat exchangers for this research in order to set the necessary constraints discussed in Section 3.2.2.1. After the recuperator system, the working fluid is condensed before it is fed through a series of heat exchangers via a pump. While for the LP cycle, the working fluid goes through a condenser and is fed through a set of heat exchangers via a pump after existing from the LP turbine. Figure 4.19 shows the process flow diagram of the MBPP that was modelled for this research. The following subsections will detail the algebraic EO model of the MBPP and the validation of the SM model.

4.4.1 MBPP Algebraic Equation-Oriented Model

The Magmamax Binary Power Plant model was developed based on the design specifications obtained from [3, 7], as shown in Table 4.7. Given the limited information about the plant, there were some assumptions that were made about the plant in order to model and optimize the plant under various working fluids. It is assumed that there are no changes to the pressure across the heat exchangers, the pump inlet

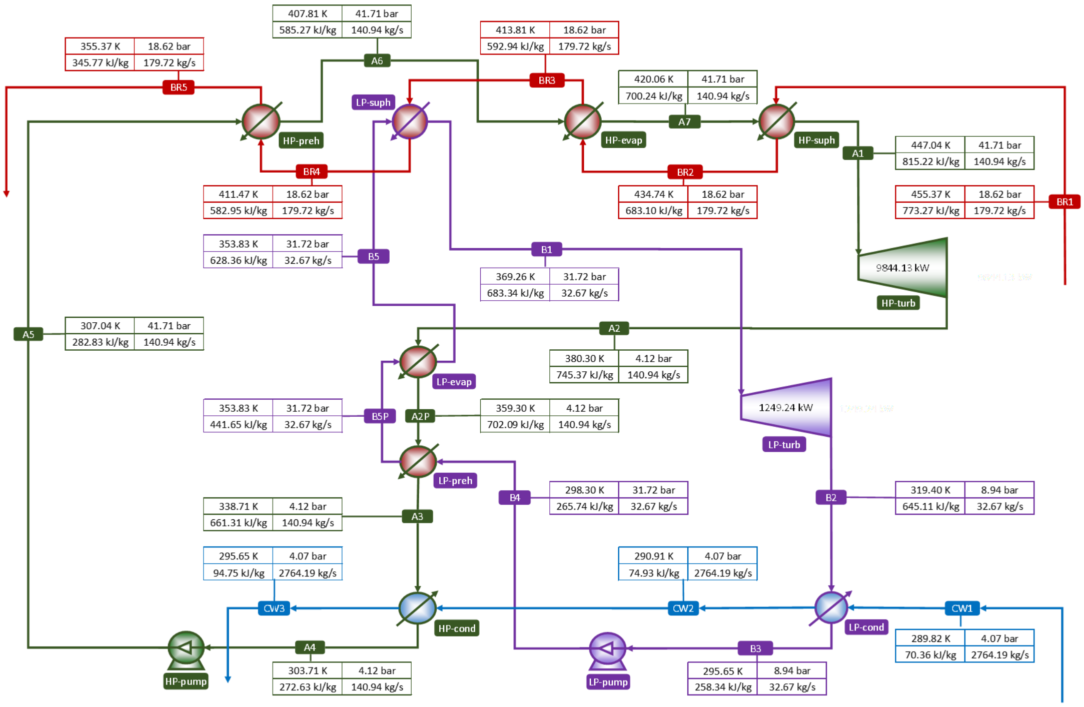


Figure 4.19: The process flow diagram of the Magmamax Binary Power Plant with R600a for the HP cycle and R290 for the LP cycle. The state-point properties were obtained from the solved SM model, see Section 4.4.2.

Table 4.7: The MBPP design specifications obtained from [3, 7]. Refer to Figure 4.19 for the state labels.

State	Temperature	Pressure	Entropy	Enthalpy	Mass Flow
label	[K]	[bar]	[kJ/kgK]	[kJ/kg]	[kg/s]
BR1	455.37	18.62	2.16	773.41	179.72
BR2	-	-	-	-	-
BR3	-	-	-	-	-
BR4	-	-	-	-	-
BR5	355.37	-	-	-	-
A1	447.04	34.47	2.71	835.77	129.90
A2	380.37	4.14	2.75	745.48	-
A2P	-	-	-	-	-
A3	338.71	-	-	-	-
A4	303.71	-	-	-	-
A5	307.04	41.71	1.26	282.82	-
A6	408.15	-	-	-	-
A7	-	-	-	-	-
B1	369.26	31.72	2.43	683.36	34.52
B2	319.26	8.89	2.50	644.97	-
B3	295.65	-	-	-	-
B4	298.15	-	-	-	-
B5P	-	-	-	-	-
B5	353.71	-	-	-	-
CW1	289.82	4.07	0.25	70.35	1577.26
CW2	291.76	-	-	-	-
CW3	299.54	-	-	-	-

pressures are equal to the saturated pressures at the condenser outlet temperatures, the enthalpy at A7 is equal to the average enthalpy of A1 and A6, the temperature at A6 is equal to the critical temperature, and the temperature difference $\Delta T_{\mathrm{recu,A3-A4}}$ between T_{A3} and T_{A4} is 35 K as per [3]. Note that $\Delta T_{\mathrm{recu,A3-A4}}$ will be changed to 25 K for the EO model to allow for a wider degree of freedom for the optimizer to find a better solution. These assumptions resulted in the following modified design specifications in Table 4.8, which was used as the basis of the algebraic EO model and for the validation of the SM model.

Table 4.8: The modified MBPP design specifications of Table 4.7. Refer to Figure 4.19 for the state labels.

State label	Temperature [K]	Pressure [bar]	Entropy [kJ/kgK]	Enthalpy [kJ/kg]	Mass Flow [kg/s]
BR1	455.37	18.62	2.16	773.41	179.72
BR2	-	18.62	-	-	179.72
BR3	-	18.62	-	-	179.72
BR4	-	18.62	-	-	179.72
BR5	355.37	18.62	1.10	345.77	179.72
A1	447.04	41.71	2.65	815.22	129.90
A2	380.37	4.12	2.75	745.51	129.90
A2P	-	4.12	-	-	129.90
A3	338.71	4.12	2.52	661.31	129.90
A4	303.71	4.12	1.25	272.63	129.90
A5	307.04	41.71	1.26	282.82	129.90
A6	407.81	41.71	2.10	585.27	129.90
A7	420.06	41.71	2.38	700.24	129.90
B1	369.26	31.72	2.43	683.34	34.52
B2	319.26	8.94	2.50	644.83	34.52
B3	295.65	8.94	1.20	258.34	34.52
B4	298.15	31.72	1.21	265.35	34.52
B5P	353.83	31.72	1.75	441.65	34.52
B5	353.83	31.72	2.28	628.36	34.52
CW1	289.82	4.07	0.25	70.35	1577.26
CW2	291.76	4.07	0.28	78.48	1577.26
CW3	299.54	4.07	0.39	111.01	1577.26

With the intention of keeping the plant model parameters as close to the original plant design specifications when optimizing under various operating conditions, the constants listed in Table 4.9 were used in both the algebraic EO model and the SM model. Note that the outlet temperature of the cooling water was decreased to the outlet temperature of the LP condenser (i.e., T_{B3}) to ensure that the thermodynamic violation discussed in the Section 3.2.2.1 does not occur in the condensers.

Since, the original working fluids of the plant were isobutane (for the HP cycle) and propane (for the LP cycle), they will be used for the purpose of presenting the development of the MBPP model in this section.

Table 4.9: The Magmamax Binary Power Plant constant parameters.

Plant Constants	**Value**	**Unit**
$\dot{m}_{\mathrm{BR}}$	179.72	kg/s
T_{BR1}	455.37	K
T_{BR5}	355.37	K
P_{BR}	18.62	bar
η_{turbA}	63.97	%
η_{turbB}	61.88	%
η_{pumpA}	67.38	%
η_{pumpB}	61.76	%
T_{A4}	303.71	K
P_{A4}	$P_{f@\mathrm{T_{A4}}}$	bar
T_{B3}	295.65	K
P_{B3}	$P_{f@\mathrm{T_{B3}}}$	bar
T_{CW1}	289.82	K
T_{CW3}	T_{B3}	K
P_{CW}	4.07	bar

The algebraic EO model of the MBPP consists of the energy balance equations that were derived from the process flow diagram in Figure 4.19, which resulted in the following equations:

$$\begin{aligned}
\dot{m}_{\mathrm{A}}h_{\mathrm{A1}} - \dot{W}_{\mathrm{turbA}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A2}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A2}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A2P}} + \dot{m}_{\mathrm{B}}h_{\mathrm{B5P}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B5}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A2P}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A3}} + \dot{m}_{\mathrm{B}}h_{\mathrm{B4}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B5P}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A3}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A4}} + \dot{m}_{\mathrm{CW}}h_{\mathrm{CW2}} - \dot{m}_{\mathrm{CW}}h_{\mathrm{CW3}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A4}} + \dot{W}_{\mathrm{pumpA}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A5}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A5}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A6}} + \dot{m}_{\mathrm{BR}}h_{\mathrm{BR4}} - \dot{m}_{\mathrm{BR}}h_{\mathrm{BR5}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A6}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A7}} + \dot{m}_{\mathrm{BR}}h_{\mathrm{BR2}} - \dot{m}_{\mathrm{BR}}h_{\mathrm{BR3}} &= 0,\\
\dot{m}_{\mathrm{A}}h_{\mathrm{A7}} - \dot{m}_{\mathrm{A}}h_{\mathrm{A1}} + \dot{m}_{\mathrm{BR}}h_{\mathrm{BR1}} - \dot{m}_{\mathrm{BR}}h_{\mathrm{BR2}} &= 0,\\
\dot{m}_{\mathrm{B}}h_{\mathrm{B1}} - \dot{W}_{\mathrm{turbB}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B2}} &= 0,\\
\dot{m}_{\mathrm{B}}h_{\mathrm{B2}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B3}} + \dot{m}_{\mathrm{CW}}h_{\mathrm{CW1}} - \dot{m}_{\mathrm{CW}}h_{\mathrm{CW2}} &= 0,\\
\dot{m}_{\mathrm{B}}h_{\mathrm{B3}} + \dot{W}_{\mathrm{pumpB}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B4}} &= 0,\\
\dot{m}_{\mathrm{B}}h_{\mathrm{B5}} - \dot{m}_{\mathrm{B}}h_{\mathrm{B1}} + \dot{m}_{\mathrm{BR}}h_{\mathrm{BR3}} - \dot{m}_{\mathrm{BR}}h_{\mathrm{BR4}} &= 0,
\end{aligned} \tag{4.23}$$

where $\dot{m}$ is the mass flow rate, h is the enthalpy, $\dot{W}$ is the mechanical work. In order for the model to be expressed algebraically, the turbine and pump work terms need to be approximated as a function of enthalpy and/or pressure. Following the assumptions made above and keeping in line with the design specifications, the isentropic work of the HP was calculate and approximated at $T_{\mathrm{A1}} \in [T_{\mathrm{crit}} + 30, 447.04]\,\mathrm{K}$, $P_{\mathrm{A1}} = P_{\mathrm{A5}} \in [P_{\mathrm{crit}} + 2, 41.71]\,\mathrm{bar}$, and $P_{\mathrm{A2}} = P_{\mathrm{A4}}$, as shown in Figure 4.20. Since the inlet temperature term is not explicitly used in (4.23), the corresponding inlet enthalpy values were used for the approximation. This resulted in the following expressions for the HP turbine work:

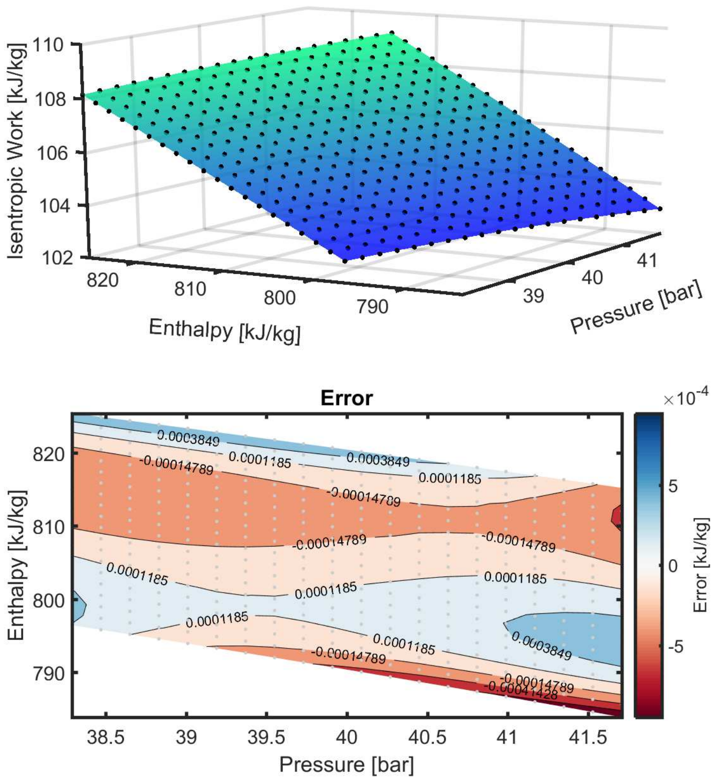


Figure 4.20: The regression fit and error plot of the MBPP HP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

$$\begin{aligned}
\Delta\hat{h}_{\text{is,turbA}} =& 0.000365P_{\text{A1}}^3 - 0.0452P_{\text{A1}}^2 + 2.59\times10^{-5}h_{\text{A1}}^2 - \\
& 1.75\times10^{-5}P_{\text{A1}}^2h_{\text{A1}} - 5\times10^{-6}P_{\text{A1}}h_{\text{A1}}^2 + \\
& 0.0122P_{\text{A1}}h_{\text{A1}} - 2.79P_{\text{A1}} + 0.0104h_{\text{A1}}, \\
\dot{W}_{\text{turbA}} =& \eta_{\text{turbA}}\dot{m}_{\text{A}}\Delta\hat{h}_{\text{is,turbA}},
\end{aligned} \tag{4.24}$$

where the isentropic work $\hat{h}_{\text{is,turbA}}$ was approximated using a cubic polynomial surface. Similarly, the isentropic work of the LP turbine was calculated and approximated at $T_{\text{B1}} \in [T_{g@31.72\,\text{bar}}, 369.26]\,\text{K}$, $P_{\text{B1}} = P_{\text{B4}} \in [P_{\text{B3}} + 2, 31.72]\,\text{bar}$, and $P_{\text{B2}} = P_{\text{B3}}$, as shown in Figure 4.21. This resulted in the following expressions for the LP turbine:

$$\begin{aligned}
\Delta\hat{h}_{\text{is,turbB}} =& 0.00311P_{\text{B1}}^3 - 0.259P_{\text{B1}}^2 - 2.71\times10^{-5}P_{\text{B1}}^2h_{\text{B1}} + \\
& 0.00836P_{\text{B1}}h_{\text{B1}} + 3.98P_{\text{B1}} - 0.0605h_{\text{B1}} - 24.1, \\
\dot{W}_{\text{turbB}} =& \eta_{\text{turbB}}\dot{m}_{\text{B}}\Delta\hat{h}_{\text{is,turbB}},
\end{aligned} \tag{4.25}$$

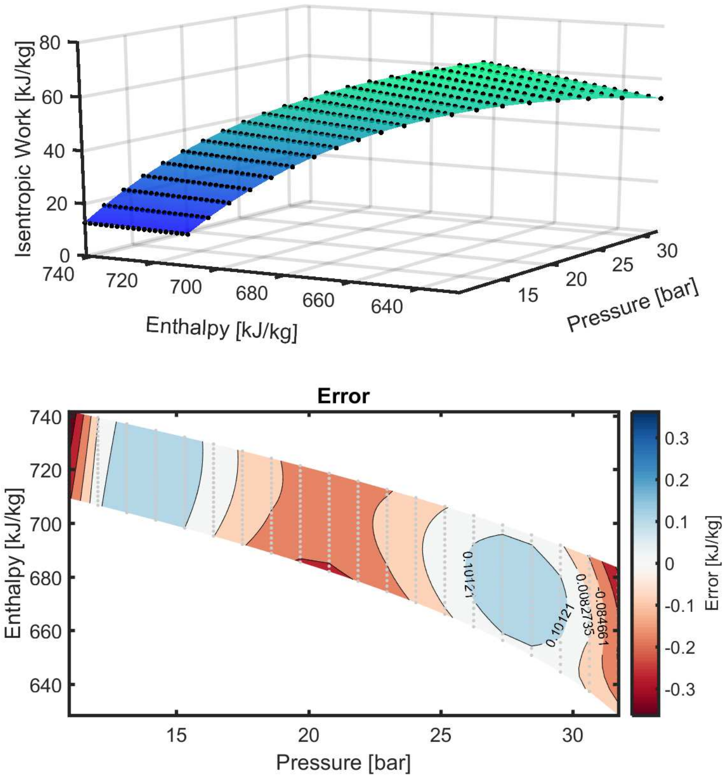


Figure 4.21: The regression fit and error plot of the MBPP LP turbine isentropic work as a function of the inlet pressure and enthalpy. The black dots (top plot) and the grey dots (bottom plot) are the REFPROP thermodynamic calculations.

where $\Delta\hat{h}_{\text{is,turbB}}$ is the approximated isentropic work.

The HP and LP pump isentropic work can be approximated as a function of the outlet pressure using the same regression procedure that was implemented on the turbines. The isentropic work of the HP pump was calculated at P_{A4} and $P_{\text{A1}} = P_{\text{A5}} \in [P_{\text{crit}} + 2, 41.71]$ bar, as shown in Figure 4.22. This corresponds to the following expressions for the approximated HP pump work:

$$\begin{aligned} \Delta\hat{h}_{\text{is,pumpA}} &= 0.182 P_{\text{A1}} - 0.718, \\ \dot{W}_{\text{pumpA}} &= \dot{m}_{\text{A}} \Delta\hat{h}_{\text{is,pumpA}} / \eta_{\text{pumpA}}, \end{aligned} \tag{4.26}$$

where the approximated isentropic work $\hat{h}_{\text{is,pumpA}}$ was fitted using a linear curve. Similarly, the isentropic work of the LP pump was calculated at P_{B3} and $P_{\text{B1}} = P_{\text{B4}} \in [P_{\text{B3}} + 2, 31.72]$ bar, as shown in Figure 4.23. This resulted in the following expressions for the LP pump work:

$$\begin{aligned} \Delta\hat{h}_{\text{is,pumpB}} &= 0.201 P_{\text{B1}} - 1.79, \\ \dot{W}_{\text{pumpB}} &= \dot{m}_{\text{B}} \Delta\hat{h}_{\text{is,pumpB}} / \eta_{\text{pumpB}}, \end{aligned} \tag{4.27}$$

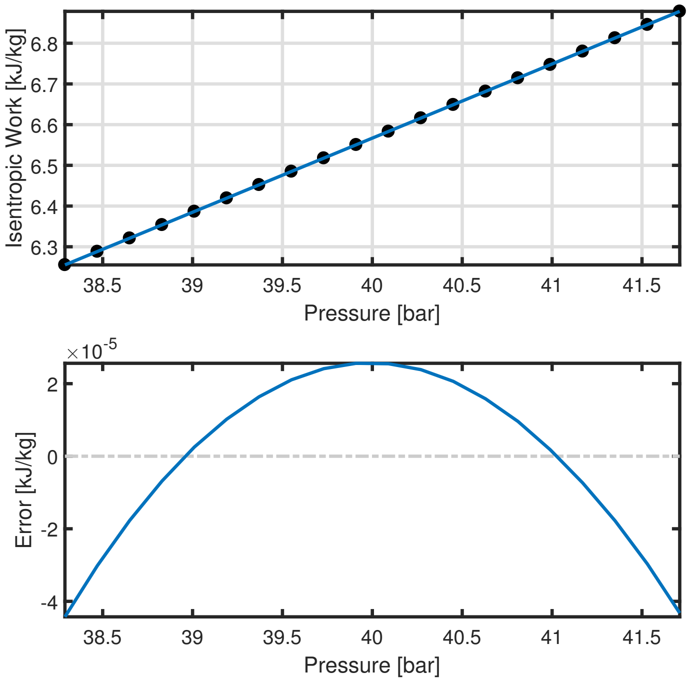


Figure 4.22: The regression fit and error plot of the MBPP HP pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

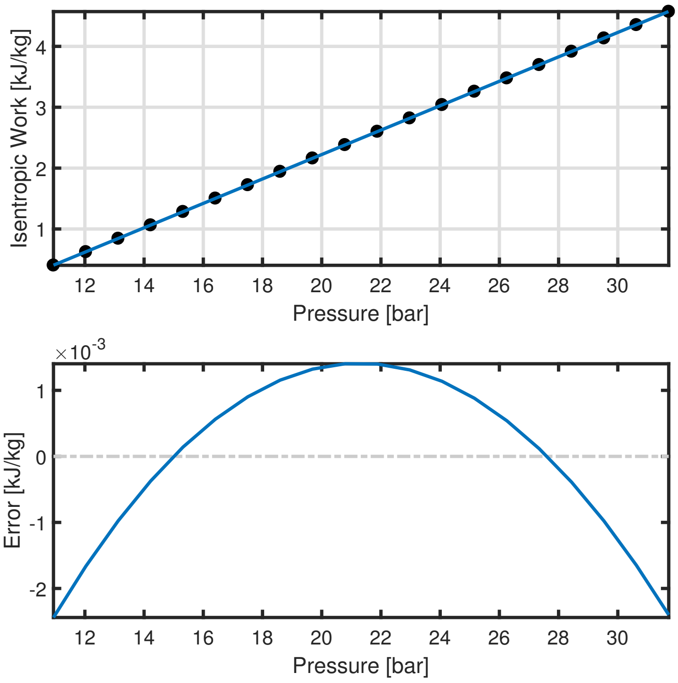


Figure 4.23: The regression fit and error plot of the MBPP LP pump isentropic work as a function of the outlet pressure. The black dots (top plot) are the REFPROP thermodynamic calculations.

where the approximated isentropic work $\hat{h}_{\text{is,pumpB}}$ was fitted using a linear curve.

In order for the MBPP model to operate as a feasible ORC system and comply with the assumptions made about the plant at the beginning of this subsection, the model was subjected to the following operational constraints:

$$\begin{aligned}
&h_{\text{A7}} = (h_{\text{A6}} + h_{\text{A1}})/2, \\
&h_{\text{A6}} = \hat{h}_{@T_{\text{crit}},\ P_{\text{A1}} \in [P_{\text{crit}}+2,\ 41.71]\,\text{bar}}, \\
&h_{\text{A1}} \leq \hat{h}_{@447.04\,\text{K},\ P_{\text{A1}} \in [P_{\text{crit}}+2,\ 41.71]\,\text{bar}}, \\
&h_{\text{A1}} \geq \hat{h}_{@T_{\text{crit}}+30\,\text{K},\ P_{\text{A1}} \in [P_{\text{crit}}+2,\ 41.71]\,\text{bar}}, \\
&T_{\text{A4}} = \hat{T}_{\text{A3}} - \Delta T_{\text{recu,A3-A4}}
\end{aligned} \tag{4.28}$$

for the HP cycle;

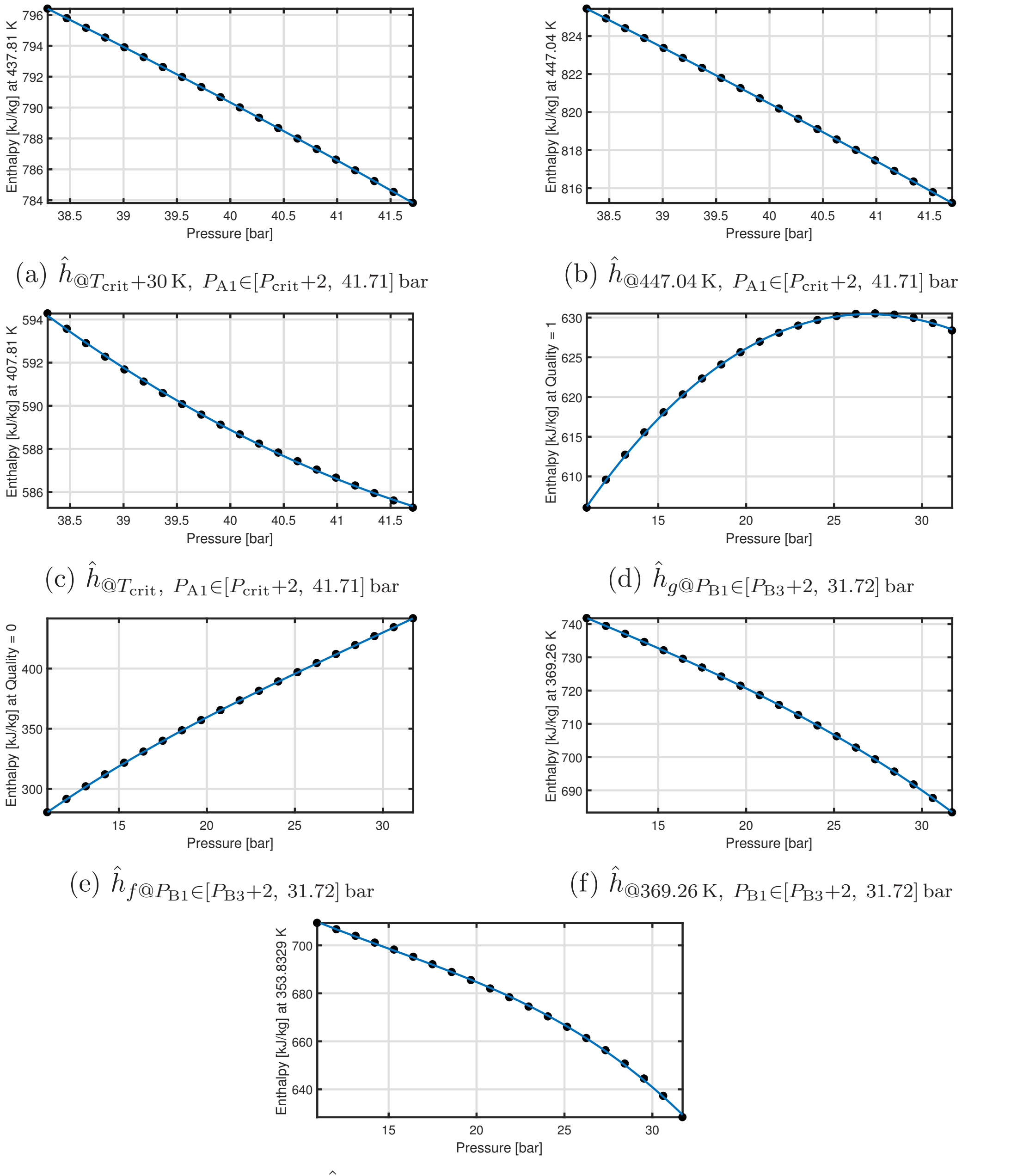
$$\begin{aligned}
h_{\text{B5}} &= \hat{h}_{g@P_{\text{B1}} \in [P_{\text{B3}}+2,\ 31.72]\,\text{bar}}, \\
h_{\text{B5P}} &= \hat{h}_{f@P_{\text{B1}} \in [P_{\text{B3}}+2,\ 31.72]\,\text{bar}}, \\
h_{\text{B1}} &\leq \hat{h}_{@369.26\,\text{K},\ P_{\text{B1}} \in [P_{\text{B3}}+2,\ 31.72]\,\text{bar}}, \\
h_{\text{B1}} &\geq \hat{h}_{@T_{g@31.72\,\text{bar}},\ P_{\text{B1}} \in [P_{\text{B3}}+2,\ 31.72]\,\text{bar}}, \\
(h_{\text{B1}} - h_{\text{B5}})\dot{m}_{\text{B}} &\geq 0.02(h_{\text{A2}} - h_{\text{A3}})\dot{m}_{\text{A}}
\end{aligned} \tag{4.29}$$

for the LP cycle; and

$$\begin{aligned}
h_{\text{BR3}} &\leq h_{\text{BR2}}, \\
h_{\text{BR4}} &\leq h_{\text{BR3}}
\end{aligned} \tag{4.30}$$

for the brine system. Note that some of the thermodynamic terms on the right-hand side of (4.28) and (4.29) were approximated and are shown in Figure 4.24 and Figure 4.25b.

In order to ensure that the heat exchangers transfer heat from the hot fluid to the cold fluid, the HP cycle temperatures at the inlet and outlet of each heat exchanger



(a) $\hat{h}_{@T_{\mathrm{crit}}+30\,\mathrm{K}},\ P_{\mathrm{A1}}\in[P_{\mathrm{crit}}+2,\ 41.71]$ bar

(b) $\hat{h}_{@447.04\,\mathrm{K}},\ P_{\mathrm{A1}}\in[P_{\mathrm{crit}}+2,\ 41.71]$ bar

(c) $\hat{h}_{@T_{\mathrm{crit}}},\ P_{\mathrm{A1}}\in[P_{\mathrm{crit}}+2,\ 41.71]$ bar

(d) $\hat{h}_{g@P_{\mathrm{B1}}\in[P_{\mathrm{B3}}+2,\ 31.72]}$ bar

(e) $\hat{h}_{f@P_{\mathrm{B1}}\in[P_{\mathrm{B3}}+2,\ 31.72]}$ bar

(f) $\hat{h}_{@369.26\,\mathrm{K}},\ P_{\mathrm{B1}}\in[P_{\mathrm{B3}}+2,\ 31.72]$ bar

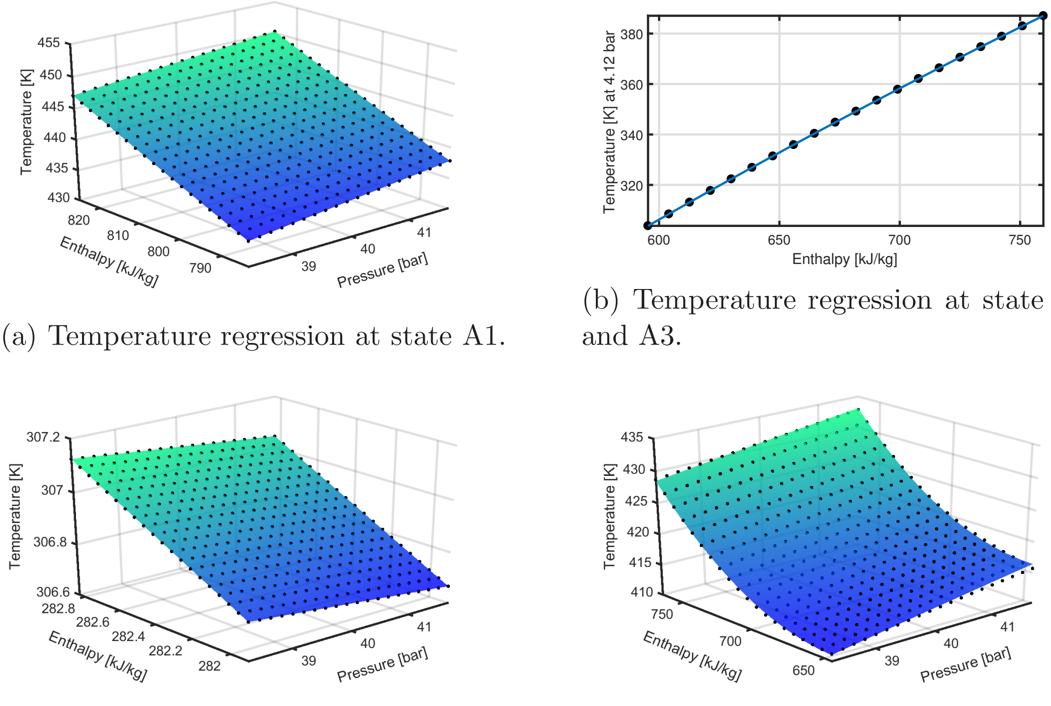
(g) $\hat{h}_{@T_{g@31.72\,\mathrm{bar}}},\ P_{\mathrm{B1}}\in[P_{\mathrm{B3}}+2,\ 31.72]$ bar

Figure 4.24: The thermodynamic regressions for the operational constraints in (4.28) and (4.29). The black dots are the REFPROP thermodynamic calculations.

were constrained as

$$\begin{aligned}
\hat{T}_{\mathrm{A3}} - T_{\mathrm{CW3}} &\geq 1, \\
T_{\mathrm{A4}} - \hat{T}_{\mathrm{CW2}} &\geq 1, \\
\hat{T}_{\mathrm{A2}} - \hat{T}_{\mathrm{B5}} &\geq 1, \\
\hat{T}_{\mathrm{A2P}} - \hat{T}_{\mathrm{B5P}} &\geq 1, \\
\hat{T}_{\mathrm{A3}} - \hat{T}_{\mathrm{B4}} &\geq 1, \\
T_{\mathrm{BR1}} - \hat{T}_{\mathrm{A1}} &\geq 1, \\
\hat{T}_{\mathrm{BR2}} - \hat{T}_{\mathrm{A7}} &\geq 1, \\
\hat{T}_{\mathrm{BR3}} - T_{\mathrm{A6}} &\geq 1, \\
\hat{T}_{\mathrm{BR4}} - T_{\mathrm{A6}} &\geq 1, \\
T_{\mathrm{BR5}} - \hat{T}_{\mathrm{A5}} &\geq 1,
\end{aligned} \tag{4.31}$$

where the temperature regressions are shown in Figure 4.25 and Figure 4.27. Similarly, the LP cycle temperatures at the inlet and outlet of each heat exchanger were constrained as



(a) Temperature regression at state A1.

(b) Temperature regression at state A2 and A3.

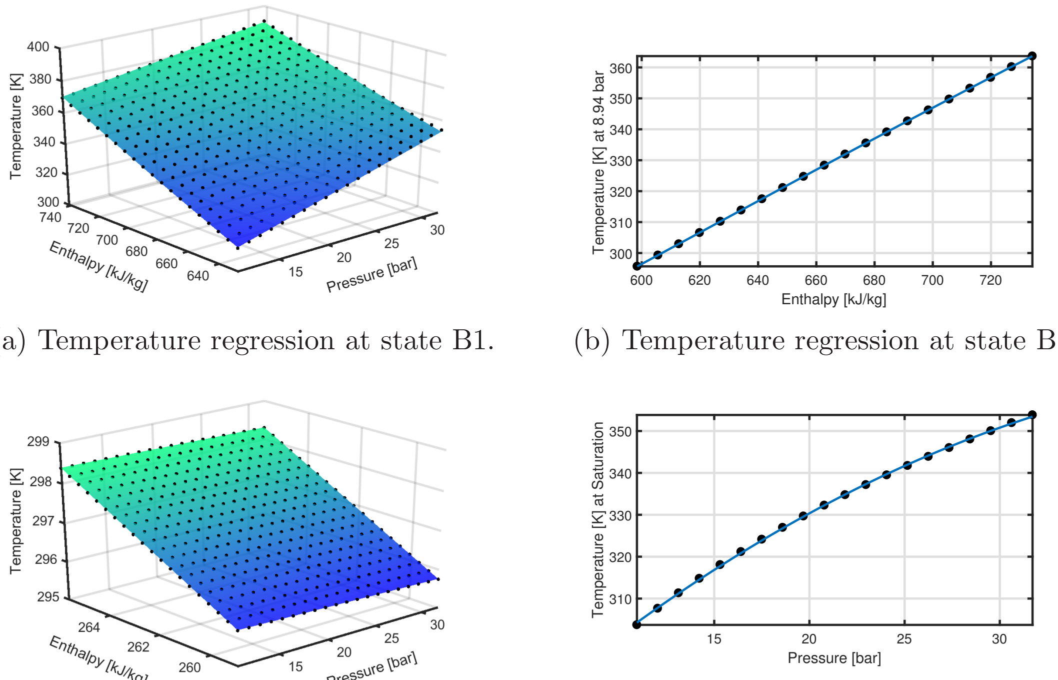
(c) Temperature regression at state A5.

(d) Temperature regression at state A7.

Figure 4.25: The temperature regressions of the MBPP HP cycle. The black dots are the REFPROP thermodynamic calculations.

$$\begin{aligned}
\hat{T}_{\mathrm{B2}} - \hat{T}_{\mathrm{CW2}} &\geq 1, \\
\hat{T}_{\mathrm{BR3}} - \hat{T}_{\mathrm{B1}} &\geq 1, \\
\hat{T}_{\mathrm{BR4}} - \hat{T}_{\mathrm{B5}} &\geq 1,
\end{aligned} \tag{4.32}$$

where the temperature regressions are shown in Figure 4.26 and Figure 4.27.



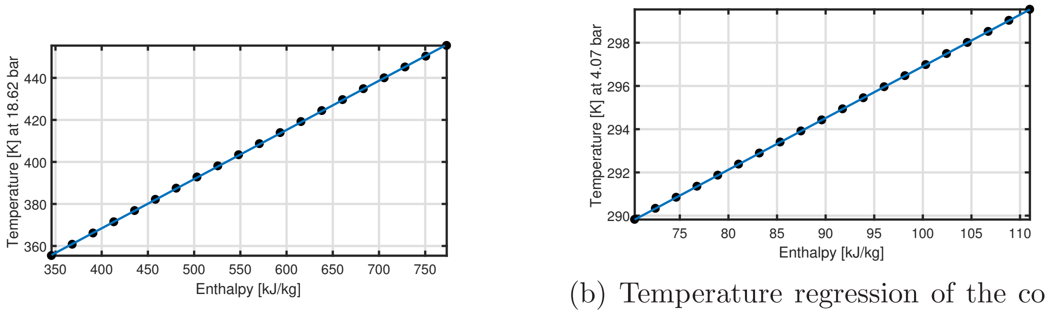
(a) Temperature regression at state B1.

(b) Temperature regression at state B2.

(c) Temperature regression at state B4.

(d) Temperature regression at state B5P and B5.

Figure 4.26: The temperature regressions of the MBPP LP cycle. The black dots are the REFPROP thermodynamic calculations.



(a) Temperature regression of the brine.

(b) Temperature regression of the cooling water.

Figure 4.27: The temperature regressions of the MBPP brine and cooling water. The black dots are the REFPROP thermodynamic calculations.

Lastly, in order to reduce the search space of the region, all the decision variables were bounded within a reasonable range as

$$\begin{aligned}
38.29 \leq P_{\mathrm{A1}} &\leq 41.71,\\
783.82 \leq h_{\mathrm{A1}} &\leq 825.44,\\
595.41 \leq h_{\mathrm{A2}} &\leq 759.67,\\
595.41 \leq h_{\mathrm{A2P}} &\leq 759.67,\\
595.41 \leq h_{\mathrm{A3}} &\leq 759.67,\\
281.91 \leq h_{\mathrm{A5}} &\leq 282.83,\\
585.27 \leq h_{\mathrm{A6}} &\leq 594.17,\\
645.31 \leq h_{\mathrm{A7}} &\leq 765.39,\\
100.00 \leq \dot{m}_{\mathrm{A}} &\leq 400.00,\\
6890.9 \leq \dot{W}_{\mathrm{turbA}} &\leq 12797,\\
1007.1 \leq \dot{W}_{\mathrm{pumpA}} &\leq 1870.36
\end{aligned} \tag{4.33}$$

for the HP cycle;

$$\begin{aligned}
10.94 \leq P_{\mathrm{B1}} &\leq 31.72,\\
628.36 \leq h_{\mathrm{B1}} &\leq 741.74,\\
598.47 \leq h_{\mathrm{B2}} &\leq 734.22,\\
259.00 \leq h_{\mathrm{B4}} &\leq 265.75,\\
280.39 \leq h_{\mathrm{B5P}} &\leq 441.65,\\
606.05 \leq h_{\mathrm{B5}} &\leq 630.51,\\
10.00 \leq \dot{m}_{\mathrm{B}} &\leq 200.00,\\
874.48 \leq \dot{W}_{\mathrm{turbB}} &\leq 1624.0,\\
169.29 \leq \dot{W}_{\mathrm{pumpB}} &\leq 314.39
\end{aligned} \tag{4.34}$$

for the LP cycle; and

$$\begin{aligned}
345.77 \leq h_{\mathrm{BR}} &\leq 773.27,\\
70.36 \leq h_{\mathrm{CW2}} &\leq 94.75,\\
2000.0 \leq \dot{m}_{\mathrm{CW}} &\leq 3000.0,\\
6588.9 \leq \dot{W}_{\mathrm{net}} &\leq 12237
\end{aligned} \tag{4.35}$$

for the brine, cooling water, and the net output power. These bounds were derived based on the plant's approximations and the general operation of an ORC system. Note that the bounds are for the default working fluids, namely, isobutane and propane, and thus they are subjected to change if a different set of working fluids is to be used.

4.4.2 MBPP Sequential-Modular Model Validation

The SM model of the MBPP was constructed using the JSteam MATLAB Interface modelling framework that complied with the same model specifications as the algebraic EO model detailed in Section 4.4.1. The same two references, namely [3, 7], that formed the modified design specifications in Table 4.8 were used to construct the SM model. The pre-built unit operation functions in JSteam were used to construct the SM model and were arranged in a sequential order of the plant process, where the output of one function is the input argument of the proceeding function.

The state-point properties of the working fluids in Table 4.8 were entered into the SM model, namely, P_{A1}, P_{B1}, T_{A1}, T_{A3}, T_{A7}, T_{B1}, T_{B5P}, and T_{B5}, and solved to a feasible operating point using MATLAB's nonlinear equation solver `fsolve`. Note that some of the temperature or pressure values taken from the reference papers do not coincide with the right saturation point when different working fluids were used. Therefore, the temperature at state B5P and B5 was recalculated to ensure that the state is at the correct liquid/vapour saturation point for all simulations. The process flow diagram in Figure 4.19 shows the state-point properties of the solved SM model using the original working fluid pair, i.e., isobutane (HP) and propane (LP).

Table 4.10 shows the comparison between the modified MBPP design specifications and the SM model. The discrepancies between the design specifications and the SM model were expected due to the simplifications and assumptions that were made about the plant. Given the limited information about the plant and the decrease in the outlet temperature of the cooling water, the errors are relatively larger than the DOEP model. However, the overall accuracy of model is satisfactory for the purpose of this research because the model still retains the characteristics of a dual-fluid ORC system as indicated by the Ts diagram in Figure 4.28 and as described in [3].

Similar to the previous two plants, this solved SM model will be used as the base case for the optimization problem given that it was solved using the original operating point of the plant. The same approach will also apply when the model is subjected to different working fluid pairs.

Furthermore, to check that there are no violations of the second law of thermodynamics as discussed in Section 3.2.2.1, the temperature profile of the heat exchangers can be plotted and examined. As shown in Figure 4.29, the heat exchangers do not show any thermodynamic violations during the heat exchanging process.

Table 4.10: A comparison between the modified MBPP design specifications (nominal design) and the JSteam SM Model.

Plant Parameter	Nominal Design	SM Model (Base case)	Error [%]
LP Preh/Evap Duty, [kW]	10938	11847	8.31
LP Superheater Duty, [kW]	1898.1	1796.5	5.36
LP Turbine Power, [kW]	1329.6	1249.2	6.04
LP Condenser Duty, [kW]	13342	12636	5.29
LP Pump Power, [kW]	241.89	241.84	0.02
LP Working Fluid, [kg/s]	34.52	32.67	5.36
HP Preheater Duty, [kW]	39288	42626	8.49
HP Evaporator Duty, [kW]	14935	16204	8.50
HP Superheater Duty, [kW]	14935	16204	8.50
HP Turbine Power, [kW]	9054.4	9844.1	8.72
HP Recuperator Power, [kW]	10938	11847	8.31
HP Condenser Duty, [kW]	50490	54781	8.50
HP Pump Power, [kW]	1324.6	1438.7	8.61
HP Working Fluid, [kg/s]	129.9	140.94	8.50
Cooling Water, [kg/s]	1577.3	2764.2	75.25
Total Condenser Duty, [kW]	63831	67418	5.62
Total Input Duty, [kW]	71056	76830	8.13
Net Power, [kW]	8817.4	9412.8	6.75
Thermal Efficiency, [%]	12.41	12.25	1.29
Specific Machinery Cost, [k$/MW]	262.71	255.71	2.66

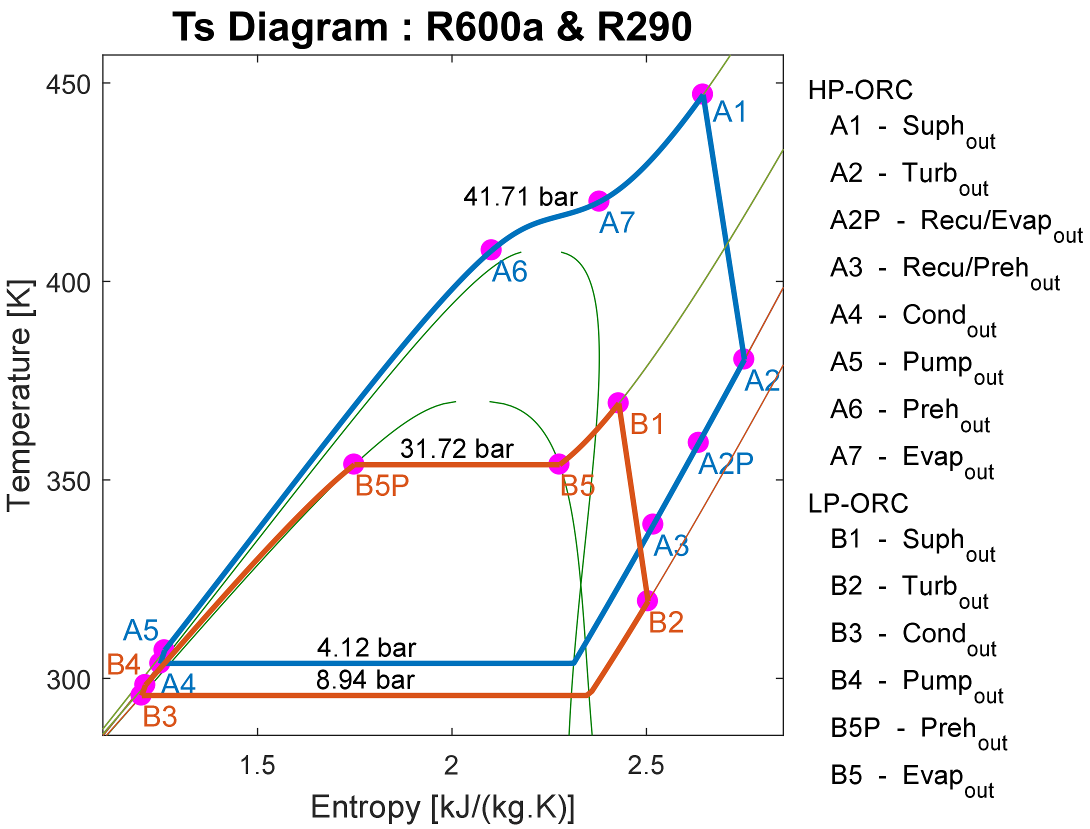


Figure 4.28: The Ts diagram of the Magmamax Binary Power Plant's sequential-modular model.

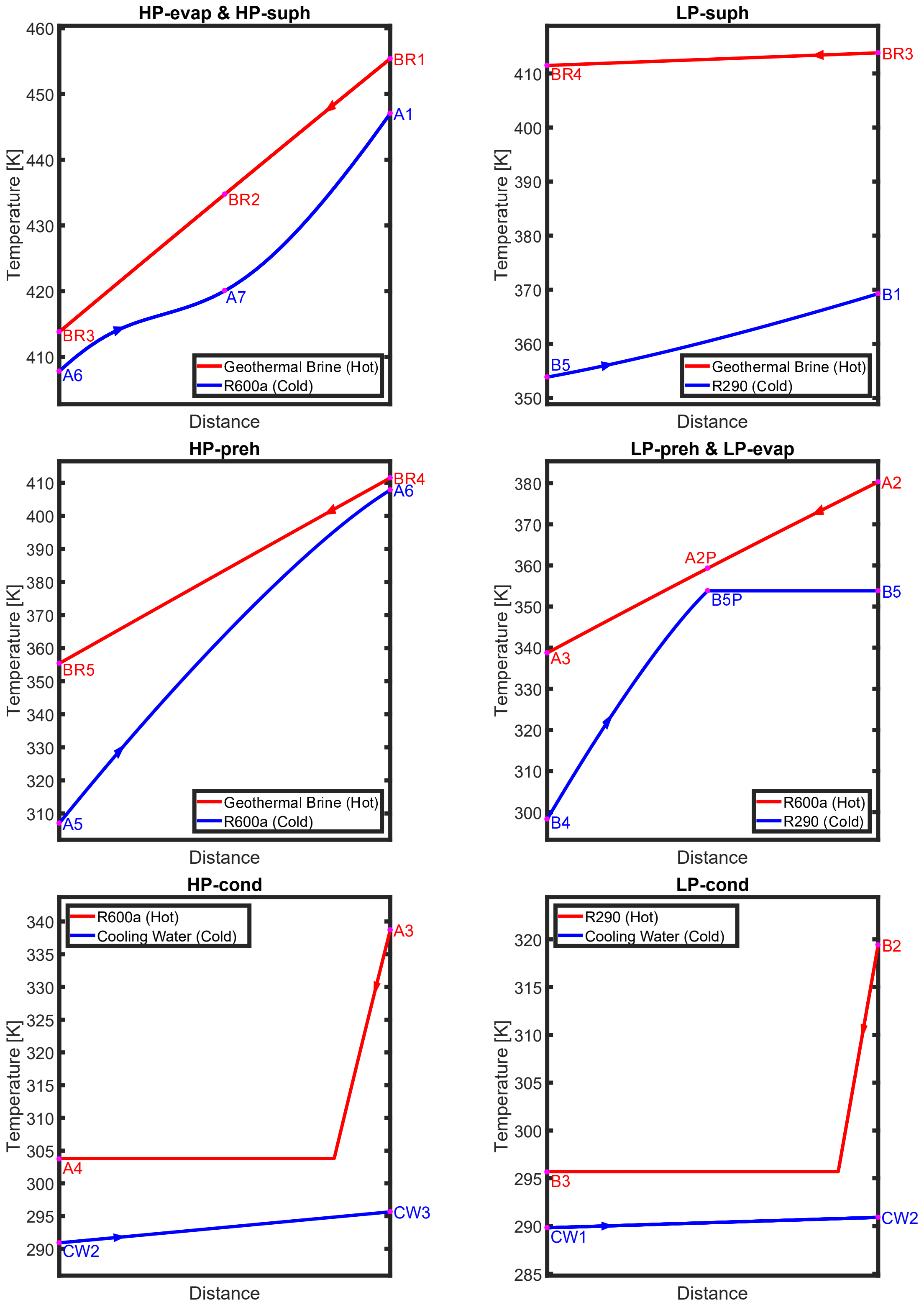


Figure 4.29: Diagrams showing the heat exchange between the hot medium and cold medium. The states refer to Figure 4.19. The x-axis represents the path of the fluid flow in the heat exchanger.

4.5 Summary

This chapter details the algebraic EO models of three real-world ORC systems. The modelling process is similar to what is described in Chapter 3, where the models are expressed as a set of equations with the thermodynamic properties and unit operations approximated using regression analysis. The models were constructed using the SymBuilder framework via the OPTI toolbox to leverage of the algebraic structure and automatically generate the required Jacobian and Hessian matrices, as well as the sparsity information, for the optimization solvers. Each plant model was constructed with flexibility in mind, thus they can be subjected to different working fluids without having to do any major modifications to the models.

In addition, the SM models of all three ORC systems were also constructed using the JSteam MATLAB Interface. The SM models were validated against the original nominal design information that was taken directly from the corresponding references of each plant. The discrepancies between the nominal design and the SM model were adequate for the purpose of this research given the assumptions that were made about the plant due to the insufficient information available for some of the plants. Furthermore, the thermodynamic process of each plant model was analysed by plotting the Ts diagram of the ORC system. From this analysis, the Ts diagrams indicated that the SM models are functioning as feasible ORC systems and do not violate any thermodynamic laws. This information is very important for this research given that the SM models are used to validate the algebraic EO models. Therefore, it is necessary to ensure that the SM models are a good representation of the original plants.

The next chapter will analyse the optimized results of the three algebraic EO models and examine the optimization performance of each model.

Chapter 5

Optimization of the Organic Rankine Cycle System Case Studies

In order to effectively illustrate the benefits of the proposed algebraic equation-oriented modelling approach, the three real-world binary cycle power plants described in Chapter 4 will be optimized in this chapter. The first three sections of the chapter will be focused on the optimization results and performance of the three plants, where the solve times and the global optimality of the solution will be analysed. The optimized results will be compared against each other and to their base-case scenarios that were derived from their respective literature where possible. The plant models will be optimized with different working fluids and two objective functions to examine any possible improvement to the plant. The last section of the chapter will highlight the automating optimization procedure of selecting the optimal working fluid for an ORC system using the proposed algebraic EO modelling approach.

5.1 Introduction

Once the algebraic EO models in Chapter 4 are constructed and formulated into an optimization problem, each plant will subsequently result in a different number of variables and constraints, as shown in Table 5.1. These dissimilarities between the plant models highlight the level of nonlinearity that can result from modelling complex ORC systems using the proposed algebraic EO modelling approach. Similar to Chapter 3, two objective functions will be investigated, namely, the net output power and the specific rotating machinery cost. While it is acknowledged that it might seem not applicable to optimize the specific rotating machinery cost of existing ORC systems given that the components have already been purchased, the optimization problem is aimed to investigate the future replacement of existing pumps and turbines that are damaged or in need of an upgrade. Also, both objective functions can be a reference for future greenfield optimizations that have a similar construction to the three presented binary cycle power plants. It is important to note that the proposed algebraic EO model does not restrict the objective functions presented in this research, and they can be modified to adhere to the different optimization problems.

Table 5.1: A comparison between the total number of variables and constraints of the binary cycle power plants in Chapter 4.

	DOEP	**USGP**	**MBPP**
Variables	21	27	26
Constraints	32	42	42
Linear	11	9	7
Quadratic	15	31	26
Nonlinear	6	2	9

All three plants were optimized with respect to different working fluids that were selected based on their availability via JSteam, frequent use in the literature, and their compatibility with the plant's design specifications and assumptions that were stated in Chapter 4. These can involve ensuring that condenser pressure is not higher than the inlet turbine pressure, the critical pressure and temperature are not higher than the upper pressure and temperature limits, the inlet turbine saturation temperature is not higher than the upper-temperature limit, etc. Refer to the appropriate sections in Chapter 4 for the process flow diagram and more information about the individual plant.

5.2 DOE Pilot Plant Optimization

For the DOE Pilot Plant, both the algebraic EO model and the SM model will be optimized in order to compare the differences in the optimization performance. This is aimed to highlight the issues associated with the optimization of SM models and show the significant contributions of this research. The algebraic EO model will be optimized using NLP solvers with both black-box and white-box solvers. Whereas, the equivalent SM model will be optimized with the black-box NLP solvers, but not the white-box solvers, due to its non-algebraic structure.

5.2.1 DOEP Algebraic Equation-Oriented Model Optimization

The algebraic EO model of the DOEP that was discussed in Section 4.2.1 was constructed using the SymBuilder framework. The MATLAB code of the algebraic EO model can be found in Appendix B.1. As a result, the built SymBuilder object of the DOEP reported the following model statistics:

```
SymBuilder Object
BUILT in 0.778s with:
- 21 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 32 constraint(s)
        - 11 linear
        - 15 quadratic
        -  6 nonlinear
- 42 bound(s)
-  0 integer variable(s)
```

Note that there are more variables and constraints than the basic ORC system presented in Chapter 3. Overall, the optimization problem consists of only continuous variables and constraints that are made up of linear, quadratic and nonlinear terms, thus resulting in an NLP problem.

The net output power of the DOEP was optimized with respect to different working fluids using NLP solvers and then substituted into the SM model. The optimized results and the base case values are shown in Table 5.2. Interestingly, all the black-box solvers managed to converge to the same solution as the white-box solver. Note that the base case for each working fluid was obtained from solving the SM model using the original state-point properties as discussed in Section 4.2.2.

The comparison between the optimized results and the base case values are shown in Figure 5.1. Analysing the optimized results, the working fluid with the best net output power of 4554.2 kW was R600a, whereas R218 yielded the worst output power of 1703.3 kW. However, in terms of the biggest improvement from the base case, R218 had the highest increase of 9.64% from the base case, while RC318 had the lowest increase of only 0.14%.

The average solve time of each solver is shown in Table 5.3. One observation that is very noticeable in Table 5.3 is that all the black-box solvers converged to a solution significantly faster than the white-box solvers. The fastest solve time was obtained by FILTERSD with only 0.009s, while the longest time was obtained by

Table 5.2: The optimized net output powers [kW] and the base case values of the DOE Pilot Plant for all the NLP solvers.

	Base Case	Optimized	Relative Improvement
R600a	4541.5	4554.2	0.28%
R134a	2773.0	2844.7	2.59%
R152a	3314.6	3350.2	1.07%
R218	1553.5	1703.3	9.64%
R227ea	3314.0	3360.3	1.40%
RC270	3038.5	3096.3	1.90%
RC318	3889.0	3894.4	0.14%
CF3I	4019.1	4028.6	0.24%

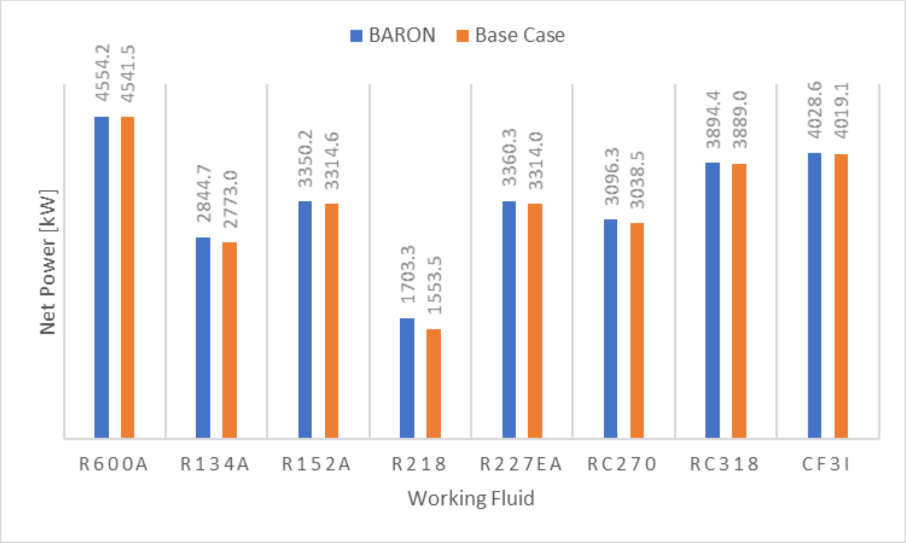


Figure 5.1: The comparison between BARON's optimized net output powers and the base case values of the DOE Pilot Plant.

BARON with 10.43s. This is a significant difference from the solve time for the basic ORC system examined in Chapter 3 where the highest value for the white-box solver was only 0.5s. This shows the complexity of this system is relatively harder to solve compared to the basic ORC system.

Taking the optimization problem further and minimizing the specific rotating machinery cost of the plant, which converts the objective function into a nonlinear expression, the solutions of all the solvers can be summarised in Table 5.4 and Figure 5.2. Similarly, all NLP solvers converged to the same solution for each working fluid, where R218 showed the biggest improvement of 3.5% from the base case, and CF3I had the smallest change of 0.66%.

Analysing the solve times for this second objective function in Table 5.5, the solvers on average took longer converging to a solution than the net output power objective function. As expected, the black-box solvers were much faster than the

Table 5.3: The solve times [s] of Table 5.2.

	IPOPT	fmincon	FILTERSD	BARON	SCIP
R600a	0.020	0.041	0.009	5.82	3.39
R134a	0.020	0.043	0.011	0.59	0.71
R152a	0.020	0.037	0.011	2.34	0.99
R218	0.020	0.043	0.011	0.49	0.68
R227ea	0.019	0.035	0.010	1.00	1.07
RC270	0.019	0.032	0.011	0.61	0.98
RC318	0.019	0.038	0.010	10.43	4.87
CF3I	0.019	0.040	0.011	2.31	1.41

Table 5.4: The optimized specific rotating machinery costs [k$/MW] and the base case values of the DOE Pilot Plant for all the NLP solvers.

	Base Case	Optimized	Relative Improvement
R600a	330.89	325.84	1.53%
R134a	373.00	367.71	1.42%
R152a	348.99	346.19	0.80%
R218	495.51	478.15	3.50%
R227ea	372.41	368.22	1.13%
RC270	352.18	347.73	1.26%
RC318	357.08	348.12	2.51%
CF3I	342.27	340.01	0.66%

white-box solvers, with the shortest time of 0.011s by FILTERSD. The solve times for the white-box solvers were noticeably longer than in Table 5.3, with the longest time of 31.63s by SCIP. This could be due to the nonlinearity of the second objective function being more difficult to solve than the former objective function.

Table 5.5: The solve times [s] of Table 5.4.

	IPOPT	fmincon	FILTERSD	BARON	SCIP
R600a	0.026	0.116	0.013	7.08	31.63
R134a	0.021	0.051	0.012	0.79	2.86
R152a	0.021	0.042	0.011	2.02	3.44
R218	0.021	0.171	0.012	0.67	4.50
R227ea	0.020	0.042	0.011	1.90	9.44
RC270	0.019	0.041	0.011	0.92	4.18
RC318	0.021	0.049	0.013	1.65	12.23
CF3I	0.019	0.042	0.012	6.74	9.29

Furthermore, comparing the relative error of the optimized algebraic EO model and the SM model, the average discrepancies are no more than 0.4% for each of the solvers and both objective functions. This is a good indication that the algebraic EO model is an accurate representation of the rigorous simulation model over a range

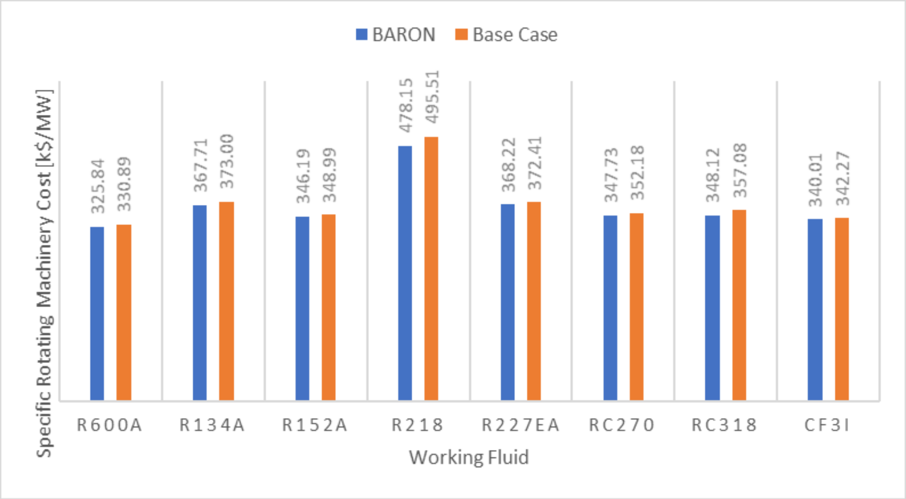


Figure 5.2: The comparison between BARON's optimized specific rotating machinery costs and the base case values of the DOE Pilot Plant.

of different operating points and working fluids.

For additional thermodynamic validation, the heat exchange process of the optimized DOE Pilot Plant can be plotted and analysed to ensure that the laws of thermodynamics and conventions are not being violated. The temperature profile between the geothermal brine/cooling water and the working fluid in Figure 5.3 shows that there are no thermodynamic violations during the heat exchange process, hence indicates that this ORC model does not show any anomalies discussed in Section 3.2.2.1.

5.2.2 DOEP Sequential-Modular Model Optimization

In order to illustrate the inefficiency of the SM model for optimization, especially for large and complex systems, the DOEP SM model was optimized with respect to different working fluids and subjected to the same two objective functions as in the previous section. The SM model and optimization parameters adhered to the same assumptions and simplifications as stated in Section 4.2.1. This resulted in 6 variables and 21 nonlinear constraints for the optimization problem, which was solved using IPOPT, `fmincon` and FILTERSD with the relative and absolute convergence tolerances reduced to 1×10^{-2}.

The tolerances were relaxed after noticing that the solvers were failing to find a solution and taking a significantly long time to solve the optimization problem. From observing the iterations of the solvers, the objective function evaluations were constantly fluctuating around the optimal point, which indicated that the solvers did not recognize that it has found a minimum as they could not satisfy the small

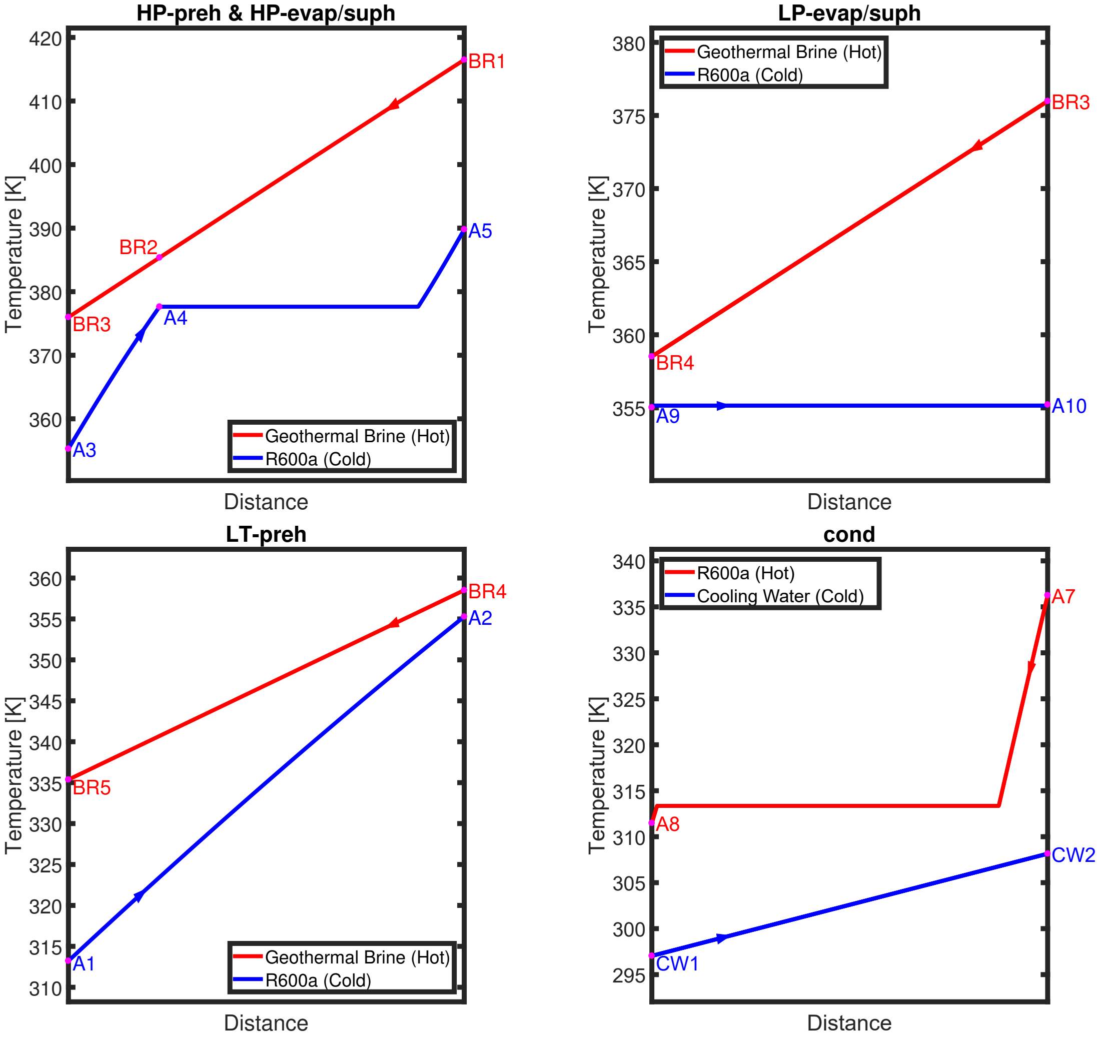


Figure 5.3: Diagrams showing the heat exchange process between the geothermal brine/cooling water and the working fluid (R600a) of the DOE Pilot Plant, where the specific rotating machinery cost was optimized using SCIP. The states refer to Figure 4.1. The x-axis represents the path of the fluid flow in the heat exchanger

.

tolerances.

Table 5.6 shows the optimized results of the net output power objective function. As can be seen, the performances of the black-box solvers were significantly worse than the performance of the algebraic EO model in Section 5.2.1. In some cases, the solvers had difficulties converging to a feasible solution and terminated prematurely. This is very prominent with FILTERSD where it was unable to find a solution for all of the working fluids, whereas IPOPT and `fmincon` were more successful but still did not find a solution for all the working fluids.

In addition to failing to converge to a solution for all the working fluids, for those instances where the solver did succeed, the solve times were considerably longer than the algebraic EO model (see Table 5.3). On average, it took IPOPT more than 6800x longer to optimize the SM model than the algebraic EO model. While

Table 5.6: The optimized net output powers of the DOEP SM model, where $\mathcal{J}$ is the objective function in [kW] and Time is the solve time in [s]. The relative and absolute convergence tolerances were set to 1×10^{-2}. FNS: Found No Solution.

	IPOPT		`fmincon`		**FILTERSD**	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**
R600a	FNS	-	4550.1	15.53	FNS	-
R134a	2844.3	31.76	2844.2	28.20	FNS	-
R152a	3349.8	423.80	3349.7	43.11	FNS	-
R218	1702.6	217.52	FNS	-	FNS	-
R227ea	3359.9	79.29	FNS	-	FNS	-
RC270	3095.9	46.66	3095.9	35.73	FNS	-
RC318	3893.1	94.37	FNS	-	FNS	-
CF3I	4028.1	11.22	FNS	-	FNS	-

`fmincon` took more than 800x longer on average to optimize the SM model. This is a major drawback for the SM model in regards to optimization, especially dealing with an exhaustive optimization that consists of the same number of variables and constraints but different equation coefficients and constants.

Taking the optimization further and minimizing the specific rotating machinery cost, the black-box solvers yielded the same poor, if not worse, optimization performance. As shown in Table 5.7, IPOPT managed to successfully converge to a solution for all the working fluids, but the average solve time was longer than maximizing the net output power by around 4x for all the solved instances for both objective functions. Unlike the first objective function, `fmincon` did not manage to solve for R152a and the average solve time was slower by more than 2x. Compared to the algebraic EO model counterpart and only considering all the instances that were solved in both models, the average solve time increased by more than 29000x for IPOPT, more than 860x for `fmincon`, and more than 2500x for FILTERSD to optimize the SM model.

Table 5.8 shows the average number of iterations needed to optimize the SM model and the algebraic EO model, which shows that all the black-box solvers required more iterations to optimize the SM model than the algebraic EO model. The biggest difference is observed with the IPOPT solver that can require up to 62x more iterations to optimize the SM model. Therefore, this would have contributed to the longer execution times shown in Tables 5.6 and Table 5.7.

The poor optimization performance of the SM model was due to the direct implementation of the external thermodynamic routines (REFPROP) and the unit operation functions that prevented the analytical gradient and Hessian information to be supplied to the solver. Consequently, this resulted in the use of the finite

Table 5.7: The optimized specific rotating machinery costs of the DOEP SM model, where $\mathcal{J}$ is the objective function in [k\$/MW] and Time is the solve time in [s]. The relative and absolute convergence tolerances were set to 1×10^{-2}. FNS: Found No Solution.

	IPOPT		`fmincon`		**FILTERSD**	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**
R600a	325.88	1322.4	326.27	89.69	330.08	32.88
R134a	367.76	26.50	367.97	40.01	FNS	-
R152a	346.22	24.06	FNS	-	FNS	-
R218	478.32	2225.4	FNS	-	FNS	-
R227ea	368.26	265.26	FNS	-	FNS	-
RC270	347.77	47.96	347.98	49.77	FNS	-
RC318	348.19	605.40	FNS	-	FNS	-
CF3I	340.02	362.77	FNS	-	FNS	-

Table 5.8: The average number of iterations that the black-box solvers took to solve the SM model and the algebraic EO model of the DOE Pilot Plant. This table only considers the working fluids that were solved in both models.

	IPOPT		`fmincon`		**FILTERSD**	
	EO	**SM**	**EO**	**SM**	**EO**	**SM**
Net Output Power	13	183	11	22	-	-
Specific Rotating Machinery Cost	14	878	13	39	9	31

difference method to obtain the required derivative information that led to various convergence issues, such as premature termination and more iterations, as discussed in Section 2.9. Furthermore, the implementation of a nonlinear equation solver, `fsolve`, which was needed to solve the ORC system recycles at each optimization iteration, is another factor that contributed the poor optimization performance. This nonlinear equation solver added extra complexity to the SM optimization problem that resulted in a slower and less efficient performance, despite having a lower number of decision variables and constraints than the algebraic EO model.

The optimization performance of the SM model presented in this subsection illustrates the major optimization issues associated with the SM model and highlight the importance and contributions of the proposed algebraic EO model. From comparing the optimization performance between the SM model and the algebraic EO model, it is evident that the formulation of the ORC model plays a significant role in the overall performance and solution of the optimization problem. Therefore, given the poor performance presented in this subsection, the SM models of the next two plants will be not optimized but will be used to validate the algebraic EO models.

5.3 USGeo Plant Optimization

The algebraic EO model of the USGP that was discussed in Section 4.3.1 was constructed using the SymBuilder framework. The MATLAB code of the algebraic EO model can be found in Appendix B.2. As a result, the built SymBuilder object of the USGP reported the following model statistics:

```
SymBuilder Object
BUILT in 1.329s with:
- 27 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 42 constraint(s)
        - 9 linear
        - 31 quadratic
        -  2 nonlinear
- 54 bound(s)
-  0 integer variable(s)
```

Similar to the DOEP, the algebraic EO model was optimized with respect to different working fluids, namely, R601a, R245ca, R245fa, and C5F12, that complied with the model specifications discussed Section 4.3.1. When optimizing the net output power of the plant, all the NLP solvers managed to converge to the same solution for each working fluid, as shown in Table 5.9 and Figure 5.4. The working fluid that had the highest improvement in the net output power of 1.58% was C5F12, whereas R601a did not show any significant improvement and converged to the base case operating point as this was already the optimal solution. The small discrepancy between the optimized result and the base case for R601a is due to the approximation made to the algebraic EO model that rendered the two results different.

Table 5.9: The optimized net output powers [kW] and the base case values of the USGeo Plant.

	Base Case	**Optimized**	**Relative Improvement**
R601a	14307.2	14311.3	0.03%
R245ca	12715.4	12842.9	1.00%
R245fa	10713.1	10853.1	1.31%
C5F12	11448.3	11629.4	1.58%

Analysing the solve times for the net power optimization problem in Table 5.10, the average execution times for the black-box solvers were faster than the white-box solvers and exhibit a similar performance as the DOEP model.

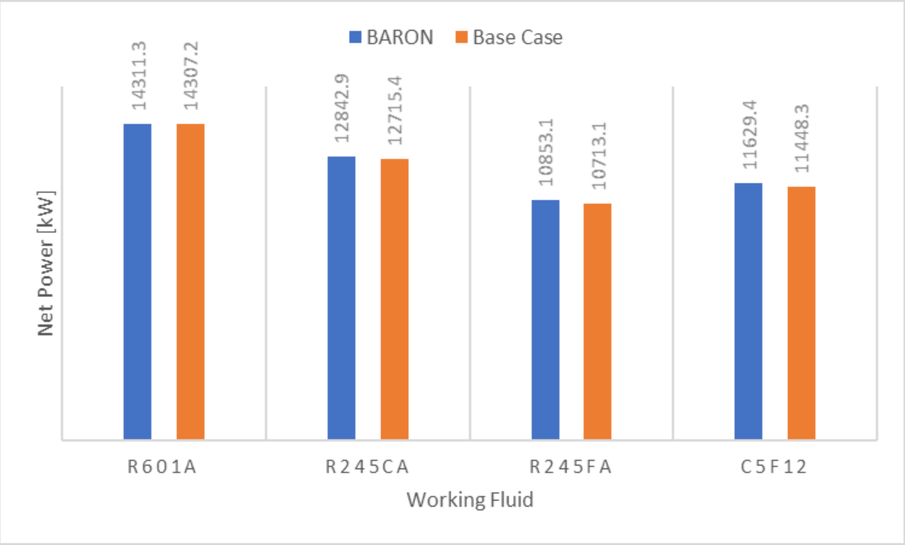


Figure 5.4: The comparison between BARON's optimized net output powers and the base case values of the USGeo Plant.

Table 5.10: The solve times [s] of Table 5.9.

	IPOPT	fmincon	FILTERSD	BARON	SCIP
R601a	0.029	0.219	0.017	9.83	28.16
R245ca	0.023	0.065	0.013	0.64	3.14
R245fa	0.022	0.058	0.012	8.51	3.66
C5F12	0.021	0.044	0.014	0.69	3.18

Taking the optimization problem further and changing the objective function to the specific rotating machinery cost, the following optimized results in Table 5.11 were obtained. Again, the black-box solvers managed to find the same global solution as the white-box solvers. The comparison between the global solutions and the base case values is shown in Figure 5.5. Similar to the previous objective function, C5F12 yielded the best result with an improvement of 2.11% from the base case, whereas R601a only had a small improvement of 0.86%.

Table 5.11: The optimized specific rotating machinery costs [k$/MW] and the base case values of the USGeo Plant.

	Base Case	Optimized	Relative Improvement
R601a	215.66	213.80	0.86%
R245ca	219.47	215.37	1.87%
R245fa	230.93	226.71	1.83%
C5F12	220.85	216.18	2.11%

When analysing the solve times of the optimized specific rotating machinery cost, it took `fmincon` significantly longer to solve the model than the other black-

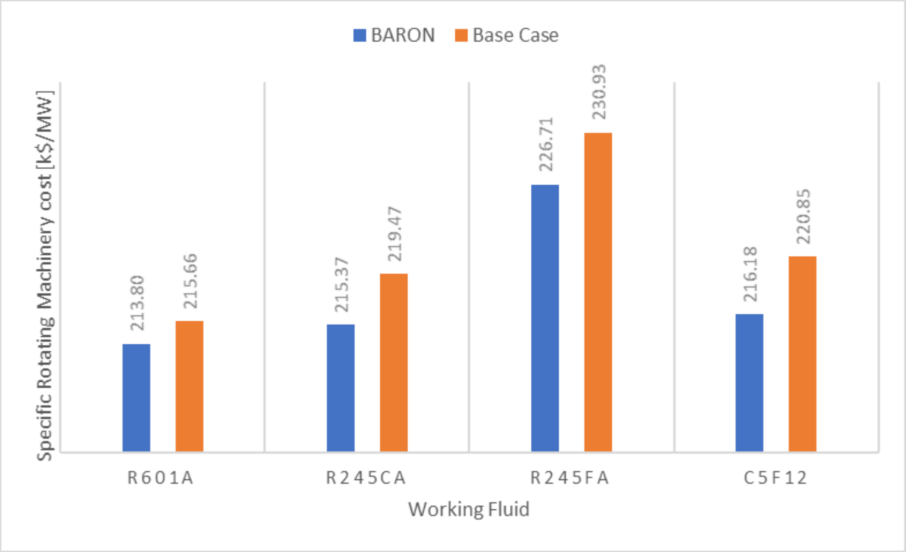


Figure 5.5: The comparison between BARON's optimized specific rotating machinery costs and the base case values of the USGeo Plant.

box solvers. As shown in Table 5.12, `fmincon` took on average 0.395s to solve the optimization problem, while IPOPT and FILTERSD took around 0.025s and 0.015s, respectively. The long execution time of `fmincon` compared to other black-box solvers could be due to a number of factors, including the supplied initial guess, the complexity of the problem, the number of variables and constraints, or the algorithm convergence criteria, and it can be very difficult to narrow down the actual causes as it can be a combination of factors that affect the overall performance of the solver. Furthermore, for interior-point algorithms (which is used in `fmincon`), can set the initial points to the midpoint of the finite bounds if they are outside the supplied bounds or the algorithm can attempt to find a "central path" (the path that the algorithm takes to find the optimal solution) midpoint between the finite bounds [125]. This can result in the solver using an initial point or a central path that is further away from the optimum point and take more iterations to converge to a solution, as shown in Table 5.13.

Table 5.12: The solve times [s] of Table 5.11.

	IPOPT	`fmincon`	**FILTERSD**	**BARON**	**SCIP**
R601a	0.024	0.192	0.015	1.67	13.85
R245ca	0.024	0.227	0.014	0.63	3.72
R245fa	0.026	0.921	0.015	1.93	6.22
C5F12	0.025	0.240	0.016	0.53	2.71

Despite the long execution times of the `fmincon` solver, the average discrepancies between the optimized algebraic EO model and the SM model were less than 0.15% for each of the solvers and both objective functions. This shows that the algebraic EO model of the USGP is an accurate representation of the rigorous simulation

Table 5.13: The average number of iterations that the black-box solvers took to solve the algebraic EO model of the USGeo Plant.

	IPOPT	`fmincon`	**FILTERSD**
Net Output Power	19	19	8
Specific Rotating Machinery Cost	20	72	9

model over a range of different operating points and working fluids.

For an additional validation check, the heat exchange process of the optimized USGeo Plant can be plotted and analysed to ensure that the laws of thermodynamics and conventions are not being violated. The temperature profile between the geothermal brine/cooling water and the working fluid in Figure 5.6 shows that there are no thermodynamic violations during the heat exchange process, hence indicates that this ORC model does not show any anomalies that were discussed in Section 3.2.2.1.

5.4 Magmamax Binary Power Plant Optimization

The algebraic EO model of MBPP that was discussed in Section 4.4.1 was constructed using the SymBuilder framework. The MATLAB code of the algebraic EO model can be found in Appendix B.3. As a result, the built SymBuilder object of the MBPP reported the following model statistics:

```
SymBuilder Object
BUILT in 1.592s with:
- 26 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 42 constraint(s)
        -  7 linear
        - 26 quadratic
        -  9 nonlinear
- 52 bound(s)
-  0 integer variable(s)
```

Given that the plant uses two different working fluids, the model will be subjected to two sets of working fluids, namely, R600a, R236ea, R236fa and RC318 for the HP cycle, and R290, R32 and R143a for the LP cycle. These working fluids were selected based on their availability via JSteam and their thermodynamic properties complying with the model specifications that were discussed in Section 4.4.1. Table 5.14 shows the optimized net output powers of the algebraic EO model with

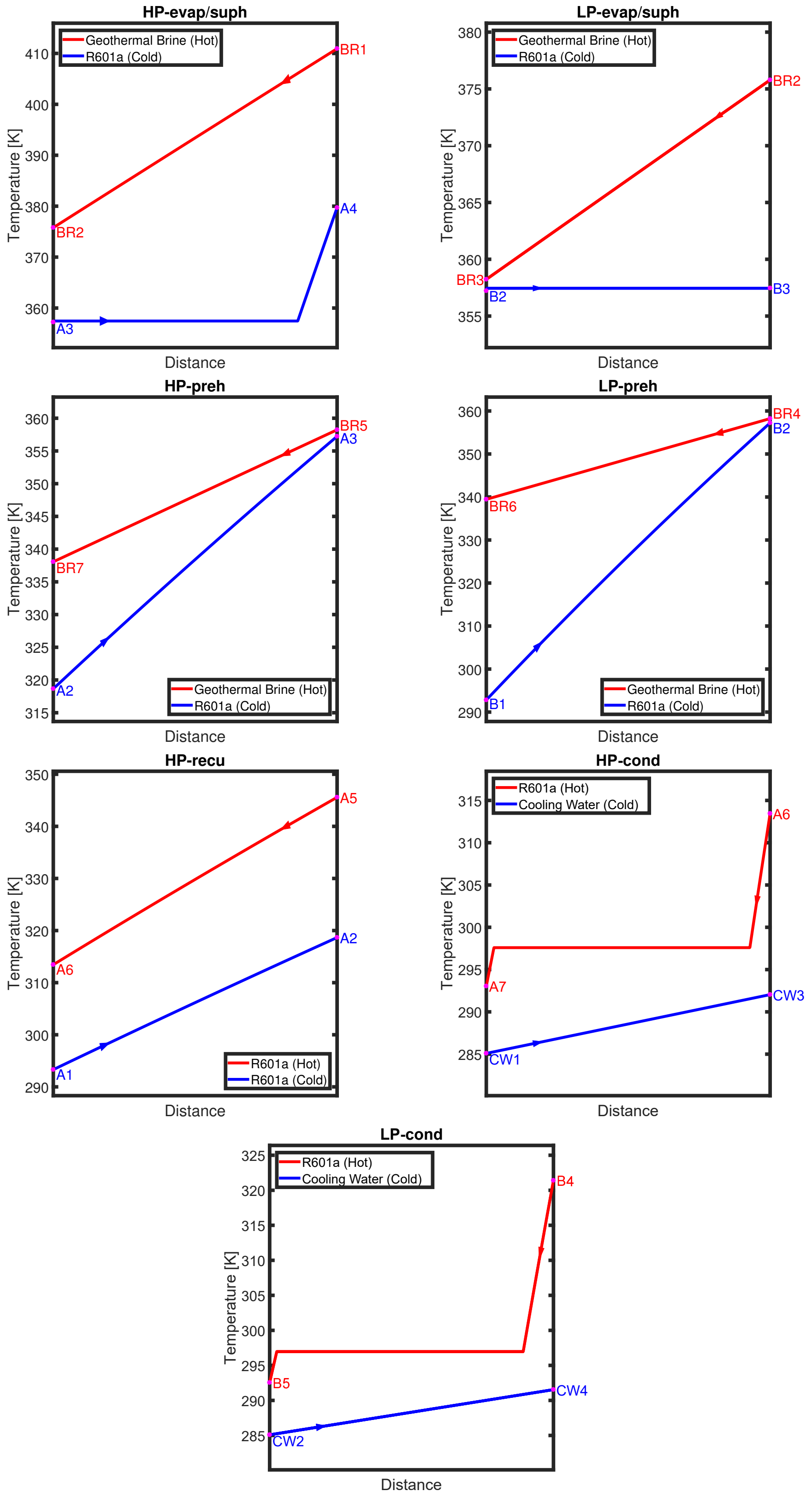


Figure 5.6: Diagrams showing the heat exchange between the hot medium and cold medium, where the specific rotating machinery cost of the USGeo Plant was optimized using SCIP. The states refer to Figure 4.9. The x-axis represents the path of the fluid flow in the heat exchanger.

respect to the different working fluid pairs using the NLP solvers. In this case, all the black-box solvers managed to converge to the same global solution as the white-box solvers.

Table 5.14: The optimized net output powers [kW] of the Magmamax Binary Power Plant.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	9625.7	9157.0	9272.9
	R236ea	9641.8	9205.0	9314.4
	R236fa	9282.6	8766.3	8894.7
	RC318	8553.9	7854.5	8031.3

The comparison between the optimized results and the base case values is shown in Figure 5.7. From the optimized results, the pair of working fluids that gave the highest net output power of 9641.8 kW is R236ea for the HP cycle and R290 for the LP cycle, which is higher than the optimized original plant working fluids (namely, R600a for the HP cycle and R290 for the LP cycle) by 0.17%. The pair of working fluids that yielded the biggest improvement from the base case by 3.23% is RC318 for the HP cycle and R143a for the LP cycle. For this optimization problem, the average discrepancy between the optimal results and the base case values is around 0.05%, which indicates that the algebraic EO model is good representation of the original system.

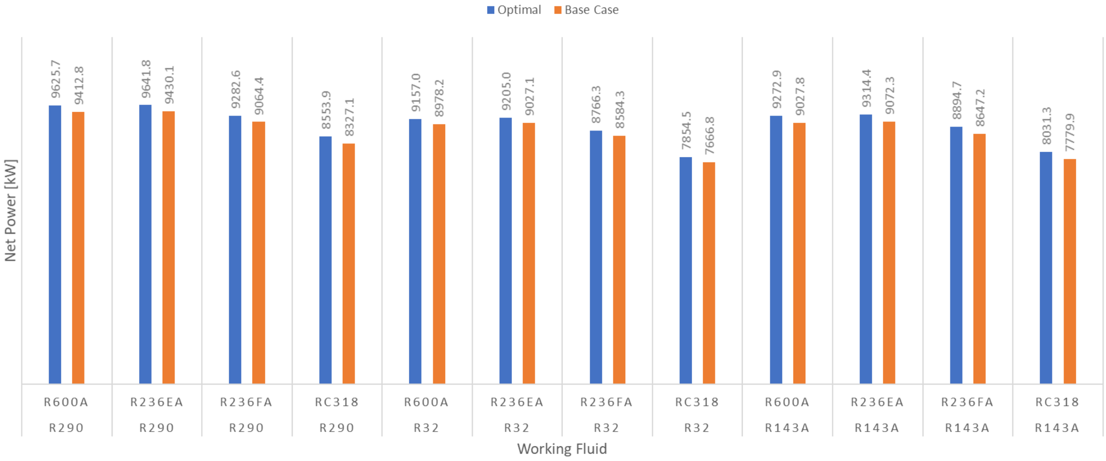


Figure 5.7: The comparison between the optimized net output powers and the base case values of the Magmamax Binary Power Plant.

Collectively the black-box solvers managed to converge to a solution within an average solve time of 0.032s for all the working fluid pairs, as shown in Table 5.15. While both of the white-box solvers on average took 87s to converge to a solution, which is significantly longer than the black-box solvers. The poor performance could be due to the larger number of nonlinear constraints compared to the previous two binary cycle power plants. Given that the algorithm of the white-box solvers is the spatial branch-and-bound method that requires solving a convex relaxed problem and the original nonconvex problem repeatedly, it can be very computationally expensive to solve highly nonlinear problems and can result in a long execution time.

Table 5.15: The solve times [s] of Table 5.14.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	17.2	71.6	24.3
		R236ea	40.4	210.1	90.9
		R236fa	57.8	296.0	217.5
		RC318	348.6	234.4	106.0
	SCIP	R600a	5.6	7.4	20.5
		R236ea	47.2	13.6	24.8
		R236fa	40.3	17.9	20.0
		RC318	78.6	22.3	76.2
	IPOP	R600a	0.031	0.022	0.024
		R236ea	0.023	0.022	0.019
		R236fa	0.019	0.019	0.019
		RC318	0.023	0.023	0.025
	fmincon	R600a	0.207	0.057	0.058
		R236ea	0.050	0.046	0.054
		R236fa	0.040	0.038	0.043
		RC318	0.042	0.038	0.043
	FILTERSD	R600a	0.023	0.014	0.018
		R236ea	0.014	0.018	0.013
		R236fa	0.015	0.012	0.011
		RC318	0.012	0.018	0.014

Taking the optimization problem further and minimizing the specific rotating machinery cost of the plant, a similar outcome can be observed in the optimized results. The black-box solvers managed to converged to the same global solution as SCIP, as shown in Table 5.16. However, in this case, BARON was not able to find a solution to all working fluid pairs and terminated after reaching the maximum solve time limit of 1 hour.

Collectively, on average, it took white-box solvers around 515s to solve the optimization problem. Whereas, the black-box solvers managed to converge to a solution for all the working fluid pairs within an average solve time of 0.156s, as shown in Table 5.17, which is about 5x slower than the net output power objective function. Consequently, the solvers required more number of iterations to optimize the problem compared to the first objective function, as shown in Table 5.18.

Table 5.16: The optimized specific rotating machinery costs [k$/MW] of the Magmamax Binary Power Plant.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	253.00	247.08	252.99
	R236ea	247.66	241.22	247.42
	R236fa	257.25	251.20	257.69*
	RC318	276.89*	273.10	280.05

*BARON: Maximum Time Reached

Table 5.17: The solve times [s] of Table 5.16. MTR: Maximum Time Reached.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	22.2	8.7	24.5
		R236ea	106.1	9.4	951.5
		R236fa	895.4	464.2	MTR
		RC318	MTR	337.1	1328.7
	SCIP	R600a	46.5	17.0	50.1
		R236ea	104.4	14.0	47.4
		R236fa	784.8	157.4	2804.5
		RC318	541.1	320.1	2303.4
	IPOPT	R600a	0.033	0.022	0.029
		R236ea	0.034	0.028	0.023
		R236fa	0.039	0.028	0.031
		RC318	0.052	0.024	0.027
	fmincon	R600a	1.062	0.132	0.173
		R236ea	0.866	0.159	0.641
		R236fa	0.353	0.150	0.789
		RC318	0.298	0.144	0.260
	FILTERSD	R600a	0.021	0.016	0.018
		R236ea	0.019	0.020	0.020
		R236fa	0.016	0.025	0.016
		RC318	0.017	0.013	0.019

Table 5.18: The average number of iterations of the black-box solvers for the Magmamax Binary Power Plant.

	IPOPT	fmincon	**FILTERSD**
Net Output Power	18	14	9
Specific Rotating Machinery Cost	26	75	12

Analysing the comparison between the optimized results and the base case values in Figure 5.8, the highest improvement of 2.84% from the base case was obtained with RC318 for the HP cycle and R143a for the LP cycle. While the lowest improvement of 1.06% was obtained with R600a for the HP cycle and R290 for the

LP cycle. The lowest specific rotating machinery cost of 241.22 k\$/MW was obtained with R236ea for the HP cycle and R32 for the LP cycle, which is a 4.66% improvement from the original working fluid pair.

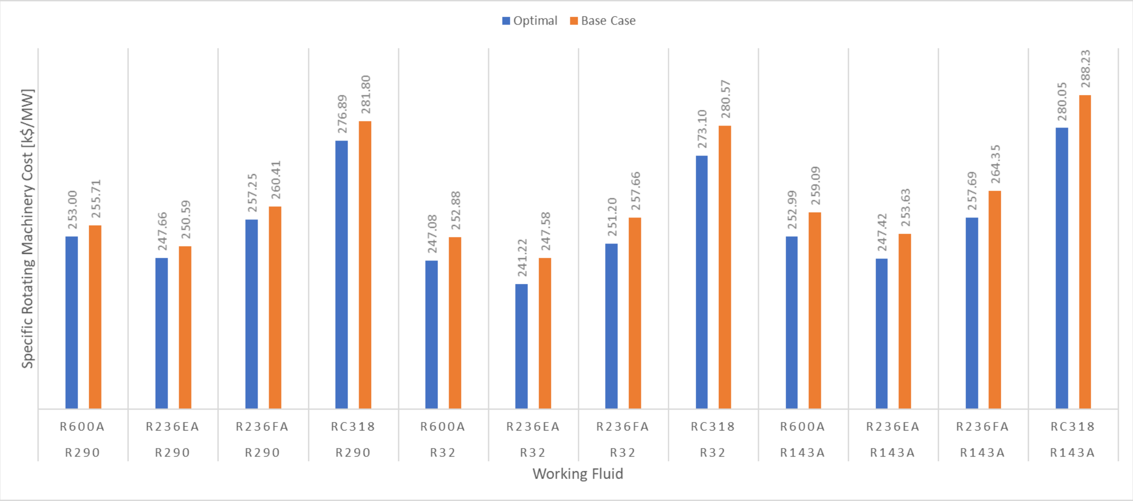


Figure 5.8: The comparison between the optimized specific rotating machinery costs and the base case values of the Magmamax Binary Power Plant.

To ensure that the thermodynamic violation discussed in Section 3.2.2.1 does not occur, the temperature profiles of the heat exchangers can be plotted and analysed. As shown in Figure 5.9, there are no anomalies during the heat exchange process that indicate the second law of thermodynamics is violated.

Despite the poor performance of the white-box solvers, the average discrepancy between the optimized algebraic EO model and the SM model for the specific rotating machinery cost was <0.01% for each of the solvers. This indicates that the thermodynamic approximations did not deviate the algebraic EO model drastically from its SM model counterpart over a range of different operating points and working fluids.

It is widely acknowledged that nonlinear problems are hard to solve and reducing the nonlinearity of the problem can help improve the performance of the solver. Often practitioners applied various linear and integer techniques, see [126, 127, 24], to the nonlinear problem in order to reduce the complexity and result in a more favourable outcome. However, some of these techniques can reduce the accuracy of the model by using linear approximations on the nonlinear terms, which can be a drawback if the accuracy is of high importance. In the next chapter, we will address this issue by implementing piecewise fit approximations to reduce the nonlinearity of the optimization problem without compromising on the accuracy of the model.

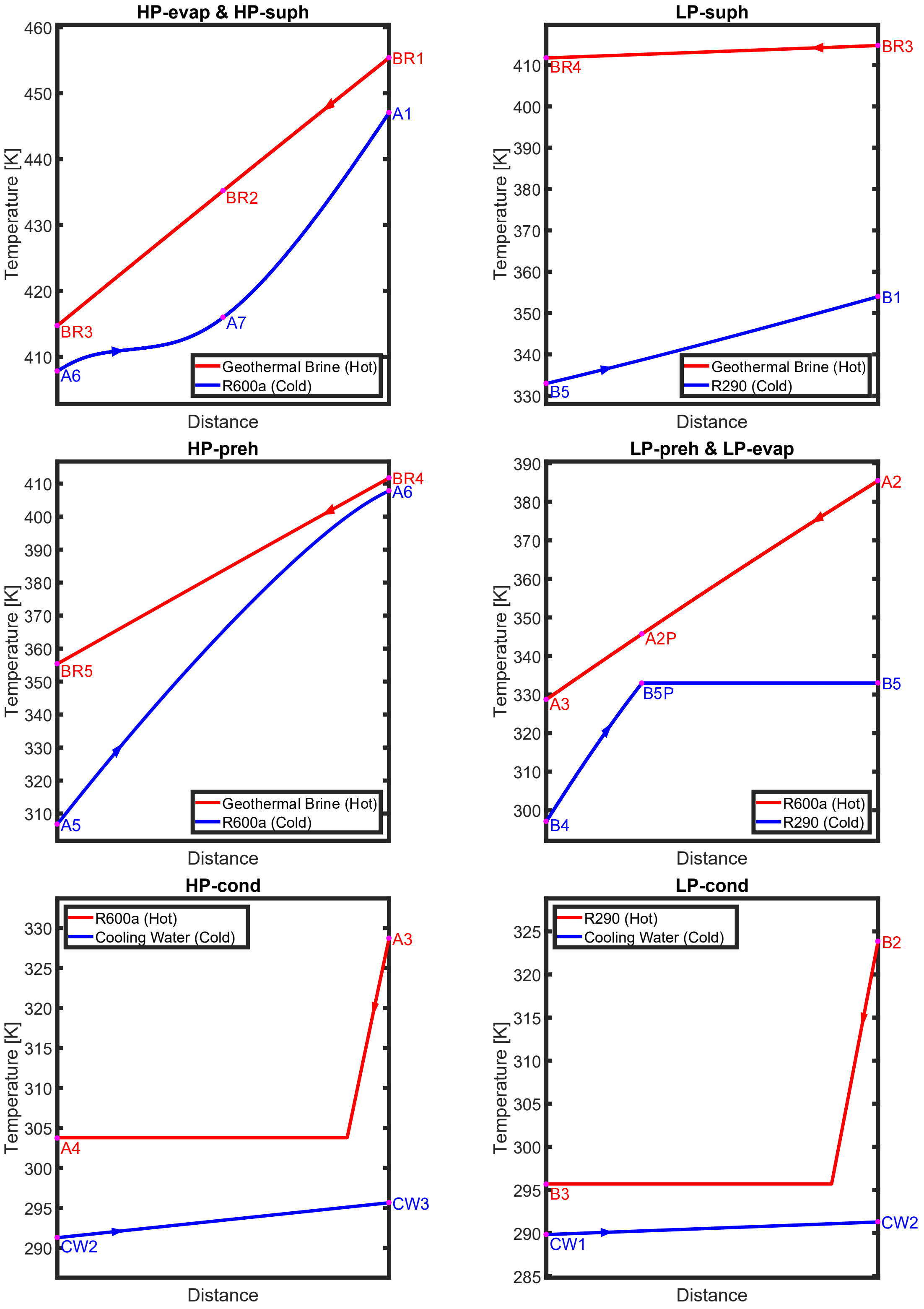


Figure 5.9: Diagrams showing the heat exchange between the hot medium and cold medium, where the specific rotating machinery cost of the Magmamax Binary Power Plant was optimized using SCIP. The states refer to Figure 4.19. The x-axis represents the path of the fluid flow in the heat exchanger.

5.5 Automating the Working Fluid Selection

While in the literature there are extensive analysis and comparisons between different working fluids on various ORC systems, there is no one best working fluid that can meet all the thermodynamic constraints, environmental regulations, cost limitations and different heat source temperatures of every ORC system [53]. Also, the best working fluid that generates the highest efficiency for one ORC system may not have the same outcome under different operating conditions, plant configurations, and working fluids. There will always be compromises that need to be considered when selecting a working fluid. However, leveraging off the proposed algebraic EO model structure, the optimization problem can be integrated with an outer loop statement to automatically and efficiently select the optimum working fluid from a set of working fluids.

Constructing an outer loop around the ORC optimization problem amounts to gathering the potential working fluids and optimizing the model with respect to each working fluid. At the end of each iteration, the optimized result will be compared to the previous result and replaced if the new working fluid offers a more superior outcome. As a result, the optimum working fluid will be automatically selected from a set of fluids for the ORC system.

Table 5.19 shows the total computational optimization times of the construction and optimization of the algebraic EO models. Evidently, the total solve times are reasonably fast despite solving the optimization problem multiple times. Suppose the same optimization procedure was to be implemented using an SM model, the optimization times would be significantly longer and, in some cases, have issues converging to a solution (as shown in Section 5.2.2). The construction of the optimization problem (i.e., SymBuilder object) took significantly longer than the optimization time. This is mainly due to the construction of the Jacobian and Hessian matrices using the MATLAB's Symbolic Math Toolbox and, thus, will subsequently take much longer to build for larger problems. Therefore, it might be beneficial for future work to consider optimizing the code of the SymBuilder framework or

Table 5.19: The total computational optimization times of the algebraic EO models with respect to different working fluids using IPOPT, where the net output power was optimized.

	DOEP	**USGP**	**MBPP**
Build Time	6.224s	5.316s	19.10s
Solve Time	0.136s	0.095s	0.269s
Total	6.360s	5.411s	19.37s

investigate in other alternative modelling platforms, such as GAMS or AMPL, and explore if they can offer a faster approach at constructing the optimization model.

5.6 Summary

This chapter detailed the optimized results and performance of three real-world ORC systems using the algebraic EO formulation discussed in Chapter 4. Each ORC model was optimized with respect to different working fluids and two objective functions, namely, the net output power and the specific rotating machinery cost. Overall, the black-box solvers were relatively faster than the white-box solvers and converged to the same global optimum.

In addition, this chapter also highlighted the optimization inefficiency of the SM model compared to the proposed algebraic EO model, especially for large and complex systems, by optimizing the SM model of the DOE Pilot Plant using NLP black-box solvers. The results showed that the optimization performance of the SM model was poor and resulted in long solve times and convergence issues. This is due to the use of the finite difference method and the requirement of a nonlinear equation solver to converge the ORC flowsheet at each optimization iteration.

While the overall performance of the black-box solvers and accuracy of the algebraic EO models were very favourable, the white-box solvers did not perform as well in some cases, especially with the Magmamax Binary Power Plant. The following chapter will explore the implementation of piecewise functions for approximating the thermodynamics and unit operations in order to reduce the nonlinearity of the problem without compromising on the accuracy. This alternative approach is beneficial for when both the accuracy of the model and the optimization performance are of high importance, but also want to retain the same algebraic characteristic that is differentiable and compatible with the white-box solvers.

Chapter 6

Model Improvement Using Piecewise Fit Approximation

This chapter will introduce a piecewise fit formulation to approximate univariate and bivariate functions for global optimization. The first section will present a description of the piecewise fit approximation algorithm with a simple univariate and bivariate function as an example. This is then followed by the discussion on how the accuracy of the approximation can be further improved by optimizing the allocation of the piecewise fit breakpoints. Lastly, the advantages and limitations of the presented piecewise fit approach will also be highlighted, and the chapter will conclude with the implementation of the piecewise fit on the ORC systems.

6.1 Introduction

In the algebraic EO models presented thus far, some of the operational constraints and the isentropic work of the turbine and pump were approximated using only a single curve or surface fit. This approach allows an algebraic EO model of an ORC system to be formulated and globally optimized to a reasonable level of accuracy. However, utilizing a single surface/curve fit might not always provide the level of accuracy that is adequate for different working fluids, chemical processes, or highly nonlinear thermodynamic properties. While it is possible to increase the accuracy of the fit by using a more nonlinear fit or increase the order of the polynomial function, the improvement in accuracy might not be sufficient and can increase the nonlinearity of the optimization problem that can lead to poor optimization performance. An alternative approach is to use a piecewise fit and formulate an algebraic mixed integer nonlinear programming (MINLP) model that consists of low-level nonlinear fits with better regression accuracy. Leveraging the modelling approach presented in the Chapter 3 and Chapter 4, the proposed piecewise fit formulation will not compromise on the algebraic structure of the optimization problem and will still be compatible with the white-box solvers.

6.2 Piecewise Fit Approximation

This section will present the piecewise fit algorithm that was developed to increase the accuracy of the approximations of univariate and bivariate functions. Unlike other piecewise fit approximation in [128, 127], this approach is not limited to linear approximations, but allows higher order polynomial approximations to achieve better accuracy without increasing the number of breakpoints. In addition, this approach can easily be implemented to both univariate and bivariate functions and requires fewer binary variables and a simpler *special ordered set* (SOS) type constraint. This method was developed in MATLAB using the object-oriented approach, where the name of the MATLAB class is called `pwfit`.

6.2.1 The `pwfit` Algorithm

The algorithm of `pwfit` utilizes the `lsqlin` MATLAB function [129], which solves constrained least-squares curve fitting problems of the form

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{C}\mathbf{x} - \mathbf{d}||_2^2 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{A}_{\text{eq}}\mathbf{x}_{\text{eq}} = \mathbf{b}_{\text{eq}}. \end{aligned} \tag{6.1}$$

In this case, the linear constraints are associated with the piecewise subfunction continuity constraints, i.e., the gradient and function evaluation of the adjacent subfunctions must be the same at the breakpoint, which will be discussed next in Section 6.2.2.

For illustration purposes, the data in (6.2) will be used to explain the `pwfit` algorithm, which was generated from a basic exponential function $y(x) = e^x$. Figure. 6.1 shows the data as black dots and the two quadratic polynomial fits separated by the breakpoint. The breakpoint for this example was fixed at 2, thus dividing the number of data points evenly. Note the location of the breakpoint was selected arbitrarily for this example, but it will be optimized later in Section 6.2.3 to further improve the accuracy of the regression fit.

$$\mathbf{x}_{\text{data}} = \begin{bmatrix} 1.00 \\ 1.22 \\ 1.44 \\ 1.67 \\ 1.89 \\ 2.11 \\ 2.33 \\ 2.56 \\ 2.78 \\ 3.00 \end{bmatrix}, \mathbf{y}_{\text{data}} = \begin{bmatrix} 2.72 \\ 3.39 \\ 4.24 \\ 5.29 \\ 6.61 \\ 8.26 \\ 10.31 \\ 12.88 \\ 16.08 \\ 20.09 \end{bmatrix}. \tag{6.2}$$

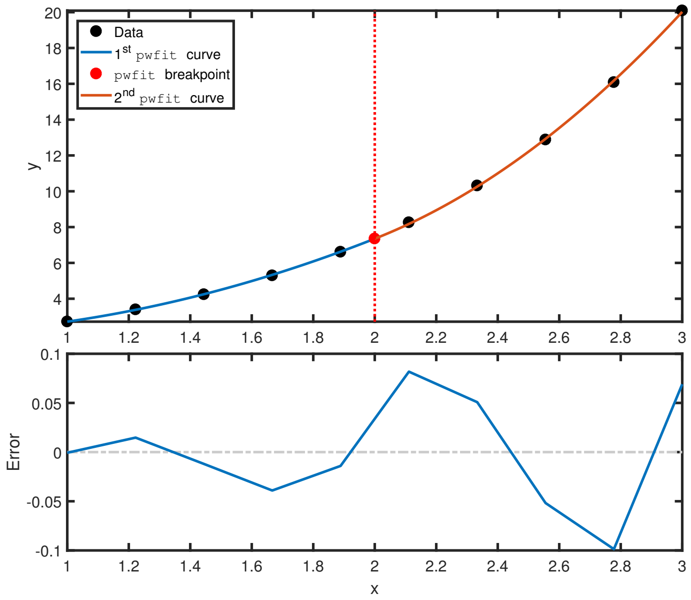


Figure 6.1: Approximating $y(x) = e^x$ with `pwfit` using two quadratic polynomial functions with a fixed breakpoint at 2.

The constant vector $\mathbf{d}$ is equal to the $\mathbf{y}_{\text{data}}$, whereas the multiplier matrix $\mathbf{C}$ can be constructed as a set of Vandermonde matrices for the $\mathbf{x}_{\text{data}}$ that are arranged diagonally in a matrix as

$$\mathbf{C} = \begin{bmatrix} \mathbf{V}_{1,1} & 0 & \cdots & 0 \\ 0 & \mathbf{V}_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{V}_{m,n} \end{bmatrix}. \tag{6.3}$$

Each Vandermonde matrix $\mathbf{V}_{m,n}$ is associated to one of the polynomial curves/surfaces of the piecewise fit and is defined as

$$\mathbf{V}_{m,n} = \begin{bmatrix} x^{N}_{\text{data}_1} & x^{N-1}_{\text{data}_1} & \cdots & x^{0}_{\text{data}_1} \\ x^{N}_{\text{data}_2} & x^{N-1}_{\text{data}_2} & \cdots & x^{0}_{\text{data}_2} \\ \vdots & \vdots & \ddots & \vdots \\ x^{N}_{\text{data}_i} & x^{N-1}_{\text{data}_i} & \cdots & x^{0}_{\text{data}_i} \end{bmatrix}, \tag{6.4}$$

where N is the degree of the polynomial function fit. The dimension of each Vandermonde matrix depends on the location of the breakpoints and the number of data points between two adjacent breakpoints. Therefore, the value of the multiplier matrix $\mathbf{C}$ and the constant vector $\mathbf{d}$ for (6.2) are given as

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1.49 & 1.22 & 1 & 0 & 0 & 0 \\ 2.09 & 1.44 & 1 & 0 & 0 & 0 \\ 2.78 & 1.67 & 1 & 0 & 0 & 0 \\ 3.57 & 1.89 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.46 & 2.11 & 1 \\ 0 & 0 & 0 & 5.44 & 2.33 & 1 \\ 0 & 0 & 0 & 6.53 & 2.56 & 1 \\ 0 & 0 & 0 & 7.72 & 2.78 & 1 \\ 0 & 0 & 0 & 9 & 3 & 1 \end{bmatrix} \text{ and } \mathbf{d} = \begin{bmatrix} 2.72 \\ 3.39 \\ 4.24 \\ 5.29 \\ 6.61 \\ 8.26 \\ 10.31 \\ 12.88 \\ 16.08 \\ 20.09 \end{bmatrix}. \tag{6.5}$$

6.2.2 Subfunction Continuity Constraints

In order to meet the function continuity constraints, the gradient and the function evaluation of two adjacent subfunctions must be equal at their conjoining breakpoint. This involves deriving a set of linear equality constraints that equate the geometric progression terms of the two adjacent subfunctions and their gradients. This results in the following general matrix and vector:

$$\mathbf{A}_{\text{eq}} = \begin{bmatrix} \mathbf{v}_{1,1} & -\mathbf{v}_{2,1} & 0 & \cdots & 0 \\ \mathbf{v}'_{1,1} & -\mathbf{v}'_{2,1} & 0 & \cdots & 0 \\ 0 & \mathbf{v}_{2,2} & -\mathbf{v}_{3,2} & \cdots & 0 \\ 0 & \mathbf{v}'_{2,2} & -\mathbf{v}'_{3,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{v}_{i-1,j} & -\mathbf{v}_{i,j} \\ 0 & 0 & 0 & \mathbf{v}'_{i-1,j} & -\mathbf{v}'_{i,j} \end{bmatrix}, \mathbf{b}_{\text{eq}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{6.6}$$

where i is the number of curves/surfaces, j is the number of breakpoints, $\mathbf{v}_{i,j} = [x_{\text{bk}_j}^N, x_{\text{bk}_j}^{N-1}, ..., x_{\text{bk}_j}^0]$ is the geometric progression terms of the piecewise subfunction at x_{bk_j} breakpoint, $\mathbf{v}'_{i,j} = [Nx_{\text{bk}_j}^{N-1}, (N-1)x_{\text{bk}_j}^{(N-1)-1}, ..., 0]$ is the geometric progression terms of the piecewise subfunction gradient at x_{bk_j} breakpoint, and N is the degree of the polynomial function fit. Every odd row of (6.6) deals with the function evaluation continuity, and every even row deals with the gradient continuity. Note that for a linear approximation, all the even rows of $\mathbf{A}_{\text{eq}}$ that deal with gradient continuity are omitted because it will result in a straight line, and the sum of squared errors (SSE) will be very large for highly nonlinear approximations.

Applying (6.6) to the example data in (6.2) where the location of the breakpoint is at 2, the geometric progression terms of the polynomial subfunctions and their corresponding gradients can be written as

$$\mathbf{A}_{\text{eq}} = \begin{bmatrix} 4 & 2 & 1 & -4 & -2 & -1 \\ 4 & 1 & 0 & -4 & -1 & 0 \end{bmatrix} \text{ and } \mathbf{b}_{\text{eq}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{6.7}$$

Once $\mathbf{C}$, $\mathbf{d}$, $\mathbf{A}_{\text{eq}}$, and $\mathbf{b}_{\text{eq}}$ are derived and entered into the `lsqlin` function, the coefficients of the quadratic polynomial subfunctions $\mathbf{x}$ can be solved for the piecewise function. For the example data in (6.2), this results in the following piecewise function:

$$\hat{y}(x) = \begin{cases} 2.13x^2 - 1.76x + 2.35 & : 0 \leq x \leq 2 \\ 5.90x^2 - 16.80x + 17.40 & : 2 < x \leq 3 \end{cases} \tag{6.8}$$

that can be used to approximate the correlation between $\mathbf{x}_{\text{data}}$ and $\mathbf{y}_{\text{data}}$ in (6.2).

In order to automate the piecewise fit discussed above, the algorithm was developed using the object-oriented approach in MATLAB, where the user only requires to enter the data ($\mathbf{x}_{\text{data}}$, $\mathbf{y}_{\text{data}}$), type of fit model, and the breakpoint allocations as follows:

```
pwfitobj = pwfit(xdata,ydata,'poly2',[1,2,3]);
```

The single line of code above will automatically generate the $\mathbf{C}$, $\mathbf{d}$, $\mathbf{A}_{\text{eq}}$, and $\mathbf{b}_{\text{eq}}$ matrices that are specific to the input arguments. The `pwfit` function will then output an object that contains properties of the piecewise function, such as the polynomial coefficients and the sum of squared errors. The `pwfit` object can be passed to one of the `pwfit` methods, namely, `feval`, `print`, and `plot`, to evaluate the piecewise function at a specified point, to print the piecewise fit subfunctions, and to plot the piecewise fit (see Figure 6.1). For instance, the user can type the following lines in the command window:

```
>> feval(pwfitobj,2.2)

ans =

        8.9400

>> print(pwfitobj,'%0.3g')

ans =

        2x1 cell array

        '2.13*xd^2 + -1.76*xd + 2.35'
        '5.9*xd^2 + -16.8*xd + 17.4'

>> plot(pwfitobj)
```

The same pwfit algorithm described above can also be implemented to bivariate functions using polynomial surface fit models, as shown Figure 6.2. However, unlike univariate functions, the input data must be supplied to pwfit as matrices, not vectors, thus the user must enter the data as follows:

```
pwfitobj = pwfit(Xdata,{Ydata,Zdata},'poly21',[2,3.5,5]);
```

The MATLAB code above will generate all the required matrices for the lsqlin function and output the following piecewise function:

$$\hat{z}(x,y) = \begin{cases} 6.26x^2 - 17.8x + 2.7y + 6.85 & : 2 \leq x \leq 3.5 \\ 33.6x^2 - 209x + 2.7y + 342 & : 3.5 \leq x \leq 5 \end{cases} \tag{6.9}$$

that corresponds to Figure 6.2. The current pwfit algorithm can only divide the bivariate function along the x-axis, thus the breakpoints specify the locations on the x-axis where two subfunctions meet. Consequently, the Vandermonde matrices in (6.3) are derived based on where the breakpoints divide the $\mathbf{X}_{\text{data}}$ matrix.

For the example in Figure 6.2, the pwfit algorithm automatically converts $\mathbf{X}_{\text{data}}$ matrix into a single column vector $\mathbf{x}_{\text{data}}$ using the colon operator in MATLAB and divides it into two equal sets of data because the breakpoints are evenly spaced. Subsequently, the $\mathbf{Y}_{\text{data}}$ is also converted into a single column vector and is divided at the same position as where the $\mathbf{x}_{\text{data}}$ vector was divided. The two column vectors are then used to generate the Vandermonde matrices similar to the univariate function example described above. For the function continuity constraints, the $\mathbf{A}_{\text{eq}}$ matrix and the $\mathbf{b}_{\text{eq}}$ vector are derived using the breakpoint, i.e., 3.5 in this case, and the values from the first column of the $\mathbf{Y}_{\text{data}}$ matrix. Lastly, the $\mathbf{d}$ vector in (6.1) is derived by converting the $\mathbf{Z}_{\text{data}}$ matrix into single column vector. These generated

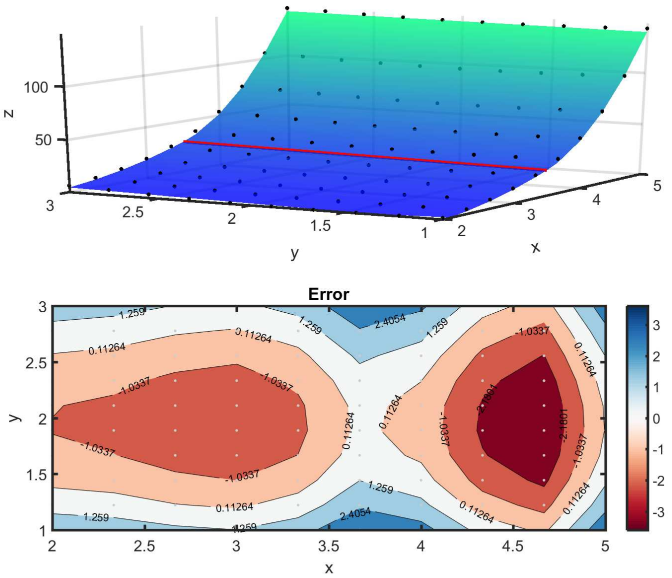


Figure 6.2: Approximating $z(x, y) = -(\sin y)^2 + e^x$ with `pwfit` using two quadratic polynomial surfaces (`poly21`) with a fixed breakpoint at 3.5. The black dots are the input data and the red solid line is the breakpoint.

vectors and matrices are subsequently entered into `lsqlin` function and solved for the coefficients of the piecewise function.

6.2.3 Optimizing Breakpoint Allocations

The accuracy of the regression can be improved by optimizing the location of the breakpoints, which differs from the arbitrarily fixed locations in Section 6.2.1. This is done by formulating an outer optimization problem to allocate the breakpoints in order to minimize the sum of all the SSE of each subfunction. Given that `lsqlin` output the squared 2-norm of the residuals, which is equivalent to the SSE value, and is defined as

$$\texttt{resnorm} = ||\mathbf{C}\mathbf{x} - \mathbf{d}||_2^2, \tag{6.10}$$

this value can be used as part of the objective function for the outer optimization problem. In addition, in order to penalize the distance between two adjacent breakpoints from being too close to one another, the product of the sum of the inverse difference between two adjacent breakpoints and a weighted constant was added to the objective function. Therefore the `pwfit` breakpoint objective function becomes

$$\mathcal{J} = \texttt{resnorm} + w_{\text{bk}} \sum_{i=1}^{n-1} \frac{1}{x_{\text{bk}_i} - x_{\text{bk}_{i+1}}}, \tag{6.11}$$

where `resnorm` is the squared 2-norm of the residuals (or the SSE value) from the `lsqlin`, x_{bk_i} is the breakpoint, w_{bk} is the weighted constant (0.01 by default but can be changed by the user), and n is the number of breakpoints including the minimum and maximum values of the x_{data}, i.e., the breakpoint vector consists of $\mathbf{x}_{\text{bk}} = [x_{\text{data}_{\text{min}}}, \ldots, x_{\text{data}_{\text{max}}}]$.

Each x_{bk_i} is bounded using the minimum and maximum x_{data} value and is subjected to an inequality constraint to ensure that they are not greater than the following breakpoint, which can be expressed as

$$x_{\text{bk}_i} \leq x_{\text{bk}_{i+1}}. \tag{6.12}$$

The breakpoint optimization problem is solve using the optimizer NLopt [130] (algorithm: AUGLAG, subalgorithm: LN_COBYLA), which can be invoked by introducing a fifth input argument, i.e., ‘optimized’, to the `pwfit` function, as shown below:

```
pwfitobj = pwfit(xdata,ydata,‘poly2’,[1,2,3],‘optimized’);
```

Using the data points in (6.2) as an example, `pwfit` will generate the required matrices for the `lsqlin` function, formulate the breakpoint optimization problem and use the fixed point, i.e., [1,2,3], as the initial point for the optimization. Figure 6.3 shows the piecewise fit of the example data in (6.2) with the optimized location of the breakpoint.

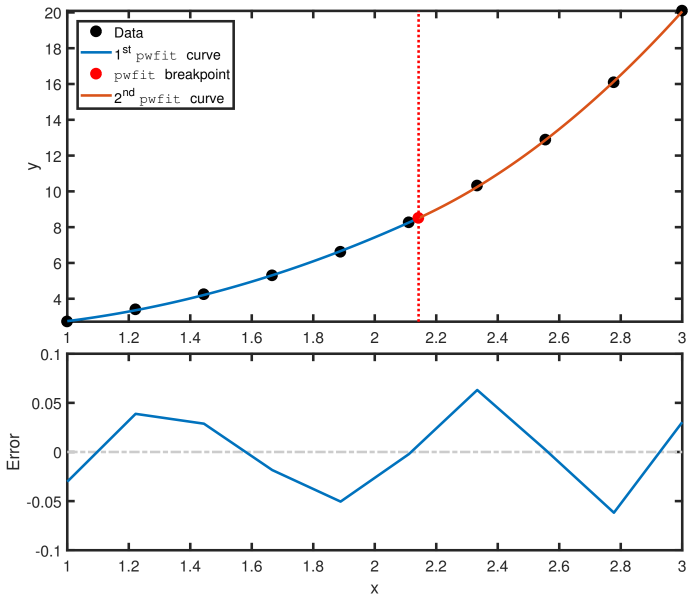


Figure 6.3: Approximating $y(x) = e^x$ with `pwfit` using two quadratic polynomial functions with an optimized breakpoint location.

By optimizing the breakpoint location from 2 to 2.14, the SSE value has decreased from 0.0285 to 0.0149 without increasing the complexity of the piecewise function. As a result, the piecewise approximation for the example data in (6.2) can now be expressed as

$$\hat{y}(x) = \begin{cases} 2.51x^2 - 2.83x + 3.08 & : 0 \leq x \leq 2.14 \\ 6.50x^2 - 19.90x + 21.40 & : 2.14 \leq x \leq 3 \end{cases} \tag{6.13}$$

using the same quadratic polynomial fit functions.

The same optimization problem can also be used to optimize the breakpoint allocation along the x-axis of a bivariate function to improve its accuracy. Using the bivariate function example in Figure 6.2, the SSE of the regression decreased from 239.01 to 162.22 by optimizing the location of the breakpoint. This resulted in a piecewise function that can be expressed as

$$\hat{z}(x, y) = \begin{cases} 10.2x^2 - 38.6x + 2.7y + 33.7 & : 2 \leq x \leq 3.8 \\ 41.4x^2 - 278x + 2.7y + 494 & : 3.8 \leq x \leq 5 \end{cases} \tag{6.14}$$

with the breakpoint optimized from 3.5 to 3.8, as shown in Figure 6.4.

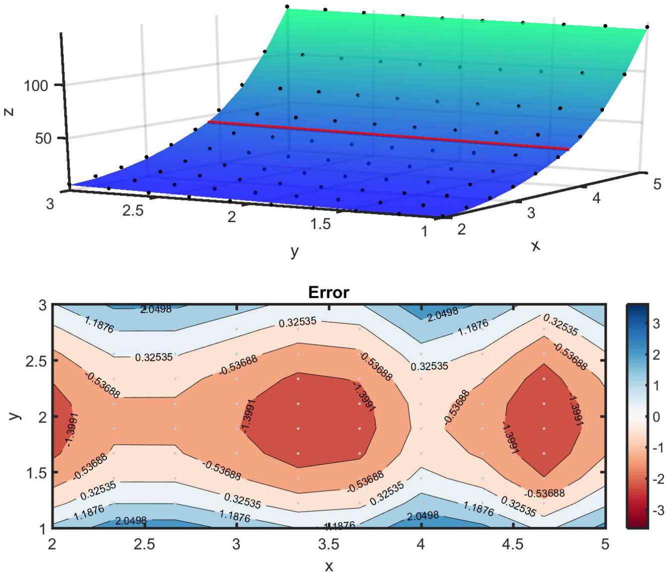


Figure 6.4: Approximating $z(x, y) = -(\sin y)^2 + e^x$ with `pwfit` using two quadratic polynomial surface (`poly21`) with an optimized breakpoint location. The black dots are the input data and the red solid line is the breakpoint.

By optimizing the locations of the breakpoints, the accuracy of the piecewise fit can be improved without increasing the number of fitting curves/surfaces. This can very beneficial when dealing with a large optimization problem that requires many regressions because it reduces the number of auxiliary variables and constraints needed to implement the piecewise fit.

6.2.4 Limitations of the `pwfit` Algorithm

There are some limitations to the current `pwfit` algorithm. First, the subfunction can only be a polynomial function (see Appendix C), and not exponential, power, or trigonometric functions, because it is not possible to construct the Vandermonde Matrix for non-polynomial functions that are not linear in the parameters. While it is acknowledged that there might be problems outside this research where non-polynomial functions could be beneficial, the proposed ORC systems can be approximated quite sufficiently with only polynomial functions. However, it will be on the agenda for future work to incorporate other nonlinear functions. Second, the order/degree of the polynomial model fit has to be same for every subfunction. This means that some parts of the regression might be overcompensated by the high order nonlinear model fit, which can add extra complexity to the optimization problem when a linear fit might suffice. Third, for bivariate functions, the current `pwfit` algorithm can only divide the piecewise function along one of the axes. Therefore, increasing the number of breakpoints might not always improve the accuracy of the regression in some cases.

For this research, the current `pwfit` algorithm is more than sufficient to approximate the thermodynamic properties of the working fluids over the proposed optimization range. The REFPROP thermodynamic data that was generated in Chapter 3 and Chapter 4 did not present extreme cases of nonlinearity or discontinuity that were impossible to approximate using polynomial functions. Furthermore, the bivariate function approximations in this research did not exhibit high nonlinearity along the y-axis (i.e., looking at the plot from the Y-Z view) and thus there was no need to divide the piecewise function along the y-axis.

6.3 Piecewise Function Optimization Formulation

In order to utilize the piecewise fit approximation, it needs to be introduced to the optimization problem via integer programming techniques. A simple way to incorporate the piecewise fit is to make the piecewise function equal to the sum of the product of the subfunction and binary variable, and set all the binary variables as special order set type 1 (SOS1) variables. This will result in the following constraints:

$$\begin{aligned} \hat{f} &= \sum_{i=1}^{n} g_i b_i, \\ \sum_{i=1}^{n} b_i &= 1, \\ b_i &\in \{0,1\}, \end{aligned} \tag{6.15}$$

where $\hat{f}(x)$ is the approximated piecewise function, g_i is the subfunction, and b_i is the binary variable. Since only one binary variable can equal to 1, this will equate $\hat{f}(x)$ to only one of the subfunctions and set all the others to zero. While this approach can be easily implemented, it is inefficient to multiply a binary variable with another variable because it introduces extra nonlinearity into the optimization problem, as noted by [127].

An alternative method is to use a slightly modified integer programming technique from AIMMS modelling guide (in Section 7.7) [127], also known as the Glover's linearization scheme [24]. This involves introducing a new variable v and enforcing it to take the value of $g_i b_i$ using the following linear constraints for each of the subfunctions:

$$\begin{aligned} v_i &\leq u_i b_i, \\ v_i &\leq g_i + M(1 - b_i), \\ v_i &\geq g_i - M(1 - b_i), \\ v_i &\geq l_i b_i, \end{aligned} \tag{6.16}$$

where u_i and l_i are the upper and lower bounds of the corresponding subfunction g_i, M is the "big-M" constant (a constant that has a large value) and b_i is the binary variable. When $b_i = 0$, this implies

$$\left.\begin{aligned} v_i &\leq 0 \\ v_i &\leq g_i + M \\ v_i &\geq g_i - M \\ v_i &\geq 0 \end{aligned}\right\} \Rightarrow v_i = 0, \tag{6.17}$$

and when $b_i = 1$, it implies

$$\left.\begin{aligned} v_i &\leq u \\ v_i &\leq g_i \\ v_i &\geq g_i \\ v_i &\geq l \end{aligned}\right\} \Rightarrow v_i = g_i. \tag{6.18}$$

Consequently, the piecewise function now becomes

$$\hat{f} = \sum_{i=1}^{n} v_i, \tag{6.19}$$

where n is the number of subfunctions. Since only one subfunction can be selected, this can be enforced as follows

$$\begin{aligned} x &\geq \sum_{i=1}^{n} x_{\text{bk}_i} b_i, \\ x &\leq \sum_{i=1}^{n} x_{\text{bk}_{i+1}} b_i, \\ &\sum_{i=1}^{n} b_i = 1, \end{aligned} \tag{6.20}$$

where x is the x-axis variable of the subfunction, x_{bk_i} is the breakpoint value (including the minimum and maximum values of x_{data}, i.e, $x_{\text{bk}_1} = x_{\text{data}_{\text{min}}}$ and $x_{\text{bk}_{n+1}} = x_{\text{data}_{\text{max}}}$), and n is the number of subfunctions.

Applying (6.16), (6.19) and (6.20) to the piecewise function example in (6.13), the first subfunction results in

$$\begin{aligned} &v_1 \leq 8.5b_1, \\ &v_1 \leq (2.51x^2 - 2.83x + 3.08) + 20.1(1 - b_1), \\ &v_1 \geq (2.51x^2 - 2.83x + 3.08) - 20.1(1 - b_1), \\ &v_1 \geq 2.7b_1, \end{aligned} \tag{6.21}$$

and the second subfunction results in

$$\begin{aligned} &v_2 \leq 20.1b_2, \\ &v_2 \leq (6.50x^2 - 19.90x + 21.40) + 20.1(1 - b_2), \\ &v_2 \geq (6.50x^2 - 19.90x + 21.40) - 20.1(1 - b_2), \\ &v_2 \geq 8.5b_2. \end{aligned} \tag{6.22}$$

In this case, the u_i and l_i are equal to the maximum and minimum subfunction evaluations, and M is equal to the maximum piecewise function evaluation (but it can be a larger value). Consequently, the piecewise function can be expressed as

$$\hat{f} = v_1 + v_2, \tag{6.23}$$

and the piecewise function conditions can be formulated as

$$\begin{aligned} x &\geq 1.00b_1 + 2.14b_2, \\ x &\leq 2.14b_1 + 3.00b_2, \\ b_1 + b_2 &= 1. \end{aligned} \tag{6.24}$$

The bivariate piecewise functions can be implemented into an optimization problem in the same matter as illustrated above.

6.4 The Implementation of `pwfit` on ORC Systems

This section will reformulate the three ORC models presented in Chapter 4 using the `pwfit` function to highlight two contributions of the proposed piecewise fit approximation. First, the accuracy of the thermodynamic approximation can be improved by using a piecewise fit instead of a single fit. Second, the performance of certain white-box solvers can be improved by using `pwfit` to decrease the deleterious effect of the nonlinearity of the optimization problem.

6.4.1 Improving the Accuracy of the Optimization Model

In order to demonstrate the first contribution, the enthalpy values of R227ea were calculated over a range of pressures, i.e., $P \in [11.97, 26.31]\,\text{bar}$ at $T = 369.72\,\text{K}$, which is one of the operating constraints of the DOE Pilot Plant model. Figure 6.5 shows the regression fit and error plot of the enthalpy values using a cubic function. As such, the enthalpy can be approximated using

$$\hat{h} = -0.00942P^3 + 0.443P^2 - 8.18P + 449, \tag{6.25}$$

where P is the pressure value. Alternatively, the same enthalpy can be approximated using `pwfit`, as shown in Figure 6.6. As a result, the enthalpy can be approximated using in the following piecewise function:

$$\hat{h} = \begin{cases} -0.0549P^2 + 0.392P + 401 & : 11.97 \leq P_{\text{A1}} \leq 23.14 \\ -0.542P^2 + 22.9P + 140 & : 23.14 \leq P_{\text{A1}} \leq 26.31 \end{cases} \tag{6.26}$$

that has two quadratic polynomial fits.

Comparing the two regression fits, the piecewise fit achieved a lower maximum

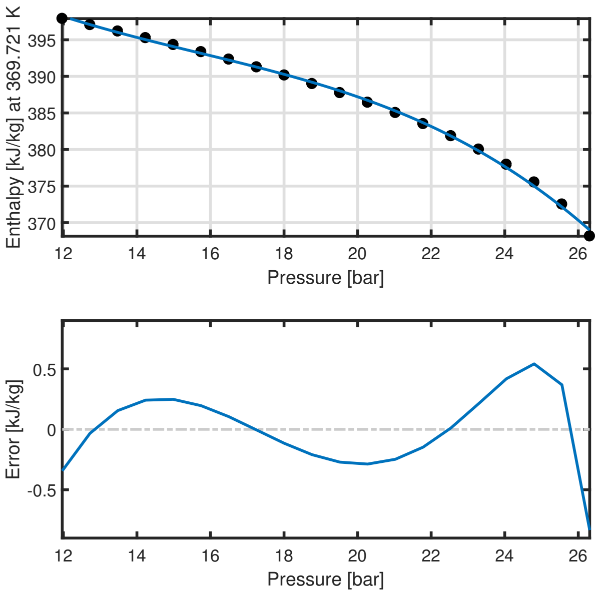


Figure 6.5: Approximating the enthalpy of R227ea at $P \in [11.97, 26.31]$ bar and $T = 369.72\,\mathrm{K}$ using a single fit. The black dots are the REFPROP data and the blue solid line (top plot) is the approximated cubic polynomial fit.

error than the single fit as observed in the two error plots. Also, by calculating the SSE of both regressions, the piecewise fit obtained a lower value of 0.37 compared to the single fit that yielded an SSE of 1.95. This shows that by using a piecewise fit, it is possible to achieve a more accurate approximation than using a single fit, despite using a lower order polynomial fit. This has an important useful implication for problems with multiple regressions that are coupled together by constraints, as will be shown in Section 6.4.2. If the regressions are not accurate and highly nonlinear, they can lead to an unreliable solution, poor optimization performance or, in some cases, can render the optimization problem infeasible.

6.4.2 Reducing the Deleterious Effect of the Nonlinearity of the Optimization Model

From the previous subsection, the accuracy of the regression fit can be improved by using a piecewise fit that has a lower order polynomial fit. This attribute can be used to reduce the deleterious effect of the nonlinearity of the optimization problem without compensating on the accuracy of the model. As will be presented in this subsection, this can improve the performance of some optimization solvers. In order to highlight this contribution, all three case studies in Chapter 5 will be reformulated using piecewise fit approximations.

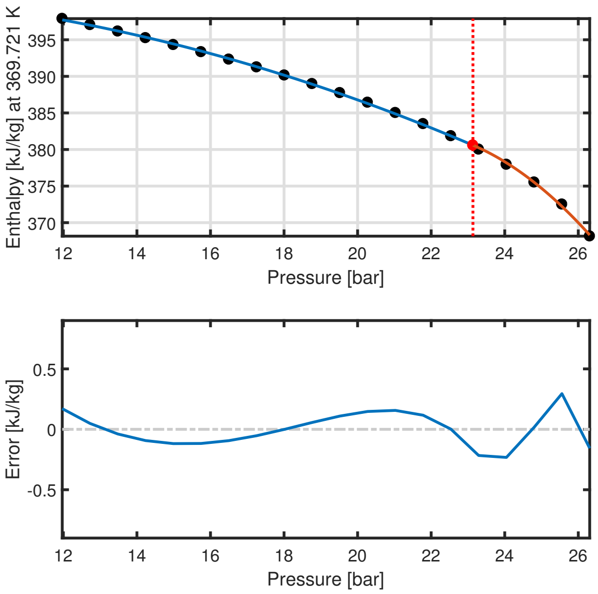


Figure 6.6: Approximating the enthalpy of R227ea at $P \in [11.97, 26.31]$ bar and $T = 369.72$ K using `pwfit`. The black dots are the REFPROP data and the solid lines (top plot) are the approximated quadratic polynomial fits. The error plot is using the same scale as Figure 6.5.

6.4.2.1 DOE Pilot Plant

For the DOE Pilot Plant, all the single cubic polynomial approximations were replaced with piecewise quadratic approximations, which resulted in the following SymBuilder object:

```
SymBuilder Object
BUILT in 2.858s with:
- 45 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 98 constraint(s)
        - 57 linear
        - 41 quadratic
        -  0 nonlinear
- 90 bound(s)
- 12 integer variable(s)
        -  0 integer
        - 12 binary
```

As can be seen, the number of nonlinear constraints has decreased from 6 to 0, but on the other hand, the total number of variables and constraints have increased. Compared to the NLP model in Chapter 5, the number of variables has increased by 24, including 12 binary variables, and the number of constraints increased by 66. While the optimization formulation is technically a mixed-integer quadratically constrained linear program (MIQCLP), it does consist of *equality* quadratic constraints that are not supported by the mixed-integer quadratically constrained quadratic programming (MIQCQP) solvers, like CPLEX and GUROBI. Therefore, an MINLP solver will need to be used to solve this optimization problem. In addition, given that the model will also be optimized with respect to the specific rotating machinery cost, which is a nonlinear function, this optimization formulation will be solved as a mixed-integer nonlinear program.

Optimizing the net output power of the new reformulated algebraic EO model using three MINLP solvers, namely, BARON, SCIP and BONMIN, yielded the optimized results shown in Table 6.1. All three solvers managed to solve the optimization problem under a reasonable time of 2.4s, despite the larger number of variables and constraints. On average, it took BARON around 1.46s and SCIP around 1.16s to find a solution. Comparing these average solve times to their NLP counterparts in Table 5.3, where on average BARON took around 2.95s and SCIP took around 1.76s, it was observed that BARON and SCIP are 2x and 1.5x faster, respectively.

Table 6.1: The optimized net output powers $\mathcal{J}$ [kW] and the solve times [s] of the DOE Pilot Plant using piecewise approximations.

	BARON		**SCIP**		**BONMIN***	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**
R600a	4555.4	1.34	4555.4	0.81	4555.4	0.069
R134a	2844.7	1.35	2844.7	0.95	2844.7	0.085
R152a	3350.2	1.24	3350.2	0.83	3350.2	0.084
R218	1697.9	1.44	1697.9	1.17	1625.7	0.099
R227ea	3360.3	1.42	3360.3	1.85	3217.9	0.082
RC270	3096.3	1.27	3096.3	0.86	3096.3	0.079
RC318	3891.8	2.38	3891.8	2.09	3852.0	0.075
CF3I	4028.6	1.25	4028.6	0.74	4028.6	0.087

*Obtained using outer approximation algorithm.

In contrast, BONMIN did not manage to solve faster than the black-box NLP solvers. On average, it took BONMIN 0.083s to find a solution, whereas it took the NLP solvers (IPOPT, `fmincon`, and FILTERSD) collectively around 0.023s. While this might seem unfavourable for the MINLP formulation, it is important to note that the MINLP and NLP solvers use different strategies to find the optimal solution. Therefore, it is quite hard to compare the solvers' performance when they are fundamentally different in terms of the underlying algorithm, as one can be innately more computationally expensive than the other even when solving the same optimization problem. This is different from the white-box solvers that use the spatial branch and bound algorithm in both formulations. It would be more appropriate to optimize the two formulations with several numbers of solvers of various algorithms and take the average solve time, instead of one MINLP solver (BONMIN) in this case. Furthermore, since the solve times are very small, it would be more ideal to also solve larger and more complex ORC problems to reduce the influence of the solvers' initial non-deterministic setup procedures to the overall solve time. As such, these additional adjustments would give a better view of the optimization performance collectively. However, given the limited number of MINLP solvers available in the OPTI Toolbox and the time and financial constraints to develop OPTI interfaces to other MINLP solvers, this investigation was unable to be carried out but will be left for future work. Therefore, for the remainder of this chapter, BONMIN will only be presented to give the reader a rough idea of the optimization performance and solution of the MINLP formulations using a black-box solver, but it will not be assessed against the NLP solvers.

Taking the problem further and optimizing the plant with respect to the specific rotating machinery cost, the MINLP solvers obtained the following optimized results shown in Table 6.2. As can be seen again, all the solvers managed to converge to a solution with a reasonable solve time of less than 2.6s. On average, it took BARON around 1.32s and SCIP around 1.80s to find a solution. Compared to the NLP formulation in Table 5.5, the average solve time of BARON and SCIP were around 2.72s and 9.70s, respectively, which corresponds to about 2.1x and 5.4x slower than this proposed MINLP formulation. In addition to the improvement in the computational times, the average number of nodes needed to solve the MINLP formulation using the white-box solvers was significantly less than the NLP formulation in Chapter 5, as shown in Table 6.3. This illustrates that the MINLP model with only linear and quadratic constraints can be much more efficient and faster to solve using white-box solvers than its equivalent NLP counterpart that has nonlinear constraints.

Table 6.2: The optimized specific rotating machinery costs $\mathcal{J}$ [k\$/MW] and the solve times [s] of the DOE Pilot Plant using piecewise approximations.

	BARON		**SCIP**		**BONMIN**	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time [s]**
R600a	325.83	1.48	325.83	2.55	325.83	1.03
R134a	367.71	1.05	367.71	1.47	367.71	0.83
R152a	346.19	1.12	346.19	1.93	346.19	0.78
R218	478.16	1.48	478.16	1.97	478.60	0.91
R227ea	368.22	1.77	368.22	1.89	368.22	1.40
RC270	347.73	0.95	347.73	1.62	347.73	0.73
RC318	348.14	1.00	348.14	1.35	348.14	1.02
CF3I	340.00	1.72	340.00	1.60	340.00	1.23

Note that some of the optimized solutions in both Table 6.1 and Table 6.2 are

Table 6.3: The average number of nodes needed for the white-box solvers to optimize the NLP and MINLP formulations of the DOE Pilot Plant.

	BARON		**SCIP**	
	NLP	**MINLP**	**NLP**	**MINLP**
Net Output Power	788	2	255	106
Specific Rotating Machinery Cost	867	6	1654	472

slightly different from their NLP solution counterparts due to the implementation of the `pwfit` approximations, instead of the single fit approximations. As a result, the average discrepancies between the SM model and the optimized algebraic EO model had slightly improved by around 0.02 percentage point, as shown in Table 6.4.

Table 6.4: The average relative errors between the optimized algebraic EO model and the SM model of the DOE Pilot Plant using the white-box solvers.

	NLP	**MINLP**
Net Output Power	0.40%	0.38%
Specific Rotating Machinery Cost	0.17%	0.15%

6.4.2.2 USGeo Plant

For the USGeo Plant, only the turbine regressions were replaced with the piecewise fit approximations. This resulted in the following SymBuilder object:

```
SymBuilder Object
BUILT in 1.957s with:
- 37 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 68 constraint(s)
        - 25 linear
        - 43 quadratic
        -  0 nonlinear
- 74 bound(s)
- 5 integer variable(s)
        -  0 integer
        -  5 binary
```

Similar to the DOEP model, the new USGP formulation has no nonlinear constraints but a higher number of variables and constraints than the NLP formulation. In total, the number of variables has increased by 10 with 5 of them being binary variables, and the number of constraints has increased by 26.

Optimizing the net output power of the new USGP formulation, the MINLP solvers obtained the following solutions and solve times shown in Table 6.5. As can be observed, all the solvers managed to converged a solution within 3s for all the working fluids. On average, it took BARON around 1.28s and SCIP around 1.47s to find a solution. This is about 3.8x and 6.5x faster than the NLP formulation, where the average solve time of BARON and SCIP were around 4.92s and 9.53s, respectively.

Table 6.5: The optimized net output powers $\mathcal{J}$ [kW] and the solve times [s] of the USGeo Plant using piecewise approximations.

	BARON		**SCIP**		**BONMIN**	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**
R601a	14311.1	1.45	14311.1	2.23	14311.1	0.46
R245ca	12842.9	1.19	12842.9	1.59	12842.9	0.38
R245fa	10852.9	1.42	10852.9	1.09	10852.9	0.39
C5F12	11629.4	1.08	11629.4	0.96	11629.4	0.42

The same conclusion can also be observed from optimizing the specific rotating machinery cost of the plant. As shown in Table 6.6, the average solve time of BARON was around 0.80s and SCIP was around 0.85s. Consequently, this is about 1.5x and 7.8x faster than the NLP formulation, where the average solve time of BARON and SCIP were around 1.19s and 6.62s, respectively.

Table 6.6: The optimized specific rotating machinery costs $\mathcal{J}$ [k\$/MW] and the solve times [s] of the USGeo Plant using piecewise approximations.

	BARON		**SCIP**		**BONMIN**	
	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**	$\mathcal{J}$	**Time**
R601a	213.80	0.94	213.80	0.94	213.80	0.29
R245ca	215.37	0.62	215.37	0.83	215.37	0.24
R245fa	226.71	0.96	226.71	0.98	226.71	0.16
C5F12	216.19	0.66	216.19	0.66	216.19	0.25

In addition, similar to the DOE Pilot Plant, the average number of nodes needed to solve the MINLP formulation was significantly smaller than the NLP counterpart in Chapter 5, as shown in Table 6.7. Furthermore, along with improving the performance of the optimization, the accuracy of the algebraic EO model was not compromised and improved slightly by up to 0.11 percentage point, as shown in Table 6.8.

Table 6.7: The average number of nodes needed for the white-box solvers to optimize the NLP and MINLP formulations of the USGeo Plant.

	BARON		**SCIP**	
	NLP	**MINLP**	**NLP**	**MINLP**
Net Output Power	875	5	7324	138
Specific Rotating Machinery Cost	14	4	1548	244

Table 6.8: The average relative errors between the optimized algebraic EO model and the SM model of the USGeo Plant using the white-box solvers.

	NLP	**MINLP**
Net Output Power	0.15%	0.04%
Specific Rotating Machinery Cost	0.03%	0.03%

6.4.2.3 Magmamax Binary Power Plant

The Magmamax Binary Power Plant had the most nonlinear constraints out of all the ORC plants, thus this resulted in a higher number of variables and constraints than the previous two plants when the piecewise fit approximations were implemented. Using the SymBuilder framework, the new MBPP formulation resulted in the following object:

```
SymBuilder Object
BUILT in 4.817s with:
- 68 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 153 constraint(s)
        - 83 linear
        - 70 quadratic
        -  0 nonlinear
- 136 bound(s)
- 21 integer variable(s)
        -  0 integer
        - 21 binary
```

where the number of nonlinear constraints has decreased from 9 to 0, but the total number of variables and constraints have increased by 42 and 111, respectively.

Optimizing the net output power of the plant using the new formulation, the MINLP solvers obtained the following optimized results shown in Table 6.9. Similar to the NLP formulation, both BARON and SCIP managed to find a solution for all the working fluid pairs within the solve time limit of 1 hour. Likewise, BONMIN also managed to solve for all the working fluid pairs but converged to a local optimum for one of the cases.

Table 6.9: The optimized net output powers [kW] of the Magmamax Binary Power Plant using piecewise approximations.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	9625.3	9157.3	9268.9
	R236ea	9641.8	9205.3	9310.9*
	R236fa	9282.5	8766.6	8890.4
	RC318	8553.8	7854.8	8027.1

*BONMIN: 9308.2 kW

As shown in Table 6.10, it took BARON and SCIP on average around 5.34s and 11.43s, respectively, to find a solution, which is around 27x and 3x faster than using the NLP formulation. This is a significant improvement in the performance for the white-box solvers.

Table 6.10: The solve times [s] of Table 6.9.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	4.84	3.30	7.51
		R236ea	4.19	5.19	8.70
		R236fa	3.85	3.85	7.39
		RC318	5.45	3.22	6.59
	SCIP	R600a	4.61	6.38	8.35
		R236ea	22.61	6.48	11.46
		R236fa	6.45	21.26	10.36
		RC318	24.93	4.47	9.77
	BONMIN	R600a	7.44	5.03	7.18
		R236ea	7.08	4.53	8.95
		R236fa	7.68	5.30	6.39
		RC318	8.01	5.48	6.58

Taking the optimization problem further and optimizing the specific rotating machinery cost of the plant, the MINLP solvers obtained the following optimized results in Table 6.11 and the solve times in Table 6.12. Both white-box solvers managed to converge to a solution for all the working fluid pairs within the maximum solve time limit of 1 hour, as opposed to the NLP formulation. The optimization time of BARON and SCIP is on average 5.96s and 7.02s, respectively, which is around 70x and 85x faster than the NLP formulation. In addition, they required less number of nodes to solve the MINLP formulation than the NLP formulation for both objective functions, as shown in Table 6.13. Furthermore, for the second objective function, BONMIN also managed to converge the same solution as the white-box solver for all the working fluid pairs.

Table 6.11: The optimized specific rotating machinery costs [k$/MW] of the Magmamax Binary Power Plant using piecewise approximations.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	252.99	247.02	252.96
	R236ea	247.66	241.15	247.36
	R236fa	257.24	251.13	257.67
	RC318	276.88	273.05	280.02

Generally speaking, it is possible for an optimization problem to have more than one global optimum, i.e., the ORC can output the same global solution at two or more different operating points. Therefore, it is not impossible for both BARON and SCIP to find a global solution but at a different operating point from each other, such as at a different enthalpy and pressure level. Consequently, when the optimized solutions of the algebraic EO model are entered into the SM model, there is a possibility that there could be a slight difference between the global solutions because the errors of the regressions are not constant over the approximated range.

Table 6.12: The solve times [s] of Table 6.11.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	5.29	2.20	3.99
		R236ea	5.97	1.85	4.20
		R236fa	15.77	5.35	11.82
		RC318	9.31	4.73	10.28
	SCIP	R600a	4.09	3.36	6.07
		R236ea	4.35	2.51	4.63
		R236fa	18.80	7.77	9.17
		RC318	6.74	8.78	7.92
	BONMIN	R600a	3.18	2.36	4.43
		R236ea	2.39	2.89	2.62
		R236fa	4.62	5.86	6.32
		RC318	4.37	3.76	5.25

Given that the accuracy of the NLP formulation of the MBPP is already considerably small due to the use of high order polynomial function fits, there is not a significant improvement in the accuracy of the MINLP formulation, as shown in Table 6.14. However, this illustrates that the piecewise fit approximation can significantly improve the optimization performance without compromising on the accuracy of the model.

Table 6.13: The average number of nodes needed for the white-box solvers to optimize the NLP and MINLP formulations of the Magmamax Binary Power Plant. This table only considers the working fluid pairs that were solved in both formulations.

	BARON		**SCIP**	
	NLP	**MINLP**	**NLP**	**MINLP**
Net Output Power	23927	22	11066	3286
Specific Rotating Machinery Cost	54270	165	443728	2121

Table 6.14: The average relative errors between the optimized algebraic EO model and the SM model of the Magmamax Binary Power Plant using the white-box solvers. This table only considers the working fluid pairs that are solved in both NLP and MINLP formulations.

	BARON		**SCIP**	
	NLP	**MINLP**	**NLP**	**MINLP**
Net Output Power	0.05%	0.04%	0.05%	0.04%
Specific Rotating Machinery Cost	<0.01%	0.01%	<0.01%	0.01%

6.5 Investigating the Effect of the Working Fluid Mixtures

In addition to optimizing the ORC system with respect to pure working fluids, the algebraic EO model can be used to investigate the effect of the working fluid mixtures and the composition ratio that will output the best performance. This investigation illustrates one of the contributions of the proposed modelling approach where the optimization problem can be extended further to efficiently exploit the existing thermodynamic data of different mixtures that are readily available in REFPROP and other thermodynamic packages. This adds another dimension to the optimization problem and allows the practitioners to explore this non-trivial mixture analysis of the ORC system without making any major changes to the model.

This study can be carried out very efficiently using the MINLP formulation given the improvement in the optimization performance from utilizing piecewise fit approximations. In order to demonstrate this contribution, the DOE Pilot Plant was used as an example to show the effect of different mixtures and their compositions on the net output power. As will be shown later in this section, the proposed MINLP model presented in this chapter allows a large ORC system to be optimized 215 times within a reasonable computational time frame. The investigation presented in this section exclusively considers the effect of the working fluid mixtures on the plant's net output power, and not the cost, environmental impact, safety, and other aspects of the working fluid mixtures as they are outside the scope of this research and are subjective to the location and build of the plant.

For this optimization problem, the selected working fluid mixtures consisted of two working fluids where the composition ratio was increased by 5% increments from 0% to 100%, and the optimization problem was solved at every increment using BARON. Given that mixtures change phase at variable temperatures, different from the pure fluids that condense and evaporate at a constant temperature, the pressure value at the condenser's outlet was calculated using the saturated enthalpy and entropy value at 311.48 K. This is to address the issue with JSteam that outputs the saturated vapour pressure value by default (see Figure 6.7), instead of the saturated liquid pressure, when the temperature value is used directly in the thermodynamic routine (i.e, `JSteamMEX('PsatmT',mixture,311.48)`). Consequently, the optimized results for the pure fluids in the tables below will be slightly different from what is stated in the previous section as there is a slight difference between the condenser's outlet pressure.

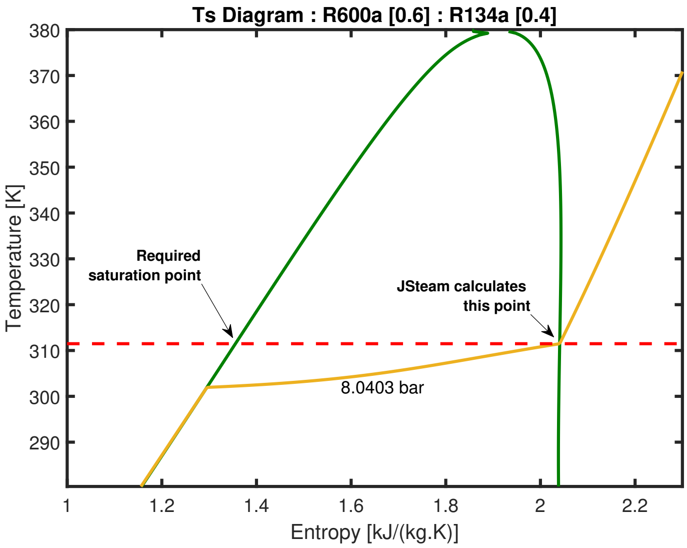


Figure 6.7: The Ts diagram of a mixture with 60% R600a and 40% R134a, where the mixture evaporates and condenses at variable temperatures.

Table 6.15 shows the maximum optimal net output powers of the DOEP for different mixtures, where the composition of one of the fluids, namely, the ones stated in the column, was changed from 0% to 100%. The composition of each mixture that corresponds to the highest net output power is shown in Table 6.16. Interestingly, most of the mixtures did not obtain a higher net output power than the pure fluid. While the composition of the mixtures did affect the net output power of the plant, the highest net output power for most mixtures was obtained when the composition of one of the fluids was either 0% or 100%. Out of all the 10 mixtures that were investigated, only one yielded a higher net output power than its pure fluid components, namely, RC318 and CF3I.

Table 6.15: The maximum optimal net output powers [kW] of the DOE Pilot Plant that were obtained from varying the composition of the column fluid from 0% to 100%, where the optimization problem was solved using BARON.

	R600a	**R134a**	**RC270**	**RC318**	**CF3I**
R600a	4541.31	4541.31	4541.31	4541.31	4541.31
R134a		2831.00	3090.13	3863.23	3993.12
RC270			3090.13	3863.23	3993.12
RC318				3863.23	**4004.22**
CF3I					3993.12

Table 6.16: The composition of each mixture that obtained the highest net output power. The ratio number on the left and right associates with the row and column fluid, respectively. For example 15%:85% means 15% RC318 and 85% CF3I.

	R600a	**R134a**	**RC270**	**RC318**	**CF3I**
R600a	100%	100%:0%	100%:0%	100%:0%	100%:0%
R134a		100%	0%:100%	0%:100%	0%:100%
RC270			100%	0%:100%	0%:100%
RC318				100%	**15%:85%**
CF3I					100%

Figure 6.8 shows the trendline of the optimized net output powers as the composition of one fluid in the mixture changes from 0% to 100%. Evidently, a different composition ratio results in a different output power, and the trendline is not the same for every mixture. There are two types of correlations that can be observed from changing the composition of the mixtures. First, some of the trendlines resemble a linear correlation, as shown by R600a:RC270, and RC270:CF3I, where the net output power either decreases or increases as the mixture composition changes from 0% to 100%. Second, some of the trendlines resemble a concave correlation, as shown by R600a:R134a and RC318:CF3I, where there is a minimum or maximum net output power at a specific composition ratio as the mixture shift from one pure fluid to another.

The total build time for this optimization problem was 300.09s and the total solve time was 287.73s. This is very impressive given that in total there were 215 optimization problems that were solved, where 210 were for mixtures and 5 were for pure fluids. Furthermore, the optimization problems were solved using BARON, thus the overall solve time would be even faster if a derivative-based MINLP solver was used.

A major limitation associated with this study depends on the accuracy and uncertainty of REFPROP to reliably calculate the thermodynamic properties of the working fluid mixtures. While REFPROP is developed by the National Institute of

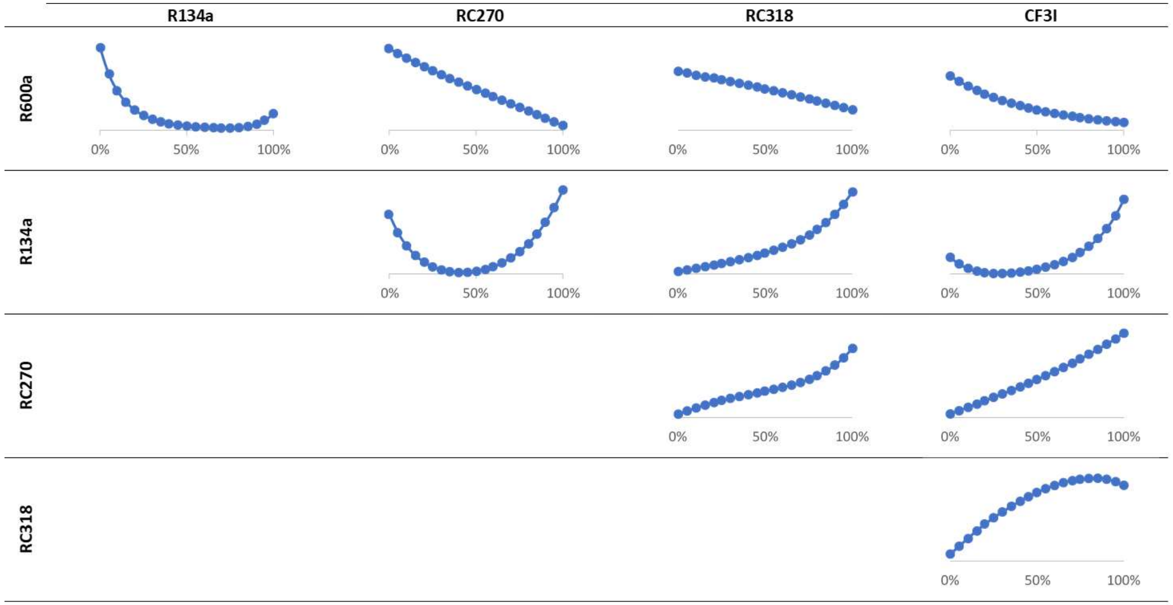


Figure 6.8: The trendline of the optimized net output powers of the DOE Pilot Plant subjected to different mixture composition ratios. The abscissa indicates the percentage of the working fluid of that column.

Standards and Technology (NIST), and based on the most accurate models available for pure fluids and mixtures, it is not perfect and can cause issues for the optimization problem. It is stated in the REFPROP manual (version 9.1) that "certain calculations, especially saturation calculations, may fail without generating warnings" [35], thus in some cases the optimization problem might fail as the enthalpy or pressure cannot be calculated. This is the reason why the number of working fluids for the DOEP was reduced from 9 to 5 because REFPROP failed to calculate the required thermodynamic properties for the corresponding mixtures.

It is important to again highlight that there is no one best working fluid that can meet all the criteria of an ORC system and some compromises have to be made, and this undoubtedly extends to mixtures. Furthermore, given the possible number of fluids that can be used in a mixture and the variation in the composition, selecting the optimal mixture is not a trivial task and will have to rely on an efficient optimization approach, like the one proposed in this thesis, to investigate different combinations within a manageable time frame.

6.6 Summary

This chapter details the implementation of the piecewise fit approximation in order to improve the accuracy of the algebraic EO model and the performance of certain white-box solvers. The piecewise fit algorithm `pwfit` and the optimization formulation were discussed and implemented in MATLAB using the object-oriented approach. With the proposed `pwfit` class, both univariate and bivariate functions can be approximated using generic polynomial functions. The available fit models are listed in Appendix C. In order to further improve the accuracy of the regression, the breakpoints can be specified manually by the user or they can be optimized using a built-in feature that aims to minimize the total SSE value. Thus, this can help reduce the order of the polynomial fits, reduce the overall number of curve/-surface fits, and decrease the complexity of the optimization problem by lowering the number of auxiliary variables and constraints.

All three ORC systems in Chapter 5 were reformulated using the piecewise fit approximation and optimized using three MINLP solvers, namely, BARON, SCIP, and BONMIN. As a result, this converted the optimization problem from an NLP problem to an MINLP problem due to the introduction of binary variables. While the new formulations had more variables and constraints, the overall nonlinearity of the optimization problem was decreased by using a lower order polynomial fit without compromising on the accuracy of the model. From the optimized results, it was found that the performance of the white-box solvers had improved significantly from the NLP formulation counterparts, especially for the Magmamax Binary Power Plant. Leveraging off the excellent performance of the white-box solvers using this formulation, the model of the DOE Pilot Plant was optimized with respect to different mixtures and compositions using BARON. The results showed that in most cases the pure fluids yielded a better result than the mixtures. Furthermore, some mixtures had a linear correlation between the composition ratio and the optimal solution, while others resembled a concave correlation.

The following chapter will build on the implementation of the piecewise fit approximation and extend it to consider piecewise linear fit approximation. The incentive is to investigate if the performance of the white-box solvers can be further improved by using a mixed-integer linear programming (MILP) formulation. In addition, there are more MILP solvers than there are MINLP solvers, thus providing an alternative method and the tools to model the ORC systems or other optimization problems outside this research would be beneficial for cases when the NLP or MINLP solvers fail.

Chapter 7

Mixed-Integer Linear Programming Formulation

This chapter will discuss the implementation of various linear and integer programming techniques that converts a nonlinear programming (NLP) model into an equivalent mixed-integer linear programming (MILP) model. The key incentive of this chapter is to investigate the performance of the MILP formulation compared to the NLP and MINLP formulation, and also to provide an alternative modelling approach for ORC systems that can exploit the availability of MILP solvers. The chapter will start with a detailed description of the MILP formulation modelling procedure that implements piecewise linear approximations and different linear and integer programming techniques. This will be applied first to a simple optimization problem and then later to the Magmamax Binary Power Plant of Section 4.4.1. Lastly, the complexity and the optimized results of the MILP model will be discussed and compared to the NLP and MINLP counterpart in Chapters 5 and 6.

7.1 Introduction

Often one type of optimization formulation can be more advantageous than the others due to financial constraints, compatibility issues with the modelling software, the availability of optimization solvers and/or the importance of establishing global optimality. Therefore, practitioners invest a considerable amount of time reformulating their optimization problem into another equivalent formulation in order to achieve better practical or optimization performance, as shown in [131, 132]. Historically, the availability and computational efficiency of linear programming (LP) solvers, or even mixed-integer linear programming solvers, were significantly better than some nonlinear programming solvers, which was a key incentive to reformulate nonlinear optimization problems to LPs or MILPs.

In general, it can be very difficult to predict if the nonlinear solver will converge to a solution, especially with highly nonlinear and complex problems. This is evident in Section 5.4 where the global solvers (namely, BARON and SCIP) failed to find a solution for most the MBPP's working fluid pairs within the optimization time limit of one hour. Therefore, it would be beneficial to provide an alternative modelling approach for ORC systems in replacement of the NLP and MINLP models and potentially improve the optimization performance. However, given the recent advancement in nonlinear optimization algorithms and the improvement in optimization modelling platforms and computational performance of desktop hardware, it is questionable if there are now sufficient compelling advantages for certain real-world nonlinear chemical process systems to be completely reformulated into LPs or MILPs.

In order to formulate the MILP model, the thermodynamic properties, the unit operations and the bilinear terms in the energy balance equations need to be regressed using piecewise linear approximations in conjunction with linear and integer programming techniques. This inevitably will introduce additional auxiliary binary and continuous variables and constraints that will need to be incorporated into the optimization problem. This chapter also aims to compare the computational advantages and disadvantages of linear and nonlinear formulations, and indicate if it is advisable to formulate linear models for real-world ORC systems.

7.2 Mixed-Integer Linear Programming Formulation Procedure

The following subsections will discuss the implementation of linear and integer programming techniques in order to convert the NLP model into an equivalent MILP model. This assumes that the NLP model is formulated using the modelling approach discussed in Chapter 3. While this chapter focuses on the conversion from NLP to MILP model, the same concept would also be applicable to the MINLP models in Chapter 6. In addition, all the functions in this proposed MILP formulation are algebraic and provided explicitly to the solvers, thus they will be compatible with global solvers that can deterministically find the global solution. However, it is important to note the accuracy of the MILP will likely to be compromised due to the linear approximations. While the accuracy can be improved by increasing the number of piecewise breakpoints, this can add extra complexity by introducing more additional auxiliary variables and constraints that can make the problem difficult to solve and troubleshoot.

7.2.1 Nonlinear Term Approximations

All the nonlinear terms in the NLP model will need to be identified and approximated using linear approximations. For this research, these nonlinear terms are mainly the thermodynamic routines, unit operation processes, and cost functions. While it is possible to use a piecewise linear fit to approximate the thermodynamics and unit operations via `pwfit`, the errors can become too large in some cases for bivariate functions and cause a degrading concern to the model's accuracy. This is because `pwfit` can only divide the breakpoints of the piecewise function along one of the axis and not both at the same time. Therefore, in order to address this issue, an alternative method is needed for the MILP formulation. One of the best piecewise linear approximations is the Triangle Method described in [128].

A comparison between the `pwfit` method and the Triangle Method shows that the errors are significantly less for the Triangle Method, as shown in the error plots of Figure 7.1 and Figure 7.2. In addition, the sum of squared errors (SSE) has decreased from 22.9 to 13.2 from implementing the Triangle Method. However, the drawback with the Triangle Method is the number of auxiliary variables (both binary and continuous) and constraints are higher than the `pwfit` method for the same number of breakpoints along the x-axis. Therefore, the Triangle Method will only be used to approximate the nonlinear terms that cannot be accurately approximated using the `pwfit` approach.

The Triangle Method can be incorporated into the optimization problem using the following formulation:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}f(x_i,y_j) &= \hat{f},\\ \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}x_i &= x,\\ \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}y_j &= y,\\ \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij} &= 1, \qquad \alpha_{ij}\in[0,1],\end{aligned} \tag{7.1}$$

where α_{ij} is the auxiliary variable associated with each breakpoint (i.e., the 16 black dots in the top plot of Figure 7.2), x_i is the fixed breakpoint value along the x-axis, y_j is the fixed breakpoint value along the y-axis, $\hat{f}$ is the approximated value of

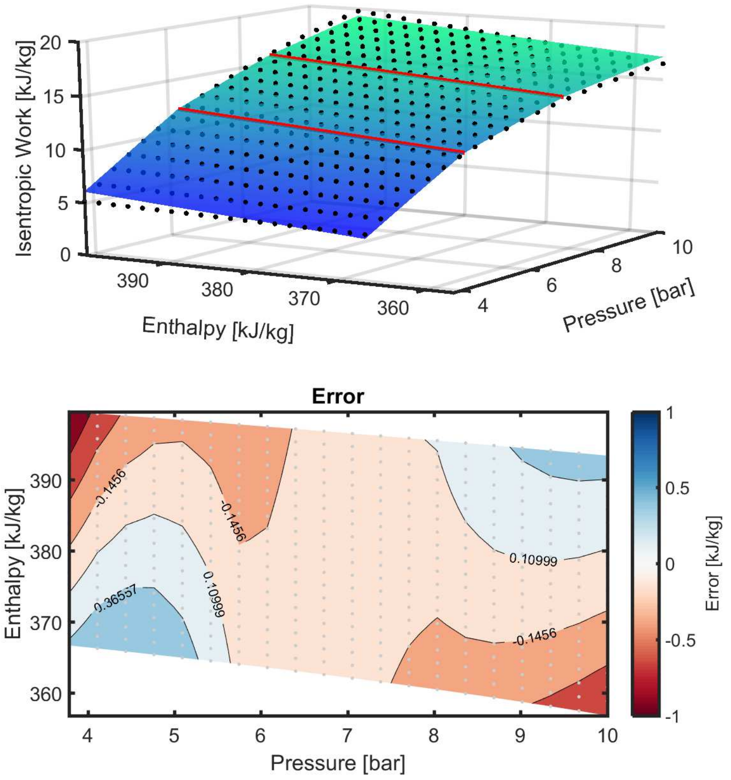


Figure 7.1: Approximating the turbine isentropic work of the Basaran ORC system using the `pwfit` approach with linear surface fits, where the working fluid was R227ea.

the nonlinear function evaluation at point x and y, and $f(x_i, y_j)$ is the function evaluation at the breakpoints x_i and y_j. For the above formulation to work, the α_{ij} variables need to be defined as special order set type 3 (SOS3) variables, thus at most three adjacent α_{ij} variables can be nonzero. This can be enforced using the following constraints:

$$\begin{aligned}
&\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}(b^u{}_{ij} + b^l{}_{ij}) = 1,\\
&\alpha_{ij} \leq b^u{}_{ij} + b^l{}_{ij} + b^u{}_{i,j-1} + b^l{}_{i-1,j-1} + b^u{}_{i-1,j-1} + b^l{}_{i-1,j},\\
&i = \{1, 2, 3, ..., n\},\\
&j = \{1, 2, 3, ..., m\},
\end{aligned} \tag{7.2}$$

where b^u_{ij} and b^l_{ij} are the binary variables associated with the upper and lower triangle, respectively, and the variables $b^*{}_{0*}$, $b^*{}_{*0}$, $b^*{}_{n*}$, and $b^*{}_{*m}$ are set to zero.

Conversely, for the nonlinear univariate functions, such as the thermodynamic properties of the working fluid shown in Figure 7.3, they will be approximated using the `pwfit` method. This is because the accuracy of the approximations is generally

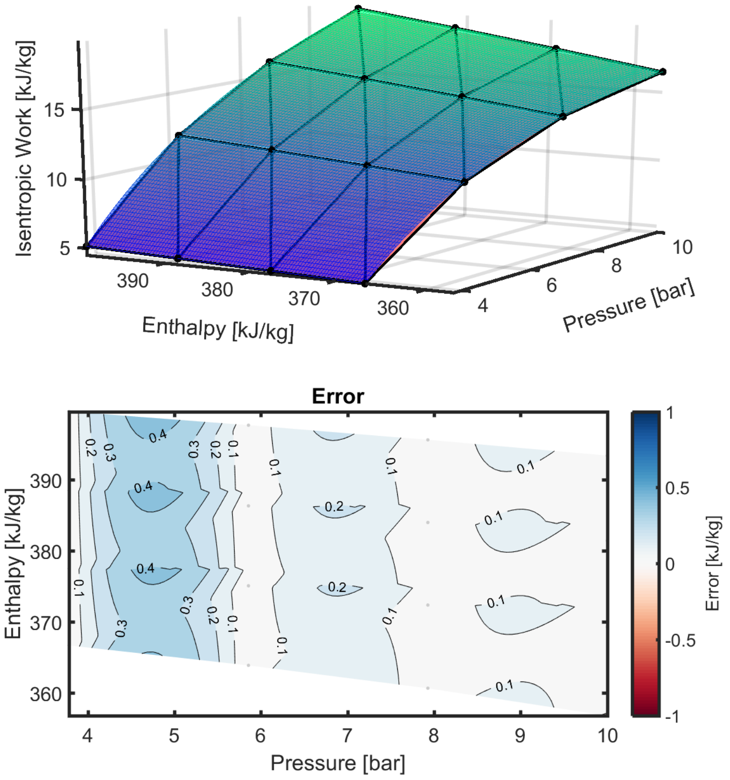


Figure 7.2: Approximating the turbine isentropic work of the Basaran ORC system using the Triangle Method, where the working fluid was R227ea.

sufficient for the purpose of this research.

7.2.2 Bilinear Term Approximation

Once all the nonlinear functions have been approximated using the Triangle Method or the `pwfit` approach, there will be bilinear terms remaining in the optimization problems that are mainly the products of the mass and enthalpy term in the energy balance equations. Since some of the bilinear terms are nonseparable functions, they need to be converted into a separable function as follows:

$$m_{\mathrm{A1}} h_{\mathrm{A1}} = {y_1}^2 - {y_2}^2, \tag{7.3}$$

where

$$\begin{aligned} y_1 &= \frac{1}{2}(m_{\mathrm{A1}} + h_{\mathrm{A1}}), \\ y_2 &= \frac{1}{2}(m_{\mathrm{A1}} - h_{\mathrm{A1}}). \end{aligned} \tag{7.4}$$

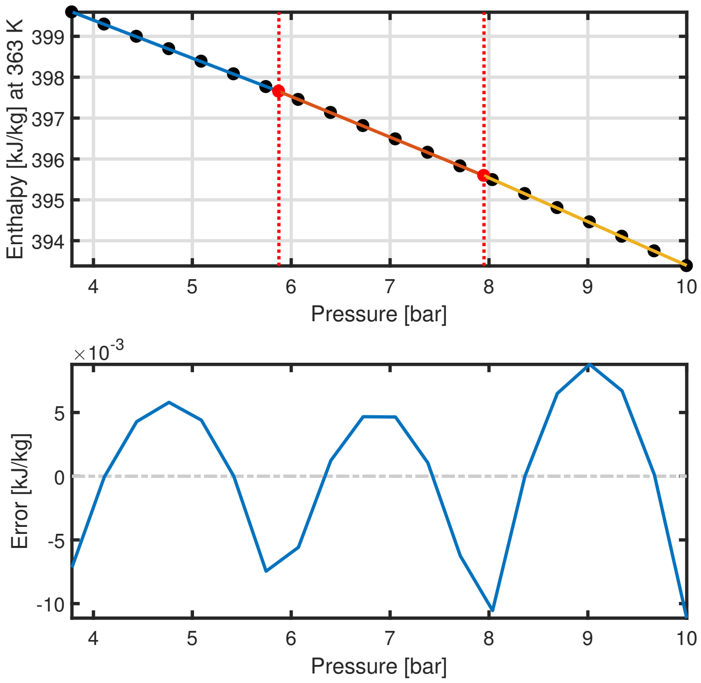


Figure 7.3: Piecewise linear approximation of R227ea enthalpy values at $P \in [3.78, 10]$ bar and $T = 363\,\mathrm{K}$.

If $l_{m_{\mathrm{A1}}} \leq m_{\mathrm{A1}} \leq u_{m_{\mathrm{A1}}}$ and $l_{h_{\mathrm{A1}}} \leq h_{\mathrm{A1}} \leq u_{h_{\mathrm{A1}}}$, both y_1 and y_2 are bounded as

$$\begin{aligned} \frac{1}{2}(l_{m_{\mathrm{A1}}} + l_{h_{\mathrm{A1}}}) \leq y_1 \leq \frac{1}{2}(u_{m_{\mathrm{A1}}} + u_{h_{\mathrm{A1}}}), \\ \frac{1}{2}(l_{m_{\mathrm{A1}}} - u_{h_{\mathrm{A1}}}) \leq y_2 \leq \frac{1}{2}(u_{m_{\mathrm{A1}}} - l_{h_{\mathrm{A1}}}). \end{aligned} \tag{7.5}$$

Once the bilinear term is converted into the form shown in (7.3), the separable terms (i.e., ${y_1}^2$ and ${y_2}^2$) can be approximated using the λ-Formulation [127, 128] (see Figure 7.4) as follows:

$$\begin{aligned} \sum_{i=1}^{n} \lambda_i f(x_i) = \hat{f}, \\ \sum_{i=1}^{n} \lambda_i x_i = x, \\ \sum_{i=1}^{n} \lambda_i = 1, \qquad \lambda_i \in [0, 1], \end{aligned} \tag{7.6}$$

where $\hat{f}$ is the approximated value of the separable term evaluation at x, $f(x_i)$ is the separable term function evaluation at the breakpoint x_i, and λ_i is the auxiliary variable that need to defined as a special order set type 2 (SOS2). This can be enforced as follows:

$$\begin{aligned} &\sum_{i=1}^{n-1} b_i = 1, \\ &\lambda_i \leq b_{i-1} + b_i, \qquad i = \{1,2,3,...,n\}, \end{aligned} \tag{7.7}$$

where b_i is the binary variable associated with the interval between two adjacent breakpoints, and the variables b_0 and b_n are set to zero. The λ-Formulation (7.6) can be viewed as the two-dimensional space of the Triangle Method formulation given the similarities between the two formulations.

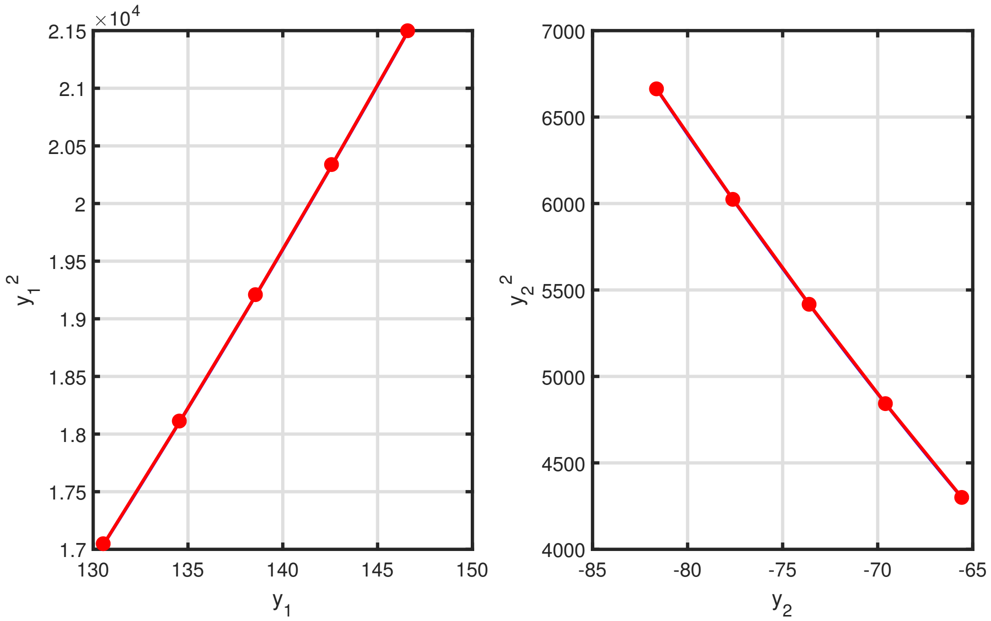


Figure 7.4: Piecewise linear approximation of ${y_1}^2$ and ${y_2}^2$ in (7.3) using the λ-Formulation.

In general, a product of two variables can be replaced with one variable by equating/constraining the two approximated separable terms to a new single variable, i.e., $x = \hat{f}_1 - \hat{f}_2 \approx {y_1}^2 - {y_2}^2$, thus this methodology can easily be extended to products of more than two terms. However, it is important to note that the accuracy of the overall approximation will decrease as the number of variables increases.

7.2.3 Fractional Objective Function

For optimization problems dealing with the rate of return, productivity ratios, and economic analysis, the objective function is usually a fractional term, which is a nonlinear function even if it is a ratio of two linear terms. In order to convert the model into a regular linear model, the Charnes-Cooper transformation [25] can be implemented, provided that both the numerator, denominator, and constraints are all linear terms. However, since there are binary variables in the proposed ORC models due to the piecewise fit approximations, the Glover's linearization scheme [24] needs to also be incorporated into the optimization problem.

If all the nonlinear terms have been linearized up to this point, the optimization problem should be

$$\begin{aligned} \min \quad & \frac{\mathbf{a}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} + c}{\mathbf{d}^\top \mathbf{x} + \mathbf{e}^\top \mathbf{y} + f} \\ \text{s.t.} \quad & \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{y} \leq \mathbf{k}, \\ & \mathbf{G}_{\text{eq}}\mathbf{x} + \mathbf{H}_{\text{eq}}\mathbf{y} = \mathbf{k}_{\text{eq}}, \\ & \mathbf{x} \in \mathbb{R}, \\ & \mathbf{y} \in \{0, 1\}, \end{aligned} \tag{7.8}$$

where $\mathbf{d}^\top \mathbf{x} + \mathbf{e}^\top \mathbf{y} + f > 0$. If a new variable t is introduced and equated to

$$t = \frac{1}{\mathbf{d}^\top \mathbf{x} + \mathbf{e}^\top \mathbf{y} + f}, \tag{7.9}$$

the fractional linear optimization problem in (7.8) can be written in terms of t as

$$\begin{aligned} \min \quad & \mathbf{a}^\top \mathbf{x}t + \mathbf{b}^\top \mathbf{y}t + ct \\ \text{s.t.} \quad & \mathbf{G}\mathbf{x}t + \mathbf{H}\mathbf{y}t \leq \mathbf{k}t, \\ & \mathbf{G}_{\text{eq}}\mathbf{x}t + \mathbf{H}_{\text{eq}}\mathbf{y}t = \mathbf{k}_{\text{eq}}t, \\ & \mathbf{d}^\top \mathbf{x}t + \mathbf{e}^\top \mathbf{y}t + ft = 1, \\ & \mathbf{x}t \in \mathbb{R}, \\ & \mathbf{y} \in \{0, 1\}, \end{aligned} \tag{7.10}$$

to eliminate the fractional term. If $\mathbf{x}t$ can be rewritten as $\mathbf{z} = \mathbf{x}t$, and $\mathbf{y}t$ can be linearized using the Glover's linearization scheme for each binary variable as

$$\begin{aligned} & v_i \leq u y_i, \\ & v_i \leq t, \\ & v_i \geq t - u(1 - y_i), \\ & v_i \geq 0, \end{aligned} \tag{7.11}$$

the optimization problem in (7.10) can be reformulated as

$$
\begin{aligned}
\min \quad & \mathbf{a}^\top \mathbf{z} + \mathbf{b}^\top \mathbf{v} + ct \\
\text{s.t.} \quad & \mathbf{G}\mathbf{z} + \mathbf{H}\mathbf{v} \leq \mathbf{k}t, \\
& \mathbf{G}_{\text{eq}}\mathbf{z} + \mathbf{H}_{\text{eq}}\mathbf{v} = \mathbf{k}_{\text{eq}}t, \\
& \mathbf{d}^\top \mathbf{z} + \mathbf{e}^\top \mathbf{v} + ft = 1, \\
& v_i \leq u y_i, \\
& v_i \leq t, \\
& v_i \geq t - u(1 - y_i), \\
& v_i \geq 0, \\
& \mathbf{z} \in \mathbb{R}, \\
& \mathbf{y} \in \{0, 1\}, \\
& i = \{1, 2, 3, ..., n\},
\end{aligned}
\tag{7.12}
$$

where $\mathbf{v} = [v_1, v_2, ..., v_n]^\top$, u is the upper bound of t, and n is the number of binary variables.

This MILP formulation was implemented in [133] that had the following the objective function:

$$
\mathcal{J} = \frac{\dot{W}_{\text{net}}}{\dot{m}_{\text{CW}} + \sum_{i=1}^{n} \dot{m}_{\text{WF},i}}, \tag{7.13}
$$

where $\dot{W}_{\text{net}}$ is the net output power of the plant, i.e., total turbine work minus total pump work, $\dot{m}_{\text{CW}}$ is the cooling water mass flow, and $\dot{m}_{\text{WF}}$ is the working fluid mass flow.

7.2.4 Automating the Conversion from NLP to MILP

The linearization formulations described in the previous subsections can be very tedious and time-consuming to implement on large and complex ORC systems. Thus it is more efficient to automate the procedure and build the MILP model from an existing NLP model. As a result, a function called `conv2milp` was constructed in MATLAB to automatically convert a SymBuilder NLP model into an equivalent MILP model. The `conv2milp` algorithm that was developed for this research is given in Algorithm 1.

Algorithm 1: The algorithm of the `conv2milp` function that automatically converts an NLP SymBuilder object to an equivalent MILP SymBuilder object.

```
Check that the user entered an NLP SymBuilder object.
Extract and store the required NLP information, such as constraints,
  expressions, objective function, etc.
Create a new SymBuilder object/template for the MILP model.
if user entered piecewise linear fit approximations then
    replace the equivalent nonlinear functions with the piecewise linear fit
      approximations.
for all the constraints and objective function terms do
    identify and replace bilinear terms with their equivalent linear
      approximations.
Add the old and new linear constraints, expressions, objective function
  terms, constants, and bounds to the new SymBuilder object.
Build the new SymBuilder object.
if the new SymBuilder model has a fractional objective function then
    apply the Charne-Cooper transformation and the Glover's linearization
      scheme. Rebuilt the new SymBuilder object.
Output the MILP SymBuilder object.
```

For a simple example, consider the following nonlinear program:

$$
\begin{aligned}
\min \quad & 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2 - 2x_1 - 3x_2 - x_3 \\
\text{s.t.} \quad & x_1 + e^{x_2} + x_3 + x_4 \leq 2, \\
& 3x_1 - 2x_2 - 3x_3^3 \leq 1, \\
& x_1 - 3x_2 + 2x_2 \leq 1, \\
& 0 \leq x \leq 2,
\end{aligned}
\tag{7.14}
$$

which has a quadratic objective function that is subjected to two nonlinear inequality constraints and one linear inequality constraint. The problem can be built using the SymBuilder framework as follows:

```
% New SymBuilder object
nlp = SymBuilder();

% Add objective
nlp.AddObj(‘0.5*x1^2 + 0.5*x2^2 + 0.5*x3^2 - 2*x1 - 3*x2 - x3’);

% Nonlinear expressions
nlp.AddExpression(‘fx3 = x3^3’);
nlp.AddExpression(‘fx2 = exp(x2)’);

% Add constraints
nlp.AddCon(‘x1 + fx2 + x3 + x4 <= 2’);
nlp.AddCon(‘3*x1 - 2*x2 - 3*fx3 <= 1’);
nlp.AddCon(‘x1 - 3*x2 + 2*x2 <= 1’);

% Add bounds
nlp.AddBound(‘0 <= x <= 2’);

% Build SymBuilder object
Build(nlp)
```

This resulted in the following model statistics:

```
SymBuilder Object
BUILT in 0.068s with:
- 4 variables
- 1 objective
        -  0 linear
        -  1 quadratic
        -  0 nonlinear
- 3 constraint(s)
        -  1 linear
        -  0 quadratic
        -  2 nonlinear
- 8 bound(s)
- 0 integer variable(s)
```

Solving the NLP formulation of (7.14) using SCIP, the solver converged to -2.0196 for the objective function within 0.0234s that resulted in $x_1 = 0.4079$, $x_2 = 0.4651$, $x_3 = 0$, and $x_4 = 0$. Leveraging off the SymBuilder framework, the NLP formulation can be converted into an equivalent MILP formulation as follows:

```
%% MILP model
% Approximating the nonlinear expressions using piecewise linear fit
x = linspace(0,2,20);
fx2Fit = pwfit(x,exp(x),‘poly1’,linspace(0,2,5),‘optimized’);
fx3Fit = pwfit(x,x.^3,‘poly1’,linspace(0,2,5),‘optimized’);

% Compiling the piecewise linear fit objects
pwfitObj = {‘fx2(x2)’,fx2Fit;‘fx3(x3)’,fx3Fit};

% Bilinear term approximation settings
pwOptions = pwset(‘allTerms’,... %apply to all bilinear terms
 ‘drawPlot’,‘off’,... %don’t plot regression
 ‘numBreakpoint’,5); %number of breakpoints

% Convert NLP to MILP
milp = conv2milp(nlp,‘pwfitObj’,pwfitObj,‘pwsetOpts’,pwOptions);
```

where the two nonlinear terms (i.e., x_3^3 and e^{x_2}) need to be explicitly approximated using piecewise linear fit before it can be entered into `conv2milp`. As a result, the SymBuilder object of the MILP formulation reported the following statistics:

```
SymBuilder Object
```

```
BUILT in 1.319s with:
- 50 variables
-  1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 68 constraint(s)
        - 68 linear
        -  0 quadratic
        -  0 nonlinear
- 100 bound(s)
- 20 integer variable(s)
        -  0 integer
        - 20 binary
```

Noticeably, the number of variables and constraints of the MILP formulation increased dramatically from the NLP counterpart due to the implementation of linear and integer programming techniques discussed in the previous subsections. Optimizing the MILP formulation using SCIP, the solver converged to -2.0561 for the objective function within 0.036s that resulted in $x_1 = 0.3978$, $x_2 = 0.4945$, $x_3 = 0$ and $x_4 = 0$. Evidently, the optimal result of the MILP formulation is close to the NLP result with minor differences due to the approximations made to the nonlinear and bilinear terms.

7.3 The MILP Formulation of the Magmamax Binary Power Plant

From the previous chapter, the Magamax Binary Power Plant MINLP model had the highest average solve time out of all three ORC systems. Therefore, the MBPP model was used as the case study for this chapter to see if a reformulation would help improve the optimization performance. Utilizing the SymBuilder framework structure of the NLP formulation, the MILP formulation of the MBPP was constructed as follows:

```
%% Building the MILP formulation of the Magmamax Binary Power Plant
% Compiling the piecewise linear fit objects
pwfitObj = {...
 ‘turbAIsenWork(PA,hA1)’,linTurbAIsenWorkFit;
 ‘turbBIsenWork(PB,hB1)’,linTurbBIsenWorkFit;
 ‘HmPTcA(PA)’,linHmTcAFit;
 ‘HmPTA1UbA(PA)’,linHmPTA1UbAFit;
 ‘HmPTA1LbA(PA)’,linHmPTA1LbAFit;
 ‘HsatVapB(PB)’,linHsatVapBFit;
 ‘HsatLiqB(PB)’,linHsatLiqBFit;
 ‘HmPTB1UbB(PB)’,linHmPTB1UbBFit;
 ‘HmPTB1LbB(PB)’,linHmPTB1LbBFit;
 ‘TA1(PA,hA1)’,linTA1Fit;
 ‘TA2(hA2)’,linTA2A3Fit;
 ‘TA2P(hA2P)’,linTA2A3Fit;
 ‘TA3(hA3)’,linTA2A3Fit;
 ‘TA5(PA,hA5)’,linTA5Fit;
 ‘TA7(PA,hA7)’,linTA7Fit;
 ‘TB1(PB,hB1)’,linTB1Fit;
 ‘TB2(hB2)’,linTB2Fit;
 ‘TB4(PB,hB4)’,linTB4Fit;
 ‘TB5(PB)’,linTB5Fit;
 ‘TB5P(PB)’,linTB5Fit;
 ‘CturbA(WturbA)’,CturbAFit;
 ‘CturbB(WturbB)’,CturbBFit;
 ‘CpumpA(WpumpA)’,CpumpAFit;
 ‘CpumpB(WpumpB)’,CpumpBFit};

% Bilinear term approximation settings
pwOptions = pwset(‘allTerms’,... %apply to all bilinear terms
 ‘drawPlot’,‘off’,... %don’t plot regressions
 ‘numBreakpoint’,10); %number of breakpoints

% Convert NLP to MILP
milpMBPP = conv2milp(eoMBPP,‘pwfitObj’,pwfitObj,‘pwsetOpts’,pwOptions);
```

where the objective function was the net output power. This resulted a linear optimization problem as a SymBuilder object that consisted of

```
SymBuilder Object
BUILT in 281.574s with:
- 1064 variables
-    1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 999 constraint(s)
        - 999 linear
        -  0 quadratic
        -  0 nonlinear
- 2128 bound(s)
-  498 integer variable(s)
        -  0 integer
        - 498 binary
```

Evidently, compared to the MINLP formulation, the number of binary variables had increased significantly from 21 to 498 due to the implementation of the piecewise linear approximation on both the nonlinear and bilinear terms. Consequently, this resulted in an increase in the total number of variables and constraints by 996 and 846, respectively.

The optimized net output powers of the MILP model using BARON, SCIP, CPLEX, `intlinprog`, and GUROBI are shown in Table 7.1. These solvers were selected based on their availability for academic use and their compatibility with OPTI and MATLAB. One noticeable observation is that all the solvers converged successfully to a solution for every working fluid combination. Please refer to Table 5.14 for the optimized net output powers of the NLP formulation, and Table 6.9 for the optimized net output powers of the MINLP formulation.

Table 7.1: The optimized net output powers [kW] of the Magmamax Binary Power Plant's MILP model.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	9616.3	9153.5	9270.6
	R236ea	9645.0[b]	9199.1	9312.3[a]
	R236fa	9259.9	8767.1	8893.3
	RC318	8545.4	7834.8	8027.3

[a] `intlinprog`: 9311.7 kW

[b] CPLEX and `intlinprog`: 9644.8 kW

In two scenarios, the black-box solvers, namely, `intlinprog` and CPLEX, found a local optimum as opposed to the global optimum, which is sometimes expected as they cannot guarantee the global solution.

Analysing the solve times of the solvers in Table 7.2 showed that the local solvers (namely, GUROBI and CPLEX) were on average faster than the global solvers. CPLEX had the fastest average solve time of 11.8s, followed by GUROBI, BARON, SCIP and `intlinprog`, respectively. Compared to the MINLP formulation, the average solve time of BARON and SCIP were 4.8x and 4.2x times slower, respectively, for the MILP formulation. In addition, the performance of the MILP local solvers were inferior to both the NLP and MINLP solver counterparts. It took the MILP local solvers collectively (CPLEX, GUROBI, and `intlinprog`) on average 41.74s to solve the problem, while it took the NLP local solvers (black-box) 0.032s and the MINLP local solver (BONMIN) 6.64s. This does not support the initial assumption that the MILP formulation can match or outperform the nonlinear formulations, despite the use of advanced state-of-the-art MILP solvers like CPLEX and GUROBI. This could be due to the adverse effect of the larger number of variables and constraints that increased the complexity of the MILP formulation [134] compared to the nonlinear counterparts.

Reformulating the MILP model to minimize the specific rotating machinery cost, the new model resulted in the following SymBuilder object:

```
SymBuilder Object
BUILT in 967.611s with:
- 1563 variables
-    1 objective
        -  1 linear
        -  0 quadratic
        -  0 nonlinear
- 3044 constraint(s)
        - 3044 linear
        -  0 quadratic
        -  0 nonlinear
- 3126 bound(s)
-  498 integer variable(s)
        -  0 integer
        - 498 binary
```

Due to the fractional term of the objective function, the number of variables and constraints had increased drastically from the net output power objective. While the number of binary variables remained the same at 498, the total number of variables and constraints had increased by 499 and 2045, respectively. The optimized results of the plant's specific rotating machinery cost are shown in Table 7.3. Please refer to Table 5.16 for the optimized specific rotating machinery costs of the NLP formulation, and Table 6.11 for the optimized results of the MINLP formulation.

In one of the cases, the optimized result of `intlinprog` was slightly better than the global solvers' solution. This is due to the inaccuracy of the linear approximations rendering the global optimum of the algebraic EO model different from the SM model. For instance, the optimized result of R236fa:R32 for the algebraic EO model using `intlinprog` was 251.38 k\$/MW, and the optimized result of BARON was 251.37 k\$/MW, which shows that BARON did find a better solution. However, when these optimal results were substituted into the SM model, the specific rotating machinery costs were different due to the linear approximations, hence why the optimized result of `intlinprog` is slightly better than BARON in Table 7.3. Given that the accuracy of the MILP model is compromised over the linearity of the model, unlike the MINLP model in the previous chapter (see Table 7.7), there is a higher possibility of having a different global optimum between the algebraic EO model and the SM model. It is possible to address this issue by increasing the number of breakpoints of the piecewise fit, but this can introduce more auxiliary variables and constraints that can make the optimization even harder to solve.

With the additional increase of variables and constraints, both the local and global solvers performed poorer than the former objective function. It took GUROBI, `intlinprog`, and CPLEX about 6.7x, 13.3x, and 5.5x longer, respectively, to optimize the problem and corresponds to a higher average number of nodes, as shown in Table 7.4. In regard to the global solvers, it took BARON and SCIP about 9.0x and 9.1x longer, respectively, to optimize the second objective function. Similarly, the average number of nodes for the global solvers had also increased from the previous objective function, as shown in Table 7.5.

While most of the MILP solvers were successful in finding a solution for every

Table 7.2: The solve times [s] of Table 7.1.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	28.7	23.6	16.1
		R236ea	15.8	21.7	18.9
		R236fa	34.3	27.5	36.4
		RC318	26.1	27.6	30.2
	SCIP	R600a	29.1	47.2	30.1
		R236ea	30.5	70.7	43.9
		R236fa	53.2	55.1	69.4
		RC318	51.2	46.5	48.8
	CPLEX	R600a	7.4	7.5	6.9
		R236ea	16.8	12.0	14.3
		R236fa	13.6	11.5	10.8
		RC318	8.1	17.7	14.6
	intlinprog	R600a	24.1	103.1	68.3
		R236ea	112.3	171.5	52.2
		R236fa	60.7	88.1	77.0
		RC318	167.4	163.8	114.9
	GUROBI	R600a	3.9	4.0	4.9
		R236ea	4.6	3.8	5.0
		R236fa	4.5	59.1	52.3
		RC318	3.5	7.4	4.6

Table 7.3: The optimized specific rotating machinery costs [k$/MW] of the Magmamax Binary Power Plant's MILP model.

		LP Cycle		
		R290	R32	R143a
HP Cycle	R600a	253.0	247.2	253.0
	R236ea	247.6	241.3	247.3[a]
	R236fa	257.4	251.4[b]	257.8
	RC318	277.1	273.1[c]	280.1

[a] `intlinprog`: Maximum Time Reached

[b] `intlinprog`: 251.3 k$/MW

[c] `intlinprog`: 273.2 k$/MW

working fluid pairs, the performance of the solvers was still inferior to the MINLP formulation, as shown by the solve times in Table 7.6. On average, the solve times of BARON and SCIP were 38.7x and 62.0x slower, respectively, when optimizing the MILP formulation. This is also evident in Table 7.5 where the number of nodes needed to solve the MILP formulation using the global solvers is on average higher than the MINLP formulation. However, the BARON was able to converge to a solution for all the working fluid pairs, which is significantly better than the NLP counterpart.

Furthermore, the local MILP solvers again did not perform better than the nonlinear counterparts. On average, it took the local MILP solvers (CPLEX, `intlinprog`, and GUROBI) 488.4s to solve the problem, while it took the NLP solvers (black-box solvers) 0.16s and the MINLP solver (BONMIN) 4.0s. As previously discussed in Section 6.4.2.1, the average solve time comparison between the local solvers would be more appropriate and reliable if more solvers were used, especially in regards to the MINLP solvers, given that local solvers have different underlying algorithms that can also result in different computational performance. Therefore, the comparison between the local solvers presented in this chapter is a preliminary assessment of the solvers' performance.

The average relative errors between the algebraic EO model and the SM model were larger with the MILP formulation than the MINLP counterpart, as shown in Table 7.7. This highlights one of the disadvantages of the linearization procedure. With the attempt to achieve an MILP model, the accuracy of the model was compromised due to the linear approximations of the nonlinear and bilinear terms. Inevitably, given the nonlinearity of the thermodynamic properties of the working fluids, it is difficult to accurately model the ORC system using linear functions.

Generally, there is a possibility that MILP solvers can run into a time limitation with a large number of binary variables, especially if there is a large difference between the feasible solution and the optimal relaxed LP solution [64]. In addition, given that MILP problems are NP-hard, there is no known algorithm that can solve the problem in polynomial time. This means that it is much harder to solve large MILP problems as the size of the problem increases. While the number of binary variables can be reduced by decreasing the number of breakpoints of the piecewise

Table 7.4: The average number of nodes needed for the local solvers to optimize the Magmamax Binary Power Plant's MILP formulation.

	CPLEX	`intlinprog`	**GUROBI**
Net Output Power	365204	112716	1689
Specific Rotating Machinery Cost	1392052	584738	32605

Table 7.5: The average number of nodes needed for the global solvers to optimize the Magmamax Binary Power Plant's MINLP and MILP formulations.

	BARON		**SCIP**	
	MINLP	**MILP**	**MINLP**	**MILP**
Net Output Power	22	107	3286	3199
Specific Rotating Machinery Cost	165	315	2121	35276

Table 7.6: The solve times [s] of Table 7.3. MTR: Maximum Time Reached.

			LP Cycle		
			R290	R32	R143a
HP Cycle	BARON	R600a	168.1	90.9	274.2
		R236ea	271.7	202.5	259.8
		R236fa	238.1	156.3	321.8
		RC318	200.3	186.5	397.7
	SCIP	R600a	154.4	152.1	187.2
		R236ea	356.0	109.1	444.0
		R236fa	423.5	739.8	1043.4
		RC318	514.5	306.8	792.4
	CPLEX	R600a	79.1	32.4	26.1
		R236ea	117.7	39.1	50.7
		R236fa	76.9	69.8	85.8
		RC318	39.5	38.5	121.7
	intlinprog	R600a	1735.7	2968.6	730.9
		R236ea	2633.6	1019.0	MTR
		R236fa	958.7	712.0	3431.8
		RC318	449.5	489.8	136.2
	GUROBI	R600a	47.6	22.8	88.8
		R236ea	116.3	67.0	155.6
		R236fa	85.6	89.7	132.9
		RC318	68.1	86.9	89.1

approximation, it might not be sensible as it could render the MILP model infeasible or give an inaccurate representation of the ORC system. Furthermore, unlike NLP solvers, supplying a good initial guess to MILP solvers does not always improve the performance because all the MILP solvers in this research can automatically handle this internally [135, 100]. Consequently, having a large number of variables would require more effort from the solver to finding a feasible starting point and likely leads to a longer CPU time. For the NLP and MINLP solvers, they benefited more from the algebraic model structure; therefore, their short computational time can be attributed to the availability of accurate derivatives and bypassing the use of the finite difference method.

Table 7.7: The average relative errors between the optimized algebraic EO model and the SM model of the Magmamax Binary Power Plant using SCIP.

	MILP	**MINLP**
Net Output Power	0.15%	0.04%
Specific Rotating Machinery Cost	0.10%	0.01%

Evidently, the performance of the MILP formulation cannot justify the effort to linearize the nonlinear ORC formulation. However, the framework was developed with flexibility in mind and, thus, can be implemented to other NLP problems outside this research that can benefit from the MILP formulation. In addition, given that NLP and MINLP problems generally require the user to enter a good initial point, it might be beneficial to supply the initial point obtained from the MILP model to the NLP model where the initial guess is hard to obtain. This might potentially improve the performance and optimality of the solution.

7.4 Summary

This chapter discussed the MILP formulation of an ORC system, specifically the Magmamax Binary Power Plant, using piecewise linear approximations and linear/integer programming techniques. Leveraging off the algebraic structure of NLP model in Chapter 5, the equivalent MILP model was constructed using an automated approach through a MATLAB function `conv2milp` that was developed for this research. This open up the ORC optimization problem to a whole set of MILP solvers that can be utilized. In addition, the MILP formulation still retains the algebraic structure as the NLP formulation and, thus, it is compatible with global solvers that can deterministically guarantee the global optimal solution.

Due to the implementation of the piecewise linear approximations and the linearization programming techniques, the number of variables and constraints increased significantly compared to the nonlinear counterparts. Consequently, this had an adverse effect on the solvers' performance despite the decrease in the nonlinearity of the optimization problem. While the results of the optimized MILP formulation showed that most of the solvers managed to converge to a solution for all the different working fluid pairs, the performance of the local solvers was inferior to the NLP and MINLP counterparts. Similarly, the global solvers were also able to find a solution for all the working fluid pairs using the MILP formulation, which is significantly better than the NLP formulation, but their performance was inferior to the MINLP counterpart. Therefore, based on this study, it is not advisable to reformulation the nonlinear ORC model into an MILP model if the MINLP model is obtainable.

Chapter 8

Conclusions and Future Work

This chapter will conclude and review the work in this thesis. A summary of key contributions and a critique of the research questions will be presented. This is then followed by a discussion on the recommendations for future work.

8.1 Conclusion

This thesis has described different optimization formulations of ORC systems that aim to provide an efficient and robust optimization approach to address some of the common issues associated with sequential-modular optimization. Three different real-world ORC systems that operated as industrial binary cycle power plants were modelled and optimized in order to highlight the flexibility of the modelling framework. These plants represent a number of advanced binary cycles that are aimed to reduce thermodynamic losses, thus they provide a range of innovations in the topology and complexity that can exist in an ORC design. Each plant was subjected to a number of different working fluids that were compatible with the design specifications of the plant. Two objective functions were investigated, namely, the net output power and the specific rotating machinery cost, in order to illustrate a linear and nonlinear objective function in this research.

The proposed optimization model is an algebraic equation-oriented model that was derived using thermodynamic and regression analysis. The optimization results showed that the algebraic EO model performed significantly better than the common SM approach, which supports the existing literature. Furthermore, leveraging off the algebraic structure of the EO model, a piecewise fit algorithm was developed to reduce the nonlinearity of the model but further improve the accuracy of the model and the optimization performance. Subsequently, this led to the final investigation of this research where the nonlinearity of the model was reduced down to only linear terms, namely, in the form of a mixed-integer linear programming model.

8.2 Key Contributions

The following subsections highlight the key contributions of this thesis. The first two contributions address the issues associated with the conventional optimization for ORC systems, which is mainly due to the use of the SM approach. This is achieved by formulating an algebraic EO model and attempt to strategically make careful approximations to better suit the optimizers. The third contribution reviews the implementation of the `pwfit` algorithm to improve the regressions' accuracy and reduce the nonlinearity of the model, which consequently enhance the overall optimization performance. Finally, the last contribution reviews the equivalent MILP formulation of one of the ORC models that was constructed using an automatic conversion algorithm, `conv2milp`, and highlights the advantages and disadvantages compared to its nonlinear counterparts.

8.2.1 Algebraic Equation-Oriented Models

Most optimization problems in the literature focus on the improvements of the optimized results, and not on the efficiency or optimality of the optimization problem. Traditionally, process systems are modelled and optimized using the SM method due to its intuitive approach to modelling. This is evident in the literature, where most of the ORC systems are modelled and optimized using SM models. This is widely acknowledged that this can be inefficient and lead to various optimization issues, such as convergence issues, long execution time, and poor performance, compared to the EO models. This thesis proposed an algebraic EO modelling approach for ORC systems that is tailored for efficient and robust optimization, which have not been extensively investigated in the literature, especially for large industrial ORC systems. This involves deriving a set of equations that describe the process of the system using thermodynamic and regression analysis.

From the optimization studies carried out in this research, the algebraic EO approach significantly outperformed the SM approach as expected. Utilizing the derivative-based black-box solvers, the algebraic EO models of the ORC systems can be optimized within a few seconds. This was carried out on three real-world ORC systems that cover a wide range of different configurations and operating conditions. The structure of the algebraic model allows for analytical derivative information to be obtained and supplied to the solver, which resulted in a more robust optimization and faster computational time. In addition, there is no need for a nonlinear equation solver to converge the system cycle(s) as the EO model structure allows the cycle(s) to be treated as any other equation.

In order to illustrate the inefficiency of the SM model for optimization, one of the ORC plants was optimized using the SM approach. Consequently, even with a smaller number of variables and constraints, and reduced optimality tolerances, the derivative-based solvers still took significantly longer to converge to a solution and, in some cases, were unsuccessful to find an optimal solution. This is due to the use of the finite difference method that can only give an approximation of the derivative information, which can lead to an infeasible point that causes the solver to fail or terminate prematurely.

8.2.2 Global Optimality

In addition to achieving efficient and robust optimization, the algebraic EO model is compatible with certain white-box solvers that can deterministically guarantee the global optimum of the optimization problem. This is a major advantage in optimization as it can translate to a large difference in the overall financial cost and performance of the plant, which can be the deciding factor between a viable system and one that is not. In addition, this study also helped to reinforce the fundamental interest in the literature surrounding global optimality and the development of global solvers. There is a need from practitioners for these solvers to exist, and hence it is important to highlight the different applications where they can be implemented and/or improved.

While the differences between the local and global results presented in this research are small, this does not hinder the significant importance of achieving global optimality for an ORC system. Given that ORC systems have low thermal efficiencies and a long operation life span of 25+ years, even a small difference can have a significant competitive advantage. In addition, this research only considered two objective functions, however, there are many other aspects of the ORC system that can be optimized and analysed that can lead to a larger difference between the local and global optimality. Therefore, providing a modelling approach that this tailored for global optimization is a very advantageous contribution.

8.2.3 Regression Accuracy and Optimization Performance Improvement

In order for the ORC model to be explicitly algebraic and analytically differentiable, the thermodynamic routines and unit operation processes need to be approximated using regression analysis. In some cases, the nonlinearity of the thermodynamics or nonlinear functions cannot be accurately approximated using a single curve/surface fit, which can lead to an inaccurate model. While it might be possible to increase the order of the polynomial fit to obtain better accuracy, this can increase the nonlinearity of the model and might have an adverse effect on the performance of the optimization solver. In order to address this issue, a piecewise fit algorithm, called `pwfit`, was developed in MATLAB to approximate nonlinear univariate and bivariate functions. Unlike other piecewise fit approximation proposed in the literature, `pwfit` is not limited to linear approximations and allows higher order polynomial functions to be used. This inherently offers more level of flexibility and accuracy for the overall regression. As shown in Chapter 6, the accuracy of the regression fit can often be improved by using the proposed piecewise fit approximation instead of the single fit approximation. Furthermore, the location of the breakpoints can also be allocated manually by the user or optimized automatically in order to further improve the accuracy of the regression.

In addition to improving the accuracy of the model, the performance of the solver can also be improved by reducing the nonlinearity of the optimization problem. Leveraging off the piecewise fit algorithm, the order of the polynomial fits can be decreased in some cases without compromising on the accuracy of the model. As shown in Chapter 6, the performance of the white-box solvers improved significantly from decreasing all the constraints to only quadratic and linear functions. This was clearly prominent with the Magmamax Binary Power Plant model, whereby the BARON was able to find a solution for all the working fluids pairs as opposed to the NLP model counterpart.

8.2.4 Linearization of the Nonlinear Model

A nonlinear ORC model was successfully converted into an equivalent MILP problem using an algorithm developed for this research, which involved implementing a combination of piecewise linear approximations and linear/integer programming techniques, such as the Glover's linearization scheme and the Charnes-Cooper transformation. While the linear structure was perceived as the ideal formulation for optimization, the adverse effect of the substantial increase in the number of auxiliary variables and constraints due to the piecewise linear approximations and linearization techniques resulted in an inferior computational performance compared to the nonlinear counterparts.

The implication of the MILP optimized results and performance highlighted that it is not necessarily advantageous to linearize the nonlinear ORC model. In some cases, it is enough to linearize and/or reduce the order of some nonlinear terms to achieve a more favourable optimization performance. While the thesis mainly focused on modelling and optimizing ORC systems, the framework was developed with flexibility in mind and can be applied to other nonlinear problems outside this research. Therefore, the proposed MILP formulation can be used as an alternative approach for when the nonlinear optimization fails. In addition, the MILP model can exploit the available MILP solvers that could potentially help improve the performance of some nonlinear solvers by generating the initial points that are hard to obtain.

8.3 An Assessment of the Proposed Research Questions

To conclude the contributions of this research, the research questions in Section 1.5 will be assessed below:

1. *Can we formulate equation-orientated models of large and complex ORC systems that are algebraic in structure and tailored for efficient and robust optimization?*

   Three large and complex ORC systems were successfully modelled using an equation-oriented approach. By approximating the thermodynamic routines and the unit operation process using polynomial functions, the model can be expressed algebraically that can be exploited by the optimization solvers. The algebraic structure allows the solvers to obtain accurate derivatives and matrix sparsity information that lead to efficient and robust optimization. In addition, the algebraic structure consists of functions that are compatible with certain white-box solvers, thus the global optimum of the optimization problem can be found using solvers such as SCIP and BARON.

2. *Can the performance of some optimization solvers be improved by reducing the deleterious effect of the nonlinearity of the optimization ORC model without compromising, if not improving, on the accuracy of the approximated model?*

The number of nonlinear constraints of the algebraic EO model can be reduced by using piecewise regressions to approximate the nonlinear model. Consequently, the optimization performance of the white-box solvers was improved by decreasing the nonlinearity of the model, despite having more variables and constraints from implementing the piecewise fit approximations. In addition, the accuracy of the ORC models was not compromised but was improved by using quadratic polynomial piecewise fits, instead of the common piecewise linear fit approximation methods proposed in the literature, and optimizing the allocation of the breakpoints. These piecewise fit features were incorporated into the `pwfit` algorithm that was developed specifically for this research.

3. *Is it possible to reformulate the nonlinear algebraic EO model of an ORC system into an equivalent mixed-integer linear programming model using linear/integer programming techniques? In addition, given the improvement in nonlinear solver algorithms and the advancement in computer hardware, can the mixed-integer linear model improve or match the performance of the nonlinear counterpart as traditionally concluded in the literature?*

   Leveraging off the algebraic EO model structure of the ORC model and using the linear/integer programming techniques and piecewise linear approximations, the nonlinear model was successfully converted into an equivalent mixed-integer linear programming model. This linearization process was automated using a MATLAB function called `conv2milp` that was developed specifically for this research. Given the significant amount of auxiliary variables and constraints that were introduced to the optimization problem due to the linearization methods, the performance of the MILP model was inferior to the nonlinear model counterparts that were presented in thesis research.

8.4 Recommendation for Future Work

There are several areas that can be further developed and investigated as a natural extension to this research or as a possible application for the proposed modelling framework. However, given the time and financial constraints, it was not possible to include them in this research. However, they are highlighted below for future work and for researchers that are interested in further work in this area.

8.4.1 Binary Cycle Power Plant Optimization

The plant models presented in this thesis were assumed to operate in ideal conditions, e.g., there are no pressure drops or heat loss across the exchangers, which is common in the literature. While this might be suitable for most preliminary design and optimization studies, it would be more practical to investigate plant models that are not ideal. This might involve obtaining real operating data from existing plants, power companies, or plant suppliers over a significant period of time. However, this can be difficult given that usually this information is confidential and cannot be published.

A simpler alternative is to model the plant using more realistic unit operation modules/models from existing chemical modelling platforms, such as Aspen Plus and VMGSim, that were developed from empirical and test data. This will also include using the real thermodynamic properties of the brine and its compositions, instead of pure water. These unit operation models and thermodynamic properties will then be approximated using the same procedures discussed in this thesis to obtain the algebraic EO model and then optimized. It would be interesting to compare the optimal results between an ideal and a realistic model to see if there is a significant difference to raise a concern surrounding the use of ideal models.

In addition, it would be worthwhile to investigate other objective functions, such as a multi-objective function that is associated with the system's cost, yield and profit combined. This will help to further test the limitation and flexibility of the proposed framework and indicate which areas need to be improved or if they are not applicable to this work. Furthermore, the plant models should be optimized with several numbers of solvers that vary in different algorithms for each of the optimization problems to provide a more reliable performance comparison across the different formulations.

8.4.2 Piecewise Approximation of Univariate and Bivariate Functions

The current `pwfit` algorithm has some limitations that can be addressed to improve the accuracy of the regression fit. Given that the proposed modelling framework in thesis relies heavily on an accurate and flexible regression toolbox, it would be significantly beneficial for the overall performance and accuracy of the ORC model if the `pwfit` algorithm can be further improved.

First, the current algorithm only allows the breakpoints to be allocated along the x-axis for bivariate functions, which can limit the use of some nonlinear functions and result in poor regression accuracy. Therefore, to accommodate a wider range of other nonlinear functions, the algorithm should also allow the user to specify and optimize the allocations of the breakpoints along the y-axis.

Second, the current algorithm does not allow each subfunction to have a different polynomial function from one another. This means that some parts of the regression can be overcompensated by the high order nonlinear fit model, which can add unnecessary complexity to the optimization problem. It would be more sensible to allow the user to select the fit model for each subfunction or automate the selection process to get the best fit possible with the lowest order and number of polynomial fits.

Third, the `pwfit` algorithm should incorporate other nonlinear functions, such as the exponential functions, logarithmic functions, trigonometric functions, etc., that are currently not available and also allow the users to enter their own fit model. This might be difficult to implement, given that the algorithm requires the Vandermonde matrix to be derived, which is not possible with non-polynomial functions that are not linear in the parameters. Therefore, another alternative approach will be required to incorporate the non-polynomial functions with the current algorithm.

8.4.3 Optimization of the MILP Formulation

The poor performance of the mixed-integer linear programming formulation was due to the substantial increase in the number of binary variables, which subsequently resulted in more auxiliary decision variables and constraints. Therefore, investigating ways to reduce the number of binary variables in the MILP model (and also in the MINLP model) will improve the optimization performance and lead to a shorter solve time. It might also be possible to utilize the built-in special order set type feature in some of the optimization solvers to reduce the number of binary variables and constraints.

While the optimization performance of the MILP model presented in this research did not perform better than their nonlinear counterparts. It would be worthwhile to implement the functionality of the `conv2milp` function to other optimization problems outside this research. Given that most of the process engineering problems (see Table 2.2) are NLP and MINLP problems, there might be cases where the MILP formulation is more beneficial.

Bibliography

[1] A. Basaran and L. Ozgener, “Investigation of the effect of different refrigerants on performances of binary geothermal power plants,” *Energy Conversion and Management*, vol. 76, pp. 483–498, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.enconman.2013.07.058

[2] Inverse Problem Ltd, “JSteam Excel Add-In,” 2017. [Online]. Available: https://www.inverseproblem.co.nz/JSteam.html [Accessed: 2017-10-11]

[3] R. DiPippo, *Geothermal Power Plants Principles, Applications, Case Studies and Environmental Impact*, 4th ed. Waltham, MA: Joe Hayton: Elsevier, 2016.

[4] L. T. Biegler, “New nonlinear programming paradigms for the future of process optimization,” *AIChE Journal*, vol. 63, no. 4, pp. 1178–1193, 2017. [Online]. Available: http://dx.doi.org/10.1002/aic.15674

[5] M. M. Martín, *Introduction to Software for Chemical Engineers.* Baton Rouge, United States: Chapman and Hall/CRC, 2014.

[6] Y. K. Yeo, *Chemical Engineering Computation with MATLAB®.* Milton, United Kingdom: CRC Press, 2017.

[7] R. DiPippo, “Geothermal Power Plants Around the World,” in *Sourcebook on the production of electricity from geothermal energy*, J. Kestin, Ed. Washington, D.C.: U.S. Dept. of Energy, Assistant Secretary for Resource Applications, Division of Geothermal Energy., 1980, ch. 10, pp. 870–972.

[8] World Energy Council, “World Energy Resources: Geothermal 2016,” Tech. Rep., 2016. [Online]. Available: https://www.worldenergy.org/wp-content/uploads/2017/03/WEResources_Geothermal_2016.pdf

[9] New Zealand Geothermal Association, “Electricity Generation,” 2018. [Online]. Available: https://nzgeothermal.org.nz/geothermal-energy/electricity-generation/ [Accessed: 2018-11-24]

[10] New Zealand Geothermal Association, “NZ Geothermal Fields,” 2018. [Online]. Available: https://nzgeothermal.org.nz/nz_geo_fields/ [Accessed: 2018-11-24]

[11] S. Lecompte, H. Huisseune, M. Van Den Broek, B. Vanslambrouck, and M. De Paepe, “Review of organic Rankine cycle (ORC) architectures for waste heat recovery,” *Renewable and Sustainable Energy Reviews*, vol. 47, pp. 448–461, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2015.03.089

[12] R. Bertani, “Geothermal power generation in the world 2010-2014 update report,” *Geothermics*, vol. 60, pp. 31 – 43, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0375650515001558

[13] AspenTech, “Aspen Plus,” 2015. [Online]. Available: http://www.aspentech.com/products/engineering/aspen-plus/ [Accessed: 2015-08-24]

[14] Electric General, “GateCycle,” 2014. [Online]. Available: https://getotalplant.com/GateCycle/docs/GateCycle/index.html [Accessed: 2016-08-10]

[15] H. Ghasemi, M. Paci, A. Tizzanini, and A. Mitsos, “Modeling and optimization of a binary geothermal power plant,” *Energy*, vol. 50, no. 1, pp. 412–428, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2012.10.039

[16] K. Satanphol, W. Pridasawas, and B. Suphanit, “A study on optimal composition of zeotropic working fluid in an Organic Rankine Cycle (ORC) for low grade heat recovery,” *Energy*, vol. 123, pp. 326–339, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2017.02.024

[17] MathWorks, “fmincon,” 2017. [Online]. Available: http://au.mathworks.com/help/optim/ug/fmincon.html?searchHighlight=fmincon&s_tid=doc_srchtitle [Accessed: 2017-12-02]

[18] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[19] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.* Society for Industrial and Applied Mathematics, 2010. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898719383

[20] R. Smith, *Chemical Process Design and Integration*, 2nd ed. West Sussex, United Kingdom: John Wiley & Sons Inc, 2016.

[21] T. Achterberg, “SCIP: Solving constraint integer programs,” *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.

[22] M. Tawarmalani and N. Sahinidis, “A polyhedral branch-and-cut approach to global optimization,” *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, 2005.

[23] W. R. Huster, D. Bongartz, and A. Mitsos, “Deterministic Global Optimization of the Design of a Geothermal Organic Rankine Cycle,” *Energy Procedia*, vol. 129, pp. 50–57, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.egypro.2017.09.181

[24] F. Glover, “Improved Linear Integer Programming Formulations of Nonlinear Integer Problems,” *Management Science*, vol. 22, no. 4, pp. 455–460, 1975.

[25] A. Charnes and W. W. Cooper, "Programming with linear fractional functionals," *Naval Research Logistics Quarterly*, vol. 9, no. 3-4, pp. 181–186, 1962. [Online]. Available: http://dx.doi.org/10.1002/nav.3800090303

[26] National Geographic Society, "Non-renewable Energy." [Online]. Available: https://www.nationalgeographic.org/encyclopedia/non-renewable-energy/ [Accessed: 2018-05-30]

[27] United Nation Climate Change, "The Paris Agreement," 2018. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement [Accessed: 2018-11-06]

[28] A. Kecebas and H. Gokgedik, "Thermodynamic evaluation of a geothermal power plant for advanced exergy analysis," *Energy*, vol. 88, pp. 746–755, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2015.05.094

[29] D. Tempesti, G. Manfrida, and D. Fiaschi, "Thermodynamic analysis of two micro CHP systems operating with geothermal and solar energy," *Applied Energy*, vol. 97, pp. 609–617, sep 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261912001067

[30] M. Astolfi, M. C. Romano, P. Bombarda, and E. Macchi, "Binary ORC (Organic Rankine Cycles) power plants for the exploitation of medium-low temperature geothermal sources - Part A: Thermodynamic optimization," *Energy*, vol. 66, pp. 423–434, 2014.

[31] M. Astolfi, M. C. Romano, P. Bombarda, and E. Macchi, "Binary ORC (Organic Rankine Cycles) power plants for the exploitation of medium-low temperature geothermal sources - Part B: Techno-economic optimization," *Energy*, vol. 66, pp. 435–446, 2014.

[32] J. Clarke and J. T. McLeskey, "Multi-objective particle swarm optimization of binary geothermal power plants," *Applied Energy*, vol. 138, pp. 302–314, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0306261914011222

[33] H. D. Madhawa Hettiarachchi, M. Golubovic, W. M. Worek, and Y. Ikegami, "Optimum design criteria for an Organic Rankine cycle using low-temperature geothermal heat sources," *Energy*, vol. 32, no. 9, pp. 1698–1706, 2007.

[34] D. Maraver, J. Royo, V. Lemort, and S. Quoilin, "Systematic optimization of subcritical and transcritical organic Rankine cycles (ORCs) constrained by technical parameters in multiple applications," *Applied Energy*, vol. 117, pp. 11–29, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2013.11.076

[35] E. W. Lemmon, M. L. Huber, and M. O. McLinden, "NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 9.1, National Institute of Standards and Technology," 2013. [Online]. Available: https://www.nist.gov/srd/refprop

[36] Virtual Materials Group Inc., "VMGSim." [Online]. Available: http://www.virtualmaterials.com/ [Accessed: 2015-11-17]

[37] D. Fiaschi, A. Lifshitz, G. Manfrida, and D. Tempesti, "An innovative ORC power plant layout for heat and power generation from medium- to low-temperature geothermal resources," *Energy Conversion and Management*, vol. 88, pp. 883–893, 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0196890414007821

[38] D. Budisulistyo and S. Krumdieck, "Thermodynamic and economic analysis for the pre-feasibility study of a binary geothermal power plant," *Energy Conversion and Management*, vol. 103, pp. 639–649, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0196890415006172

[39] A. Coskun, A. Bolatturk, and M. Kanoglu, "Thermodynamic and economic analysis and optimization of power cycles for a medium temperature geothermal resource," *Energy Conversion and Management*, vol. 78, pp. 39–49, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.enconman.2013.10.045

[40] N. Shokati, F. Ranjbar, and M. Yari, "Exergoeconomic analysis and optimization of basic, dual-pressure and dual-fluid ORCs and Kalina geothermal power plants: A comparative study," *Renewable Energy*, vol. 83, pp. 527–542, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.renene.2015.04.069

[41] M. Kanogu and Y. A. Çengel, "Retrofitting a Geothermal Power Plant to Optimize Performance: A Case Study," *Journal of Energy Resources Technology*, vol. 121, no. 4, p. 295, 1999. [Online]. Available: http://energyresources.asmedigitalcollection.asme.org/article.aspx?articleid=1413872

[42] S. Ogriseck, "Integration of Kalina cycle in a combined heat and power plant, a case study," *Applied Thermal Engineering*, vol. 29, no. 14-15, pp. 2843–2848, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1359431109000519

[43] R. Dipippo and P. Moya, "Geothermics Las Pailas geothermal binary power plant , Rincón de la Vieja , Costa Rica : Performance assessment of plant and alternatives," *Geothermics*, vol. 48, pp. 1–15, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.geothermics.2013.03.006

[44] A. Nemati, H. Nami, F. Ranjbar, and M. Yari, "A comparative thermodynamic analysis of ORC and Kalina cycles for waste heat recovery: A case study for CGAM cogeneration system," *Case Studies in Thermal Engineering*, vol. 9, no. 1, pp. 1–13, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.csite.2016.11.003

[45] Z. Shengjun, W. Huaixin, and G. Tao, "Performance comparison and parametric optimization of subcritical Organic Rankine Cycle (ORC) and transcritical power cycle system for low-temperature geothermal power generation," *Applied Energy*, vol. 88, no. 8, pp. 2740–2754, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2011.02.034

[46] Z. Q. Wang, N. J. Zhou, J. Guo, and X. Y. Wang, "Fluid selection and parametric optimization of organic Rankine cycle using low temperature waste heat," *Energy*, vol. 40, no. 1, pp. 107–115, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2012.02.022

[47] S. Quoilin, S. Declaye, B. F. Tchanche, and V. Lemort, “Thermo-economic optimization of waste heat recovery Organic Rankine Cycles,” *Applied Thermal Engineering*, vol. 31, no. 14, pp. 2885–2893, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.applthermaleng.2011.05.014

[48] Aspen Technology, “Aspen Plus Dynamics.” [Online]. Available: https://www.aspentech.com/en/products/pages/aspen-plus-dynamics [Accessed: 2018-07-05]

[49] Modelica, “Modelica and the Modelica Association.” [Online]. Available: https://www.modelica.org/ [Accessed: 2018-07-05]

[50] M. J. Proctor, W. Yu, R. D. Kirkpatrick, and B. R. Young, “Dynamic modelling and validation of a commercial scale geothermal organic rankine cycle power plant,” *Geothermics*, vol. 61, pp. 63–74, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.geothermics.2016.01.007

[51] F. Felgner, L. Exel, and G. Frey, “Component-oriented ORC plant modeling for efficient system design and profitability prediction,” *3rd International Conference on Clean Electrical Power: Renewable Energy Resources Impact, ICCEP 2011*, pp. 196–203, 2011.

[52] S. Quoilin, R. Aumann, A. Grill, A. Schuster, V. Lemort, and H. Spliethoff, “Dynamic modeling and optimal control strategy of waste heat recovery Organic Rankine Cycles,” *Applied Energy*, vol. 88, no. 6, pp. 2183–2190, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2011.01.015

[53] H. Chen, D. Y. Goswami, and E. K. Stefanakos, “A review of thermodynamic cycles and working fluids for the conversion of low-grade heat,” *Renewable and Sustainable Energy Reviews*, vol. 14, no. 9, pp. 3059–3067, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2010.07.006

[54] R. Rayegan and Y. X. Tao, “A procedure to select working fluids for Solar Organic Rankine Cycles (ORCs),” *Renewable Energy*, vol. 36, no. 2, pp. 659–670, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.renene.2010.07.010

[55] B. F. Tchanche, G. Papadakis, G. Lambrinos, and A. Frangoudakis, “Fluid selection for a low-temperature solar organic Rankine cycle,” *Applied Thermal Engineering*, vol. 29, no. 11, pp. 2468–2476, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359431108004900

[56] R. Scaccabarozzi, M. Tavano, C. M. Invernizzi, and E. Martelli, “Comparison of working fluids and cycle optimization for heat recovery ORCs from large internal combustion engines,” *Energy*, vol. 158, pp. 396–416, 2018. [Online]. Available: https://doi.org/10.1016/j.energy.2018.06.017

[57] Y. Lu, A. P. Roskilly, L. Jiang, and X. Yu, “Working fluid selection for a small-scale organic Rankine cycle recovering engine waste heat,” *Energy Procedia*, vol. 123, pp. 346–352, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.egypro.2017.07.266

[58] H. Yu, X. Feng, and Y. Wang, “Working Fluid Selection for Organic Rankine Cycle (ORC) Considering the Characteristics of Waste Heat Sources,” *Industrial and Engineering Chemistry Research*, vol. 55, no. 5, pp. 1309–1321, 2016.

[59] B. T. Liu, K. H. Chien, and C. C. Wang, “Effect of working fluids on organic Rankine cycle for waste heat recovery,” *Energy*, vol. 29, no. 8, pp. 1207–1217, 2004.

[60] K. Yang, H. Zhang, Z. Wang, J. Zhang, F. Yang, E. Wang, and B. Yao, “Study of zeotropic mixtures of ORC (organic Rankine cycle) under engine various operating conditions,” *Energy*, vol. 58, pp. 494–510, 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0360544213004477

[61] F. Heberle, M. Preißinger, and D. Brüggemann, “Zeotropic mixtures as working fluids in Organic Rankine Cycles for low-enthalpy geothermal resources,” *Renewable Energy*, vol. 37, no. 1, pp. 364–370, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.renene.2011.06.044

[62] S. Milora and J. Tester, *Geothermal energy as a source of electric power: thermodynamic and economic criteria.* Cambridge, MA: MIT Press, 1976.

[63] L. T. Biegler, “Recent Advances in Chemical Process Optimization,” *Chemie Ingenieur Technik*, vol. 86, no. 7, pp. 943–952, 2014. [Online]. Available: http://doi.wiley.com/10.1002/cite.201400033

[64] L. T. Biegler and I. E. Grossmann, “Retrospective on optimization,” *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1169–1192, 2004.

[65] R. Hooke and T. A. Jeeves, “‘ Direct Search’ Solution of Numerical and Statistical Problems,” *J. ACM*, vol. 8, no. 2, pp. 212–229, apr 1961. [Online]. Available: http://doi.acm.org/10.1145/321062.321069

[66] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, 1975. [Online]. Available: https://books.google.co.nz/books?id=JE5RAAAAMAAJ

[67] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: http://science.sciencemag.org/content/220/4598/671

[68] I. E. Grossmann and L. T. Biegler, “Part II. Future perspective on optimization,” *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1193–1218, 2004.

[69] W. Andreas and L. T. Biegler, “Global and Local Convergence of Line Search Filter Methods for Nonlinear Programming,” *CAPD Technical Report B-01-09*, pp. 1–37, August 2001, revised May 2002.

[70] E. Balas and M. Fischetti, “Chapter 5: Integer (Linear) Optimization,” in *Advances and Trends in Optimization with Engineering Applications*, 2017, pp. 49–63. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611974683.ch5

[71] Gurobi Optimization, “Gurobi 8 Performance Benchmarks,” 2019. [Online]. Available: http://www.gurobi.com/pdfs/benchmarks.pdf [Accessed: 13/02/2019]

[72] M. R. Bussieck and S. Vigerske, “MINLP Solver Software,” *Wiley Encyclopedia of Operations Research and Management Science*, pp. 1–17, 2011. [Online]. Available: http://doi.wiley.com/10.1002/9780470400531.eorms0527

[73] M. A. Duran and I. E. Grossmann, “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, Oct 1986. [Online]. Available: https://doi.org/10.1007/BF02592064

[74] R. Fletcher and S. Leyffer, “Solving mixed integer nonlinear programs by outer approximation,” *Mathematical Programming*, vol. 66, no. 1, pp. 327–349, Aug 1994. [Online]. Available: https://doi.org/10.1007/BF01581153

[75] I. E. Grossmann and Z. Kravanja, “Mixed-integer nonlinear programming techniques for process systems engineering,” *Computers and Chemical Engineering*, vol. 19, no. SUPPL. 1, pp. 189–204, 1995.

[76] A. Bonomi, O. Cavalett, M. Pereira da Cunha, M. A. P. Lima, and M. P. da Cunha, *Virtual Biorefinery : An Optimization Strategy for Renewable Carbon Valorization.* Cham, Switzerland: Springer, 2015.

[77] A. W. Westerberg, “A retrospective on design and process synthesis,” *Computers and Chemical Engineering*, vol. 28, no. 4, pp. 447–458, 2004.

[78] J. C. Kantor and T. F. Edgar, “Computing Skills in the Chemical Engineering Curriculum,” in *Computers in Chemical Engineering Education*, B. Carnahan, Ed. Austin, Texas: CACHE Corp., 1996, ch. 2, pp. 9–56.

[79] P. van Cotten, J. W. M. Steeman, and J. A. de Leeuw den Bouter, “Computer simulation of coal gasification and combined-cycle electric power generation,” *Computers & Chemical Engineering*, vol. 3, no. 1, pp. 405–411, 1979. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0098135479800654

[80] P. D. Babcock and L. F. Stutzman, “New Developments in Chemical Process Simulation - FLOWSIM,” *Mathematical Modelling*, vol. 3, pp. 559–572, 1982.

[81] Chemical Online, “gPROMS.” [Online]. Available: https://www.chemicalonline.com/doc/gproms-0001 [Accessed: 2018-06-21]

[82] A. C. Dimian, “Introduction in Process Simulation,” in *Integrated Design and Simulation of Chemical Processes*, ser. Computer Aided Chemical Engineering. Amsterdam, The Netherlands: Elsevier, 2003, vol. 13, pp. 33–58. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570794603800263

[83] E. M. ROSEN, “Steady State Chemical Process Simulation: A State-of-the-Art Review,” in *Computer Applications to Chemical Engineering*, ch. 2, pp. 3–36. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/bk-1980-0124.ch001

[84] C. C. Pantelides, M. Nauta, and M. Matzopoulos, “Equation-Oriented Process Modelling Technology : Recent Advances & Current Perspectives,” in *5th Annual TRC-Idemitsu Workshop*, Abu Dhabi, 2015.

[85] J. van Baten, "What is COCO?" 2015. [Online]. Available: http://www.cocosimulator.org/ [Accessed: 2016-02-09]

[86] J. D. Currie, "Practical Applications of Industrial Optimization : From High-Speed Embedded Controllers to Large Discrete Utility Systems," PhD Thesis, Dept. Elect. Eng., AUT University, Auckland, 2014.

[87] B. Braunschweig and R. Gani, *Software Architectures and Tools for Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering. Elsevier Science, 2002. [Online]. Available: https://books.google.co.nz/books?id=GVzsJ_a1m0EC

[88] AMPL Optimization Inc, "AMPL Home Page." [Online]. Available: https://ampl.com [Accessed: 2018-02-01]

[89] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, "GAMS Home Page." [Online]. Available: https://www.gams.com/ [Accessed: 2017-10-26]

[90] S. Poles, M. Vassileva, and D. Sasaki, *Multiobjective Optimization Software.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 329–348. [Online]. Available: https://doi.org/10.1007/978-3-540-88908-3_12

[91] COIN-OR, "FilterSD," 2018. [Online]. Available: https://projects.coin-or.org/filterSD [Accessed: 2018-11-28]

[92] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Knitro: An Integrated Package for Nonlinear Optimization.* Boston, MA: Springer US, 2006, pp. 35–59. [Online]. Available: https://doi.org/10.1007/0-387-30065-1_4

[93] P. E. Gill, W. Murray, M. A. Saunders, and E. Wong, "User's Guide for SNOPT 7.7: Software for Large-Scale Nonlinear Programming," Department of Mathematics, University of California, San Diego, La Jolla, CA, Center for Computational Mathematics Report CCoM 18-1, 2018.

[94] Mathworks, "Optimization Toolbox User's Guide R2017a," Tech. Rep., 2017.

[95] C. Audet, S. Le Digabel, and C. Tribes, "NOMAD user guide," Les cahiers du GERAD, Tech. Rep. G-2009-37, 2009. [Online]. Available: https://www.gerad.ca/nomad/Downloads/user_guide.pdf

[96] A. I. F. Vaz and L. N. Vicente, "A particle swarm pattern search method for bound constrained global optimization," *Journal of Global Optimization*, vol. 39, no. 2, pp. 197–219, oct 2007. [Online]. Available: https://doi.org/10.1007/s10898-007-9133-5

[97] R. Misener and C. A. Floudas, "ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations," *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 503–526, 2014.

[98] A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, F. Serrano, Y. Shinano, J. M. Viernickel, S. Vigerske, D. Weninger, J. T. Witt, and J. Witzig, "The SCIP Optimization Suite 5.0," ZIB, Takustr. 7, 14195 Berlin, Tech. Rep. 17-61, 2017.

[99] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, "Branching and bounds tightening techniques for non-convex MINLP," *Optimization Methods and Software*, vol. 24, no. 4-5, pp. 597–634, 2009.

[100] IBM, "IBM ILOG CPLEX Optimization Studio." [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio [Accessed: 2018-07-14]

[101] IBM, "Boolean Quadric Polytope (BQP) cuts." [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.1/ilog.odms.cplex.help/CPLEX/UsrMan/topics/discr_optim/mip/cuts/28_BQPcuts.html [Accessed: 2019-02-10]

[102] J. Wang, Z. Yan, M. Wang, S. Ma, and Y. Dai, "Thermodynamic analysis and optimization of an (organic Rankine cycle) ORC using low grade heat source," *Energy*, vol. 49, pp. 356–365, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544212008675

[103] Y. Dai, J. Wang, and L. Gao, "Parametric optimization and comparative study of organic Rankine cycle (ORC) for low grade waste heat recovery," *Energy Conversion and Management*, vol. 50, no. 3, pp. 576–582, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.enconman.2008.10.018

[104] E. Martelli, F. Capra, and S. Consonni, "Numerical optimization of Combined Heat and Power Organic Rankine Cycles - Part A: Design optimization," *Energy*, vol. 90, pp. 310–328, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2015.06.111

[105] J. Currie, "OPTI Toolbox," 2017. [Online]. Available: https://www.inverseproblem.co.nz/OPTI/ [Accessed: 2017-10-25]

[106] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[107] J. Löfberg, "YALMIP: Global optimization," 2016. [Online]. Available: https://yalmip.github.io/tutorial/globaloptimization/ [Accessed: 2018-08-02]

[108] Google, "Google Optimization Tools (OR-Tools)," 2018. [Online]. Available: https://developers.google.com/optimization/ [Accessed: 2018-07-15]

[109] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo - Optimization Modeling in Python*, 1st ed. Springer Publishing Company, Incorporated, 2012.

[110] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A Modeling Language for Mathematical Optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[111] R. D. Neidinger, "Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming," *SIAM Review*, vol. 52, no. 3, pp. 545–563, 2010.

[112] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson,

F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "SymPy: symbolic computing in Python," *PeerJ Computer Science*, vol. 3, p. e103, jan 2017. [Online]. Available: https://doi.org/10.7717/peerj-cs.103

[113] Mathworks, "Symbolic Math Toolbox User's Guide," Tech. Rep., 2018.

[114] B. Linnhoff and E. Hindmarsh, "The pinch design method for heat exchanger networks," *Chemical Engineering Science*, vol. 38, no. 5, pp. 745–763, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0009250983801857

[115] M. Shacham, S. Macchieto, L. F. Stutzman, and P. Babcock, "Equation oriented approach to process flowsheeting," *Computers and Chemical Engineering*, vol. 6, no. 2, pp. 79–95, 1982.

[116] D. Meinel, C. Wieland, and H. Spliethoff, "Effect and comparison of different working fluids on a two-stage organic rankine cycle (ORC) concept," *Applied Thermal Engineering*, vol. 63, no. 1, pp. 246–253, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.applthermaleng.2013.11.016

[117] Y. A. Çengel and M. A. Boles, *Thermodynamics: An Engineering Approach*, 6th ed. New York: McGraw-Hill, 2008.

[118] A. Franco and M. Villani, "Optimal design of binary cycle power plants for water-dominated, medium-temperature geothermal fields," *Geothermics*, vol. 38, no. 4, pp. 379–391, 2009.

[119] H. Ghasemi, E. Sheu, A. Tizzanini, M. Paci, and A. Mitsos, "Hybrid solar-geothermal power generation: Optimal retrofitting," *Applied Energy*, vol. 131, pp. 158–170, 2014.

[120] V. Am, J. Currie, and D. I. Wilson, "A systematic approach to modeling organic Rankine cycle systems for global optimization," in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, May 2017, pp. 487–492.

[121] D. S. Remer and L. H. Chai, *Process Equipment, Cost Scale-up*, ser. Chemical Processing and Design Encyclopedia, J. J. McKetta, Ed. Taylor & Francis, 1993.

[122] D. E. Garrett, "Chemical Engineering Economics," in *Chemical Engineering Economics*. Van Nostrand Reinhold, 1989, ch. Appendix 1, pp. 255–308.

[123] E. Wit, E. van den Heuvel, and J. W. Romeijn, "'All models are wrong...': An introduction to model uncertainty," *Statistica Neerlandica*, vol. 66, no. 3, pp. 217–236, 2012.

[124] J. D. Currie, "Inverse Problem Ltd," 2017. [Online]. Available: https://www.inverseproblem.co.nz/ [Accessed: 2017-10-12]

[125] MATLAB, "Solver Takes Too Long," 2018. [Online]. Available: https://www.mathworks.com/help/optim/ug/solver-takes-too-long.html [Accessed: 2018-08-29]

[126] H. P. Williams, *Model building in mathematical programming.* John Wiley & Sons Ltd, 2013.

[127] J. Bisschop, "Integer Linear Programming Tricks," in *AIMMS: Optimization Modeling.* AIMMS B.V., 2016, ch. 7, pp. 75–85.

[128] C. D'Ambrosio, A. Lodi, and S. Martello, "Piecewise linear approximation of functions of two variables in MILP models," *Operations Research Letters*, vol. 38, no. 1, pp. 39–46, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.orl.2009.09.005

[129] MathWorks, "lsqlin," 2017. [Online]. Available: https://au.mathworks.com/help/optim/ug/lsqlin.html [Accessed: 2017-12-02]

[130] S. G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://ab-initio.mit.edu/nlopt [Accessed: 2016-12-18]

[131] F. Borrelli, D. Subramanian, A. Raghunathan, and L. Biegler, "MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles," *2006 American Control Conference*, pp. 5763–5768, 2006.

[132] D. Yue, G. Guillén-Gosálbez, and F. You, "Global optimization of large-scale mixed-integer linear fractional programming problems: A reformulation-linearization method and process scheduling applications," *AIChE Journal*, vol. 59, no. 11, pp. 4255–4272, 2013. [Online]. Available: http://dx.doi.org/10.1002/aic.14185

[133] V. Am, J. Currie, and D. I. Wilson, "A Comparison between NLP and MILP Formulations of Organic Rankine Cycle Systems for Optimization," in *13th International Symposium on Process Systems Engineering (PSE 2018)*, ser. Computer Aided Chemical Engineering, M. R. Eden, M. G. Ierapetritou, and G. P. Towler, Eds. Elsevier, 2018, vol. 44, pp. 991 – 996.

[134] D. S. Hochbaum, "Complexity and algorithms for nonlinear optimization problems," *Annals of Operations Research*, vol. 153, no. 1, pp. 257–296, 2007.

[135] Gurobi Optimization, "Gurobi Optimizer." [Online]. Available: http://www.gurobi.com/products/gurobi-optimizer [Accessed: 2-5-2018]

[136] MathWorks, "patternsearch," 2017. [Online]. Available: http://au.mathworks.com/help/gads/patternsearch.html?searchHighlight=patternsearch&s_tid=doc_srchtitle [Accessed: 2017-12-02]

[137] MathWorks, "intlinprog," 2017. [Online]. Available: http://au.mathworks.com/help/optim/ug/intlinprog.html?searchHighlight=intlinprog&s_tid=doc_srchtitle [Accessed: 2017-12-02]

[138] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008.

Appendix A

The Explicit Optimization Problem of Basaran ORC System

The following optimization problem is detailed in Section 3.2. It is assumed that the objective function is the net output power of the plant and the working fluid is R227ea.

$$
\begin{aligned}
\min \quad & -\dot{W}_{\text{net}} \\
\text{s.t.} \quad & \dot{m}_{\text{WF}} h_{\text{A1}} + \dot{m}_{\text{BR}} h_{\text{BR2}} - \dot{m}_{\text{WF}} h_{\text{A2}} - \dot{m}_{\text{BR}} h_{\text{BR3}} = 0, \\
& \dot{m}_{\text{WF}} h_{\text{A2}} + \dot{m}_{\text{BR}} h_{\text{BR1}} - \dot{m}_{\text{WF}} h_{\text{A3}} - \dot{m}_{\text{BR}} h_{\text{BR2}} = 0, \\
& \dot{m}_{\text{WF}} h_{\text{A3}} - \dot{W}_{\text{turb}} - \dot{m}_{\text{WF}} h_{\text{A4}} = 0, \\
& \dot{m}_{\text{WF}} h_{\text{A4}} + \dot{m}_{\text{CW}} h_{\text{CW1}} - \dot{m}_{\text{WF}} h_{\text{A5}} - \dot{m}_{\text{CW}} h_{\text{CW2}} = 0, \\
& \dot{m}_{\text{WF}} h_{\text{A5}} + \dot{W}_{\text{pump}} - \dot{m}_{\text{WF}} h_{\text{A1}} = 0, \\
& \dot{W}_{\text{pump}} = \frac{\Delta \hat{h}_{\text{is,pump}} \dot{m}_{\text{WF}}}{\eta_{\text{pump}}}, \\
& \dot{W}_{\text{turb}} = \Delta \hat{h}_{\text{is,turb}} \eta_{\text{turb}} \dot{m}_{\text{WF}}, \qquad\qquad \text{(A.1)} \\
& h_{\text{A3}} \leq \hat{h}_{@369\,\text{K}, P_{\text{A1}} \in [P_{\text{A5}}+1, 10]\,\text{bar}}, \\
& h_{\text{A3}} \geq \hat{h}_{@T_{g@10\,\text{bar}}, P_{\text{A1}} \in [P_{\text{A5}}+1, 10]\,\text{bar}}, \\
& h_{\text{A2}} = \hat{h}_{f@P_{\text{A1}} \in [P_{\text{A5}}+1, 10]}, \\
& T_{\text{BR1}} - \hat{T}_{\text{A3}} \geq 1, \\
& \hat{T}_{\text{BR2}} - \hat{T}_{\text{A2}} \geq 1, \\
& T_{\text{BR3}} - \hat{T}_{\text{A1}} \geq 1, \\
& \hat{T}_{\text{A4}} - T_{\text{CW2}} \geq 1, \\
& \dot{W}_{\text{net}} = \dot{W}_{\text{turb}} - \dot{W}_{\text{pump}},
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta\hat{h}_{\mathrm{is,turb}} ={}& 0.0206P_{\mathrm{A1}}^{3} - 0.41P_{\mathrm{A1}}^{2} - 0.000552P_{\mathrm{A1}}^{2}h_{\mathrm{A4}} + \\
& 0.0186P_{\mathrm{A1}}h_{\mathrm{A4}} + 0.556P_{\mathrm{A1}} - 0.0424h_{\mathrm{A4}} - 0.174, \\
\Delta\hat{h}_{\mathrm{is,pump}} ={}& -1.02 \times^{-5} P_{\mathrm{A1}}^{2} + 0.0691P_{\mathrm{A1}} - 0.192, \\
\hat{h}_{@369\,\mathrm{K},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}} ={}& -0.000526P_{\mathrm{A1}}^{3} - 0.00574P_{\mathrm{A1}}^{2} - 0.839P_{\mathrm{A1}} + 403, \\
\hat{h}_{@T_{g@10\,\mathrm{bar}},P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]\,\mathrm{bar}} ={}& -0.00432P_{\mathrm{A1}}^{3} + 0.0289P_{\mathrm{A1}}^{2} - 1.33P_{\mathrm{A1}} + 372, \\
\hat{h}_{f@P_{\mathrm{A1}}\in[P_{\mathrm{A5}}+1,10]} ={}& 0.0297P_{\mathrm{A1}}^{3} - 0.94P_{\mathrm{A1}}^{2} + 15.2P_{\mathrm{A1}} + 176, \\
\hat{T}_{\mathrm{A1}} ={}& -0.0146P_{\mathrm{A1}} + 0.88h_{\mathrm{A1}} + 97.3, \\
\hat{T}_{\mathrm{A2}} ={}& -0.312P_{\mathrm{A1}}^{2} + 9.72P_{\mathrm{A1}} + 260, \\
\hat{T}_{\mathrm{A3}} ={}& -3.77 \times 10^{-5}h_{\mathrm{A3}}^{2} - 0.0179P_{\mathrm{A1}}h_{\mathrm{A3}} + \\
& 1.21h_{\mathrm{A3}} + 8.12P_{\mathrm{A1}} - 119, \\
\hat{T}_{\mathrm{A4}} ={}& -0.000753h_{\mathrm{A4}}^{2} + 1.7h_{\mathrm{A4}} - 197, \\
\hat{T}_{\mathrm{BR2}} ={}& 0.238h_{\mathrm{BR2}} + 273,
\end{aligned}
\tag{A.2}
$$

and the bounds of the decision variables are

$$
\begin{aligned}
211.20 \le{}& h_{\mathrm{A1}} \le 211.77, \\
221.77 \le{}& h_{\mathrm{A2}} \le 263.74, \\
356.82 \le{}& h_{\mathrm{A3}} \le 399.59, \\
330.77 \le{}& h_{\mathrm{A4}} \le 395.72, \\
255.14 \le{}& h_{\mathrm{BR}} \le 401.98, \\
10.00 \le{}& \dot{m}_{\mathrm{WF}} \le 100.0, \\
1100.0 \le{}& \dot{m}_{\mathrm{CW}} \le 1300, \\
3.78 \le{}& P_{\mathrm{A1}} \le 10.00, \\
733.20 \le{}& \dot{W}_{\mathrm{net}} \le 1222.00, \\
30.17 \le{}& \dot{W}_{\mathrm{pump}} \le 50.28, \\
763.37 \le{}& \dot{W}_{\mathrm{turb}} \le 1272.28.
\end{aligned}
\tag{A.3}
$$

The remaining variables that are not listed in (A.3) are constants and can be found or calculated from Table 3.2.

Appendix B

Organic Rankine Cycle SymBuilder Models

This appendix details the MATLAB models of the three binary cycle power plants that are described in Chapter 4. Refer to the corresponding sections in Chapter 4 for the process flow diagram and more information about the individual plant. These models were constructed using the SymBuilder framework via the OPTI Toolbox. To run these models, the reader should refer to the attached files that consist of all the different optimization formulations of each individual model.

B.1 DOEP SymBuilder Model

The SymBuilder model of the DOE Pilot Plant is shown below, which is as per to the assumptions and specifications described in Section 4.2.1. The full simulation file and the thermodynamic approximations are provided in the attachment under the DOE Pilot Plant folder.

```
%% Building the algebraic EO NLP/MINLP model
eoDOEP = SymBuilder();

% Brine constants
eoDOEP.AddConstant(‘mBR’,mBR);
eoDOEP.AddConstant(‘hBR1’,hBR1);
eoDOEP.AddConstant(‘hBR5’,hBR5);
eoDOEP.AddConstant(‘TBR1’,TBR1);
eoDOEP.AddConstant(‘TBR5’,TBR5);
% Cooling water constants
eoDOEP.AddConstant(‘hCW1’,hCW1);
eoDOEP.AddConstant(‘hCW2’,hCW2);
eoDOEP.AddConstant(‘TCW1’,TCW1);
eoDOEP.AddConstant(‘TCW2’,TCW2);
% ORC constants
```

```
eoDOEP.AddConstant(‘nt’,TURB_EFF);
eoDOEP.AddConstant(‘np’,PUMP_EFF);
eoDOEP.AddConstant(‘h8’,h8);
eoDOEP.AddConstant(‘T8’,T8);
eoDOEP.AddConstant(‘z’,z);
% Cost constants
eoDOEP.AddConstant(‘C0turb’,C0turb);
eoDOEP.AddConstant(‘W0turb’,W0turb);
eoDOEP.AddConstant(‘a0turb’,a0turb);
eoDOEP.AddConstant(‘C0pump’,C0pump);
eoDOEP.AddConstant(‘W0pump’,W0pump);
eoDOEP.AddConstant(‘a0pump’,a0pump);

% Energy balance equations
eoDOEP.AddCon(‘mWF*h8 + Wpump - mWF*h1 = 0’);
eoDOEP.AddCon(‘mWF*h1 - mWF*h2 + mBR*hBR4 - mBR*hBR5 = 0’);
eoDOEP.AddCon(‘z*mWF*h3 - z*mWF*h4 + mBR*hBR2 - mBR*hBR3 = 0’);
eoDOEP.AddCon(‘z*mWF*h4 - z*mWF*h5 + mBR*hBR1 - mBR*hBR2 = 0’);
eoDOEP.AddCon(‘z*mWF*h5 - WturbHP - z*mWF*h6 = 0’);
eoDOEP.AddCon(‘mWF*h7 - mWF*h8 + mCW*hCW1 - mCW*hCW2 = 0’);
eoDOEP.AddCon(‘(1-z)*mWF*h9 - (1-z)*mWF*h10 + mBR*hBR3 - mBR*hBR4 = 0’);
eoDOEP.AddCon(‘(1-z)*mWF*h10 - WturbLP - (1-z)*mWF*h11 = 0’);
eoDOEP.AddCon(‘z*mWF*h6 + (1-z)*mWF*h11 - mWF*h7 = 0’);

% Valve constraints
eoDOEP.AddCon(‘h3 = h9’);
eoDOEP.AddCon(‘h2 = h9’);

% Turbine and pump works
eoDOEP.AddCon(‘WturbHP = hpTurbIsen*nt*mWF*z’);
eoDOEP.AddCon(‘WturbLP = lpTurbIsen*nt*mWF*(1-z)’);
eoDOEP.AddCon(‘Wpump = pumpIsen*mWF/np’);
% Net output power
eoDOEP.AddCon(‘Wnet = WturbHP+WturbLP-Wpump’);
% Costs
eoDOEP.AddExpression(‘CturbLP = C0turb*((WturbLP)/W0turb)^a0turb’);
eoDOEP.AddExpression(‘CturbHP = C0turb*((WturbHP)/W0turb)^a0turb’);
eoDOEP.AddExpression(‘Cpump = C0pump*((Wpump)/W0pump)^a0pump’);

% Adding approximations
AddPwFit(‘hpTurbIsen(P5,h5)’,hpTurbIsenWorkFit,eoDOEP);
AddPwFit(‘lpTurbIsen(P10,h10)’,lpTurbIsenWorkFit,eoDOEP);
AddPwFit(‘pumpIsen(P5)’,pumpIsenWorkFit,eoDOEP);
AddPwFit(‘hpHmPT5Ub(P5)’,hpHmPT5UbFit,eoDOEP);
AddPwFit(‘hpHmPT5Lb(P5)’,hpHmPT5LbFit,eoDOEP);
AddPwFit(‘hpSatLiq(P5)’,hpSatLiqFit,eoDOEP);
AddPwFit(‘lpHmPT10Ub(P10)’,lpHmPT10UbFit,eoDOEP);
AddPwFit(‘lpHmPT10Lb(P10)’,lpHmPT10LbFit,eoDOEP);
AddPwFit(‘lpSatLiq(P10)’,lpSatLiqFit,eoDOEP);
AddPwFit(‘T7(h7)’,T7Fit,eoDOEP);
AddPwFit(‘T1(P5,h1)’,T1Fit,eoDOEP);
AddPwFit(‘T2(P5,h2)’,T2Fit,eoDOEP);
AddPwFit(‘TBR4(hBR4)’,TBRFit,eoDOEP);
AddPwFit(‘T9(h9)’,T9Fit,eoDOEP);
AddPwFit(‘T10(P10,h10)’,T10Fit,eoDOEP);
AddPwFit(‘TBR3(hBR3)’,TBRFit,eoDOEP);
AddPwFit(‘T4(h4)’,T4Fit,eoDOEP);
AddPwFit(‘TBR2(hBR2)’,TBRFit,eoDOEP);
AddPwFit(‘T5(P5,h5)’,T5Fit,eoDOEP);

% ORC constraints
```

```
eoDOEP.AddCon(‘hpSatLiq = h4’);
eoDOEP.AddCon(‘h5 <= hpHmPT5Ub’);
eoDOEP.AddCon(‘hpHmPT5Lb <= h5’);
eoDOEP.AddCon(‘lpSatLiq = h9’);
eoDOEP.AddCon(‘h10 <= lpHmPT10Ub’);
eoDOEP.AddCon(‘lpHmPT10Lb <= h10’);
eoDOEP.AddCon(‘P10 <= P5’);
eoDOEP.AddCon(‘1 <= (TBR4-T2)’);
eoDOEP.AddCon(‘1 <= (TBR5-T1)’);
eoDOEP.AddCon(‘1 <= (T7-TCW2)’);
eoDOEP.AddCon(‘1 <= (TBR3-T10)’);
eoDOEP.AddCon(‘1 <= (TBR4-T9)’);
eoDOEP.AddCon(‘1 <= (TBR2-T4)’);
eoDOEP.AddCon(‘1 <= (TBR3-T2)’);
eoDOEP.AddCon(‘1 <= (TBR1-T5)’);
% Brine constraints
eoDOEP.AddCon(‘hBR3 <= hBR2’);
eoDOEP.AddCon(‘hBR4 <= hBR3’);

% ORC Bounds
eoDOEP.AddBounds(sym(‘h1’),h1Lb,h1Ub);
eoDOEP.AddBounds(sym(‘h2’),h9Lb,h9Ub);
eoDOEP.AddBounds(sym(‘h3’),h9Lb,h9Ub);
eoDOEP.AddBounds(sym(‘h4’),h4Lb,h4Ub);
eoDOEP.AddBounds(sym(‘h5’),h5Lb,h5Ub);
eoDOEP.AddBounds(sym(‘h6’),h7Lb,h6Ub);
eoDOEP.AddBounds(sym(‘h7’),h7Lb,h7Ub);
eoDOEP.AddBounds(sym(‘h9’),h9Lb,h9Ub);
eoDOEP.AddBounds(sym(‘h10’),h10Lb,h10Ub);
eoDOEP.AddBounds(sym(‘h11’),h7Lb,h11Ub);
eoDOEP.AddBounds(sym(‘P5’),P5Lb,P5Ub);
eoDOEP.AddBounds(sym(‘P10’),P10Lb,P10Ub);
eoDOEP.AddBounds(sym(‘mWF’),mWFLb,mWFUb);
eoDOEP.AddBounds(sym(‘Wpump’),WpumpLb,WpumpUb);
eoDOEP.AddBounds(sym(‘WturbLP’),WturbLPLb,WturbLPUb);
eoDOEP.AddBounds(sym(‘WturbHP’),WturbHPLb,WturbHPUb);
% Brine and cooling water bounds
eoDOEP.AddBounds(sym(‘hBR’),hBR5,hBR1);
eoDOEP.AddBounds(sym(‘mCW’),mCWLb,mCWUb);
% Net output power bound
eoDOEP.AddBounds(sym(‘Wnet’),WnetLb,WnetUb);

%Objective function
switch (1)
        case 1 % Net power
                eoDOEP.AddObj(‘-(Wnet)’);
        case 2 % Specific rotating machinery cost
                eoDOEP.AddObj(‘(CturbLP+CturbHP+Cpump)/(Wnet*1e-3)’);
end

%Building the optimization problem
Build(eoDOEP)
```

B.2 USGP SymBuilder Model

The SymBuilder model of the USGeo Plant is shown below, which is as per to the assumptions and specifications described in Section 4.3.1. The full simulation file and the thermodynamic approximations are provided in the attachment under the USGeo Plant folder.

```
%% Building the algebraic EO NLP/MINLP model
eoUSGP = SymBuilder();

% Brine constants
eoUSGP.AddConstant(‘mBR’,mBR);
eoUSGP.AddConstant(‘hBR1’,hBR1);
eoUSGP.AddConstant(‘hBR8’,hBR8);
eoUSGP.AddConstant(‘TBR1’,TBR1);
eoUSGP.AddConstant(‘TBR8’,TBR8);
eoUSGP.AddConstant(‘zBR’,zBR);
eoUSGP.AddConstant(‘hBRLb’,hBRLb); % The lowest brine enthalpy
% Cooling water constants
eoUSGP.AddConstant(‘TCW3’,TCW3);
eoUSGP.AddConstant(‘TCW4’,TCW4);
eoUSGP.AddConstant(‘hCW1’,hCW1);
eoUSGP.AddConstant(‘hCW2’,hCW2);
eoUSGP.AddConstant(‘hCW3’,hCW3);
eoUSGP.AddConstant(‘hCW4’,hCW4);

% ORC-A constants
eoUSGP.AddConstant(‘hA7’,hA7);
eoUSGP.AddConstant(‘TA7’,TA7);
eoUSGP.AddConstant(‘ntA’,TURBA_EFF);
eoUSGP.AddConstant(‘npA’,PUMPA_EFF);
eoUSGP.AddConstant(‘dTrec’,dTrec);
% ORC-B constants
eoUSGP.AddConstant(‘hB5’,hB5);
eoUSGP.AddConstant(‘TB5’,TB5);
eoUSGP.AddConstant(‘ntB’,TURBB_EFF);
eoUSGP.AddConstant(‘npB’,PUMPB_EFF);
% Cost constants
eoUSGP.AddConstant(‘C0turb’,C0turb);
eoUSGP.AddConstant(‘W0turb’,W0turb);
eoUSGP.AddConstant(‘a0turb’,a0turb);
eoUSGP.AddConstant(‘C0pump’,C0pump);
eoUSGP.AddConstant(‘W0pump’,W0pump);
eoUSGP.AddConstant(‘a0pump’,a0pump);

% Energy balance of ORC-A
eoUSGP.AddCon(‘hA1*mA + hA5*mA - hA2*mA - hA6*mA = 0’);
eoUSGP.AddCon(‘hA6*mA + hCW1*mCWA - hA7*mA - hCW3*mCWA = 0’);
eoUSGP.AddCon(‘hA7*mA + WpumpA - hA1*mA = 0’);
eoUSGP.AddCon(‘hA2*mA + hBR5*(1-zBR)*mBR - hA3*mA - hBR7*(1-zBR)*mBR = 0’);
eoUSGP.AddCon(‘hA3*mA + hBR1*mBR - hA4*mA - hBR2*mBR = 0’);
eoUSGP.AddCon(‘hA4*mA - WturbA - hA5*mA = 0’);

% Energy balance of ORC-B
eoUSGP.AddCon(‘hB1*mB + hBR4*zBR*mBR - hB2*mB - hBR6*zBR*mBR = 0’);
eoUSGP.AddCon(‘hB2*mB + hBR2*mBR - hB3*mB - hBR3*mBR = 0’);
eoUSGP.AddCon(‘hB3*mB - WturbB - hB4*mB = 0’);
eoUSGP.AddCon(‘hB4*mB + hCW2*mCWB - hB5*mB - hCW4*mCWB = 0’);
eoUSGP.AddCon(‘hB5*mB + WpumpB - hB1*mB = 0’);

% Turbine and pump works
eoUSGP.AddCon(‘WturbA = mA*ntA*turbAIsenWork’);
eoUSGP.AddCon(‘WturbB = mB*ntB*turbBIsenWork’);
eoUSGP.AddCon(‘WpumpA = mA*pumpAIsenWork/npA’);
eoUSGP.AddCon(‘WpumpB = mB*pumpBIsenWork/npB’);
% Net output power
eoUSGP.AddCon(‘Wnet = WturbA+WturbB-WpumpA-WpumpB’);

% Costs
eoUSGP.AddExpression(‘CturbA = C0turb*(WturbA/W0turb)^a0turb’);
eoUSGP.AddExpression(‘CturbB = C0turb*(WturbB/W0turb)^a0turb’);
eoUSGP.AddExpression(‘CpumpA = C0pump*(WpumpA/W0pump)^a0pump’);
eoUSGP.AddExpression(‘CpumpB = C0pump*(WpumpB/W0pump)^a0pump’);

% ORC-A approximations
AddPwFit(‘turbAIsenWork(PA,hA4)’,nlTurbAIsenWorkFit,eoUSGP);
AddPwFit(‘pumpAIsenWork(PA)’,nlPumpAIsenWorkFit,eoUSGP);
AddPwFit(‘HmPTA4UbA(PA)’,nlHmPTA4UbFit,eoUSGP);
AddPwFit(‘HmPTA4LbA(PA)’,nlHmPTA4LbFit,eoUSGP);
AddPwFit(‘satLiqA(PA)’,nlSatLiqHFitA,eoUSGP);
AddPwFit(‘TA1(PA,hA1)’,TA1Fit,eoUSGP);
AddPwFit(‘TA2(PA,hA2)’,TA2Fit,eoUSGP);
AddPwFit(‘TA3(hA3)’,TA3Fit,eoUSGP);
AddPwFit(‘TA4(PA,hA4)’,TA4Fit,eoUSGP);
AddPwFit(‘TA5(hA5)’,TA5A6Fit,eoUSGP);
AddPwFit(‘TA6(hA6)’,TA5A6Fit,eoUSGP);
% ORC-B approximations
AddPwFit(‘turbBIsenWork(PB,hB3)’,nlTurbBIsenWorkFit,eoUSGP);
AddPwFit(‘pumpBIsenWork(PB)’,nlPumpBIsenWorkFit,eoUSGP);
AddPwFit(‘HmPTB3Ub(PB)’,nlHmPTB3UbFit,eoUSGP);
AddPwFit(‘HmPTB3Lb(PB)’,nlHmPTB3LbFit,eoUSGP);
AddPwFit(‘satLiqB(PB)’,nlSatLiqHFitB,eoUSGP);
AddPwFit(‘TB1(PB,hB1)’,TB1Fit,eoUSGP);
AddPwFit(‘TB2(hB2)’,TB2Fit,eoUSGP);
AddPwFit(‘TB3(PB,hB3)’,TB3Fit,eoUSGP);
AddPwFit(‘TB4(hB4)’,TB4Fit,eoUSGP);
% Brine approximations
AddPwFit(‘TBR2(hBR2)’,TBRFit,eoUSGP);
AddPwFit(‘TBR3(hBR3)’,TBRFit,eoUSGP); % Same as TBR5
AddPwFit(‘TBR6(hBR6)’,TBRFit,eoUSGP);
AddPwFit(‘TBR7(hBR7)’,TBRFit,eoUSGP);

% ORC-A constraints
eoUSGP.AddCon(‘hA4 <= HmPTA4UbA’);
eoUSGP.AddCon(‘HmPTA4LbA <= hA4’);
eoUSGP.AddCon(‘hA3 = satLiqA’);
eoUSGP.AddCon(‘TA6 = TA7 + dTrec’);
eoUSGP.AddCon(‘hA1 <= hA2’);
eoUSGP.AddCon(‘hA6 <= hA5’);
eoUSGP.AddCon(‘1 <= (TA6-TCW3)’);
eoUSGP.AddCon(‘1 <= (TA5-TA2)’);
eoUSGP.AddCon(‘1 <= (TA6-TA1)’);
eoUSGP.AddCon(‘1 <= (TBR3-TA3)’);
eoUSGP.AddCon(‘1 <= (TBR7-TA2)’);
eoUSGP.AddCon(‘1 <= (TBR1-TA4)’);
eoUSGP.AddCon(‘1 <= (TBR2-TA3)’);
% ORC-B constraints
eoUSGP.AddCon(‘hB3 <= HmPTB3Ub’);
eoUSGP.AddCon('HmPTB3Lb <= hB3');
eoUSGP.AddCon('hB2 = satLiqB');
eoUSGP.AddCon('1 <= (TB4-TCW4)');
eoUSGP.AddCon('1 <= (TBR3-TB2)');
eoUSGP.AddCon('1 <= (TBR6-TB1)');
eoUSGP.AddCon('1 <= (TBR2-TB3)');
% Brine constraints
eoUSGP.AddCon('hBR6 <= hBR4');
eoUSGP.AddCon('hBR7 <= hBR5');
eoUSGP.AddCon('hBR5 = hBR4');
eoUSGP.AddCon('hBR4 = hBR3');
eoUSGP.AddCon('hBR3 <= hBR2');
eoUSGP.AddCon('hBR8 = (hBR6+hBR7)/2');

% ORC-A bounds
eoUSGP.AddBounds(sym('hA1'),hA1Lb,hA1Ub);
eoUSGP.AddBounds(sym('hA2'),hA1Lb,hA3Ub);
eoUSGP.AddBounds(sym('hA3'),hA3Lb,hA3Ub);
eoUSGP.AddBounds(sym('hA4'),hA4Lb,hA4Ub);
eoUSGP.AddBounds(sym('hA5'),hA5Lb,hA5Ub);
eoUSGP.AddBounds(sym('hA6'),hA5Lb,hA5Ub);
eoUSGP.AddBounds(sym('mA'),mALb,mAUb);
eoUSGP.AddBounds(sym('PA'),PA4Lb,PA4Ub);
eoUSGP.AddBounds(sym('WturbA'),min(WturbALbUb),max(WturbALbUb));
eoUSGP.AddBounds(sym('WpumpA'),min(WpumpALbUb),max(WpumpALbUb));

% ORC-B bounds
eoUSGP.AddBounds(sym('hB1'),hB1Lb,hB1Ub);
eoUSGP.AddBounds(sym('hB2'),hB2Lb,hB2Ub);
eoUSGP.AddBounds(sym('hB3'),hB3Lb,hB3Ub);
eoUSGP.AddBounds(sym('hB4'),hB4Lb,hB4Ub);
eoUSGP.AddBounds(sym('mB'),mBLb,mBUb);
eoUSGP.AddBounds(sym('PB'),PB3Lb,PB3Ub);
eoUSGP.AddBounds(sym('WturbB'),min(WturbBLbUb),max(WturbBLbUb));
eoUSGP.AddBounds(sym('WpumpB'),min(WpumpBLbUb),max(WpumpBLbUb));

% Brine and cooling water bounds
eoUSGP.AddBounds(sym('hBR'),hBRLb,hBR1);
eoUSGP.AddBounds(sym('mCWA'),mCWALb,mCWAUb);
eoUSGP.AddBounds(sym('mCWB'),mCWBLb,mCWBUb);
% Net output power bound
eoUSGP.AddBounds(sym('Wnet'),min(WnetLbUb),max(WnetLbUb));

% Objective function
switch (1)
        case 1 % Net power
                eoUSGP.AddObj('-(Wnet)');
        case 2 % Specific rotating machinery cost
                eoUSGP.AddObj('(CturbA+CturbB+CpumpA+CpumpB)/(Wnet*1e-3)');
end

%Building the optimization problem
Build(eoUSGP)
```

B.3 MBPP SymBuilder Model

The SymBuilder model of the Magmamax Binary Power Plant is shown below, which is as per to the assumptions and specifications described in Section 4.4.1. The full simulation file and the thermodynamic approximations are provided in the attachment under the Magmamax Binary Power Plant folder.

```
%% Building the algebraic EO NLP/MINLP model
eoMBPP = SymBuilder(); %Initializing SymBuilder object

% ORC-A constants
eoMBPP.AddConstant(‘TA4’,TA4);
eoMBPP.AddConstant(‘TA6’,TA6);
eoMBPP.AddConstant(‘hA4’,hA4);
eoMBPP.AddConstant(‘dTrec’,dTrec);
eoMBPP.AddConstant(‘turbAEff’,TURBA_EFF);
eoMBPP.AddConstant(‘pumpAEff’,PUMPA_EFF);
% ORC-B constants
eoMBPP.AddConstant(‘hB3’,hB3);
eoMBPP.AddConstant(‘TB3’,TB3);
eoMBPP.AddConstant(‘turbBEff’,TURBB_EFF);
eoMBPP.AddConstant(‘pumpBEff’,PUMPB_EFF);
% Brine constants
eoMBPP.AddConstant(‘mBR’,mBR);
eoMBPP.AddConstant(‘hBR1’,hBR1);
eoMBPP.AddConstant(‘hBR5’,hBR5);
eoMBPP.AddConstant(‘TBR1’,TBR1);
eoMBPP.AddConstant(‘TBR5’,TBR5);
% Cooling water constants
eoMBPP.AddConstant(‘hCW1’,hCW1);
eoMBPP.AddConstant(‘hCW3’,hCW3);
eoMBPP.AddConstant(‘TCW1’,TCW1);
eoMBPP.AddConstant(‘TCW3’,TCW3);
% Cost constants
eoMBPP.AddConstant(‘C0turb’,C0turb);
eoMBPP.AddConstant(‘W0turb’,W0turb);
eoMBPP.AddConstant(‘a0turb’,a0turb);
eoMBPP.AddConstant(‘C0pump’,C0pump);
eoMBPP.AddConstant(‘W0pump’,W0pump);
eoMBPP.AddConstant(‘a0pump’,a0pump);

% Energy balance equation of ORC-A
eoMBPP.AddCon(‘mA*hA1 - WturbA - mA*hA2 = 0’);
eoMBPP.AddCon(‘mA*hA2 - mA*hA2P + mB*hB5P - mB*hB5 = 0’);
eoMBPP.AddCon(‘mA*hA2P - mA*hA3 + mB*hB4 - mB*hB5P = 0’);
eoMBPP.AddCon(‘mA*hA3 - mA*hA4 + mCW*hCW2 - mCW*hCW3 = 0’);
eoMBPP.AddCon(‘mA*hA4 + WpumpA - mA*hA5 = 0’);
eoMBPP.AddCon(‘mA*hA5 - mA*hA6 + mBR*hBR4 - mBR*hBR5 = 0’);
eoMBPP.AddCon(‘mA*hA6 - mA*hA7 + mBR*hBR2 - mBR*hBR3 = 0’);
eoMBPP.AddCon(‘mA*hA7 - mA*hA1 + mBR*hBR1 - mBR*hBR2 = 0’);
% Energy balance equation of ORC-B
eoMBPP.AddCon(‘mB*hB1 - WturbB - mB*hB2 = 0’);
eoMBPP.AddCon(‘mB*hB2 - mB*hB3 + mCW*hCW1 - mCW*hCW2 = 0’);
eoMBPP.AddCon(‘mB*hB3 + WpumpB - mB*hB4 = 0’);
eoMBPP.AddCon(‘mB*hB5 - mB*hB1 + mBR*hBR3 - mBR*hBR4 = 0’);

% Turbine and pump works
```

```
eoMBPP.AddCon(‘WturbA = mA*turbAEff*turbAIsenWork’);
eoMBPP.AddCon(‘WturbB = mB*turbBEff*turbBIsenWork’);
eoMBPP.AddCon(‘WpumpA = mA*pumpAIsenWork/pumpAEff’);
eoMBPP.AddCon(‘WpumpB = mB*pumpBIsenWork/pumpBEff’);
% Net output power
eoMBPP.AddCon(‘Wnet = WturbA+WturbB-WpumpA-WpumpB’);
% Costs
eoMBPP.AddExpression(‘CturbA = C0turb*(WturbA/W0turb)^a0turb’);
eoMBPP.AddExpression(‘CturbB = C0turb*(WturbB/W0turb)^a0turb’);
eoMBPP.AddExpression(‘CpumpA = C0pump*(WpumpA/W0pump)^a0pump’);
eoMBPP.AddExpression(‘CpumpB = C0pump*(WpumpB/W0pump)^a0pump’);

% ORC-A approximations
AddPwFit(‘turbAIsenWork(PA,hA1)’,nlTurbAIsenWorkFit,eoMBPP);
AddPwFit(‘pumpAIsenWork(PA)’,nlPumpAIsenWorkFit,eoMBPP);
AddPwFit(‘HmPTcA(PA)’,nlHmTcAFit,eoMBPP);
AddPwFit(‘HmPTA1UbA(PA)’,nlHmPTA1UbAFit,eoMBPP);
AddPwFit(‘HmPTA1LbA(PA)’,nlHmPTA1LbAFit,eoMBPP);
AddPwFit(‘TA1(PA,hA1)’,TA1Fit,eoMBPP);
AddPwFit(‘TA2(hA2)’,TA2A3Fit,eoMBPP);
AddPwFit(‘TA2P(hA2P)’,TA2A3Fit,eoMBPP);
AddPwFit(‘TA3(hA3)’,TA2A3Fit,eoMBPP);
AddPwFit(‘TA5(PA,hA5)’,TA5Fit,eoMBPP);
AddPwFit(‘TA7(PA,hA7)’,TA7Fit,eoMBPP);
% ORC-B approximations
AddPwFit(‘turbBIsenWork(PB,hB1)’,nlTurbBIsenWorkFit,eoMBPP);
AddPwFit(‘pumpBIsenWork(PB)’,nlPumpBIsenWorkFit,eoMBPP);
AddPwFit(‘HsatVapB(PB)’,nlHsatVapBFit,eoMBPP);
AddPwFit(‘HsatLiqB(PB)’,nlHsatLiqBFit,eoMBPP);
AddPwFit(‘HmPTB1UbB(PB)’,nlHmPTB1UbBFit,eoMBPP);
AddPwFit(‘HmPTB1LbB(PB)’,nlHmPTB1LbBFit,eoMBPP);
AddPwFit(‘TB1(PB,hB1)’,TB1Fit,eoMBPP);
AddPwFit(‘TB2(hB2)’,TB2Fit,eoMBPP);
AddPwFit(‘TB4(PB,hB4)’,TB4Fit,eoMBPP);
AddPwFit(‘TB5(PB)’,TB5Fit,eoMBPP);
AddPwFit(‘TB5P(PB)’,TB5Fit,eoMBPP);
% Brine approximations
AddPwFit(‘TBR2(hBR2)’,TBRFit,eoMBPP);
AddPwFit(‘TBR3(hBR3)’,TBRFit,eoMBPP);
AddPwFit(‘TBR4(hBR4)’,TBRFit,eoMBPP);
% Cooling water approximations
AddPwFit(‘TCW2(hCW2)’,TCWFit,eoMBPP);

% ORC-A constraints
eoMBPP.AddCon(‘hA7 = (hA6 + hA1)/2’);
eoMBPP.AddCon(‘hA6 = HmPTcA’);
eoMBPP.AddCon(‘hA1 <= HmPTA1UbA’);
eoMBPP.AddCon(‘HmPTA1LbA <= hA1’);
eoMBPP.AddCon(‘TA3 = TA4 + dTrec’);
eoMBPP.AddCon(‘1 <= (TA3-TCW3)’);
eoMBPP.AddCon(‘1 <= (TA4-TCW2)’);
eoMBPP.AddCon(‘1 <= (TA2-TB5)’);
eoMBPP.AddCon(‘1 <= (TA2P-TB5P)’);
eoMBPP.AddCon(‘1 <= (TA3-TB4)’);
eoMBPP.AddCon(‘1 <= (TBR1-TA1)’);
eoMBPP.AddCon(‘1 <= (TBR2-TA7)’);
eoMBPP.AddCon(‘1 <= (TBR3-TA6)’);
eoMBPP.AddCon(‘1 <= (TBR4-TA6)’);
eoMBPP.AddCon(‘1 <= (TBR5-TA5)’);
% ORC-B constraints
eoMBPP.AddCon(‘hB5 = HsatVapB’);
eoMBPP.AddCon(‘hB5P = HsatLiqB’);
eoMBPP.AddCon(‘hB1 <= HmPTB1UbB’);
eoMBPP.AddCon(‘HmPTB1LbB <= hB1’);
eoMBPP.AddCon(‘(hA2-hA3)*mA*0.02 <= (hB1-hB5)*mB’);
eoMBPP.AddCon(‘1 <= (TB2-TCW2)’);
eoMBPP.AddCon(‘1 <= (TBR3-TB1)’);
eoMBPP.AddCon(‘1 <= (TBR4-TB5)’);
% Brine constraints
eoMBPP.AddCon(‘hBR3 <= hBR2’);
eoMBPP.AddCon(‘hBR4 <= hBR3’);

% ORC-A bounds
eoMBPP.AddBounds(sym(‘hA1’),hA1Lb,hA1Ub);
eoMBPP.AddBounds(sym(‘hA2’),hA2Lb,hA2Ub);
eoMBPP.AddBounds(sym(‘hA2P’),hA2Lb,hA2Ub);
eoMBPP.AddBounds(sym(‘hA3’),hA2Lb,hA2Ub);
eoMBPP.AddBounds(sym(‘hA5’),hA5Lb,hA5Ub);
eoMBPP.AddBounds(sym(‘hA6’),hA6Lb,hA6Ub);
eoMBPP.AddBounds(sym(‘hA7’),hA7Lb,hA7Ub);
eoMBPP.AddBounds(sym(‘mA’),mALb,mAUb);
eoMBPP.AddBounds(sym(‘PA’),PA1Lb,PA1Ub);
eoMBPP.AddBounds(sym(‘WturbA’),min(WturbALbUb),max(WturbALbUb));
eoMBPP.AddBounds(sym(‘WpumpA’),min(WpumpALbUb),max(WpumpALbUb));
% ORC-B bounds
eoMBPP.AddBounds(sym(‘hB1’),hB1Lb,hB1Ub);
eoMBPP.AddBounds(sym(‘hB2’),hB2Lb,hB2Ub);
eoMBPP.AddBounds(sym(‘hB4’),hB4Lb,hB4Ub);
eoMBPP.AddBounds(sym(‘hB5P’),hB5PLb,hB5PUb);
eoMBPP.AddBounds(sym(‘hB5’),hB5Lb,hB5Ub);
eoMBPP.AddBounds(sym(‘mB’),mBLb,mBUb);
eoMBPP.AddBounds(sym(‘PB’),PB1Lb,PB1Ub);
eoMBPP.AddBounds(sym(‘WturbB’),min(WturbBLbUb),max(WturbBLbUb));
eoMBPP.AddBounds(sym(‘WpumpB’),min(WpumpBLbUb),max(WpumpBLbUb));
% Brine and cooling water bounds
eoMBPP.AddBounds(sym(‘hBR’),hBR5,hBR1);
eoMBPP.AddBounds(sym(‘hCW’),hCW1,hCW3);
eoMBPP.AddBounds(sym(‘mCW’),mCWLb,mCWUb);
% Net output power bound
eoMBPP.AddBounds(sym(‘Wnet’),min(WnetLbUb),max(WnetLbUb));

% Objective function
switch (1)
        case 1 % Net power
                eoMBPP.AddObj(‘-Wnet’);
        case 2 % Specific rotating machinery cost
                eoMBPP.AddObj(‘(CturbA+CturbB+CpumpA+CpumpB)/(Wnet*1e-3)’);
end

% Building the optimization problem
Build(eoMBPP)
```

Appendix C

The `pwfit` Model Library

Table C.1 and Table C.2 list the univariate and bivariate polynomial function models that are available in `pwfit`.

Table C.1: Curve fitting models.

`poly1`	$\mathrm{t}_1 x + \mathrm{t}_2$
`poly2`	$\mathrm{t}_1 x^2 + \mathrm{t}_2 x + \mathrm{t}_3$
`poly3`	$\mathrm{t}_1 x^3 + \mathrm{t}_2 x^2 + \mathrm{t}_3 x + \mathrm{t}_4$
`poly4`	$\mathrm{t}_1 x^4 + \mathrm{t}_2 x^3 + \mathrm{t}_3 x^2 + \mathrm{t}_4 x + \mathrm{t}_5$
`poly5`	$\mathrm{t}_1 x^5 + \mathrm{t}_2 x^4 + \mathrm{t}_3 x^3 + \mathrm{t}_4 x^2 + \mathrm{t}_5 x + \mathrm{t}_6$

Table C.2: Surface fitting models.

`poly11`	$\mathrm{t}_1 x + \mathrm{t}_2 y + \mathrm{t}_3$
`poly12`	$\mathrm{t}_1 y^2 + \mathrm{t}_2 xy + \mathrm{t}_3 x + \mathrm{t}_4 y + \mathrm{t}_5$
`poly13`	$\mathrm{t}_1 y^3 + \mathrm{t}_2 y^2 + \mathrm{t}_3 xy^2 + \mathrm{t}_4 xy + \mathrm{t}_5 x + \mathrm{t}_6 y + \mathrm{t}_7$
`poly21`	$\mathrm{t}_1 x^2 + \mathrm{t}_2 xy + \mathrm{t}_3 x + \mathrm{t}_4 y + \mathrm{t}_5$
`poly22`	$\mathrm{t}_1 x^2 + \mathrm{t}_2 y^2 + \mathrm{t}_3 xy + \mathrm{t}_4 x + \mathrm{t}_5 y + \mathrm{t}_6$
`poly23`	$\mathrm{t}_1 y^3 + \mathrm{t}_2 x^2 + \mathrm{t}_3 y^2 + \mathrm{t}_4 x^2 y + \mathrm{t}_5 xy^2 + \mathrm{t}_6 xy + \mathrm{t}_7 x + \mathrm{t}_8 y + \mathrm{t}_9$
`poly31`	$\mathrm{t}_1 x^3 + \mathrm{t}_2 x^2 + \mathrm{t}_3 x^2 y + \mathrm{t}_4 xy + \mathrm{t}_5 x + \mathrm{t}_6 y + \mathrm{t}_7$
`poly32`	$\mathrm{t}_1 x^3 + \mathrm{t}_2 x^2 + \mathrm{t}_3 y^2 + \mathrm{t}_4 x^2 y + \mathrm{t}_5 xy^2 + \mathrm{t}_6 xy + \mathrm{t}_7 x + \mathrm{t}_8 y + \mathrm{t}_9$
`poly33`	$\mathrm{t}_1 x^3 + \mathrm{t}_2 y^3 + \mathrm{t}_3 x^2 + \mathrm{t}_4 y^2 + \mathrm{t}_5 x^2 y + \mathrm{t}_6 xy^2 + \mathrm{t}_7 xy + \mathrm{t}_8 x + \mathrm{t}_9 y + \mathrm{t}_{10}$

Appendix D

Optimization Solvers

The optimization solvers, and their respective release version, that were used in this research are listed in Table D.1.

Table D.1: The optimization solvers that were used in this research. NC: Noncommercial; C: Commercial.

Solver	**Developer**	**Version**	**License**	**Ref.**
IPOPT	A. Wachter and L. T. Biegler	3.12.9	NC	[18]
FILTERSD	Roger Fletcher	1.0	NC	[91]
`fmincon`	MathWorks	7.6	C	[17]
`pattersearch`	MathWorks	3.4.2	C	[136]
CPLEX	IBM	12.8.0.0	NC/C	[100]
GUROBI	Gurobi Optimization	7.5.2	NC/C	[135]
`intlinprog`	MathWorks	7.6	C	[137]
BARON	Nick Sahinidis	18.8.23	C	[22]
SCIP	Tobias Achterberg et al.	5.0.1	NC/C	[21]
BONMIN	Pierre Bonami et al.	1.8.6	NC	[138]

Appendix E

Working Fluids

Table E.1 lists some of the properties of the working fluids that are used in the thesis. These working fluids were selected based on their availability in JSteam (and REFPROP) and for being thermodynamically suitable with the plant specifications discussed in Chapter 3 and Chapter 4.

Table E.1: Properties of the working fluids that were used in the thesis.

Working Fluid Name		T_{crit} **[K]**	P_{crit} **[bar]**	**Molecular Weight**
R134a	1,1,1,2-Tetrafluoroethane	374.2	40.59	102.0
R143a	1,1,1-Trifluoroethane	345.9	37.61	84.04
R152a	1,1-Difluoroethane	386.4	45.17	66.05
R218	Octafluoropropane	345.0	26.40	188.0
R227ea	1,1,1,2,3,3,3-Heptafluoropropane	374.9	29.25	170.0
R236ea	1,1,1,2,3,3-Hexafluoropropane	412.4	34.20	152.0
R236fa	1,1,1,3,3,3-Hexafluoropropane	398.1	32.00	152.0
R245ca	1,1,2,2,3-Pentafluoropropane	447.6	39.41	134.0
R245fa	1,1,1,3,3-Pentafluoropropane	427.2	36.51	134.0
R290	Propane	369.9	42.51	44.10
R32	Difluoromethane	351.3	57.82	52.02
R600	n-Butane	425.1	37.96	58.12
R600a	IsoButane	407.8	36.29	58.12
R601a	IsoPentane	460.4	33.78	72.15
RC270	Cyclopropane	398.3	55.80	42.08
RC318	Octafluorocyclobutane	388.4	27.78	200.0
C5F12	PerFluoroPentane	420.6	20.45	288.0
CF3I	Trifluoroiodomethane	396.4	39.53	195.9

Appendix F

Attached Files

Table F.1 lists the files that are included with this thesis. These files are provided to support the work presented in this thesis. The files can be downloaded from: https://1drv.ms/f/s!Avj6NgoXKYyRgchKNsDIuswfO6XDmw

Table F.1: Files included with the thesis.

Folder	**Description**
EO Toolbox	The latest version of optimization tools and linearization functions discussed in the thesis. Requires JSteam and OPTI installed.
Basaran ORC	The SM and algebraic EO models of the Basran ORC system.
DOE Pilot Plant	The SM and algebraic EO models of the DOE Pilot Plant. Also includes the mixture study.
USGeo Plant	The SM and algebraic EO models of the USGeo Plant.
Magmamax Binary Power Plant	The SM and algebraic EO models of the Magmamax Binary Power Plant.
JSteam	The JSteam MATLAB Interface v1.72 and JSteam Excel Add-In v3.20 that were used in this research.
OPTI Toolbox	The OPTI Toolbox v2.28 that was used in this research.