Development and Comparison of Compressor Models for Heat Pump Applications in TRNSYS

Master Thesis in Energy

Thermal Machines

Elisabeth Vindenes

University of Bergen

Geophysical Institute

Western Norway University of Applied Sciences

Faculty of Engineering and Sciences

Bergen, June 2018

Development and Comparison of Compressor Models for Heat Pump Applications in TRNSYS

Elisabeth Vindenes

University of Bergen (UiB)
The Faculty of Mathematics and Natural Sciences
Geophysical Institute
Post Box 7803
5020 Bergen, Norway

In cooperation with:
Western Norway University of Applied Sciences (HVL)
The Faculty of Engineering and Sciences
Department of Mechanical and Marine Engineering
Post Box 7030
5020 Bergen, Norway

Norwegian title:	Utvikling og sammenligning av kompressormodeller for varmepumpeapplikasjoner i TRNSYS

Preface

This master thesis is the final project of the Master in Energy program at the University in Bergen (UiB). My specialization is energy technology with subject thermal machines. The subject of thermal machines is mainly based at the Western Norway University of Applied Sciences (HVL), where I have spent most days studying for exams and producing this thesis. Prior to the master's degree I graduated from a bachelor's program in energy technology at the Western Norway University of Applied Sciences. The work of this thesis was conducted from January – June 2018.

The choice of subject for this project is based on my interests, work experience and future career plans. During my studies I have worked part time at Sweco Norway with amongst others ventilation and heating systems of buildings, and I have a large interest for heat pumps and refrigeration systems. This is what I want to work with and learn even more about in my career. After finishing my master's degree, I will start working full time at Sweco Norway. I am very grateful and excited for this opportunity and hope to work with heat pumps and refrigeration systems especially.

Working on my master thesis has been both a challenging and exciting experience. The objective of the project has changed during the course of the work. The process has been very enlightening, and I will definitely use this knowledge in my future career. The software used for simulations of heat pump systems in this project, Transient System Simulation Tool TRNSYS, is a new software to HVL. Many hours have been spent on figuring out the "do's and don'ts" of the program. In addition, programming was a new experience.

I would like to thank my supervisors Norbert Lümmen at HVL, Jørgen B. Skauge at Sweco, and Martin Fernø at UiB for their help with this thesis. Norbert Lümmen has been a great support, helping me to learn programming and reading numerous codes and drafts of the thesis. Jørgen B. Skauge has given me great guidance, answering many questions regarding the practical dimensioning and structure of heat pump and refrigeration systems. Martin Fernø has provided good input on the structure of the report. I would also like to thank my fellow master students in room D425 at HVL for great support and friendship. A special thanks to Hanne who also worked with TRNSYS, for good conversations and support during this project.

My husband Martin has been a great support, listening to all my thoughts and problems during the work of this thesis. Out two beautiful children, Alma and Sverre, deserve great thanks for forcing me to take my mind off this project from time to time. I could not have done this without you!

Bergen, 01.06.2018

Elisabeth Vindenes

Elisabeth Vindenes

Abstract

Heat pump systems may serve as a replacement for many heating systems using fossil fuels for heating purposes. Heat pumps are widely used in Norway where electric energy is abundant and inexpensive. Research and development of heat pump components are important to further improve component and system performance. The introduction of alternative refrigerants set new demands to heat pump components and the compressor especially.

The compressor is the driving component of the heat pump cycle and requires work input. Improving compressor performance is of great interest, as the system performance is highly affected by the compressor power demand. In addition, the proper choice of compressor for the given system and operational conditions are of great importance. The Transient System Simulation Tool (TRNSYS) is used for simulations of the vapor compression cycle in this project. Components for heat pump simulations on component level are not available in the component library included in the software. The source codes of TRNSYS components are available to the user, and components may be changed or created. The available components in TRNSYS are written in the Fortran90 programming language.

The aim of this project is to create models of variable speed compressors and other heat pump components for use in TRNSYS simulations. The compressor models are integrated into simple vapor compression refrigeration cycles, which work in heat pump mode. The behavior and properties of the heat pumps implemented with different compressor models are investigated as function of the time dependent heat demand of a simple building located in Bergen, Norway.

Two compressor components are created. The compressor models differentiate in the method of calculating the compressor power input. One compressor model is based on a 10-coefficient polynomial for the compressor power input, the other compressor model is based on a polytropic compression process. In addition, an evaporator and condenser component, a throttling device component and a simplified building load component are created. Various simulations are conducted in order to verify that the models and components are consistent with theoretical predictions and published data. The report provides an extensive description of TRNSYS and the process of creating and implementing new components.

The components perform as expected in terms of calculating compressor power input and heat transfer rates of the condenser and evaporator. These results are consistent with the energy balance for heat pump systems. The total energy consumption of the polytropic compressor is 21 % lower than the total energy consumption of the polynomial based compressor at standard conditions. The components and models have certain limitations in terms of simulations of the heat pump and compressor performance at varying operational conditions. This is due to the calculations and simplifications of the various components. The components are considered a good starting point for further development, making the components and TRSNYS models better reflect the varying operational conditions of a real heat pump system.

Elisabeth Vindenes

Sammendrag

Varmepumper kan erstatte mange oppvarmingssystemer basert på fossile brensler. Varmepumper er i utstrakt bruk i Norge, mye grunnet god tilgjengelighet på rimelig elektrisitet. Forskning på og videreutvikling av varmepumpekomponenter er viktig for å forbedre virkningsgrader og ytelse for både komponenter og system. Nye kjølemidler setter andre krav til varmepumpekomponenter, og spesielt kompressoren.

Kompressoren krever et tilført arbeid, og er den drivende komponenten i varmepumpesyklusen. Utvikling og forbedring av kompressorytelse er viktig fordi systemets ytelse i stor grad er avhengig av det nødvendige tilførte arbeidet til kompressoren. I tillegg er valg av kompressor til det bestemte system og driftsforhold viktig. Til simuleringer av damp-kompresjonssyklusene i prosjektet brukes Programmet Transient System Simulation Tool (TRNSYS). De nødvendige komponenter til å utføre simuleringer av en varmepumpe på komponentnivå er ikke tilgjengelig i komponentbiblioteket som medfølger TRNSYS. Kodene til alle komponentene i TRNSYS er tilgjengelig for brukeren. Brukeren kan da selv modifisere eksisterende komponenter eller lage nye komponenter. Komponentkodene er skrevet i programmeringsspråket Fortran90.

Målet med dette prosjektet er å lage modeller av frekvensregulerte kompressorer og andre varmepumpekomponenter til å bruke i TRNSYS simuleringer. Kompressorkomponentene settes inn i enkle damp-kompresjonssykluser som opererer i varmepumpemodus. Egenskapene til disse varmepumpene med ulike kompressorkomponenter undersøkes som funksjon av det varierende varmebehovet til en enkel bygning i Bergen, Norge.

To ulike kompressorkomponenter er laget. De to kompressorkomponentene benytter ulike metoder for å beregne kompressorens effektbehov. Kompressorens effektbehov beregnes ved et polynom i den ene modellen, og basert på en polytropisk kompresjonsprosess i den andre. I tillegg er en fordamper- og kondensatorkomponent, en ekspasjonsventil-komponent og en forenklet komponent for bygningslasten laget. Ulike simuleringer er utført for å verifisere modellene opp mot teoretiske betraktninger og tidligere publiserte data og resultater. En utfyllende beskrivelse av TRNSYS og fremgangsmåte for å lage og implementere nye komponenter er også inkludert.

Beregning av effektforbruket til kompressoren og varmeoverføringsratene i fordamper og kondensator er som forventet basert på en energibalanse for varmepumper. Det totale energiforbruket til de to kompressormodellene er ikke konsistent. Den polytropiske kompressoren krever 21 % mindre energi enn den polynom-baserte kompressoren. Komponentene og modellene har noen begrensninger når det gjelder simulering av ytelsen til både kompressor og varmepumpe ved varierende driftsforhold. Dette kan skyldes de valgte beregningsmetodene og forenklingene som er gjort i de ulike komponentene. Komponentene er ansett å være et godt utgangspunkt for videre utvikling, for å bedre kunne representere de varierende driftsforholdene til et virkelig varmepumpesystem.

Table of Contents

Nomenclature

E	Energy	kJ
$\dot{E}$	Power	W
$\dot{Q}$	Heat transfer rate	W or kW
$\dot{W}$	Power input	W or kW
$\dot{m}$	Mass flow rate	kg/s or kg/h
h	Specific enthalpy	kJ/kg
u	Velocity	m/s
g	Gravitational acceleration	m/s^2
z	Height	m
COP	Coefficient of performance	–
T	Temperature	K or °C
SCOP	Seasonal coefficient of performance	–
η	Efficiency	–
q	Specific heat	kJ/kg
w	Specific work	kJ/kg
s	Specific entropy	kJ/kgK
v	Specific volume	m^3/kg
P	Pressure	Pa, kPa or MPa
C	Constant	–
γ	Isentropic exponent	–
n	Polytropic exponent	–
$\dot{V}$	Volumetric flow rate	m^3/s
V	Volume	m^3
μ	Joule-Thomson coefficient	K/Pa
VC	Volumetric capacity	kJ/m^3
ρ	Density	kg/m^3
rpm	Rotational speed	1/min
f	Frequency	Hz
ω	Revolutions per second	rev/s
$C_1 - C_{10}$	Compressor coefficients	–
CR	Compression ratio	–
r_{loss}	Fractional heat loss	–
UA	Overall heat loss coefficient	W/°C
DUT	Dimensioning ambient temperature	°C
e	Tracking error	–
y	Controlled variable	–
y_{set}	Setpoint	–
W	Energy consumption	kWh
t	Simulation time step	hours
Q	Heat energy	kWh

Elisabeth Vindenes

1 Introduction

Increased focus on minimizing the use of fossil fuels drives the development of new technologies forward. In developed countries, about 40% of the primary energy is used in buildings, where 30% is for heating and cooling purposes [1, 2]. Over 9 million heat pumps are installed in the European Union (including Norway) by January 2018, and Scandinavia has the highest number of heat pump installed per 1000 households [3]. Heat pump technology is an important part of decreasing energy consumption and the related emissions. Heat pumps may be used for both cooling- and heating purposes, even both simultaneously.

Due to the large amount of hydropower in Norway, electricity is abundant and quite inexpensive. It is common to use electricity directly for both residential heating and domestic hot water. Heat pumps are increasingly popular in both residential, commercial, and industrial applications. This is due to both increased focus on environmental issues, economy, and incentives from the government to install such systems. From 2020 it is prohibited to use fossil fuels for heating purposes in Norway [4], and heat pumps may be a good replacement for such systems. In other European countries electricity is more expensive, and it is still common to use fossil fuels for both residential heating and domestic hot water.

Environmental considerations demand a transition from the traditional refrigerants to a new generation of refrigerants. The first synthetic refrigerants were introduced in the 1930's and became popular due to their stability. These refrigerants are non-toxic, non-flammable, compatible with materials and lubrication and has the right thermodynamic properties giving good cooling and heating capacity and system performance. The main drawback with the traditional refrigerants is their impact on the atmosphere and climate. The first synthetic refrigerants where prohibited in 1987 due to their impact on the ozone layer. A second generation of synthetic refrigerants with no ozone depletion potential (ODP) where developed and are widely used today. The Kigali Amendment to the Montreal Protocol from 2007 started the phase out of the second generation of synthetic refrigerants because of their high global warming potential (GWP) [5]. EU regulation no. 517 (F-gas directive) aims to decrease the emissions in EU of fluorinated greenhouse gases by almost 70% by 2030 [6]. The regulation will reduce the amount of high GWP refrigerants available and introduce prohibitions on the use of these refrigerants. In 2016 fluoric gases stands for 3 % of the Norwegian greenhouse gas (GHG) emissions, and most of this comes from high-GWP refrigerants in heat pump- and refrigeration systems [7].

The new generation of refrigerants consist of both natural and synthetic refrigerants. The natural refrigerants (e.g. hydrocarbons, ammonium, and carbon dioxide) all have various challenges regarding performance, flammability, toxicity and compatibility with lubricants and materials. The synthetic alternatives are being developed as replacements for older refrigerants in existing systems. These synthetic refrigerants do not give the same performance and cooling and heating capacity as the traditional refrigerants and hence demands larger systems and components. The natural refrigerants may have different requirements to system configuration and components and may not be able to directly replace the refrigerant in an existing system.

Due to the various applications of heat pumps and the different refrigerants available there are many different component types and system configurations. This project mainly focuses on the compressor in the heat pump system. The net performance of the heat pump is highly dependent on the

compressor performance [8]. The compressor drives the thermodynamic cycle and consumes electric energy. Many different compressor types have been developed, covering all the various applications and requirements. Vapor compression facilities mostly operate on part load, due to large variations in heating- or cooling load [9-11]. As compressors are designed to have optimal performance at full load, there are increased losses at part load [9, 12]. The optimal scenario would be to obtain a proportional relationship between the absorbed power to the compressor and the capacity of the system [13].

Research and studies of heat pump systems and the related components are important to further improve the efficiencies and performance of the components. Better knowledge of system and component performance facilitates for choosing the correct components for the given operational conditions and refrigerant. Optimization in terms of regulation and compressor control is considered important for performance improvements. The work of this project may be further developed to model a complete heat pump system and building load heat demand, covering most parameters and variables of a real system. The main focus of this project is the compressor in the heat pump or refrigeration system. The compressor is an important component for heat pump performance, and the energy consumption of the system highly depends on the compressor efficiency [9, 14, 15]. Variable speed drive compressors may give significant energy savings compared to the conventional on/off thermostatic control [9]. Simulations of compressors in heat pump systems may provide important knowledge and experience regarding the choice of compressor for any application.

1.1 Objectives

The main aim of this thesis is to simulate a variable speed heat pump system in software Transient System Simulation Tool (TRNSYS). Any components necessary that are not available in TRNSYS component libraries are to be created. TRNSYS is a software for simulations of almost any energy system. An objective of this thesis is to investigate the possibilities regarding the simulation of heat pump systems on component level in TRNSYS. The possible methods and approaches to modelling compressors are to be explored and the components implemented in TRSNYS simulations. The source codes of all TRNSYS components are avaible, enabling the user to modify existing or create new components. The process of creating new components for TRNSYS simulations is explored and described. As TRNSYS is a new program at Western Norway University of Applied Sciences and Engineering (HVL), the program and the approach to creating new components should be described.

The created components are to be verified based on results from appropriate simulations in TRNSYS, theoretical predictions, and published data. TRNSYS mainly focuses on energy systems, and the ability of the program to simulate systems on component level is to be investigated. Recommendations and suggestions for further development of the models to better represent the varying operational conditions of a real heat pump system must be included.

.

2 Background

This chapter provides a short historical review of both compressors and heat pump and refrigeration systems, a brief introduction to classification of compressor models and a literature review presenting some of the work done on compressor and heat pump performance, new refrigerant alternatives, compressor design improvements and compressor models. Finally, the structure of the report is presented.

2.1 Historical development

The first compressors emerged in the 15th century and were developed to compress air [16]. Compressors have been in use since the start of the industrial revolution in the 16th century [17]. In the 17th century further developments included multistage compression and cooling of the compression process. The idea of artificial cooling emerged in the 18th century. William Cullen, a Scottish physician and professor of medicine, discovered the cooling effect of evaporating fluids and described the process in the book "Of the Cold Produced by Evaporating Fluids and of Some Other Means of Producing Cold" published in 1756 [18]. In 1805 Oliver Evans suggested using evaporation of a volatile fluid in a closed cycle to produce ice [19]. Richard Trevithick designed an air-cycle refrigeration system in 1828, and Jacob Perkins built and patented a vapor-compression machine in the 1830s [19]. Perkins is considered the inventor of the vapor-compression cycle used today in refrigerators and heat pumps. The first large scale compressor in the United States ws built in 1866 [16] In the 20th century refrigeration technology revolutionized the food industry and peoples eating habits, especially with the introduction of domestic refrigerators. Vapor-compression machines for heating purposes where introduced in the 1930s, amongst others in Switzerland [20]. The first screw compressor was designed by the Swedish engineer Alf Lysholm in the 1930s. The idea was further developed by Svenske Rotor Maskiner SRM. In 1946 a Scottish company got the first licence for commercial use of screw compressors [13]. Further developments of (ground source) heat pumps took place mainly in North-America, but low fuel-prices kept the interest for developing heat pumps at a moderate level. In Europe, the interest in developing vapor-compression technology increased greatly after the first oil crisis in 1973 when oil became more expensive. As the next oil crisis hit in the early 1980s, heat pump technology was more developed, and heat pumps replaced many oil-fired boilers in heating systems. As the heat pumps were installed in existing systems, and often in addition to other heat sources, the control and regulation of the systems were complicated. In many European countries, the popularity of heat pump systems dropped, and research and development on heat pump technology nearly stopped. The interest increased once more in the following years, and in the 1990s guidelines and legislations were compiled [20]. Parallel to the development of heat pumps and refrigerators, a lot of research on developing better refrigerants was conducted.

Today heat pumps and refrigerators are well known technology, with numerous applications. Other cycles, such as the absorption-compression cycle, have been developed and are increasingly used. Focus of research is still to improve the efficiencies of the components and developing better refrigerants, to improve system performance and efficiency. Refrigerants must also meet demands in terms of safety, environmental considerations, and costs. Efficient cooling of the compressor, improved refrigerant properties, and proper capacity control regulation are examples of measures to improve efficiency and optimize operation of heat pump systems.

2.2   Compressor models

There are many different approaches to model a compressor. The appropriate method must be chosen based on the information available and the desired outputs. The compressor models can in general be divided into three groups, categorized by the amount of knowledge necessary to develop the model [21]. The black box models are completely empirical and based on experimental or statistical data. These models do not describe any physical processes in the compressor and concerns only the inlet and outlet state of the refrigerant. White box models are based on the fundamental equations only, e.g. conservation of mass, energy, and momentum. The compressor is divided into several control volumes and such models are usually very complicated. All models in between these two categories are called grey box models. These models combine both empirical and physical relations to describe the compressor. Another classification of compressor models is provided by Byrne et al. [22], where the models are divided into geometrical, semi-empirical and empirical models. The geometrical models are the most comprehensive, as they include the geometrical properties of the compressor. The semi-empirical are geometrical models that include some empirical relations. The more comprehensive and accurate models require much longer computational time. Depending on the information available and the aim of the study, the compressor model is always a compromise between accuracy and computational time and costs. Modern computers have improved both computational time and the costs, and these parameters may not be an issue in most cases.

2.3   Literature review

Numerous models have been developed for compressors, and many studies focus on the development and improvement of such models. Koury et al. [23] did a numerical study of a variable speed refrigeration system, but do not specify the type of compressor. The model calculates the refrigerant mass flow rate, compressor power, and volumetric efficiency. The isentropic efficiency must be provided as an input to the model. The numerical model is validated by measurements on a prototype refrigeration system. The study also focused on the other components of the refrigeration cycle, especially regulating the flow sectional throttling area to control the degree of superheating, refrigerant mass flow rate and hence the cooling capacity. Start-up conditions were investigated, and the refrigerant mass flow rate and evaporation temperature was registered. In addition, the study investigated a step-wise increase in compressor speed, and how this affects the performance of the system. Balduzzi et al. [24] present a two-dimensional CFD (computational fluid dynamics) modeling strategy for large bore reciprocating compressors. The model uses a simplified geometry of the compressor in order to represent the refrigerant flow in only two dimensions. This gives lower computational time and reduced costs. The results are compared to experimental measurements and the model is found to provide satisfactory accuracy. Roskosch et al. [25] developed a differential model for reciprocating compressors which includes both compressor geometry and fluid properties. The aim was to develop a model for fluid replacement studies, where the compressor performance with various refrigerants may be investigated. Both fluid friction and heat transfer rate between fluid and compressor wall and compressor wall and the ambient environment are considered. The outputs of the model are the isentropic and volumetric efficiency. The efficiencies are calculated from the inlet conditions and outlet pressure. The model was developed for constant speed drive. The model was validated by experimental measurements on a heat pump with various refrigerants and showed good accuracy with the measured results. Castaing-Lasvignottes et al. [26] presented a model for calculating the isentropic and volumetric efficiencies of a reciprocating compressor based on energy and mass

balance equations. The aim was to create a simple model requiring short computational time. The model does not consider leakages or the influence of lubrication. The model is validated by experimental measurements on two different heat pumps using different refrigerants. The validation shows that the model provides satisfactory accuracy.

Li [27] analyzed a simplified semi-empirical steady-state model for variable speed reciprocating compressors, scroll compressors and piston rotary compressors. The compression process was considered polytropic, and the polytropic exponent was determined by experimental data. The compressor efficiencies and power input at variable speed were calculated from polynomial functions of the normalized rotational speed of the compressor. The model was validated for both constant and variable speed compressors using experimental data from other studies. The modeling results have a good accuracy with experimental data. Mendoza-Miranda et al. [8] developed a compressor model based on dimensional analysis. The model calculates the isentropic and volumetric efficiencies and is validated with experimental data for a reciprocating compressor. The variables used to develop the dimensionless parameters were chosen based on experimental data analysis. The model was also tested for some other compressor types and the results were compared to experimental data from other studies. The comparison showed that the model may be used for other compressor types if calibrated for the different compressor characteristics. Nunes et al. [2] developed a dimensionless mathematical model for a vapor compression refrigeration system and its components. The model combines thermodynamics, heat and mass transfer and empirical relations, and was validated by experiments. The compression process was modelled as a polytropic process. Due to the dimensionless variables, the results are normalized and may apply to other systems and designs. Rasmussen et al. [21] presented an overview of compressor models and a method of categorizing the different models based on the level of knowledge necessary to develop them. The paper also presents different characterization variables for compressor performance.

Kinab et al. [28] developed a system model for a heat pump in TRNSYS and sub-models for all necessary components. The compressor model calculates the power consumption and mass flow rate using 10-coefficient polynomials given in AHRI Standard 540 [29]. The inputs of the model are the evaporating and condensing temperatures and coefficients for mass flow rate and compressor power. The calculated efficiencies were plotted as functions of the compression ratio, and by regression the curves were fitted to second-order polynomials of the compression ratio. For variable speed drive, the efficiencies are related to both compression ratio and frequency. The model was validated by experimental measurements on a test bench and provides acceptable accuracy. Byrne et al. [1] performed simulations in TRNSYS of heat pumps and its components. The compressor model is based on empirical relations and experimental data. The inputs to the compressor model are the suction and discharge pressures and polytropic exponent. The refrigerant properties at the different states are calculated using the Refprop database [30]. The model was validated by experimental data, and the compressor model was fitted to the experimental results. The uncertainty of the mass flow rate is calculated to be 7%. The component models were produced by the authors and transferred to TRNSYS to model the system as a whole.

Many studies have been conducted on compressor efficiency and performance under varying operational conditions. Aprea et al. [9] performed an experimental analysis on a hermetic scroll compressor with varying speed, and compared the power consumption to that of an on/off controlled or single speed scroll compressor. When utilizing varying speed control, significant energy savings

where obtained compared to the on/off control [9]. Several other studies [8, 12, 14, 23, 31] have compared variable speed compressors to single-speed systems by experiments or simulations and have all concluded that variable speed control gives energy savings. Aprea et al. [32] focused on finding the optimal frequency for the different cooling and heating loads for scroll and reciprocating compressors, minimizing costs and power demand. The optimal frequency is important if energy savings compared to single speed drive are to be obtained. Cuevas et al. [15] tested a variable speed scroll compressor, with focus on the inverter efficiency. The study concluded that losses induced by the inverter are negligible. Wang et al. [33] investigated two different methods for capacity control of a screw compressor, namely frequency regulation and sliding valve regulation. Their conclusion is that frequency control with an inverter (i.e. variable speed) is the best option for capacity control of a screw compressor.

Studies have been performed to compare different types of compressors for use in heat pump or refrigeration applications. Suess et al. [34] recommended compressor types with respect to different refrigerants and applications. Tassou and Qureshi [10] compared an open-type reciprocating compressor, a semi-hermetic reciprocating compressor and an open-type rotary vane compressor in terms of their variable-speed performance. The study concluded that of the three, the open-type reciprocating compressor is best suited for variable-speed drive. As the study was executed in 1998, there may have been progress in technology that would affect the conclusion if the comparison was repeated today. The same argument can be made for the study conducted by Brendeng in 1979 [35]. Brendeng compared the capital and operating costs of reciprocating compressors and screw compressors. The results show that screw compressors have higher running costs than reciprocating compressors.

Further developing and improving both compressor- and heat pump technology is of great interest. Many studies focus on improving efficiencies and optimizing operation. Zhang et al. [36] provides a review of advances in heat pump systems in cold regions and identifies the most promising measures to improve efficiency and performance. For a single-stage system the oil injected compressor has the largest potential to improve efficiency, as the oil both seals and cools the compressor. Ahamed et al. [37] summarizes research on exergy analysis of vapor compression systems. The review finds that most of the exergy destruction in a vapor-compression cycle occurs in the compressor. The exergy destruction is proportional to the temperature difference between suction and discharge of the compressor. Hence, cooling of the compressor reduces the exergy destruction and improves efficiency. Another modification of refrigeration cycles to improve efficiency is the ejector. Sarkar [38] gives a review of ejector enhanced refrigeration cycles. Incorporating an ejector in the refrigeration cycle reduces compressor work and hence increases COP.

Due to environmental considerations, popular refrigerants are being phased out as they have high global warming potential (GWP). Alternative refrigerants are the subject of many studies, as their properties may differ from the traditional refrigerants. Calm [19] provides a summary of refrigerant history and requirements for the next generation of refrigerants. Mendoza-Miranda et al. [8] compared the performance of some HFO (hydrofluoroolefins) synthetic refrigerants to R134a performance. The study concluded that the HFO refrigerants provide lower cooling capacity than R134a, but mixtures of HFO refrigerants with R134a gives good cooling capacity and a low GWP. Nawaz et al. [39] evaluated hydrocarbons as natural refrigerants in residential heat pumps. Their conclusion was that R290 (propane) may replace R134a in existing systems, as their properties and

system requirements are comparable. Choudhari and Sapali [40] concluded that R290 is a good substitution for HCFC refrigerant R22. Venzik et al. [41] focused on various mixtures of hydrocarbons as refrigerants. These mixtures are zeotropic, i.e. the saturation temperature is a function of the vapor quality and is not constant during evaporation and condensation. The temperature glide will theoretically increase COP of the system, but experiments show that this increase is smaller than expected due to pressure drop in the heat exchangers reducing the temperature glide. Byrne et al. [22] created a scroll compressor model for hydrocarbons as refrigerants. The compressor model is based on polytropic compression, and accounts for heat loss from the compressor to the ambient air. The model was created for refrigerant R407c and adapted to hydrocarbon refrigerant and other compressor sizes. The model is validated for R407c and propane, providing accuracies less than ± 10 %. Cecchinato et al. [42] compared carbon dioxide as refrigerant to R134a for water heating, and concluded that CO_2 is a viable alternative if the system is properly designed. CO_2 also has the potential to deliver very high temperature heat and may be advantageous in industrial applications. For comprehensive information on transcritical carbon dioxide refrigeration cycles the review by Ma et al. [43] is recommended. Ciconkov [44] summarized the advantages and disadvantages of new refrigerants, both natural (ammonium, CO_2 and hydrocarbons) and synthetic (HFO's).

2.4   Structure

The remaining chapters of the report are the main parts of this thesis, presenting the work and results of this project. Chapter 3 introduces the underlying basis of the compressor models and other components created. This chapter includes information regarding refrigeration cycles, heat pump components and refrigerants. TRNSYS software is introduced and described in Chapter 4. The user written components are introduced in detail in Chapter 5. Various parameter values and the choice of these values are covered in this chapter. The results from the various simulations conducted are presented in Chapter 6. The results and models are compared based on their performance. The results are discussed in Chapter 7. The components are compared to results from literature, and their limitations are evaluated. Chapter 7 also provides suggestions and recommendations for further development of the components. Conclusions are provided in Chapter 8, followed by a summary of recommendations for further development of the work.

3 Theory

The following chapter presents some necessary theoretical considerations of heat pump technology. The vapor compression cycle is presented first, along with the relevant processes and components. Compression processes are described next, followed by a brief description of other heat pump components. Important refrigerant properties are presented in the following section. Finally, a brief introduction to control and regulation strategies of heat pump systems is provided.

3.1 Vapor compression cycle

The vapor compression cycle is the most commonly used principle for refrigeration and heat pump applications [13]. Figure 3.1 shows the basic components and energy flows in the vapor compression cycle. The order of which the refrigerant pass through the components is the same for both heat pump and refrigeration purposes. If the system is used to provide heat $\dot{Q}_{heat}$ to a building or process, the machine is called a heat pump. If the aim is to remove heat $\dot{Q}_{cool}$ from a refrigerated space, the system is called a refrigerator. Power $\dot{W}_{in}$ must be supplied to the compressor.
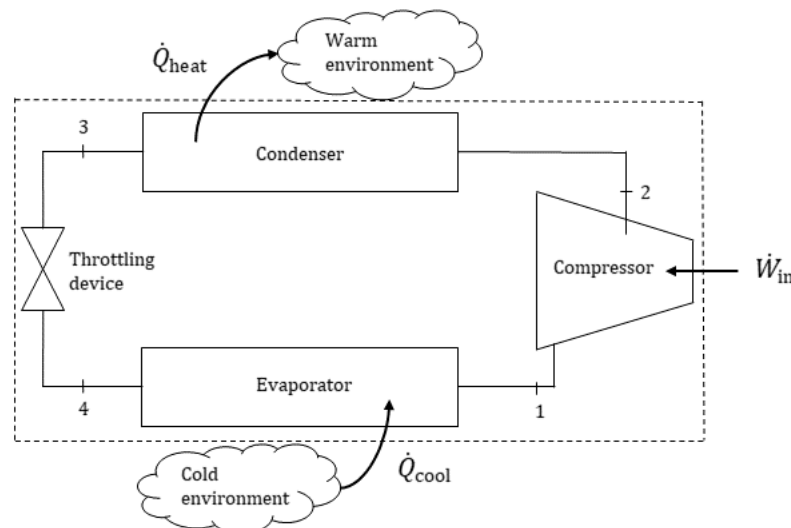
Figure 3.1: *Components and energy flows in the vapor compression cycle. Heat transfer occurs in the evaporator and condenser, the compressor requires work input. Dashed lines show the system boundaries.*

An energy balance may express the energy flow of a vapor compression cycle. The energy balance is derived from the first law of thermodynamics, stating that energy is preserved [16]. The system boundaries must be defined and are shown in Figure 3.1. Assuming a steady state process for a fixed frequency, the energy balance is shown in Eq.3-1 and 3-2. For a steady state process, the change in the system energy with time is zero.

$$\frac{dE_{system}}{dt} = \dot{E}_{in} - \dot{E}_{out} = 0 \qquad (3\text{-}1)$$

$$\dot{E}_{in} = \dot{E}_{out} \qquad (3\text{-}2)$$

where $\frac{dE_{system}}{dt}$ is the change in the energy of the system with time, and $\dot{E}_{in}$ and $\dot{E}_{out}$ is the sum of energy flows in and out of the system respectively. The energy flows that may occur across a system boundary is heat transfer, work transfer and energy transfer with a flowing fluid. The energy balance is shown in Eq.3-3.

$$\dot{Q}_{in}+\dot{W}_{in}+\dot{m}_{in}\left(h_{in}+\frac{u_{in}^2}{2}+gz_{in}\right) = \dot{Q}_{out}+\dot{W}_{out}+\dot{m}_{out}\left(h_{out}+\frac{u_{out}^2}{2}+gz_{out}\right) \tag{3-3}$$

where $\dot{Q}$ is heat transfer rate, $\dot{W}$ is work transfer rate, $\dot{m}$ is the mass flow rate, h is the specific enthalpy of the flowing fluid, u is the velocity of the fluid, z is the height of the fluid in relation to some reference height, and g is the gravitational acceleration. There is no fluid flow in or out of the system boundaries, and no work flow out of the system shown in Figure 3.1. Simplifying and rearranging the expression above yields Eq. 3-4.

$$\dot{W}_{in}+\dot{Q}_{cool}=\dot{Q}_{heat} \tag{3-4}$$

where $\dot{W}_{in}$ is the compressor power input, $\dot{Q}_{cool}$ is the heat transfer rate from the cold environment to the evaporator and $\dot{Q}_{heat}$ is the heat transfer rate from the condenser to the warm environment.

3.1.1 Reversed Carnot cycle

The basis of the vapor compression cycle is the reversed Carnot cycle, or the Carnot refrigeration cycle. The reversed Carnot cycle and the energy transfers involved is presented on a Temperature-Entropy diagram in Figure 3.2. The cycle consists of the following four processes;

1 – 2 Isentropic compression
2 – 3 Isothermal heat rejection
3 – 4 Isentropic expansion
4 – 1 Isothermal heat absorption

The isentropic processes are the vertical lines on the diagram, the isothermal processes are the horizontal lines on the diagram. As the isothermal processes of the Reversed Carnot cycle occur under the saturation dome, the pressure is also constant during the heat transfer processes.
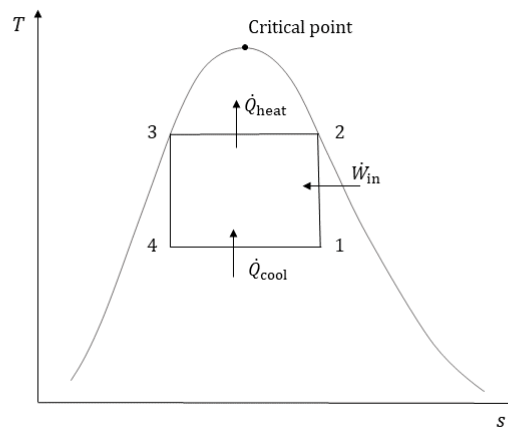
Figure 3.2: Reversed Carnot cycle on a Temperature-Entropy diagram. The saturation dome and critical point are also presented. The vertical and horizontal lines are isentropic and isothermal processes respectively.

The reversed Carnot cycle is not achievable in real cycles, mainly due to irreversibilities [16]. Reversible heat transfer is not possible, as it must occur without any temperature difference. There are losses in the compressor, pressure drops in the heat exchangers due to fluid friction and heat loss to the environment. Isothermal heat transfer is possible, and compression close to the isentropic process is achievable. The expansion process can be near the isentropic process if a turbine is used for expansion. This is normally not feasible in the vapor compression cycle as the refrigerant changes phase in the throttling process. Two-phase turbines are available but are expensive and not commonly used.

The coefficient of performance COP_{Carnot} of a Carnot refrigerator or heat pump may be expressed by the temperature levels it is operating between. The coefficient of performance is a measure of the systems effectiveness. For a real vapor compression cycle the coefficient of performance is calculated from the heat and work transfer rates of the system. This is further introduced in Chapter 3.1.4. The COP_{Carnot} is the maximum effeciency of a refrigeration cycle obtainable between two specified temperature levels. The coefficient of performance for the Carnot cycle is defined differently for heating and cooling purposes. The COP_{Carnot} may be calculated by Eq. 3-5 and 3-6.

$$COP_{\text{Carnot,heat}} = \frac{1}{1 - {T_L}/{T_H}} \tag{3-5}$$

$$COP_{\text{Carnot,cool}} = \frac{1}{{T_H}/{T_L} - 1} \tag{3-6}$$

where $COP_{\text{Carnot,heat}}$ and $COP_{\text{Carnot,cool}}$ is the Carnot coefficient of performance for heating and cooling applications respectively, T_H is the temperature of the warm environment, and T_L is the temperature of the cold environment, both in absolute temperatures. From the expressions for COP_{Carnot} it is evident that it increases as the temperature difference decreases. Real refrigeration cycles and systems may be compared using the Carnot cycle as a standard, by comparing the COP of the real cycle to the COP_{Carnot} calculated from the temperature levels the system operates between.

3.1.2 Ideal vapor compression cycle

Components used in the ideal vapor compression cycle are compressor, condenser, throttling device and evaporator. The cycle differs from the Reversed Carnot cycle by completely vaporizing the refrigerant before compression, and by expansion of the refrigerant through a throttling device. The ideal vapor compression cycle consists of the following four processes;

$1 - 2$ Isentropic compression

$2 - 3$ Isobaric heat rejection

$3 - 4$ Isenthalpic throttling

$4 - 1$ Isobaric heat absorption

Figure 3.3 presents the processes of the ideal vapor compression cycle in a pressure-specific enthalpy diagram. The saturation dome and the critical point is also shown in the diagram. The compression process follows a constant entropy line in the diagram.

Figure 3.3: *Ideal vapor compression process and the related energy flows on a Pressure-Enthalpy diagram. The saturation dome and critical point are also presented. The compression process follows a constant entropy line in the diagram.*

The refrigerant leaves the evaporator and enters the compressor as a saturated vapor. The compression process is isentropic, requiring a lower energy input than most real compression processes. Refer to Chapter 3.2 for further information about the compression process. At the discharge state of the compressor the refrigerant is a superheated vapor. The refrigerant leaves the condenser in a saturated liquid state, and the pressure and temperature are further reduced through the throttling device. The throttling process involves no heat addition or removal, and the enthalpy remains approximately constant throughout the process.

3.1.3 Real vapor compression cycle

The real vapor compression cycle differs from the ideal cycle, mainly due to irreversibilities and pressure losses in the system. The components are the same as for the ideal vapor compression cycle. The processes and energy flows related to a real vapor compression cycle are presented in a pressure-specific enthalpy diagram in Figure 3.4. The compression process in not isentropic, and there is pressure loss in both the condenser and evaporator. Both entropy generation and pressure loss may derive from fluid friction.

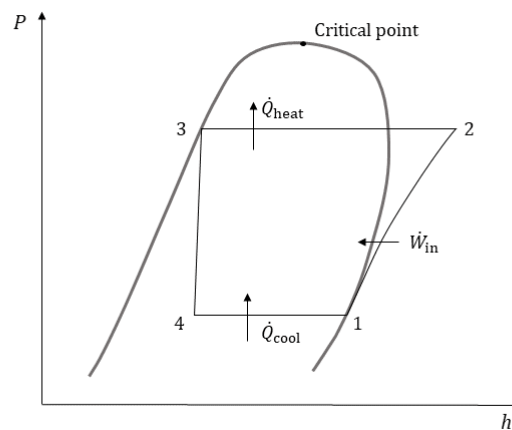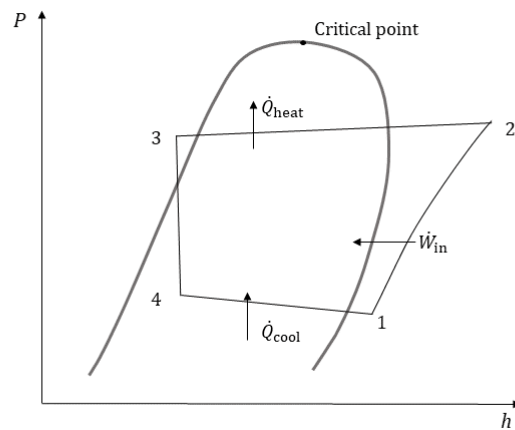**Figure 3.4:** *The real vapor compression cycle and the related energy flows in a Pressure-Enthalpy diagram. The saturation dome and critical point is also shown. There are pressure losses in both the condenser and evaporator. The compression process in not isentropic. The throttling process is considered isenthalpic.*

In practice, it may be hard to control the state of the refrigerant accurately in the different points of the cycle. To avoid two-phase flow in the compressor, the refrigerator or heat pump system is usually designed to superheat the refrigerant slightly before entering the compressor. This ensures that no liquid enters the compressor, which would cause wear and damage in the compressor. In addition, any liquid phase leaving the evaporator is a waste of potential cooling or heating capacity as it has not absorbed enough latent heat from the heat source to completely evaporate. The compression process is not isentropic as it involves both frictional losses and heat transfer processes. Cooling of the refrigerant during compression is advantageous, keeping the specific volume high and hence requiring less work input. Refrigerant cooling is not always feasible as it complicates the system structure and induces higher costs. To achieve maximum heat transfer in the condenser it is important that all the vapor is condensed. The system is usually designed to subcool the liquid slightly, to ensure the refrigerant is in pure liquid phase when entering the throttling device. The subcooling is also necessary to avoid vaporization in the pipe if heat transfer from the environment occurs. The lower temperature is an advantage when entering the evaporator, as the refrigerant may absorb more heat as it enters with lower specific enthalpy. Due to the constant enthalpy process in the throttling device, the refrigerant will enter the evaporator as a mixture of liquid and vapor. The vaporized parts of refrigerant will not contribute to the cooling or heating effect, as it cannot receive any latent heat.

In real systems, there are pressure losses in both the heat exchangers and pipes due to fluid friction. Especially in the suction line from the evaporator to the compressor the pressure drop is a problem [45]. At lower suction pressures the compressor will need higher work input to deliver the required discharge pressure, decreasing the compressor efficiency. Compressor efficiencies are further introduced in Chapter 3.2.

3.1.4   Coefficient of performance

The coefficient of performance COP is a measure of the efficiency of the system. It is defined as the ratio of the useful energy added or removed to the energy consumed in the compressor. COP is an instantaneous value that changes constantly with the operating conditions. The COP is defined

differently for heating and cooling applications, depending on the purpose of the heat transfer. COP for heating and cooling mode may be calculated by Eq. 3-7 and 3-8 respectively.

$$COP_{\text{heat}} = \frac{\text{heat added}}{\text{energy consumed}} = \frac{\dot{Q}_{\text{heat}}}{\dot{W}_{\text{in}}} \tag{3-7}$$

$$COP_{\text{cool}} = \frac{\text{heat removed}}{\text{energy consumed}} = \frac{\dot{Q}_{\text{cool}}}{\dot{W}_{\text{in}}} \tag{3-8}$$

From the expressions for COP it is evident that the compressor performance greatly influences the systems effectiveness and performance. However, it is important to note that the COP is not a measure of the compressor efficiency, but the system performance. Some definitions of COP also take other energy consuming components in the cycle into account. Such components could be fans or oil pumps, depending on the system configuration. As the COP changes with the operating conditions is it common to define a seasonal coefficient of performance $SCOP$. The $SCOP$ is an average value, calculated over e.g. an entire season or year. It may be calculated for any specified time period by Eq. 3-9. The seasonal coefficient of performance for a heat pump system delivering heat at a medium level ($45-55$ °C) is typically $3-4$ [46]. These values are for a heat source temperature of 0 °C. The seasonal coefficient of performance will be larger with higher heat source temperature.

$$SCOP = \frac{\text{total heat removed/added}}{\text{total energy consumed}} \tag{3-9}$$

How well a vapor compression cycle exploits the temperature difference between the heat source and sink may be expressed by comparing the COP of the system to the coefficient of performance for reversed Carnot cycles COP_{Carnot} by calculating the Carnot efficiency η_{Carnot}. The coefficient of performance for reversed Carnot cycles is described in Chapter 3.1.1. The Carnot efficiency, or second-law efficiency, may be calculated by Eq. 3-10.

$$\eta_{\text{Carnot}} = \frac{COP}{COP_{\text{Carnot}}} \tag{3-10}$$

3.1.5 Temperatures

The system is usually designed based on the desired temperature for cooling or heating, and the available temperature in the heat source for heat pumps. These temperatures determine the condensing and evaporating temperatures. The condensing and evaporating temperatures determine the condensing and evaporating pressure respectively, as the pressures in the condenser and evaporator corresponds to the saturation temperature of the refrigerant. The condensing and evaporating pressures determine the required compression ratio for the compressor. In addition, the proper refrigerant must be chosen. The refrigerant must be able to operate at the desired temperatures, and the refrigerant properties give further requirements to compressor size, design, and performance. Refrigerant properties are further introduced in Chapter 3.6. Changes in the condensing or evaporating temperature will affect the heating and cooling capacity of the system, and the operating conditions of the compressor. The saturation temperatures are usually not constant due to pressure loss from fluid friction.

The temperatures of especially the heat source may also vary substantially over a year. There is a large variety of heat sources for heat pumps available and in use. The most common heat source for

domestic heat pumps is the ambient air. Heat pumps for industrial or office applications often operate with other sources, for example waste heat from processes, ground source heat or seawater. A good heat source has a stable and high temperature, and abundant thermal energy available [46]. Ambient air is the heat source with the largest variation in temperature. Another drawback of ambient air as heat source is the low heating capacity due to low temperatures in periods where the heat demand is the highest. Some heat sources may have an approximately constant temperature, and a constant evaporating and condensation temperature is used in the models simulated in this project. An example of such a heat source is a district heating system, for example the district heating system of the University in Bergen (UiB). The district heating system of UiB circulates water at 20 °C, heated by heat pumps using sea water as heat source. The various buildings around the campus has heat pumps using the district heating system as heat source, delivering the required heat to the buildings [47].

3.2 Compression processes

The purpose of compression is to increase the pressure of a gas. As a consequence of the increased pressure, the specific volume is reduced and the temperature of the gas increases. Compression requires work input, and it is desirable to keep the required work input as low as possible.

3.2.1 Energy balance

The expression for the required work input to the compressor is obtained through an energy balance. Assuming a steady state compression process at a fixed frequency the energy balance is as presented in Eq. 3-1 in Chapter 3.1. The energy flow in and out of the compressor is the sum of the kinetic and potential energy, internal energy and the flow energy of the refrigerant at the inlet and outlet state respectively, and the heat transfer and work flows. The complete energy balance for a steady state system is presented in Eq.3-11. There is no change in energy flows with time during a steady state process.

$$\dot{Q}_{in} + \dot{W}_{in} + \dot{m}_{in}\left(h_{in} + \frac{u_{in}^2}{2} + gz_{in}\right) = \dot{Q}_{out} + \dot{W}_{out} + \dot{m}_{out}\left(h_{out} + \frac{u_{out}^2}{2} + gz_{out}\right) \tag{3-11}$$

where $\dot{Q}$ is heat transfer rate, $\dot{W}$ is work transfer rate, $\dot{m}$ is the refrigerant mass flow, h is the specific enthalpy of the refrigerant, u is the velocity of the refrigerant, z is the height in relation to some reference height and g is the gravitational acceleration. There is no work output from a compression process, and it is assumed that there is no heat transfer from the environment to the compressor due to the increasing temperature of the refrigerant during compression. Therefore, $\dot{Q}_{in}$ and $\dot{W}_{out}$ are zero. The changes in kinetic energy (the velocity of the refrigerant) and potential energy (the height difference of inlet and outlet) are considered negligible compared to the change in specific enthalpy and the work and heat transfer rates. The mass flow rate is considered constant through the compressor at a fixed frequency. This is consistent with the recommendations of Ding [48] regarding dynamic simulations of compressors. The resulting energy balance is stated in Eq.3-12.

$$\dot{W}_{in} = \dot{m}(h_{out} - h_{in}) + \dot{Q}_{out} \tag{3-12}$$

Figure 3.5 presents a compressor and the energy transfers involved in the compression process. Due to the increased temperature of the refrigerant during compression, the heat transfer $\dot{Q}$ is assumed to be from the compressor to the ambient environment.
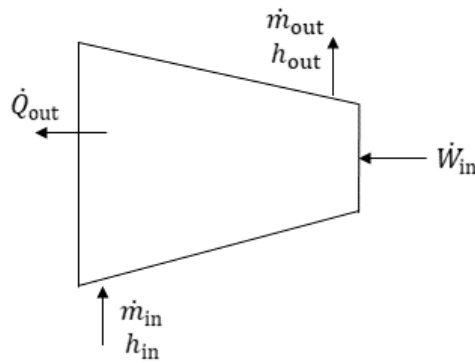
Figure 3.5: *Compressor and the related energy transfers during a steady state compression process, where $\dot{Q}_{out}$ is any heat loss from the compressor, $\dot{W}_{in}$ is the required power input, h_{in} and h_{out}, and $\dot{m}_{in}$ and $\dot{m}_{out}$ are the specific enthalpy and mass flow rate of the refrigerant at the inlet and outlet of the compressor respectively.*

3.2.2 Compression processes

The expression to calculate the work input to a compressor presented in Eq.3-12 is not very complicated. However, the discharge enthalpy h_{out} is hard to estimate without measurements obtained from an actual compressor. The discharge enthalpy depends on which compression process that is occurring. There are three basic compression processes, the isothermal, isentropic and polytropic process. Figure 3.6 shows the internally reversible isothermal, polytropic and isentropic compression processes on a pressure-specific volume diagram. Internally reversible processes are free from friction and other irreversibilities inside the system boundaries. The internally reversible process consists of a series of equilibrium states, and the reverse process may return the system to its initial state. The required work of the process equals the area between the process curve, the pressure axis, and the two vertical, dashed lines of the suction and discharge pressure.
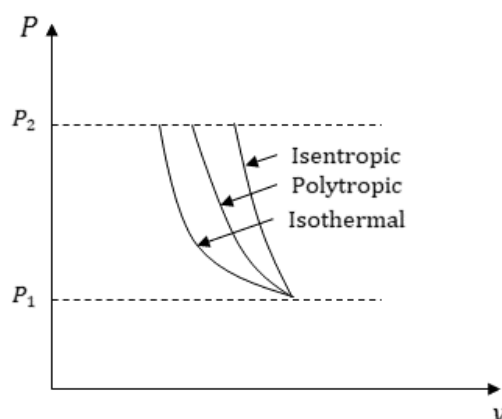
Figure 3.6: *Internally reversible isothermal, polytropic and isentropic compression process presented in a Pressure-Specific volume-diagram. The internally reversible processes are free from friction and other irreversibilities. The reverse process will return the system to its initial state. The required work of the process equals the area between the process curve, the pressure axis and the two vertical, dashed lines of the suction and discharge pressure.*

31

An isothermal process is a process that does not involve any change in temperature. To achieve an isothermal compression process the compression must proceed very slowly, and the compressor must be cooled to remove any excess heat. All reversible processes must proceed very slowly to obtain equilibrium states during the process. An isentropic process is a process where the entropy remains constant between the inlet and outlet of the compressor. There is no heat transfer between the compressor and its surroundings. This process is also called adiabatic compression, meaning there is no heat transfer occurring. The isothermal compression process requires the lowest work input of the three processes. The isentropic process requires the highest work input. The internally reversible polytropic process lies between the isothermal and isentropic process in terms of required work input. The internally reversible polytropic process involves heat removal or cooling of the refrigerant during compression.

3.2.3   Work input

The processes are presented on a pressure – specific volume diagram. The required work input may be calculated from the pressure and volume in the start and end state of the process. The required work input corresponds to the area between the process curve, the pressure axis, and the vertical lines representing the suction and discharge pressure. This may be derived from an energy balance. As previously shown the energy balance of a compressor may be expressed as in Eq. 3-13.

$$\dot{Q}_{\text{in}} + \dot{W}_{\text{in}} + \dot{m}_{\text{in}} \left(h_{\text{in}} + \frac{u_{\text{in}}^2}{2} + g z_{\text{in}} \right) = \dot{Q}_{\text{out}} + \dot{W}_{\text{out}} + \dot{m}_{\text{out}} \left(h_{\text{out}} + \frac{u_{\text{out}}^2}{2} + g z_{\text{out}} \right) \tag{3-13}$$

where $\dot{Q}$ is heat transfer rate, $\dot{W}$ is work transfer rate, $\dot{m}$ is the refrigerant mass flow, h is the specific enthalpy of the refrigerant, u is the velocity of the refrigerant, z is the height in relation to some reference height, and g is the gravitational acceleration. For a compression process there is only work input, and it is assumed that the heat is transferred from the compressor to its surroundings. The energy balance for a reversible process may be expressed in differential form as in Eq. 3-14.

$$-\delta q_{\text{rev}} + \delta w_{\text{rev}} = dh + dke + dpe \tag{3-14}$$

where q and w are the specific heat and work respectively, h is the specific enthalpy, ke is the specific kinetic energy, and pe is the specific potential energy. As previously mentioned the change in kinetic and potential energy may be neglected, as they are very small compared to the work input and heat transfer. From the defining equation for entropy and the second T-ds relation given in Eq. 3-15 and 3-16 the energy balance may be simplified.

$$\delta q_{\text{rev}} = T \, ds \tag{3-15}$$

$$T \, ds = dh - v \, dP \tag{3-16}$$

where δq_{rev} is the reversible heat transfer rate, T is the temperature and, ds is the change in specific entropy of the refrigerant. Substituting for δq in Eq.3-14, eliminating dh and integrating yields Eq. 3-17 which is an expression for the reversible work input during a compression process.

$$w_{\text{rev}} = \int_1^2 v \, dP \tag{3-17}$$

where w_{rev} is the specific work input, v is the specific volume, and dP is the change in pressure. In order to perform the integration, the specific volume v as a function of the pressure P must be known. The relationship between v and P varies for the different compression processes. Eq. 3-18 – 3-20 presents the relation between specific volume and pressure for the isothermal, isentropic, and polytropic compression process respectively.

$$Pv = C \tag{3-18}$$

$$Pv^{\gamma} = C \tag{3-19}$$

$$Pv^{n} = C \tag{3-20}$$

where C is a constant, γ is the isentropic exponent, and n is the polytropic exponent. The isentropic exponent γ equals the specific heat capacity ratio of the refrigerant, and the isothermal exponent equals one. For internally reversible processes the value of the polytropic exponent lies between the isentropic and isothermal exponent. Inserting the relation between the pressure and specific volume in Eq. 3-17 and integrating yields the following expressions for the work input to the compressor. Introducing the refrigerant mass flow rate $\dot{m}$ gives the power input to the compressor. The expressions for the isothermal, isentropic and polytropic compression power input is presented in Eq. 3-21 – 3-23.

$$\dot{W}_{\mathrm{in}} = \dot{m} P_{\mathrm{suc}} v_{\mathrm{suc}} \ln\left(\frac{P_{\mathrm{dis}}}{P_{\mathrm{suc}}}\right) \tag{3-21}$$

$$\dot{W}_{\mathrm{in}} = \dot{m} P_{\mathrm{suc}} v_{\mathrm{suc}} \frac{\gamma}{\gamma - 1}\left(\left(\frac{P_{\mathrm{dis}}}{P_{\mathrm{suc}}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right) \tag{3-22}$$

$$\dot{W}_{\mathrm{in}} = \dot{m} P_{\mathrm{suc}} v_{\mathrm{suc}} \frac{n}{n - 1}\left(\left(\frac{P_{\mathrm{dis}}}{P_{\mathrm{suc}}}\right)^{\frac{n-1}{n}} - 1\right) \tag{3-23}$$

where subscripts suc and dis correspond to suction and discharge state respectively. The integral presented in Eq. 3-17 may also be used to calculate the compressor power input of any compression process. The challenge is to estimate the proper value for the exponent. Most real compression processes are polytropic, involving some cooling or heat loss to the surroundings. The internally reversible processes are ideal processes. The polytropic exponent for a real compression process may be higher or lower than the isentropic exponent, depending on the direction of the heat transfer and the irreversibilities of the process. Real processes always involve some irreversibilites like friction or heat loss, and the value of the polytropic exponent is chosen to achieve the desired process path and result. In scientific studies the value of the polytropic exponent is usually estimated by experimental data, as in the study by Byrne et al. [22] where the polytropic exponent is experimentally determined to be about 18 % higher than the isentropic exponent.

3.2.4   Efficiency

The internally reversible processes are used as reference processes for real compression processes. An irreversible compression process will always require a higher work input than a reversible process, given the compressor operates between the same end states. Which reference process that is chosen depends on the application.

The most commonly used reference process for evaluation of a compressors energetic performance is the isentropic process [21]. The isentropic efficiency is a measure of the irreversibilities in the process. The isentropic efficiency may be calculated by Eq. 3-24.

$$\eta_{\text{is}} = \frac{\text{Isentropic compressor power}}{\text{Actual compressor power}} = \frac{\dot{W}_{\text{is}}}{\dot{W}_{\text{in}}} \qquad (3\text{-}24)$$

where $\dot{W}_{\text{is}}$ is the isentropic compressor power input and $\dot{W}_{\text{in}}$ is the actual compressor power input. The efficiency may also be calculated by the work or specific work input. As mentioned in Ch. 3.2.2 the internally reversible isothermal process requires the lowest work input. It is not common to use the isothermal process as reference for compression in heat pump applications. This is because the temperature of the refrigerant must increase during compression in order to achieve heat transfer in the condenser. If a compressor is heavily cooled the isothermal compression process might be sensible to use as reference process to evaluate the energetic performance. The polytropic process is seldom used as a reference process, due to the uncertain value of the polytropic exponent. Other efficiencies are also used, for example the exergy efficiency. The exergy efficiency is a measure of the entropy generation in the process.

3.3   Compressor characteristics

Compressors are devices used to increase the pressure and temperature of a gas. Compressors are divided into two main categories; dynamic and positive displacement compressors. Dynamic compressors compress gas by increasing its velocity to obtain a higher kinetic energy, followed by a transformation of the kinetic energy into higher static pressure. Dynamic compressors are used in for example gas turbines and turbo chargers. The positive displacement compressors compress gas by reducing its physical volume. Figure 3.7 gives an overview of compressor categories and some different compressor types.
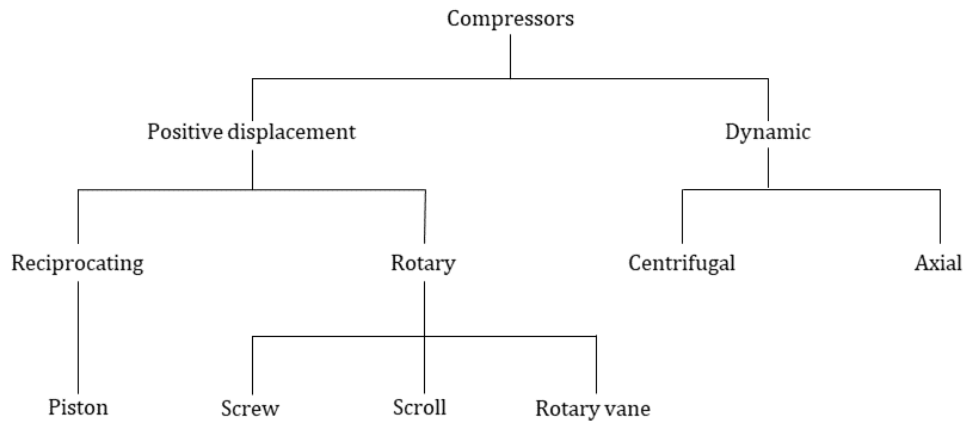
Figure 3.7: Compressor categories and types. For heat pump and refrigeration applications, the positive displacement compressors are used. The dynamic compressors are used in gas turbine engines and gas turbine power plants.

The compressors used in heat pumps are positive displacement compressors. The positive displacement compressor types operate in different ways, have various geometrical properties and some different regulation possibilities. In the previous chapter, some energetic efficiencies were described. In addition to the isentropic efficiency, compressors are described by the volumetric efficiency. The volumetric efficiency η_{vol} describes how well the physical volume of the compressor is exploited during compression. It is the ratio of the volume of suction gas entering the compressor to the swept volume of the compressor. Sometimes it is defined as the ratio of the delivered volume, i.e. discharge volume, from the compressor to the swept volume [13]. The volumetric efficiency is more critical to some compressor types than others. The volumetric efficiency varies inversely with the pressure ratio and may also change over time due to wear mechanisms. The flow characteristics, heat transfer and leakages in the compressor also affects the volumetric efficiency [21, 45]. The volumetric efficiency may be calculated by Eq. 3-25.

$$\eta_{vol} = \frac{\dot{V}_{suc}}{\dot{V}_{swept}} \tag{3-25}$$

where $\dot{V}_{suction}$ is the volumetric flow rate at the compressor inlet and $\dot{V}_{swept}$ is the swept or displaced volume rate of the compressor. Some compressor types may obtain volumetric efficiencies close to unity. Reciprocating compressors usually have lower volumetric efficiency due to the necessary clearance volume. This is elaborated in Chapter 3.3.2.

3.3.1 Built-in pressure and volume ratio

The built-in pressure and volume ratio is an important parameter for compressor dimensioning and performance. It describes the compressors ability to compress gas before discharge. The internal pressure ratio of the compressor should match the required pressure ratio of the system [13]. If the built-in pressure ratio is too high, the gas is compressed to a pressure level above the discharge pressure. This requires extra energy and gives a lower total efficiency. If the built-in pressure ratio is

too low, the compressor is not able to compress the gas to the required discharge pressure. When the compressed gas is opened to the discharge line, there is backflow from the discharge line until discharge pressure is reached in the compressor. This also requires additional energy, reducing the total efficiency. The highest efficiency is obtained when the internal pressure ratio is equal to the operating pressure ratio [45].

The built-in volume ratio is the ratio between the suction volume V_1 and the smaller discharge volume V_2. The built-in volume ratio is constant for any given compressor design, depending only on the geometry of the compressor [45]. The internal pressure ratio is related to the built-in volume ratio by Eq. 3-26.

$$\frac{P_2}{P_1} = \left(\frac{V_1}{V_2}\right)^{\gamma}$$

(3-26)

where P_1 and P_2 are the suction and discharge pressure respectively and γ is the specific heat ratio of the refrigerant. The internal pressure ratio is not constant but depends on the refrigerant in use. The specific heat ratio varies for different gases and with temperature and pressure.

3.3.2 Positive displacement compressors

The following chapter provides an introduction to various common positive displacement compressor types and their characteristics. The various compressor types may have different configurations. Compressors where both the compressor and the electric motor are contained within a gas tight casing are called hermetic or compact compressors. The casing is completely sealed, and there is no access to internal parts for service and maintenance on site. An open type compressor is connected to the electric motor by a shaft extending through the casing, with the motor placed outside the casing. This enables maintenance of the motor on site. Semi-hermetic compressors are contained in a housing with the motor. The casing is gas tight but not welded, and the casing may be opened for service and maintenance.

Reciprocating compressor

The reciprocating compressor consists of a piston in a cylinder. An electric motor drives the crankshaft that induces the reciprocating motion of the piston. On the cylinder head an inlet and an outlet valve are placed. The refrigerant is drawn into the cylinder from the suction line when the piston is at the bottom dead center. As the piston moves towards the top dead center, the refrigerant is compressed as the volume decreases. When the desired pressure is reached the outlet valve opens, and the refrigerant is transported into the high-pressure discharge line. The reciprocating compressor has a clearance volume at the top of the cylinder. The compressed gas in this clearance volume will not be delivered to the discharge line, and as the piston retracts from top dead center the refrigerant is expanded back to the lower suction pressure. The clearance volume is crucial to avoid that the piston collides with the cylinder head and valves. A very small clearance volume is possible, but will increase noise and vibrational issues [13].

The reciprocating compressors are manufactured with a various number of cylinders and are versatile in terms of applications. The number of cylinders and their size determine the capacity of the compressor, giving numerous possible capacities and applications.

Scroll compressor

The scroll compressor consists of two spirals. One is stationary and the other moves in an orbiting motion. The suction gas enters at the perimeter of the compressor and is moved towards the center by the orbiting movement. As the gas is transferred inwards, the volume in the space between the spirals decrease and the gas pressure increases. The discharge port is at the center of the spirals. The gas is isolated from the suction port during compression, and a new volume of gas may enter as the first volume moves towards the center. During operation all volumes in the scroll compressor are filled, and the compression is nearly continuous. There are no suction or discharge valves as in a reciprocating compressor. The absence of valves reduces vibration and noise. The scroll compressor may be regulated by frequency regulation, controlling the speed of the orbiting motion.

The scroll compressors are available for a large range of capacities, and are replacing reciprocating compressors in many applications [45]. The volumetric efficiency of scroll compressors is generally high as the suction and discharge gas are separated. There is no clearance volume in a scroll compressor. Internal leakage in the compressor decreases the efficiencies and may be minimized by proper compressor design and manufacturing.

Twin screw compressor

The twin screw compressor is the screw compressor with the widest application range and is usually just referred to as a screw compressor. The screw compressor consists of two meshing rotors. The male rotor has lobes that move into the female rotors flutes as the rotors turn. One rotor rotates counterclockwise, and the other clockwise. Usually the male rotor is driven by the motor and turns the female rotor by the means of gears. The rotors are not in contact with each other. The suction gas enters the space between the rotors. As the rotors revolve, the gas is moved axially along the rotors. The volume between the rotors decreases and the gas is compressed. At the end of the rotors, the compressed gas is discharged.

As the rotors turn, the suction gas is isolated from the suction line, and there is no need for either inlet or outlet valves as in reciprocating compressors. The compression of gas in a screw compressor is nearly continuous. As a volume of gas is moved away from the suction port in the axial direction, a new volume of gas enters through the suction port. The capacity of screw compressors can be controlled by varying the position of the discharge port by a sliding valve but may also be controlled by frequency regulation. Regulation by sliding valve may results in poor efficiency. Brendeng [35] states that the required work input to a screw compressor regulated by a sliding valve is about the same for part load as full load operation. The sliding valve regulation of screw compressors is beyond the scope of this project. The screw compressor does not have a clearance volume, and all the compressed gas is discharged [49]. This is an advantage compared to reciprocating compressors, giving the screw compressor a high volumetric efficiency. The screw compressors require quite complicated lubrication systems. The lubrication oil is injected into the compressor, providing both sealing and cooling of the compressor. The lubrication system requires additional components increasing both costs and energy consumption. The cooling provided by the lubrication oil enables the screw compressor to operate at high compression ratios.

3.4　Evaporator and condenser

Heat exchangers are devices that enable the transfer of heat between two separated fluids. There are various types and configurations of heat exchangers that have different applications. In a vapor compression cycle heat is transferred both from the heat source and to the heat sink. The refrigerant is enclosed in the vapor compression cycle, and the heat must be transferred by heat exchangers. As the refrigerant changes phase during the heat transfer processes, the analysis of the heat exchanger must be adapted from the analysis of heat transfer involving temperature change.

An evaporator is a device for heat transfer where the heat-receiving medium enters the device as a liquid or saturated liquid-vapor mixture. The fluid is evaporated during the heat exchange and leaves the evaporator as a saturated or superheated vapor. A condenser is a device for heat transfer where the refrigerant enters in saturated or superheated vapor state and is condensed during the heat transfer process. At the outlet of the condenser the refrigerant is in a saturated or subcooled liquid state. The heat transfer processes and the effect of superheated vapor and subcooled liquid are shown in Figure 3.4.

Heat transfer rates in heat exchangers are often calculated based on the temperature difference between inlet and outlet of both fluids involved in the heat transfer. During a phase change process, the temperature of the refrigerant is constant and using a temperature difference would provide an incorrect result. The heat transfer rate of the refrigerant may be calculated based on an energy balance as for the compressor in Chapter 3.2.1. There is no work involved in the heat exchange process. The change in kinetic and potential energy is neglible. The refrigerant mass flow rate is considered constant through the process at a fixed compressor frequency. The heat transfer rate to or from the refrigerant may be calculated by Eq. 3-27.

$$\dot{Q}_{\text{refr.}} = \dot{m}(h_{out} - h_{in}) \tag{3-27}$$

where $\dot{Q}_{\text{refr.}}$ is the heat transfer rate to or from the refrigerant, $\dot{m}$ is the refrigerant mass flow rate, and h_{out} and h_{in} is the specific enthalpy of the refrigerant at the outlet and inlet state respectively. The heat transfer rate is in this equation defined as positive if it is received by the refrigerant. When applying this equation to a condenser, the heat transfer rate calculated will be negative. In the calculations in the user created evaporator and condenser component in this project, the absolute value of the change in specific enthalpy is used to avoid a negative heat transfer rate. The heat exchanger component and the associated code is presented in Chapter 5.1.4.

Figure 3.8 shows a condenser or evaporator and the energy flows regarding the refrigerant present in the process. The heat transfer rate to or from the refrigerant $\dot{Q}_{\text{refr.}}$ is either in or out of the device, depending on if the component is an evaporator or condenser.
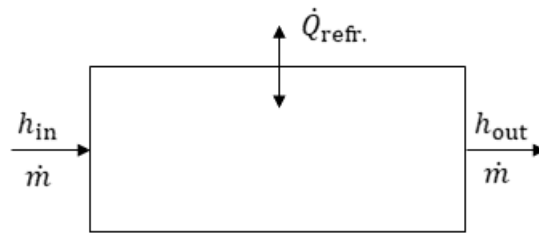
Figure 3.8: *Refrigerant energy flows in a condenser or evaporator. The heat transfer rate $\dot{Q}_{refr.}$ has direction in to the component if the device is an evaporator and out of the component if the device is a condenser.*

With time all heat exchangers experience fouling, decreasing the efficiency of the heat transfer. The efficiency is usually defined as the ratio of the actual heat transfer rate to the heat transfer rate in an ideal heat exchanger under the same operating conditions [50].

3.5 Throttling device

A throttling device is a device that decreases the pressure in a flowing fluid without producing any work. The decrease in pressure is usually followed by a decreasing temperature. In the vapor compression cycle the decrease in pressure must be accompagnied by a decrease in temperature. Common devices are the capillary tube, adjustable valve, and thermal expansion valve. An energy balance for a steady state process as presented in Eq. 3-3 in Chapter 3.1 may be used to describe the throttling process. The throttling devices are usually assumed to be adiabatic, and there is no heat transfer to or from the fluid flow. This assumption is valid because the throttling devices are quite small, and the surface area is not large enough for any substantial heat transfer to occur [16]. In addition, is there not sufficient time for heat transfer to occur. There is no work done by the throttling device, and it requires no work input. The change in potential energy between the inlet and outlet may be neglected. Decreasing the pressure of a fluid increases the fluids velocity. The change in kinetic energy is still considered insignificant in most cases. The mass flow rate is constant through the steady flow throttling device at a given compressor frequency. The energy balance reduces to the expression given in Eq. 3-28.

$$\dot{m}h_{\text{in}} \cong \dot{m}h_{\text{out}} \tag{3-28}$$

The throttling process is approximately isenthalpic. The temperature decrease during a throttling process is controlled by the Joule-Thomson coefficient μ. The Joule-Thomson coefficient is presented in Eq. 3-29.

$$\mu = \left(\frac{\delta T}{\delta P}\right)_h \tag{3-29}$$

The Joule-Thomson coefficient is defined as the change in temperature with change of pressure at constant specific enthalpy. If the Joule-Thomson coefficient is larger than zero, the temperature decreases during the throttling process.

3.6   Refrigerants

The first refrigerants were natural refrigerants, like hydrocarbons, water, ammonium, and carbon dioxide [19]. The natural refrigerants have various issues regarding flammability, corrosiveness, and toxicity. In the early 1900's, the technology and materials were not compatible with the refrigerants. The natural refrigerants were replaced by synthetic refrigerants as the CFC (chlorofluorocarbons) were introduced in the 1930's. The CFC refrigerants became popular due to their stability, safety and compatibility with materials and lubrication oils. The CFC refrigerants highly affects the ozone layer having a high ozone depletion potential (ODP). CFC refrigerants were phased out during the 1990's as agreed in the Montreal Protocol of 1987 [51]. HCFC (hydrochlorofluorocarbons) refrigerants were transitional replacements for the CFC gases, and HFC (hydrofluorocarbons) refrigerants have been the preferred choice since the phase out of CFC refrigerants. HCFC refrigerants are still in use, and HFC refrigerants are the most common refrigerants today. The HCF refrigerants do not affect the ozone layer, as they do not contain chlorine. However, they have a high global warming potential (GWP) and contribute greatly to the emission of greenhouse gases. The Kigali Amendment (2007) to the Montreal Protocol states that HFC refrigerants must be replaced by other alternatives [5]. EU Regulation no. 517, the F-gas directive, aim to decrease the emissions of fluorinated greenhouse gases in EU by almost 70% by 2030 [6]. The regulation will regulate the market for refrigerants and introduce prohibitions on the use of fluorinated refrigerants.

The choice of refrigerant is usually a compromise between various environmental concerns and safety issues. The newer low-GWP and zero-ODP refrigerants all have higher flammability and/or toxicity. In addition, the newer refrigerants generally provide lower cooling capacity. This increases the power input to the system, and the energy-related emissions increases [19]. The Montreal Protocol set a zero-tolerance for ODP of refrigerants in 1987. This zero-tolerance limit led to the phase out of refrigerants with low GWP, good performance, and ODP just slightly above zero [44].

3.6.1   Refrigerant properties

Properties of the refrigerant highly affects the system performance and the requirements of the components. An ideal refrigerant must have the proper thermodynamic properties, be non-toxic and non-flammable and not harm the environment. The refrigerant must be compatible with the materials of the system components and lubrication oils. Components of the refrigerant should be abundant. Their production should not demand too much energy and the refrigerant should have low cost. None of the refrigerants available today are ideal. They all have different properties, applications, and various disadvantages.

Thermodynamic properties

The thermal energy in a vapor compression cycle is transferred by the evaporation and condensation of the refrigerant. The system performance and configuration greatly depend on the refrigerants thermodynamic properties.

Specific heat and heat of evaporation

The specific heat c_p indicates the amount of energy needed to heat or cool one unit mass of a substance by one degree [45]. The specific heat of a substance varies with the temperature of the substance, especially when going through a phase change. The heat transferred in a process including a temperature change may be calculated by Eq. 3-30.

$$\dot{Q}_H = \dot{m}\, c_p\, \Delta T \tag{3-30}$$

where $\dot{Q}_H$ is the heat transfer rate, $\dot{m}$ is the refrigerant mass flow rate, c_p is the specific heat of the refrigerant, and ΔT is the temperature difference between the start and end of the process. When a liquid is vaporized, all the heat transferred to the liquid is used to increase the internal energy of the molecules to allow them to separate from the other molecules. This does not result in a temperature change and is called latent heat of evaporation. The specific latent heat of evaporation h_{fg} is a substance property and varies greatly with pressure [45]. The phase change from liquid to vapor takes place at the saturation temperature for the given pressure, which is constant through the process. The saturation temperature is manipulated by increasing and decreasing the pressure through the vapor compression cycle, to obtain the desired cooling or heating performance. Both the specific latent heat and saturation temperature for a given substance can be obtained from tabulated data. If additional heat is transferred to the fluid when in the saturated vapor state, the temperature will increase, and the vapor is called superheated. The various states of a substance or refrigerant may be presented in a pressure-enthalpy diagram. Figure 3.9 presents a pressure – specific enthalpy diagram with the saturation dome and critical point. The various states of the substance or refrigerant is presented in the diagram.
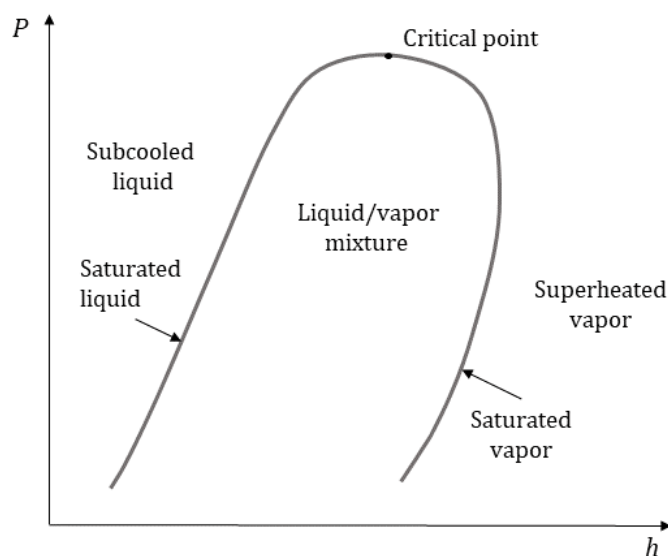
Figure 3.9: Pressure – Specific enthalpy diagram. The saturation dome, critical point and the various states of the substance or refrigerant is presented.

Vapor density

The density of a substance defines the volume one unit mass of the substance occupies. The density changes with temperature and pressure. The heat transfer rate and required compressor power input depends on the mass flow rate of the refrigerant, which is determined by the density of the refrigerant and the volume it flows through. The specific volume is defined as the inverse of the density and is often used in thermodynamic calculations instead of the density.

Volumetric capacity

The volumetric capacity VC is the product of vapor density and the specific heat of evaporation. The volumetric capacity relates to the refrigerant charge and both compressor and component sizes and may be calculated by Eq. 3-31.

$$VC = \rho h_{fg} = \frac{h_{fg}}{v} \qquad (3\text{-}31)$$

where VC is the volumetric capacity, ρ is the density of the refrigerant, h_{fg} is the latent heat of evaporation, and v is the specific volume of the refrigerant.

Thermal conductivity

The thermal conductivity is a measure of the ability of a substance to transfer heat by conduction. Heat transfer by conduction is heat transfer through a medium or substance. A refrigerant should have a high thermal conductivity, to obtain high heat transfer rates.

Critical temperature and pressure

The critical temperature and pressure defines the critical point as shown in the pressure-enthalpy diagram of Figure 3.9. The critical pressure and temperature is most important when considering the condensing operational conditions. This is because the condensing process occurs at a higher pressure than the evaporating process. It is important that the condensing pressure is kept well below the critical pressure. The saturation dome narrows towards the critical point, resulting in a decreasing latent heat of evaporation with increasing condensing pressure. At temperatures above the critical point, the refrigerant cannot condense. Some refrigeration cycles operate in the supercritical region, for example the transcritical refrigeration cycle of carbon dioxide, where the condensation process occurs at pressures above the critical pressure of carbon dioxide. The transcritical cycle of carbon dioxide is beyond the scope of this project.

Environmental properties

The development of refrigerants over the last decades has mainly been driven by environmental considerations. Legislations and protocols have been compiled to limit the emissions from refrigerants, with the aim to completely phase out harmful substances.

Ozone Depletion Potential

The ozone depletion potential (ODP) is a relative measure of a refrigerants impact on the ozone layer if leaked to the atmosphere. Refrigerant R11 (CCl_3F) is the reference with an ODP of one. According to the Montreal Protocol there is now a zero tolerance for ODP and all refrigerants in use today has an OPD of nearly zero.

Global Warming Potential

The global warming potential (GWP) of a refrigerant is a relative measure of the refrigerants impact on global warming if leaked to the atmosphere. Carbon dioxide is the reference substance with a GWP of one. Both the Kigali Amendment to the Montreal Protocol and EU regulation no. 517 aim to phase out the current refrigerants having GWP above 150 [5, 6]. The global warming potential is now the driving parameter for further development and research on refrigerants.

Safety and compatibility

Safety measures are a very important factor when considering refrigerants. The safety of the refrigerants and systems must be considered in context with the application and placement of the system.

ASHRAE safety classification

The ASHRAE (American Society of Heating, Refrigerating and Air-conditioning Engineers) safety classification [52] is the most commonly used safety classification for refrigerants. The safety classes consider both toxicity and flammability. The safety classes are presented in Table 3.1.

Table 3.1: ASHRAE safety classes for refrigerants [50]. The safety classes consider both flammability and toxicity and are the most commonly used safety classification system for refrigerants.

	Lower toxicity A	Higher toxicity B
No flame propagation 1	A1	B1
Lower flammability 2L	A2L	B2L
Lower flammability 2	A2	B2
Higher flammability 3	A3	B3

The safety classification is based on three parameters. The acute toxicity exposure limit (ATEL) is the defined as the concentration of the gas in atmospheric air that is toxic. The oxygen deprivation limit (ODL) is the concentration of the gas that results in insufficient oxygen for normal breathing. According to the ASHRAE standard the ODL is defined as the concentration of gas that will result in 19.5 % oxygen in the air when the mixture is uniform [52]. To set a maximum safe concentration limit of a gas in air both the ATEL and ODL are considered. Often the lowest value of the two is chosen as the maximum concentration, depending on the application and placement of the system. A vapor must have a certain concentration in air to be flammable or explosive if an ignition source is present. For each gas, a range of concentrations in air form a combustible mixture. The range is called flammable range and is defined by the flammability limits. The most flammable gases have very wide ranges of flammable concentrations. The lower flammability limit may also affect the maximum safe concentration if the refrigerant is flammable.

Corrosion

Some refrigerants are corrosive and require special corrosion resistant materials in the system. This may increase the cost of the system. If the materials are not corrosive resistant, the refrigerant will cause corrosive wear on the components leading to leakage and potentially failure.

Lubrication

Almost all compressor types require lubrication to minimize friction. Friction will induce wear mechanisms and increase energy consumption. The refrigerant in use must be compatible with the lubrication oil. The most common lubrication oils are hydrocarbon-based mineral oils. Synthetic lubrication oils have the advantage of a consistent composition but are more expensive. The oil also helps to seal the compressor, minimizing leakage of refrigerant. The lubrication oil may be used to provide cooling of the compressor and refrigerant during compression. Cooling of the compressor increases the specific volume of the refrigerant, resulting in a lower power input required to the compressor.

Table 3.2 present important properties of the commonly used synthetic refrigerant R134a and natural refrigerants R290 (propane) and R717 (ammonium). When moving towards refrigerants with low global warming potential the natural refrigerants are promising alternatives.

Table 3.2: Important refrigerant properties [16, 53, 54]. The various properties are presented for the commonly used synthetic refrigerant R134a and natural refrigerants R290 (propane) and R717 (ammonium).

Refrigerant	R134a	R290	R717
Type	HFC	HC	IO
Name	1,1,1,2-tetrafluoro-ethane	Propane	Ammonium
Composition	CH_2FCF_3	$CH_3CH_2CH_3$	NH_3
Saturation temperature @ 1 atm [°C]	-26.3	-41.8	-33
Heat of evaporation @ 55 °C [kJ/kg]	143.7	272,0	1032
Vapor density @ 1 atm, 25 °C [kg/m^3]	4.17	1.23	0.7
Critical temperature [°C]	101.1	96.67	132.4
Critical pressure [MPa]	4.07	4.24	11.4
Safety class	A1	A3	B2L
ATEL/ODL [kg/m^3]	0.21	0.09	0.00022
LFL [kg/m^3]	NF	0.038	0.116
ODP	0	0	0
GWP (100 y) (AR5)	1300	3	0

R134a

R 134a is a synthetic HFC refrigerant. It is the most commonly used refrigerant today, especially for medium evaporation temperature applications [8]. R134a is the refrigerant that contributes the most to global warming [55], and has a very high GWP value. The HFC refrigerants will be phased out over the next years. A lot of research and development efforts are put into finding replacements for R134a and other HFC refrigerants that may be used directly in existing systems.

R290

Propane is a hydrocarbon that is a promising alternative for the traditional refrigerants. It was used as refrigerant in the early systems, but due to the high flammability and safety issues its popularity dropped with the introduction of the safer synthetic refrigerants. The properties of propane are comparable to R134a, and it may replace R134a directly in existing systems [39]. Propane may also substitute the HCFC refrigerant R22 which is still in use in older systems [40]. The main disadvantage of propane is the high flammability, and it has a much higher safety classification than the other refrigerants listed. The flammability induces restrictions on the usage and placement of systems with R290 as refrigerant.

R717

Ammonium was used as refrigerant in the early stages before the development of safer synthetic

refrigerants. Ammonium has great thermodynamic properties for refrigeration and heating purposes. The very high specific heat of evaporation gives a high volumetric capacity. The main drawback of ammonium is its toxicity, and it is given safety classification B2L. In addition, ammonium vapor may form a combustible and explosive mixture in air. As ammonium has a very distinctive smell any leakage would normally be discovered quickly. Another drawback of ammonium is its corrosivity, which sets higher demands to system components. Ammonium systems require an oil separator on the discharge side of the compressor as ammonium and the oil are not miscible [45].

3.7 Control and regulation strategies

There are several methods of regulating both the heat pump system and the compressor. The compressor is usually the regulating component in the vapor compression cycle. System control regards the heat delivered to the heating system compared to the current heat demand. Regulation of the compressor is how the compressor is controlled in order to perform according to the desired heat capacity and the current feedback from the system. A controller is incorporated in the TRNSYS models created in this project. The controller component regulates the frequency and speed of the compressor.

3.7.1 Control strategies

Control strategies are ways to control if the heat produced by the system is satisfying compared to the heat demand of the building.

Return temperature to condenser

A common method of controlling the system is to measure the return temperature to the load side of the condenser. As the hot water flows through the building heat system, the temperature of the water drops as heat is transferred to the building. If the desired amount of heat is delivered to the building, the return temperature has a specific and known value. This value is calculated based on the known heat demand and the temperature of the water or fluid going in to the heating system of the building. If the temperature deviates from the expected temperature, the controller regulates the load of the system. If the measured temperature on the condenser inlet on the load side is lower than the design temperature, more heat was transferred to the building. Hence, the building has a higher heat demand than the heat capacity currently provided. The controller will take measures to increase the systems heating capacity. Similarly, if the return temperature is higher than expected there is less heat transferred to the building due to a lower heat demand. The controller decreases the systems heating capacity.

Ambient temperature

The system may be set to run at specific loads for various ambient temperature intervals. The desired heating capacity at each temperature interval is determined based on the buildings structure, heat transfer properties and internal gains. A required room temperature must be set. The heat demand in the building is then pre-calculated for each temperature interval. The control curve may be continuous or step-wise. This control strategy may not be continuous, and the system cannot control if it is actually providing the required heat to the heating system of the building. A compressor with continuous regulation possibilities is not necessary.

The two methods may be used in combination, allowing for a smooth regulation of the heat pump system in addition to any auxiliary heating systems in place.

3.7.2   Regulation strategies

There are several ways to regulate the operation of a compressor and heat pump system. Some methods are more complex than others are but may give great advantages in terms of energy savings. In the following sections a selection of common regulation options is presented.

On/off control

The most basic form of regulation of a heat pump system is the on/off control. The compressor in such a system may only operate at a single speed. If there is excess heat delivered to the building, the compressor is shut off. As the compressor is the driving force of the vapor compression cycle, the heat pump is also off. If there is a deficit in heat delivered to the building's heat system, the compressor is turned back on. The compressor operates on full load until there is excess heat delivered. There is usually a tolerance band of acceptable deviation from the current heat demand in the building. This is to avoid the compressor turning on and off too often. This regulation method may be used with any compressor type. The on/off control is gradually being replaced by more energy efficient methods [9]. The start-up process imposes stress and wear on the compressor, and during periods of low load the compressor will go on and off almost continuous.

Frequency regulation

The most commonly used regulation principle in newer systems is the frequency control. This regulation method is gradually replacing on/off control as the preferred regulation strategy. A frequency regulated system is also called variable speed drive system. The compressor must be able to operate at much lower loads than the design load. The frequency range the compressor tolerates varies between the different compressor types. This regulation method may be used with any compressor type. If the heat demand in the building is less than the maximum heat demand, the frequency is regulated by a frequency inverter. The inverter controls the speed of the electric motor driving the compressor, and hence the compressor itself. The lower frequency results in a lower refrigerant mass flow rate in the cycle. As the heat transfer is proportional to the mass flow rate, the delivered heating capacity decreases. Eq. 3-32 – 3-35. shows the relation between the frequency and the heat transfer rate.

$$rpm = \frac{60\,f}{n} \qquad\qquad (3\text{-}32)$$

where rpm is the rotational speed of the compressor, f is the frequency, and n is the number of pole pairs in the electric motor driving the compressor.

$$\omega = \frac{rpm}{60} \qquad\qquad (3\text{-}33)$$

$$\dot{m} = \omega\,V_{\text{swept}}\rho_{\text{suc}} \qquad\qquad (3\text{-}34)$$

where ω is the revolutions per second of the compressor, $\dot{m}$ is the mass flow rate, V_{swept} is the swept volume of the compressor, and ρ_{suc} is the density of the refrigerant at suction state. The heat transfer rate $\dot{Q}$ in the condenser may be calculated by Eq. 3-36.

$$\dot{Q} = \dot{m}(h_{\text{out}} - h_{\text{in}}) \qquad\qquad (3\text{-}35)$$

where h_{in} and h_{out} is the specific enthalpy of the refrigerant. The frequency inverter is composed of an A/D-converter and a D/A-converter. The current with a given frequency from the supply is first converted to a direct current. As the direct current is converted back to an alternating current, it is possible to set the required frequency of the current. The frequency inverter must be connected to the control component of the heating system in order to choose a proper frequency for the current heat demand.

Multiple compressors

Using multiple compressors in parallel is another common method to regulate heat pump systems. When compressors are placed in parallel, it is possible to obtain regulation in several stages. Each compressor is regulated by on/off control. At the maximum heat demand all compressors are on and operating at full speed. At the minimum load only one compressor is on and running at full speed. The number of stages possible depends on the number of compressors installed. The compressors may be of different sizes, giving even more stages when used in different combinations. Any type of compressor may be used. This regulation strategy may impose higher costs as several compressors are required.

4 TRNSYS – Transient System Simulation Tool

TRNSYS is a simulation program for transient energy systems. Many different energy systems may be simulated in TRNSYS, for example solar energy systems, ventilation and domestic water heating systems in buildings and co-generation systems. Included in the TRNSYS package is the SimulationStudio for creating projects/models, TRNBuild for creating 3D multi-zone buildings, TypeStudio for creating new components and TRNEdit for parametric studies and creating applications. The component libraries include components that are used in various energy systems. Additional component libraries may be purchased along with TRNSYS, for example the libraries developed by TESS (Thermal Energy System Specialists). The TESS libraries contain additional components, often more comprehensive and adjustable than the components included in the TRNSYS libraries. The simulations are created by connecting component types together, forming the energy system. This is done in the SimulationStudio. The main window in the SimulationStudio is the Assembly Panel, where the projects are created and modified. The projects are saved as Project files (*.tpf). The components are placed in the Assembly Panel from the component tree by clicking and dragging them into place. The components in TRNSYS are called Types and are represented by a set of equations relating the inputs and parameters to the outputs. Figure 4.1 shows an example project of a photovoltaic system from the TRNSYS documentation [56] opened in the SimulationStudio. The direct access tree containing all components is on the right-hand side of the window. The toolbar where various functions may be accessed is on the left side of the window.
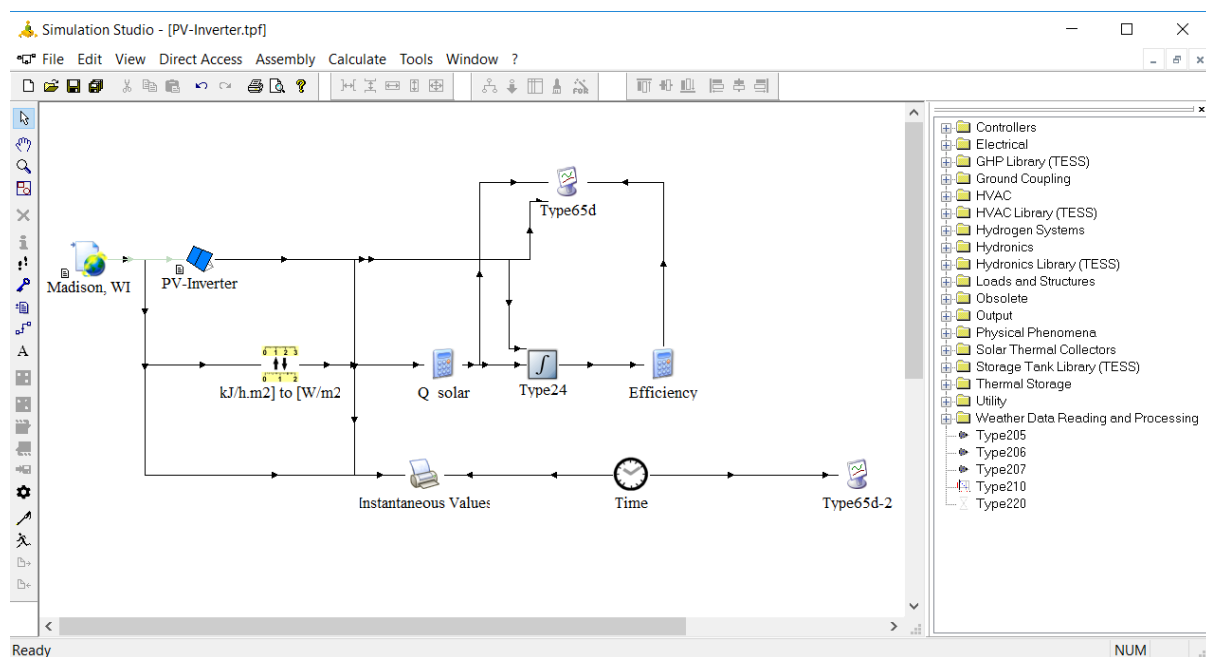
Figure 4.1: *Example project of a photovoltaic system open in SimulationStudio. The Direct Access component tree is on the right side of the window, and the Toolbar where various functions may be accessed is on the left side.*

Both parameters and inputs are inputs to the component. The parameters are constant values, while the inputs are time-dependent. All the outputs are considered time-dependent. As the component is placed in a TRNSYS-project, the user must define these parameters and connect the inputs from other components. The parameters are defined in the Variable Window, which may be accessed by double-clicking the component. The inputs may not be connected to other components and can be considered as parameters if desired. The outputs may be connected as inputs to other components. The

components are connected by the link-function, where inputs and outputs are connected. Plotters and printers may be connected to any outputs. There are many possibilities for registering outputs, both graphical, in tables and in text files. TRNSYS may also communicate with other programs such as MatLab, Excel and EES. The documentation to TRNSYS [56] is extensive and recommended for new users.

Figure 4.2 shows the parameter tab of the variable window for the compressor model Type 206. The values may be chosen by the user. Some parameters have a set default value, and some are restricted with a minimum and maximum value. TRNSYS will give an error if an invalid parameter value is set. The units may also be changed. This is not recommended without also changing the Proforma file and the source code. If needed, there are Types for unit conversion available in the TRNSYS component libraries.

		Name	Value	Unit	More	Macro
1		Evaporating temperature	15	C	More...	☑
2		Condensing temperature	55	C	More...	☑
3		Swept volume	0.002	m^3	More...	☑
4		Refrigrant	134	-	More...	☑
5		Degree of superheat	1	C	More...	☑
6		Pressureloss in suction line	1	kPa	More...	☑
7		Minimum frequency	15	1/S	More...	☑
8		r_loss	0.1	-	More...	☑
9		Logical Unit 1	152	-	More...	☑
10		Logical Unit 2	153	-	More...	☑

Figure 4.2: Variable window for the compressor model Type 206. The parameter values and initial input values may be set by the user in the variable window. The variable window is accessed by double-clicking the component.

Each Type has a Proforma file (*.tmf – Trnsys Model File). The Proforma file is the standard method for documenting component models in TRNSYS. All necessary information regarding the component and its ability to transfer information is contained in the Proforma file. The Proforma file may be accessed by right-clicking the component. The Proforma file contains general information about the component and its creator, the variables used in the component, and file paths to the source code and any external files used by the component. Figure 4.3 shows the Proforma window for the user created compressor model Type 206.
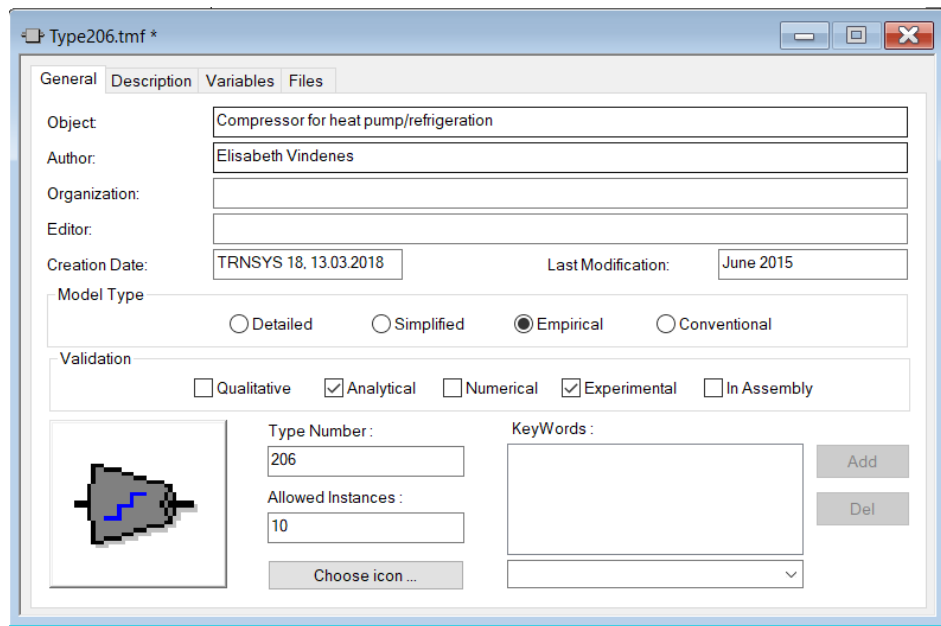
Figure 4.3: *Proforma file for the user created compressor model Type 206. The Proforma file contains general information regarding the component, the variables used in the component and file paths to the source code and any external files used by the component.*

When a simulation is started, an input file (*.dck) is created by the SimulationStudio. The input file contains information about the simulation, for example simulation settings, the types present in the project file and which Dynamic Link Libraries that are available. The simulation engine in TRNSYS is called the kernel. All components and the TRNSYS kernel are found in dynamic link library files (*.DLL). The kernel executes the simulations by calling the components in the project at each timestep of the simulation. At the start of a simulation, the kernel reads the input file to determine which Types are present in the simulation. The kernel also checks the input file for errors (e.g. syntax errors) and that all the Types in the project are available in .DLL files. If the input file is found to be correct, the simulation is initiated. The kernel calls the components in the project and registers the inputs to the components. The Types that have changed inputs are re-called. The process continues until all inputs have converged to the chosen tolerance. When all inputs have converged, the kernel continues to the next timestep. At the end of a simulation the TRNSYS kernel produces an error file called list file (*.lst). In the list file, all errors and messages that appear during the simulation are listed. Output files are written by the kernel at the end of the simulation if there are such files set up in the simulation, for example by a printer component. It is possible to trace components during a simulation. If components are traced, the calls to the traced components are logged in the *.lst-file.

In the simulation settings (accessed by the toolbar on the left side, or Assembly/Settings) the global settings for the simulation are defined. The basic settings include the simulation time and simulation time step. TRNSYS cannot simulate timesteps larger than one hour. The advanced settings include settings for the solution method, tolerances, and other global settings. For further information on the advanced settings refer to TRNSYS documentation Vol. 2 Simulation Studio [56]. The order of which the TRNSYS kernel calls the components may be chosen in the settings window. The order may affect the simulation, and an incorrect order may lead to errors when the simulation is run. For example, controller components should be called before the components it controls. All data files, for example weather files, should be called first. Figure 4.4 shows the basic settings tab of the settings window.
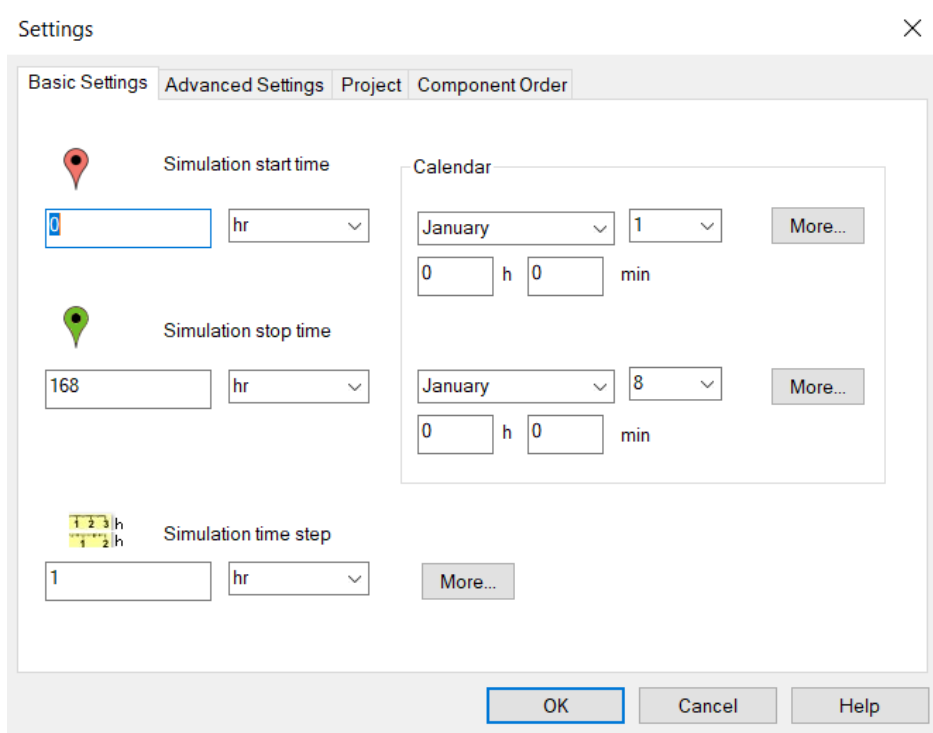
Figure 4.4: *The settings window in the SimulationStudio. The basic settings include the simulation time and simulation time step. The advanced settings are recommended to keep at default settings. The component order may be important for the simulation results.*

TRNSYS is an open source program, meaning that the source codes for all the components and the kernel are available to the user. This gives the possibility for users to either modify existing components or create new components. The documentation of TRNSYS [56] includes a Programmer's Guide, to assist users in creating new components. The source codes for the components are written in Fortran 90 programming language. Introduction notes on Fortran 90 by Davies et al. [57] are recommended for further information about programming in Fortran 90 language. TRNSYS also includes a Fortran environment called TypeStudio where programs may be written and compiled. It is also possible to write the code in another program and use the TypeStudio for compilation. The compiler checks the code for programming errors. Examples of programming errors are syntax errors, undeclared variables, and unexpected statements. The compiler states the line and point of the error. When the source code is compiled a .DLL file is created, and the component may be used in the SimulationStudio. The TypeStudio compiler only checks for programming errors. Errors may also appear in TRNSYS during simulations, even if the source code was successfully compiled. Examples of such errors are inconsistent number of parameters in the source code and Proforma file, prohibited values of parameters and if the values of the outputs do not converge.

In order to successfully implement a new component in a TRNSYS simulation, a Proforma file must be created for the component. This is done by opening the Proforma of any component. The necessary changes can be made, and the new Proforma file is saved (File/Save As) with the corresponding Type-number. The original Proforma file is contained in its prior state. The inputs, parameters and outputs of the Type is defined in the Proforma file and must match the source code. When defining the various variables, it is possible to define an upper and lower limit and set a default value. In addition, a description of the variable may be added to clarify to the user how the variable affects the component.

The user created Type must be available from the component tree to be used in simulations. When the source code is compiled, the DLL-file is created and the proforma file is set up, the component may be added to the direct access tree in Simulation Studio by clicking DirectAccess/RefreshTree. The source code should be saved in the TRNSYS18/MyTypes directory, and the proforma file in the TRNSYS18/Studio/Proformas directory. The .DLL-file is automatically saved to the proper location by the TRNEdit Fortran compiler.

The advanced simulation settings are kept at the default settings. The component order is optimized by TRNSYS. The component order is important for the simulation, especially the iterative feedback controller is sensitive to the component order. Refer to Chapter 5.2 for further information regarding the iterative feedback control. The optimal component order is starting with the data reader, obtaining the necessary input. The building load is calculated first, and the result is sent to the iterative feedback control. The compressor is called next, and the rest of the vapor compression cycle thereafter. In the models containing the variable speed heat pump components the heat pump is called after the iterative feedback control. The plotters are called right after the connected component.

5 Development of components

The following chapter presents a detailed description of the components and system models created in this project. The complete description of the parameters, inputs and calculations of the created components are presented. The complete source codes and a description of the basic structure of TRNSYS component source codes are presented in Appendix I and II respectively. Some components from the TRNSYS component library are used in the simulations, the parameter and input values of these component are presented. The various components are to be connected in a vapor compression cycle and heat pump system as presented in Figure 5.1.
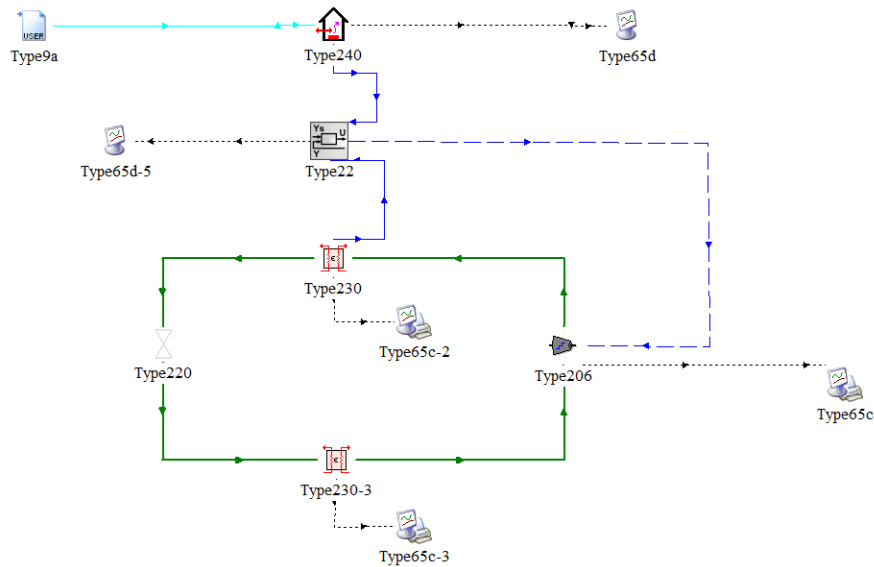
Figure 5.1: *Screenshot from SimulationStudio, TRNSYS model with compressor Type 206.The green lines connect the components of the vapor compression cycle, Type 206, 230 and 220. The dashed black lines are the connections to the online plotters Type 65c registering the output values. The blue lines show the inputs and output to and from the iterative feedback control Type 22. The light blue line is the temperature data input to the building load component Type 240.*

Table 5.1 present some local variables of thermodynamic properties used in the source codes for the created components. Various descriptive additions to the variables are used, for example `is` for isentropic, `suc` and `dis` for suction and discharge respectively, `c` and `e` for condensing and evaporating respectively and `SH` and `SC` for superheat and subcooling respectively. Local variable names for parameters, inputs and outputs are presented separately for each component.

Table 5.1: *Description of local variables used in the source codes of the created components.*

Local Variable	Description	Unit
qu	Vapor quality	-
h	Specific enthalpy	kJ/kg
v	Specific volume	m^3/kg
s	Specific entropy	kJ/kg·K
P	Pressure	kPa
T	Temperature	°C

5.1   Components created by the author

The main component that is created is the compressor. In the TRNSYS component library there is only one compressor component available. Type 167 is a multi-stage compressor developed for hydrogen systems. This compressor component is not suitable for this project as it is not adapted to variable speed drive. Recall that components in TRNSYS are called Types. In addition to the compressor Types created, an expansion valve component, an evaporator and condenser component and a simplified building load component are created. Based on the other components two heat pump components were also created.

The main objective of the compressor component is to simulate the compressor power at variable load conditions. Combined with the heat transfer rate in the condenser the performance of the system may be evaluated. Refer to Chapter 3.1 and 3.2 for further information on compressor and system performance characteristics. The compressor component is regulated by frequency control. The compressor power input, refrigerant mass flow rate and isentropic efficiency are calculated based on the current frequency value. The heating capacity of the system at the current compressor speed is compared to the heat demand of the building. If the heat delivered is not satisfying, the frequency will be adjusted. It is important to notice that the power input to the system may be higher than the compressor power input calculated in these models. This is due to loss in the electric motor and inverter, and any auxiliary equipment like fans or oil pumps requiring additional power input.

5.1.1   Type 206 – Compressor

Type 206 models a variable speed drive single stage compressor for heat pump and refrigeration applications. The main aim of the model is to calculate the compressor power. The compressor power is calculated by the 10-coefficient polynomial given in AHRI Standard 540 [29]. This polynomial is used by compressor manufacturers to produce compressor maps and calculate compressor performance. The polynomial may also be used to calculate cooling capacity and refrigerant mass flow rate, with different coefficients for each purpose. The polynomial is presented in Eq. 5-1.

$$\dot{W}_{in} = C_1 + C_2 T_e + C_3 T_C + C_4 T_e{}^2 + C_5 T_e T_c + C_6 T_c{}^2 + C_7 T_e{}^3 + C_8 T_c T_e{}^2 + C_9 T_c{}^2 T_e$$
$$+ C_{10} T_c{}^3 \tag{5-1}$$

where $\dot{W}_{in}$ is the compressor power input, T_e and T_c are the evaporating and condensing temperature respectively, and C_1-C_{10} are the coefficients for the compressor power input. The evaporating and condensing temperature is the saturation temperature at the evaporating and condensing pressure respectively.

The coefficients are different for each compressor model, refrigerant and compressor speed. The coefficients are calculated by linear regression based on measurements on each compressor model under standard test conditions given in AHRI 540 [29]. The standard method requires at least eleven data points to calculate the coefficients. The coefficients have a limited validity range. The validity is limited by the evaporating and condensing temperature and is valid only for a fixed value of suction gas super heat and degree of subcooling. A report produced for the Air-Conditioning, Heating and Refrigeration Institute (AHRI) [58] summarizes the AHRI Standard 540 method for producing

compressor maps and the accuracy of the method. The AHRI-method is also compared to other methods in the literature. The error when changing the degree of superheat is larger when calculating refrigerant mass flow rate than compressor power input [58]. The error found for the power input was lower than 2 % when changing the degree of superheat. In this model the mass flow rate is not calculated by the polynomial and related coefficients. The effect of superheat on the refrigerant mass flow rate is accounted for by the specific volume at suction state when calculating the mass flow rate.

The coefficients used in this compressor model are obtained through the Bitzer free online software [59]. The software may be used to design heat pump and refrigeration systems, and for compressor selection. There are many compressor types and models available in the software. There are however certain limitations. The main problem is the limited number of variable speed compressors available, especially for natural refrigerants. In addition, the software is mainly concerned with refrigeration. The validity range of the evaporating temperature is –25 °C – 12.5 °C, while heat pumps may experience higher evaporating temperatures depending on the temperature of the heat source. The coefficients obtained from Bitzer software are for semi-hermetic reciprocating compressor model 4NE-20.F4Y with refrigerant R134a. This compressor type is chosen simply because the compressor coefficients were available for variable speed drive at condensing and evaporating temperatures close to the desired temperature. The compressor has a maximum frequency of 87 Hz, giving a maximum heating capacity of about 79 kW and a maximum compressor power input of about 20 kW. The minimum frequency the compressor may operate with is 25 Hz. The coefficients are obtained at various frequencies with intervals of 10 Hz in the operating range of the compressor. The other components in the TRNSYS simulation are adapted to fit to the compressor design limitations. The coefficients are obtained with degree of subcooling of 0 °C and degree of superheat of 10 °C according to AHRI Standard 540 [29]. As discussed before, any change in degree of superheat in the model is accounted for when calculating the refrigerant mass flow rate. Bitzer software provides coefficients for compressor power, cooling capacity, and refrigerant mass flow rate. In addition, calculated variables are provided. Only the coefficients for compressor power input are used in this project. The calculated values are used for validation of the results from various simulations. Figure 5.2 shows a screenshot from Bitzer Software [59].
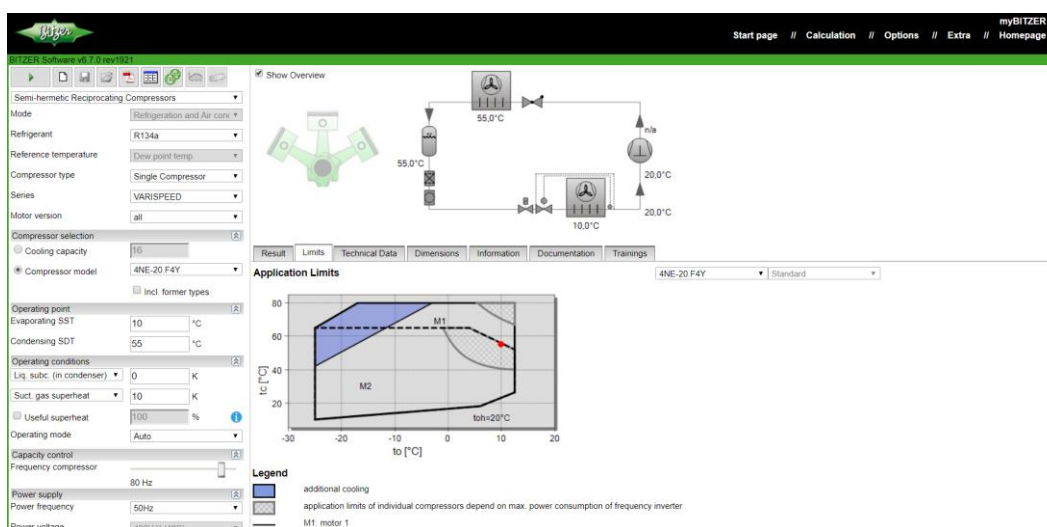
Figure 5.2: *Screenshot from Bitzer Software [59] showing the compressor performance map of semi-hermetic reciprocating compressor model 4NE-20.F4Y with $T_e = 10$ °C, $T_c = 55$ °C, $T_{SH} = 10$ °C and $T_{SC} = 0$ °C.*

The model created in this project uses linear interpolation for calculations of coefficients between the neighboring frequencies. Rice et al. [60] uses interpolation to determine the compressor power and mass flow rate between neighboring speeds. Cheung et al. [61] investigate the uncertainties of the empirical compressor maps when extrapolating outside the range of the polynomial. They conclude that the uncertainty in the compressor map is a combination of measurement inaccuracies, uncertainty due to inputs and random error. The uncertainty of measurements is the highest contributor to uncertainty within the compressor map range. The study does not include the effect of interpolation within the compressor map range. Ravi et al. [62] studies compressor maps for centrifugal compressors, and the possibility of both interpolation and extrapolation. Both interpolation and extrapolation may give accurate results, but the authors use a method for extrapolating and interpolating called cubic spline method. The cubic spline method gives a continuous curve unlike the stepped curve of linear interpolation and may be more accurate. Park et al. [63] created a compressor model utilizing compressor map data from a manufacturer. In the study, both interpolation and extrapolation are used for frequencies within and outside the range of the polynomial. The focus of most of these studies are on the effect of extrapolation outside the valid temperature range. As interpolation between frequencies is used in some studies, it is concluded that linear interpolation between frequencies do not give large errors in the calculated compressor power input.

The validity of interpolation between frequencies to obtain coefficients for any frequency has been checked for a set of coefficients for Bitzer reciprocating compressor model 4NE-20.F4Y. The coefficients are presented graphically with the frequency along the X-axis and coefficient value along the Y-axis. All ten coefficients show a linear relationship with the frequency. In addition, coefficients at some frequencies are calculated by interpolation and compared to the actual coefficients obtained from Bitzer. The largest deviation from the actual coefficients found is about three percent. This is only for one coefficient at one frequency value. The coefficient in question (C_6) has a very small value but is multiplied with the condensation temperature in the second order. Still, the error in the calculated power input is only approximately one watt. Most deviations are less than 0.5%. Figure 5.3 shows coefficient C_1 and its relationship to the frequency, showing the linear behavior. The results of this check for all coefficients $C_1 - C_{10}$ may be found in Appendix III.
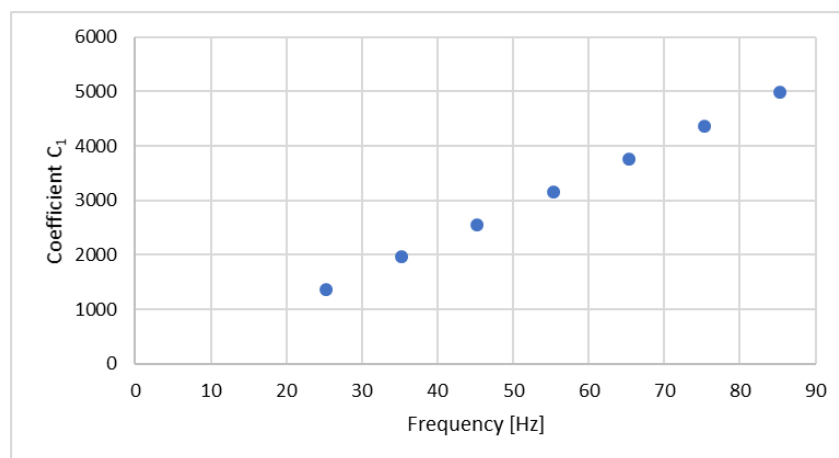
Figure 5.3: *Linear behavior of coefficient C_1 with frequency. All coefficients show a linear behavior with the frequency, and linear interpolation between frequencies is assumed to be accurate.*

The coefficients are generally not available to the public. Some coefficients may be found in the literature. Yang et al. [64] provide coefficients for three different compressor types, all constant speed drive compressors. Shao et al. [65] provides some coefficients for the previously used second order polynomial with only six coefficients. Compressor manufacturer Dorin [66] provides a free software for compressor selection were coefficients are available. The coefficients obtained from this software are equal for all tested frequencies and were considered unusable for this project. Sabroe [67] provided access to the Sabroe Matchmaster Program for compressor calculations and selection. The program does not provide coefficients for the polynomial. Mayekawa [68] could not provide access to their compressor selection tool.

The evaporating and condensing temperatures are chosen as parameters of compressor model Type 206. Other parameters are also required for Type 206 to work, these are listed in Table 5.2. The local variable is the variable names used in the source code of the component. The standard condition values are used for the simulations executed in order to compare the various compressor and heat pump models.

Table 5.2: *Parameters for compressor Type 206. The local variable is the variable name used in the source code for the given parameter. The standard condition values are used for the standard condition simulations, which other simulations are compared against.*

	Parameters	Local Variable	Standard condition value	Unit
1	Evaporating temperature	T_e	10	°C
2	Condensing temperature	T_c	55	°C
3	Swept volume	V_sw	0.0006	m^3
4	Refrigerant	nref	134	-
5	Degree of superheat	T_SH	10	°C
6	Fractional pressure loss in suction line	P_loss	0	-
7	Minimum frequency	f_min	25	1/s
8	Volumetric efficiency	n_vol	0.85	-
9	Fractional heat loss/cooling	r_loss	0	-
10	Logical Unit 1	LU1	-	-
11	Logical unit 2	LU2	-	-
12	Logical Unit 3	LU3	-	-

The refrigerant must be specified in order to calculate the thermodynamic properties of the refrigerant at the various states. The refrigerants are denoted by an integer value. The values for each refrigerant available may be found in the Programmer's Guide [56]. The integer value usually coincides with the numerical code in the refrigerant name. For this compressor model the choice of refrigerant is limited due to the coefficients for the polynomial. The coefficients are different for each refrigerant and are listed in separate text files.

The coefficients are accessed by the component through a logical unit. The external file must be connected to the component through the Proforma file. TRNSYS assigns the file a logical unit number automatically and this parameter value should not be changed by the user. Type 206 currently have

three logical unit numbers assigned to the component. This facilitates for using the component with the three refrigerants R134a, R717 and R290. Only the data file for R134a contains real coefficients. The two other files and logical unit parameters are added to facilitate for the addition of coefficients for other refrigerants if these are available. The user may connect any file containing coefficients to one of the logical units assigned by changing file path in the variable window accessed by double-clicking the component in SimulationStudio. The coefficients and data must be organized as in the data file provided Appendix IV project for the `InterpolateData` subroutine to read the data correctly. The source code calculations and the various subroutines used in the codes are presented in the next section. The data file containing the compressor coefficients are found in Appendix IV. It is also possible to add additional data files in order to use the model with other refrigerants. The process involves changes in both the source code and Proforma of the component. Instructions on how to add data files to a component can be found in Appendix V.

The degree of superheat is set to 10 °C for the standard conditions case. This is consistent with the values for degree of superheat found in literature [8, 64, 65] and the AHRI standard 540 value [29]. The minimum frequency is the lowest frequency the compressor can operate with. For frequencies lower than this minimum value the compressor will be switched off, and the power input, refrigerant mass flow rate and isentropic efficiency are set to zero. The maximum frequency of 85 Hz in the data file for the coefficients corresponds to 100 % load. The minimum frequency of 25 Hz equals a load of about 30 %. The volumetric efficiency is required to calculate the refrigerant mass flow rate in the compressor. The standard conditions volumetric efficiency is set to $\eta_{\text{vol}} = 0.85$. The value is chosen based on the simulations by Mendoza-Miranda et al. [8] and Yang et al. [64] of reciprocating compressor with refrigerant R134a, based on a compression ratio CR of 3.6 for the chosen standard conditions saturation temperatures. The compression ratio is calculated by Eq. 5-2.

$$CR = \frac{P_c}{P_e} = \frac{1492.5 \text{ kPa}}{415.3 \text{ kPa}} = 3.59 \tag{5-2}$$

where P_c and P_e is the condensing and evaporating pressure respectively of the standard conditions condensing and evaporating temperatures. The values of the saturation pressure are obtained from tables in reference [16], which are consistent with calculated saturation pressures during simulations. The swept volume is found in the documentation provided by the Bitzer software [59]. If the compressor is intentionally cooled or experiences heat loss, this is accounted for in the fractional heat loss parameter. The heat loss or cooling is considered a fraction of the compressor power input. Both the fractional pressure loss in the suction line and the fractional heat loss parameters are set to zero. The parameters are added to the component to enable future users to account for these effects if necessary. Compressor Type 206 requires only two inputs from other components. The inputs are presented in Table 5.3. The local variable is the variable name used in the source code of the component.

Table 5.3: Inputs to compressor Type 206. The local variable is the variable name used in the source code of the component.

	Inputs	Local Variable	Unit
1	Refrigerant mass flow rate	m_suc	kg/h
2	Frequency	f	1/s

The refrigerant mass flow is output from the evaporator component Type 230. Refer to Chapter 5.1.4 for information on Type 230. The mass flow is calculated and controlled by the compressor. The input from the evaporator is present in order to connect the components in a complete vapor compression cycle in TRNSYS Simulation Studio. The frequency is output from the iterative feedback controller Type 22. The values of the inputs vary with time. The frequency is controlled by the difference between the heat demand of the building load Type 240 and the delivered heat from the condenser component Type 230. Refer to Chapter 5.2.1 for more information on the iterative feedback control component. The frequency regulates the rotational speed of the compressor and hence the refrigerant mass flow. The mass flow is considered constant through the compressor at a given frequency, as the compressor is the regulating component. This is consistent with other models, for example the compressor model developed by Koury et al. [23].

Compressor Type 206 has six outputs. Some outputs are used as inputs to other components in order to connect the components in a complete vapor compression cycle in TRNSYS Simulation Studio. Some outputs are registered in a plotter or printer and used to analyze the results of the simulation and the compressor performance. An output may be connected to a plotter component and another component in the cycle simultaneously. The outputs from Type 206 are presented in Table 5.4. The local variable is the variable name used in the source code of the component.

Table 5.4: Outputs from compressor Type 206. The local variable is the variable name used in the source code of the component. Some outputs are inputs to other components, and some are registered and plotted as results of the simulations.

	Outputs	Local Variable	Unit
1	Compressor power	W_in	W
2	Isentropic efficiency	n_is	-
3	Rotational speed	rpm	1/min
4	Mass flow rate	m_suc	kg/h
5	Specific volume at inlet	v_suc	m^3/kg
6	Discharge enthalpy	h_dis	kJ/kg

The mass flow rate and discharge enthalpy are inputs to the condenser unit Type 230. The compressor power and isentropic efficiency are plotted and printed to a text file at the end of each time step.

Calculations

The following section presents the calculations performed in Type 206. The calculations start after the current input values have been retrieved and are repeated in each time step of the simulations. The input values are checked for errors by the `FoundBadInput` function. The first step in the calculations is to retrieve the coefficients for the polynomial at the given frequency. This is done by calling access function `InterpolateData`. This function can interpolate data in up to six dimensions. There can be up to ten dependent variables Y and six independent variables X. In this case the coefficients are the dependent variables (nY=10) and frequency is the independent variable (X=f, nIND=1). There are seven values of variable f in the data file (nX=7). The `InterpolateData` routine takes the frequency as the input and gives the ten coefficients as outputs. The `InterpolateData` routine cannot extrapolate beyond the values of the independent variables. If the input value of the independent variable is outside the range listed in the data file, the `InterpolateData` routine will set the

dependent variables to the minimum or maximum values. If the frequency is to low the compressor should be off. To solve this in the model the compressor power and other variables are set to zero for frequencies lower than a minimum frequency chosen by the user. In addition, the iterative feedback control has a limited range of allowed control signal values. In a real compressor the lower limit on frequency regulation varies for the different types. Reciprocating compressor model 4NE-20.F4Y has a lower frequency limit of 25 Hz. The code for this section is given below.

```fortran
If (f<f_min) Then

        f=0.0

        rpm=0.0

        W_in=0.0

        m_suc=0.0

        n_is=0.0

ElseIf (f>=f_min) Then

! Check input values

If (m_suc<0) Call FoundBadInput(1,'Fatal','The suction mass flow rate
cannot be negative')

If (f<0) Call FoundBadInput(2,'Fatal','The frequency cannot be
negative')

! Use correct data file

If (nref==134) Then

            LU=LU1

    ElseIf (nref==290) Then

            LU=LU2

    ElseIf (nref==717) Then

            LU=LU3

    EndIf


    nX=7          ! There are 7 values of variable f in the file
    X(1)=f        ! Sets X to local variable f
    nY=10         ! There are 10 coefficients to each value of f
    nIND=1        ! There is 1 independent variable (frequency)


        Call InterpolateData(LU,nIND,nX,nY,X,Y)
        C1 = Y(1)
        C2 = Y(2)
        C3 = Y(3)
        C4 = Y(4)
        C5 = Y(5)
```

```
        C6 = Y(6)

        C7 = Y(7)

        C8 = Y(8)

        C9 = Y(9)

        C10 = Y(10)

    ! Calculate compressor power

W_in=C1+(C2*T_e)+(C3*T_c)+(C4*T_e**2)+(C5*T_c*T_e)+(C6*T_c**2)+(C7*T_
    e**3)+(C8*T_c*T_e**2)+(C9*T_c**2*T_e)+(C10*T_c**3)
```

The `If` and `ElseIf` statements at the beginning of the code checks the value of the frequency input. If the value is equal to or higher than the minimum frequency the rest of the code is executed. If the frequency is lower than the minimum frequency, the variables are set to zero and no further calculations are executed.

After the compressor power is calculated the model must calculate some thermodynamic properties. These are necessary for the calculation of the efficiencies and mass flow rate. The thermodynamic properties are found by calling the utility subroutine `Coolprop_Fluid_Properties`. The Coolprop utility subroutine is described in the Programmer's Guide [56]. The routine calls the CoolProp program [69] to obtain the thermodynamic properties required. Two known properties must be supplied along with an integer value denoting which two properties are known. The refrigerant is specified by the user in the variable window of the component in SimulationStudio by an integer parameter value. The first call to the Coolprop function is to obtain the thermodynamic properties at the outlet of the evaporator prior to any superheating. The known properties are the evaporating temperature and the vapor quality. Based on these properties the saturation pressure, specific enthalpy, specific entropy, and specific volume are calculated by the Coolprop function. The code for this call is presented below.

```
        qu = 1                              ! Saturated vapor state

        prop(1)=T_e

        prop(5)=qu

    Call Coolprop_fluid_properties(prop,nref,15,iFlagr)

        If (ErrorFound()) Return

        P_e = prop(2)

        h_e = prop(3)

        s_e = prop(4)

        v_e = prop(6)
```

The thermodynamic properties at suction state are calculated next. In this step both superheating of the refrigerant and pressure loss in the suction line are accounted for, if specified by the user. Both the degree of superheat and the pressure loss must be supplied by the user as parameter values in the Simulation Studio. If the cycle has no superheating or pressure loss, the suction state properties cannot be calculated using the evaporating temperature and pressure. This is due to the refrigerant

61

being a saturated mixture. In this case the temperature and pressure are dependent properties. If there is no superheating or pressure loss in the cycle the properties at suction state is equal to the properties obtained for the evaporator outlet state. The code for this call to the Coolprop utility subroutine is stated below.

```
If ((T_SH>0.d0).or.(P_loss>0.d0)) Then
    T_suc=T_e+T_SH
    P_suc=P_e-(P_e*P_loss)
    prop(1)=T_suc
    prop(2)=P_suc
    Call coolprop_fluid_properties(prop,nref,12,iFlagr)
        If (ErrorFound()) Return
    h_suc=prop(3)
    s_suc=prop(4)
    v_suc=prop(6)
ElseIf ((T_SH==0.d0) .and. (P_loss==0.d0)) Then
    P_suc=P_e
    h_suc=h_e
    s_suc=s_e
    v_suc=v_e
EndIf
```

Based on the known properties the specific enthalpy, specific volume, and specific entropy at suction state are obtained. The chosen unit for pressure in the Proforma file for the component is kPa. The documentation to the Coolprop utility subroutine in TRNSYS Documentation Vol.7 Programmer's Guide states that the unit for pressure obtained by Coolprop is MPa. The values for pressure obtained during simulations are very high. The value of the calculated saturation pressure in the condenser is $1,4915 \times 10^6$. Interpolation of tabulated data in reference [16] gives a saturation pressure at 55 °C of 1493,38 kPa, which equals $1,4934 \times 10^6$ Pa and is consistent with the value obtained through the Coolprop utility subroutine. After conferring TRNSYS Support it is concluded that the unit for pressure in Coolprop is Pa, not MPa.

Following the calculations of the suction state properties, the refrigerant mass flow rate is calculated. The mass flow rate is an input to the condenser, and the mass flow rate in the cycle is controlled by the compressor frequency. The number of poles in the electric motor is assumed to be four. The number of poles or pole pairs could be added as a parameter to the compressor model. The user may then determine the number of poles in the electric motor freely. It is chosen not to use this as a parameter to limit the number of parameters necessary for the user to obtain, and hence keep the model as simple as possible. The reciprocating compressor model 4NE-20-F4Y for which coefficients have been obtained, is driven by an electric motor with two pole pairs [59]. The mass flow rate may be calculated by Eq. 5-3 – 5-5.

$$rpm = \frac{60\,f}{2} \tag{5-3}$$

where rpm is the rotational speed or revolutions per minute, f is the frequency and the number 2 denotes the two pole pairs in the electric motor. The revolutions per second and the refrigerant mass flow rate are then calculated.

$$\omega = \frac{rpm}{60} \tag{5-4}$$

$$\dot{m} = 3600\omega V_{\text{swept}}\eta_{\text{vol}}\frac{1}{v_{\text{suc}}} \tag{5-5}$$

where $\dot{m}$ is the refrigerant mass flow rate, ω is revolutions per second, V_{swept} is the swept volume of the compressor, η_{vol} is the volumetric efficiency and v_{suc} is the specific volume of the refrigerant at suction state. The code for calculating the refrigerant mass flow rate and compressor power is stated below.

```
rpm=((f*60.0)/2.0)

w=(rpm/60.0)

m_suc=3600.0*w*V_sw*n_vol/v_suc
```

After calculating the refrigerant mass flow rate the condensing pressure must be obtained by a call to the Coolprop utility subroutine. The known properties are the condensing temperature and vapor quality.

```
qu=0                             ! Saturated liquid

prop(1)=T_c

prop(5)=qu

Call Coolprop_fluid_properties(prop,nref,15,iFlagr)

      If (ErrorFound()) Return

P_c=prop(2)
```

The next step is to obtain the thermodynamic properties at discharge state. The properties at this stage are all unknown, and two must be estimated. The discharge pressure is assumed to be equal to the condensing pressure, as it is the minimum pressure the compressor must be able to deliver. The second property used as a known property for the discharge state is the discharge specific enthalpy. The calculation of the discharge enthalpy is derived from an energy balance between the inlet and outlet of the compressor. Refer to Chapter 3.2 for further information regarding the energy balance and compression process. The discharge enthalpy is calculated by Eq. 5-6.

$$h_{\text{dis}} = h_{\text{suc}} + \frac{\dot{W}_{\text{in}}(1 - r_{\text{loss}})}{\dot{m}_{\text{suc}}} \qquad (5\text{-}6)$$

where h_{dis} and h_{suc} is the refrigerant discharge and suction specific enthalpy respectively, r_{loss} is the fractional heat loss from the compressor, $\dot{m}_{\text{suc}}$ is the refrigerant mass flow, and $\dot{W}_{\text{in}}$ is the previously calculated compressor power input. The loss fraction is defined by the user as a parameter. If r_{loss} is set to zero, the compressor is adiabatic, i.e. there is no heat transfer to or from the compressor. Cooling of the compressor during compression is advantageous because the density is kept low, requiring lower work input to the compressor. Refer to Chapter 3.1.3 for details. Based on the pressure and enthalpy the discharge temperature and specific volume are obtained using the Coolprop function. The code for this section is stated below.

```
P_dis=P_c

h_dis = h_suc+(((W_in/1000.0)*(1-r_loss))/(m_suc/3600.0))

prop(2)=P_dis

prop(3)=h_dis

Call Coolprop_fluid_properties(prop,nref,23,iFlagr)

    If (ErrorFound()) Return

T_dis=prop(1)

v_dis=prop(6)
```

Some unit conversions are necessary to obtain the discharge specific enthalpy in unit kJ/kg. The calculated compressor power input is in unit W and must be converted to kW. The refrigerant mass flow rate must be converted from kg/h to kg/s.

The final call to the Coolprop function is made to calculate the enthalpy at discharge state in the ideal case of isentropic compression. Refer to Chapter 3.2.2 for further information on isentropic compression. The known properties of the isentropic discharge state are the discharge entropy which is equal to the suction enthalpy for isentropic compression and the discharge pressure. The isentropic efficiency is then calculated. The code for this call is stated below.

```
prop(2)=P_dis

prop(4)=s_suc

Call Coolprop_fluid_properties(prop,nref,24,iFlagr)

 If (ErrorFound()) Return

h_is = prop(3)


n_is=((h_is-h_suc)/(h_dis-h_suc))

EndIf
```

The `EndIf` statement at the end of this section belongs to the `If` and `ElseIf` statements at the very beginning of the code, where the frequency value is checked. The isentropic efficiency is calculated from specific enthalpy values obtained through the Coolprop utility subroutine. Refer to Chapter 3.2.4 and 3.3 for further information about the efficiencies and their significance to compressor performance. The isentropic efficiency is calculated by Eq. 5-7.

$$\eta_{\text{is}} = \frac{\dot{W}_{\text{is}}}{\dot{W}_{in}} = \frac{h_{\text{is}} - h_{\text{suc}}}{h_{\text{dis}} - h_{\text{suc}}} \qquad (5\text{-}7)$$

where η_{is} is the isentropic efficiency, $\dot{W}_{\text{is}}$ and $\dot{W}_{in}$ is the isentropic and calculated compressor power respectively, h_{is}, h_{suc} and h_{dis} are the isentropic, suction state and discharge state specific enthalpy respectively. When the compressor power, efficiency, and other variables for the current timestep are calculated, the model sets the outputs of the calculations and returns these to the TRNSYS simulation. The complete source code for compressor Type 206 is found in Appendix I.

5.1.2   Type 207 – Polytropic compressor

An alternative compressor model is developed due to the difficulties of finding proper coefficients for the polynomial used in Type 206. Type 207 is modelled as a polytropic compressor based on traditional thermodynamic calculations. Refer to Chapter 3.2 for further information about the polytropic compression process and the basis of these calculations. The compressor work is calculated by Eq. 5-8.

$$\dot{W}_{\text{in}} = \dot{m} P_{\text{suc}} v_{\text{suc}} \frac{n}{n-1} \left(\left(\frac{P_{\text{dis}}}{P_{\text{suc}}} \right)^{\frac{n-1}{n}} - 1 \right) \qquad (5\text{-}8)$$

where $\dot{W}_{\text{in}}$ is the compressor power input, $\dot{m}$ is the refrigerant mass flow rate, P_{suc} and P_{dis} are the suction and discharge pressure respectively, v_{suc} is the suction state specific volume, and n is the polytropic exponent. The model should simulate a real compressor and the polytropic exponent is higher than the isentropic exponent. If the polytropic exponent is lower than the isentropic exponent, as for internally reversible compression processes, the calculated isentropic efficiency will be higher than one.

Type 207 differs from Type 206 in both parameter and input variables, and calculations. The necessary parameters for this model are listed in Table 5.5. The standard conditions values are used for the simulations executed in order to compare the various compressor and heat pump models.

Table 5.5: *Parameters to Type 207 - Polytropic compressor. The local variables are the variable names used in the source code for the given parameter. The standard conditions values are used in the standard condition simulations in order to compare the various compressor and heat pump models.*

	Parameters	Local Variable	Standard conditions value	Unit
1	Evaporating temperature	T_e	10	°C
2	Condensing temperature	T_c	55	°C
3	Swept volume	V_sw	0.0006	m^3
4	Refrigerant	nref	134	-
5	Degree of superheat	T_SH	10	°C
6	Fractional pressure loss in suction line	P_loss	0	-
7	Polytropic exponent	n	1.45	-
8	Volumetric efficiency	n_vol	0.85	-
9	Minimum frequency	f_min	25	1/s

Most of the parameters are the same as for Type 206. Unlike Type 206 there are no restrictions on choice of refrigerant for this compressor model except for any restrictions in the Coolprop function. The restrictions in Type 206 are due to the external files connected to the model. There are no external files related to Type 207, hence no logical unit parameters. The polytropic exponent must be supplied by the user. Its value should exceed the value of the isentropic exponent or specific heat ratio for the refrigerant in use. The isentropic exponent for refrigerant R134a at temperature 32 °C is $\gamma = 1.251$ [45]. The average temperature of the condensing and evaporating temperatures used for the standard condition parameter values is 32.5 °C. Byrne et al. [22] use polytropic exponents that are about 18 % higher than the isentropic exponent for various refrigerants. The value of $n = 1.45$ is chosen as the standard condition value for the polytropic exponent in this model. The results of the standard conditions simulation will be compared to the standard conditions simulations of compressor Type 206. In the compressor model the volumetric efficiency is not calculated by the model but must be supplied by the user. This information may be found in compressor documentation or literature. Refer to Chapter 5.1.1 for details on the volumetric efficiency and the choice of value. For screw and scroll compressors the volumetric efficiency is close to unity. Reciprocating compressors usually have a lower volumetric efficiency due to the clearance volume. Refer to Chapter 3.3 for further information regarding the volumetric efficiency and various compressor types. Type 207 requires two inputs which are listed in Table 5.6.

Table 5.6: *Inputs to Type 207 - Polytropic compressor. The frequency is input from the iterative feedback controller Type 22. The refrigerant mass flow rate is calculated in the compressor model. The input value is added in order to connect all components in a complete cycle in TRNSYS Simulation Studio.*

	Inputs	Local Variable	Unit
1	Frequency	f	1/s
2	Refrigerant mass flow rate	m_suc	kg/h

The frequency is supplied to the model from the iterative feedback controller Type 22. The refrigerant mass flow rate is calculated within the compressor model and its value is distributed to the other components in the cycle. The input value of mass flow rate is necessary in order to connect all components in a complete cycle in TRNSYS Simulation Studio.

There are five outputs from compressor Type 207. The refrigerant mass flow rate connected as input to the condenser component, the compressor power and the isentropic efficiency are registered and printed to a text file by the plotter component Type 65. The outputs are listed in Table 5.7.

Table 5.7: Outputs from Type 207 - Polytropic compressor. The compressor power and isentropic efficiency are registered and printed to text files by the plotter component. The mass flow rate is input to the condenser component Type 230.

	Outputs	Local Variable	Unit
1	Compressor power	W_in	W
2	Isentropic efficiency	n_is	-
3	Mass flow rate	m_suc	kg/h
4	Rotational speed	rpm	1/min
5	Discharge enthalpy	h_dis	kJ/kg
6	Discharge temperature	T_dis	°C

Calculations

As for compressor Type 206 the compressor model sets various outputs to zero if the frequency provided by the iterative feedback control is lower than the minimum frequency. The first step in the calculations of Type 207 is to determine the condensing and evaporating pressures. This is achieved by calling the Coolprop utility subroutine. The known properties in both cases are the saturation temperature and the vapor quality. The pressures obtained are the saturation pressure in the evaporator and condenser. The code for these calls are stated below.

```
If (f<f_min) Then

W_in=0.0

f=0.0

rpm=0.0

m_suc=0.0

n_is=0.0

ElseIf (f>=f_min) Then


qu=1              ! Saturated vapor state

Prop(1)=T_e

Prop(5)=qu

Call coolprop_fluid_properties(prop,nref,15,iFlagr)

      If (ErrorFound()) Return

P_e=prop(2)
```

```fortran
qu=0                     ! Saturated liquid state
Prop(1)=T_c
Prop(5)=qu
Call coolprop_fluid_properties(prop,nref,15,iFlagr)
    If (ErrorFound()) Return
P_c=prop(2)
```

The next step in the calculations is to obtain the thermodynamic properties at suction state. This is done by a call to the Coolprop subroutine. The suction temperature and pressure are calculated as in Type 206. Both degree of superheating and suction line pressure loss may be accounted for here. If there is no superheating or pressure loss the refrigerant is a saturated vapor. In this case the pressure and temperature may not be used in combination as the known values, and the evaporating temperature and vapor quality is used. The code for this section is stated below.

```fortran
If ((T_SH>0.0) .or. (P_loss>0.0)) Then          ! Not saturated
    T_suc=T_e+T_SH
    P_suc=P_e-(P_e*P_loss)
    prop(1)=T_suc
    prop(2)=P_suc
Call coolprop_fluid_properties(prop,nref,12,iFlagr)
    If (ErrorFound()) Return
    h_suc=prop(3)
    s_suc=prop(4)
    v_suc=prop(6)
ElseIf ((T_SH==0.0) .and. (P_loss==0.0)) Then   ! Saturated
    qu=1.0
    prop(1)=T_e
    prop(5)=qu
Call coolprop_fluid_properties(prop,nref,15,iFlagr)
    If (ErrorFound()) Return
    P_suc=prop(2)
    h_suc=prop(3)
    s_suc=prop(4)
    v_suc=prop(6)
EndIf
```

The Coolprop subroutine returns the value of the pressure, specific enthalpy, specific entropy, and specific volume at suction conditions. The next step is to calculate the refrigerant mass flow based on

the frequency of the compressor. The mass flow rate is input to the condenser, and the mass flow rate in the cycle is controlled by the compressor frequency. The refrigerant mass flow rate is calculated as in Type 206. Refer to the previous chapter for the equations used to calculate the mass flow rate. The code for this section is stated below.

```
rpm=(f*60)/2

w=rpm/3600

m=w*V_sw*n_vol/v_suc
```

The next step is to calculate the compressor power input. The suction and discharge pressure are given in unit Pa from the Coolprop utility subroutine. To obtain the compressor power input in kW, the suction pressure in the start of the equation is divided by 1000 Pa/kPa. For the pressure ratio there are no unit conversions necessary, as the ratio is the same when both pressures are given in the same unit. The code for the calculation of the compressor power input is stated below.

```
W_in=(m_suc*(P_suc/1000)*v_suc*(n/(n-1)))*(((P_c/P_suc)**((n-1)/n))-1)
```

After the compressor power input is calculated the discharge state of the refrigerant may be found. This is cone by a call to the CoolProp subroutine, and the calculations are as in Type 206. Any heat loss or cooling of the compressor is accounted for by changing the value of the polytropic exponent, and no parameter for heat loss is included in Type 207.

```
P_dis=P_c

h_dis=h_suc+(W_in/m_suc)

prop(2)=P_dis

prop(3)=h_dis

Call coolprop_fluid_properties(prop,nref,23,iFlagr)

    If (ErrorFound()) Return

T_dis=prop(1)

v_dis=prop(6)
```

A final call to the Coolprop function is necessary, to obtain the enthalpy at discharge state in the case of ideal isentropic compression. The known properties are the discharge pressure and entropy. In the ideal case of isentropic compression, the discharge entropy is equal to the suction entropy. The code for this call is stated below.

```
P_dis=P_c

s_dis=s_suc

prop(2)=P_dis

prop(4)=s_dis

Call coolprop_fluid_properties(prop,nref,24,iFlagr)

    If (ErrorFound()) Return

h_is=prop(3)
```

The final step of the code is to calculate the isentropic efficiency. This is done as in Type 206. The mass flow rate is calculated in unit kg/s. In TRNSYS the mass flow rate is defined with unit kg/h. Before setting the mass flow rate value as output it must be converted to the correct unit. The compressor power is calculated in kW and must be converted to W before setting the output values. The code for the last two parts of the calculations is stated below.

```
n_is=(h_is-h_suc)/(h_dis-h_suc)
! Unit conversions
m=m*3600
W_in=W_in*1000
```

The complete source code for Type 207 is found in Appendix I.

5.1.3 Type 220 – Throttling device

There are various methods of throttling a fluid flow. Refer to Chapter 3.5 for further information regarding throttling and expansion devices. A source code for a throttling device is written, as there are no suitable components available in the TRNSYS or TESS component libraries. Type 220 requires four parameters which are listed in Table 5.8. The standard conditions parameter values are used for the simulations executed in order to compare the various compressor and heat pump models. The local variable is the variable name used in the source code for the given parameter.

Table 5.8: Parameters to Type 220 – Throttling device. The local variable is the local variable name used in the source code for the given parameter. The standard conditions values are used in the standard conditions simulations.

	Parameters	Local Variable	Standard conditions value	Unit
1	Refrigerant	nref	134	-
2	Evaporating temperature	T_e	10	°C
3	Condensing temperature	T_c	55	°C
4	Degree of subcooling	T_SC	8	°C

When the component is used in a vapor compression cycle the various parameters that appear in multiple components, for example the refrigerant parameter or saturation temperatures, must be given the same values. These parameters are used by the Coolprop function in each component. If the parameter values are not consistent, the results of the simulations will be incorrect without any warning or error from TRNSYS. The degree of subcooling is optional and may be set to zero. According to values provided in [8, 64, 65] the degree of subcooling for standard conditions in set to 8 °C. Pottker and Hrnjak [70] found that a degree of subcooling of $T_{SC} \approx 8\ °C$ is the value that maximizes the coefficient of performance for refrigeration cycles with various refrigerants. As for Type 207 there are no restrictions in the choice of refrigerant, expect for any restrictions in the Coolprop utility subroutine.

The input and outputs of Type 220 are listed in Table 5.9 and Table 5.10 respectively. The mass flow rate and the enthalpy of the refrigerant is constant through the throttling device. The component simply provides the values of the variables from the condenser outlet to the inlet of the evaporator.

Some additional outputs are added, for example the outlet temperature and evaporating pressure. These were used during troubleshooting and the development of the TRNSYS system model.

Table 5.9: Input to Type 220 – Throttling device. The mass flow rate is constant through the throttling device, and is also an output from the component.

	Inputs	Local Variable	Unit
1	Mass flow rate	m_suc	kg/h

Table 5.10: Outputs from Type 220 – Throttling device. The mass flow rate serves as input to the evaporator component Type 230.

	Outputs	Local Variable	Unit
1	Mass flow rate	m	kg/h
2	Evaporating pressure	P_e	kPa
3	Vapor quality at outlet	qu	-
4	Outlet temperature	T_out	°C
5	Outlet enthalpy	h_out	kJ/kg

Calculations

The first step in the calculations is to obtain the saturation pressure in the condenser. This state is prior to any subcooling of the refrigerant. The pressure is calculated by the Coolprop utility subroutine. The code is stated below.

```
qu=0                      ! Saturated liquid state
prop(1)=T_c
prop(5)=qu
Call coolprop_fluid_properties(prop,nref,15,iFlagr)
      If (ErrorFound()) Return
P_c=prop(2)
```

The next step of the code is to calculate the necessary thermodynamic properties at the inlet of the expansion valve. The two known properties are the inlet temperature and the inlet pressure. The inlet pressure is assumed to be equal to the condensing pressure. If needed a pressure loss could be added to the code at this point, similarly as in Type 206 and 207. The procedure for obtaining the necessary properties is different if there is no subcooling. If there is no subcooling in the cycle the temperature and pressure alone does not fix the state of the refrigerant. This is due to the saturation pressure and temperature being constant during the entire phase change process. If there is no subcooling the refrigerant is saturated, and the two known properties are the saturation temperature and the vapor quality. The code for this section is stated below.

```fortran
T_in=(T_c-T_SC)

If (T_SC>0.d0) Then

      prop(1)=T_in

      prop(2)=P_c

      Call coolprop_fluid_properties(prop,nref,12,iFlagr)

            If (ErrorFound()) Return

      h_in=prop(3)

      v_in=prop(6)

ElseIf (T_SC==0.d0) Then

      qu=0

      prop(1)=T_in

      prop(5)=qu

      Call coolprop_fluid_properties(prop,nref,15,iFlagr)

            If (ErrorFound()) Return

      h_in=prop(3)

      v_in=prop(6)

EndIf
```

The `If` and `ElseIf` statements check the value of the degree of subcooling, and the proper code is executed. The properties obtained are the specific enthalpy and the specific volume of the refrigerant at the inlet of the expansion valve. It is also necessary to calculate the saturation pressure in the evaporator. This pressure is calculated by the Coolprop subroutine. The known values are the evaporation temperature and vapor quality. The code is stated below.

```fortran
qu=1.0                  ! Saturated vapor state

prop(1)=T_e

prop(5)=qu

Call coolprop_fluid_properties(prop,nref,15,iFlagr)

      If (ErrorFound()) Return

P_e=prop(2)
```

When the saturation pressure in the evaporator is known, the properties at the outlet state may be calculated. The throttling process in an expansion valve is approximately isenthalpic. The outlet pressure is assumed to be equal to the saturation pressure in the evaporator. The code for the call to the Coolprop utility subroutine is stated below.

```fortran
P_out=P_e

h_out=h_in              ! Isenthalpic expansion process

prop(2)=P_out
```

```
prop(3)=h_out

Call coolprop_fluid_properties(prop,nref,23,iFlagr)

    If (ErrorFound()) Return

T_out=prop(1)

qu_out=prop(5)
```

The vapor quality output is included because it is an interesting variable in the cycle. As mentioned in Chapter 3.1.3 the vapor quality at the outlet of the expansion valve is greatly influenced by the degree of subcooling. The vapor quality affects the amount of heat gain possible in the evaporator and is therefore an important performance variable that should be optimized. At the end of the code the output values are set and control is returned to the TRNSYS kernel. The complete code for Type 220 can be found in Appendix I.

5.1.4 Type 230 – Evaporator and Condenser

The heat exchangers available in the TRSNYS libraries are not adapted to heat transfer during phase change of a substance. A component modelling an evaporator or condenser is created. It is assumed that the heat source and heat sink both have sufficient capacity to deliver and receive the required heat. The properties of the heat source and the building's heating system are not accounted for in this model. The component must not be connected to either heat source or heating system. Type 230 requires six parameters. The parameters are listed in Table 5.11. The standard conditions parameter values are used for the simulations executed in order to compare the various compressor and heat pump models.

Table 5.11: Parameters for Type 230 - evaporator and condenser. The local variables are the variable names used in the source code for the given parameter. The standard conditions values are used in the standard conditions simulations executed to compare the various compressor and heat pump models. The mode, saturation temperature and subcooling/superheat parameters are defined with two values. The first value corresponds to condenser-mode, and the second corresponds to evaporator-mode. The remaining parameters are the same for either mode.

	Parameters	Local Variable	Standard conditions value	Unit
1	Mode	mode	1 / 2	-
2	Saturation temperature	T_sat	55 / 10	°C
3	Subcooling/superheat	deltaT	8 / 10	°C
4	Refrigerant	nref	134	-
5	Efficiency	eff	1	-
6	Fractional pressure loss	P_loss	0	-

The mode parameter determines if the component is an evaporator or a condenser. The parameter is used in the code to execute the correct calculations using the proper variables, for example defining the superheat or subcooling correctly. The saturation temperature is the evaporating or condensing temperature. Most heat pump systems operate under varying conditions. The saturation temperature is usually not constant, mainly due to pressure loss due to fluid friction in the various components and pipes. For this project, the saturation temperature in both the condenser and evaporator is considered constant. It is possible to account for varying saturation temperature by changing the variable from a

parameter to a time-dependent input. The refrigerant must be the same as in the connected components, the refrigerant parameter values are found in TRNSYS documentation Vol.7 Programmer's Guide [56]. The efficiency of the condenser or evaporator is used to calculate the cooling and heating capacity. The efficiency parameter is set to one in the simulations executed in this project. The parameter is added to enable future users to account for this effect. Pressure loss is present in all real heat exchangers, and a parameter is added to make it possible to account for this loss. The pressure loss parameter is defined as a fractional loss of the saturation pressure. The pressure losses are not accounted for in this project, and the parameter value is set to zero.

The inputs and outputs of Type 230 are listed in Table 5.12 and Table 5.13 respectively. The refrigerant mass flow rate and the specific enthalpy are provided by either the compressor component or the throttling device. The local variable name is the variable name used in the source code for the given input.

Table 5.12: Inputs to Type 230 – condenser and evaporator. The refrigerant mass flow rate and the inlet specific enthalpy is provided by either the compressor component or the throttling device. The local variable is the variable name used in the source code for the given input.

	Inputs	Local Variable	Unit
1	Mass flow rate	mrefr	kg/h
2	Inlet enthalpy	h_in	kJ/kg

Table 5.13: Outputs from Type 230 – Condenser and evaporator. The heating or cooling capacity is registered and printed to a text file by the plotter component. The refrigerant mass flow rate and the outlet enthalpy serves as input to the connected component.

	Outputs	Local Variable	Unit
1	Mass flow rate	mrefr	kg/h
2	Outlet enthalpy	h_out	kJ/kg
3	Heating/cooling capacity	Q_eff	W

The refrigerant mass flow rate is equal to the inlet mass flow rate. The outlet enthalpy is used as input to other components in the model. The mode-parameter determines whether the heating or cooling capacity is calculated.

Calculations

The first step in the code is to determine the saturation pressure in the condenser or evaporator. The vapor quality is set to 0.5 at this point. The saturation temperature and the vapor quality are used to obtain the saturation pressure. As there is saturated liquid-vapor mixture present in both condenser and evaporator, the value of the vapor quality is not specified by the mode-parameter. The saturation temperature determines the saturation pressure, and the vapor quality value simply states that the refrigerant is in a saturated state.

```
qu=0.5                    ! Saturated
prop(1)=T_sat
prop(5)=qu
```

```fortran
Call coolprop_fluid_properties(prop,nref,15,iFlagr)
     If (ErrorFound()) Return
P_sat=prop(2)
```

The following code determines the refrigerant enthalpy at the outlet state. The mode parameter determines if parameter `deltaT` is degree of superheat or subcooling. When the correct outlet temperature is calculated, the outlet enthalpy is found by a call to the Coolprop utility subroutine. As for the other components the calculations of the properties at the outlet state are different depending on the values of the pressure loss and superheat or subcooling. The outlet pressure equals the saturation pressure minus the fractional pressure loss defined by the user. The pressure unit in the Coolprop utility subroutine is Pa.

```fortran
If ((deltaT/=0.0) .or. (P_loss/=0.0)) Then

     If (mode==1) Then

     T_out=T_sat-deltaT

     ElseIf (mode==2) Then

     T_out=T_sat+deltaT

     EndIf

     P_out=Psat-(P_sat*P_loss)

     prop(1)=T_out

     prop(2)=P_out

     Call coolprop_fluid_properties(prop,nref,12,iFlagr)
          If (ErrorFound()) Return

     h_out=prop(3)
Else

     If(mode==1) Then        ! condenser

       q_out=0.0

     ElseIf(mode==2) Then    ! evaporator

       q_out=1.0

     EndIf

     prop(1)=T_sat

     prop(5)=q_out

     Call coolprop_fluid_properties(prop,nref,15,iFlagr)
       If (ErrorFound()) Return

     h_out=prop(3)
EndIf
```

The next step is to calculate the heat transfer rate to or from the refrigerant and the heating or cooling capacity. To avoid any negative heat transfer rate the absolute value of the difference in specific enthalpy between inlet and outlet is used. For condenser mode, the heating capacity is the heat

transferred to the heat distribution system of the building. The heat transfer rate from the refrigerant must be multiplied with the efficiency. For the evaporator mode, the heat received by the refrigerant is smaller than the heat removed from the source. Hence, the refrigerant heat transfer rate must be divided by the efficiency to obtain the cooling capacity of the system.

```
Q_refr=(mrefr/3600.d0)*(ABS(h_in-h_out))
If(mode==1)Then
    Q_eff=Q_refr*eff         ! Heating capacity - Condenser
ElseIf(mode==2)Then
    Q_eff=Q_refr*(1/eff)     ! Cooling capacity - Evaporator
EndIf


Q_eff=Q_eff*1000
P_sat=P_sat/1000
```

Finally, the unit for the heating or cooling capacity is changed from kW to W, the pressure unit is changed from Pa to kPa, and the outputs from the component are set. The complete source code for Type 230 can be found in Appendix I.

5.1.5   Type 216/217 – Variable speed heat pumps

When connecting all components in TRNSYS to create a heat pump model on component level, the values of parameters and inputs must be consistent throughout the model. TRNSYS may give errors when running the simulations. Locating these errors can be challenging when multiple components are connected. In addition, any wrong parameters may give incorrect result without any warnings or errors from TRNSYS. Due to these issues, two heat pump models are created. In these components all necessary calculations for compressor, condenser, throttling device and evaporator are executed. The aim is to investigate if the results are different from the models on component level, and if this may be a better approach to simulations of heat pump systems in TRNSYS.

Type 216 – Based on compressor Type 206

This model is a collection of the calculations in Type 206, 220 and 230. The compressor power is calculated by the polynomial as in compressor Type 206. Thermodynamic properties are obtained by calls to the Coolprop utility subroutine. The heating and cooling capacity are calculated as in Type 230 for both condenser and evaporator. The complete source code for Type 216 is presented in Appendix I. The parameters of Type 216 and the standard condition values are presented in Table 5.14. The local variable is the variable name used in the source code for the given parameter. The standard conditions value is the parameter values used in the standard conditions simulations used to compare the heat pump models to the compressor model simulations.

Table 5.14: *Parameters of Type 216 - Variable speed heat pump. The local variable is the variable name used in the source code for the given parameter. The standard conditions value is the parameter values used in the standard conditions simulations used to compare the heat pump models to the compressor model simulations.*

	Parameters	Local Variable	Standard condition value	Unit
1	Evaporating temperature	T_e	10	°C
2	Condensing temperature	T_c	55	°C
3	Swept volume	V_sw	0.0006	m^3
4	Refrigerant	nref	134	-
5	Degree of superheat	T_SH	10	°C
6	Degree of subcooling	T_SC	8	°C
7	Fractional pressure loss in suction line	P_loss	0	-
8	Minimum frequency	f_min	25	1/s
9	Volumetric efficiency	n_vol	0.85	-
10	Fractional heat loss/cooling	r_loss	0	-
11	Efficiency	eff	1	-
12	Logical Unit 1	LU1	-	-
13	Logical unit 2	LU2	-	-
14	Logical Unit 3	LU3	-	-

Type 217 – Based on compressor Type 207

A variable speed heat pump component is created based on compressor model Type 207. The compressor is polytropic, and the power input is calculated as in Type 207. The thermodynamic properties and heating and cooling load are calculated as in Type 216. The parameters of variable speed heat pump Type 217 are presented in Table 5.15. The local variable is the variable name used in the source code for the given parameter. The standard conditions value is the parameter values used in the standard conditions simulations used to compare the heat pump models to the compressor model simulations. The complete source code for Type 217 is found in Appendix I.

Table 5.15: Parameters of Type 217 - Variable speed heat pump. The local variable is the variable name used in the source code for the given parameter. The standard conditions value is the parameter values used in the standard conditions simulations used to compare the heat pump models to the compressor model simulations.

	Parameters	Local Variable	Standard condition value	Unit
1	Evaporating temperature	T_e	10	°C
2	Condensing temperature	T_c	55	°C
3	Swept volume	V_sw	0.0006	m^3
4	Refrigerant	nref	134/290/717	-
5	Degree of superheat	T_SH	10	°C
6	Degree of subcooling	T_SC	8	°C
7	Fractional pressure loss in suction line	P_loss	0	-
8	Polytropic exponent	n	1.45	-
9	Volumetric efficiency	n_vol	0.85	-
10	Minimum frequency	f_min	25	1/s
11	Efficiency	eff	1	-

5.1.6   Type 240 – Building load

There are several alternative components for building load simulation in TRNSYS. TRNBuild may be used to create a 3D-building containing all information on building components and their thermal properties. Several building load components are available in the TRNSYS and TESS libraries. These components mostly require extensive information about the building properties and usage, for example information regarding ventilation air and internal gains. The main focus of this project is the compressor components and heat pump model and its performance. The system must be connected to a building load in order to control the frequency and capacity of the system. A simplified building load component is created. It is based on ambient temperature, desired room temperature and an overall heat loss coefficient of the building. Internal gains, solar gains and other factors are neglected. The component may be expanded to involve other parameters and variables. The component provides a varying heat demand that may be connected to the feedback control. The building load is not connected to the heat pump system directly.

Type 240 requires two parameters, which are listed in Table 5.16 . The local variables are the variable names used in the source code for the given parameter. The standard condition values are used for the simulations executed in order to compare the various compressor and heat pump models.

Table 5.16: Parameters to Type 240 - Building load. The local variables are the variable names used in the source code for the given parameter. The standard conditions values are used in the standard conditions simulations executed in order to compare the various heat pump and compressor models.

	Parameters	Local Variable	Standard condition value	Unit
1	Room temperature	T_room	20	°C
2	Overall heat loss coefficient	UA	3000	W/K

The overall heat loss coefficient UA_{max} is independent of the geometric properties and size of the building. The value is chosen to fit the maximum heating capacity of the compressor or heat pump system. Based on the desired room temperature T_{room} , the dimensioning ambient temperature DUT (DUT=-12 °C in Bergen, Norway) and the maximum heating capacity $\dot{Q}_{heat,max}$ of Bitzer compressor 4NE-20-F4Y (about 80 kW) the maximum overall heat loss coefficient is calculated by Eq. 5-9.

$$UA_{max} = \frac{\dot{Q}_{heat,max}}{T_{room} - DUT} = \frac{80 \text{ kW}}{(20 - (-12)°C} = 2,5 \text{ } kW/_{°C} \quad (5-9)$$

The dimensioning ambient temperature is defined as the lowest average temperature that occurred over a period of three days from year 1971 to 2001 [71]. According to Norwegian building standards, the heat demand of a building must be calculated for the dimensioning ambient temperature at the relevant location. The building's heating system is dimensioned and designed accordingly. The dimensioning ambient temperature occurs quite seldom, and the overall heat loss coefficient is set to 3 kW/°C at standard conditions in order to achieve maximum frequency value from the controller at somewhat higher temperatures than DUT.

The input and output of Type 240 are presented in Table 5.17 and Table 5.18 respectively. The ambient temperature is provided by the data reader component. The temperature data is provided by the Norwegian Meteorological Institute [72]. The temperature data is obtained at measuring station Florida in Bergen. The temperatures are for year 2017 and are given on an hourly basis.

Table 5.17: Input to Type 240 - Building load. The ambient temperature is provided by the data reader component.

	Inputs	Local Variable	Unit
1	Ambient temperature	T_amb	°C

Table 5.18: Output from Type 240 - Building load. The heat demand output serves as the setpoint for the iterative feedback control Type 22.

	Outputs	Local Variable	Unit
1	Heat demand	Q_dem	W

The heat demand of the building serves as the setpoint for Type 22, the iterative feedback controller. The iterative feedback control is further introduced in the next section. Th calculations in the code for Type 240 are very simple. The current heat demand is calculated based on the temperature difference between the ambient and the desired room temperature. If the temperature difference is negative, the heat demand of the building load is set to zero. The code for the calculations in Type 240 is stated below.

```
If ((T_room-T_amb)<0.d0) Then

    Q_dem=0.d0

Else

    Q_dem=UA*(T_room-T_amb)

EndIf
```

5.2 Available TRNSYS components

The remaining components necessary for the vapor compression cycle and its load and controls are found in the available TRNSYS component libraries. The components and their properties are presented in the following sections. The components from TRNSYS component library used are the iterative feedback control, the data reader and the plotter.

5.2.1 Type 22 - Iterative feedback controller

The iterative feedback control is used in the TRNSYS models to control the compressor speed. Type 22 is found in TRNSYS controllers library. The iterative feedback controller is used to make a controlled variable y track a setpoint y_{set}. A tracking error e is calculated as shown in Eq. 5-10.

$$e = y_{set} - y \qquad (5\text{-}10)$$

The controller aims to minimize the tracking error to be within the given tolerance parameter value. The controller uses a secant method to minimize the tracking error. The first iteration the controller sets the control signal u to a value within the allowed range (u_{min}, u_{max}) set by the user. The controller saves the outputs and the tracking error e the control signal provided when sent to the controlled components. The controller uses a (u,e) plane to iterate and calculate the proper control signal. The second iteration another value of the control signal is chosen by the controller and sent to the controlled components. When the second tracking error is calculated the controller extrapolates between these two points in the (u,e) plane and calculates the control signal that gives a tracking error of zero. This third value of the control signal is sent to the controlled components. In the next iteration step the controller interpolates between the second and third point, the next iteration between the third and fourth. The controller keeps interpolating between points until the tracking error is within the tolerance, or when the variation in the control signal output is within the global TRNSYS tolerances.

There are several possibilities to adjust the controller. The control signal is in this case the frequency that is given as input to the compressor component. Both the controlled variable and the setpoint are time-dependent inputs. Other inputs are necessary to adjust the control signal. The parameters and the values used in the various simulations for Type 22 are listed in Table 5.19, along with a description of each parameter.

Table 5.19: Parameters of iterative feedback control Type 22. The function of the parameters and the value used in the various simulations are described.

	Parameters	Description	Value	Unit
1	Mode	Controller mode. This parameter is not used.	0	-
2	Number of oscillations permitted	The maximum number of oscillations permitted in one timestep before the output is fixed.	0	-

The mode parameter is not yet in use and should not be changed. The second parameter controls the number of oscillations of the control signal permitted in one timestep before the output is fixed. If a large number of oscillations are permitted the simulation will be slower. The default value of zero allows the controller to do a few iterations until the global number of permitted iterations is reached. Both parameters are set to the default value. The inputs and outputs for Type 22 are listed in Table 5.20 and Table 5.21 respectively, along with descriptions and initial values of the inputs.

Table 5.20: *Inputs for the iterative feedback controller Type 22. The function of each input and the values used in the various simulations are described.*

	Inputs	Description	Value	Unit
1	Setpoint	The setpoint for the controlled variable. The controller will calculate the control signal minimizing the difference between the setpoint and the controlled variable.	100	W
2	Controlled variable	The controlled variable that should match the setpoint.	100	W
3	On/Off signal		1	-
4	Minimum control signal	The minimum value of the frequency.	0	Hz
5	Maximum control signal	The maximum value of the frequency.	85	Hz
6	Threshold for non-zero output	The minimum control signal that will give a non-zero output.	0	-
7	Tolerance on tracking error	The difference between the setpoint and the controlled variable that is acceptable.	0.01	W

Table 5.21: *Outputs of the iterative feedback controller Type 22. The function of each output is described.*

	Outputs	Description	Unit
1	Control signal	The controller output. Used as input to other components for component control.	Hz
2	Tracking error	The calculated difference between the setpoint and the controlled variable.	W
3	Controller status	Various values for controller's status. Refer to TRNSYS documentation for further information.	-
4	Unsaturated control signal	The control signal calculated before taking boundaries into account. Should not be used as output value to other components.	Hz

The simulation time step affects the iterative feedback controller. At smaller time steps the oscillations are large. Increasing the simulation time step in the global simulation settings may improve this. The tolerance may also be adjusted to reduce oscillations. A larger tolerance gives less oscillations. In addition, the component order may influence the controller. The controller should be called before the components it controls. The heat demand $\dot{Q}_{\mathrm{dem}}$ of the building load component is the setpoint for the iterative feedback controller. The heating capacity $\dot{Q}_{\mathrm{heat}}$ of the condenser is the controlled variable. The values stated for these inputs are the initial values. The remaining inputs are not connected to other components, and the chosen initial value is used throughout the simulation. The controller should regulate the system so that the heating capacity of the system matches the heat demand in the building. The maximum control signal is set to 85 Hz, as the maximum frequency of reciprocating compressor model 4NE-20.F4Y is 87 Hz [59]. The minimum control signal is set to zero. The compressor model contains a parameter to set the minimum frequency and all outputs are set to

zero if the control signal from the feedback control is lower than the allowed minimum frequency. Refer to Chapter 5.1.1 and 5.1.2 for detailed description of the compressor models. The control signal output is the frequency input to the compressor and heat pump model.

5.2.2   Type 9 – Data reader

The data reader is used in simulations to read any input data required for the simulation. The data must be provided in a text file. Type 9a provides the values as time dependent variables to other components in the model. The parameters for Type 9a is listed in Table 5.22 and the function of each parameter is described.

Table 5.22: Parameters for Type 9a- Free format data reader. The function of each parameters is described.

	Parameters	Description	Unit
1	Mode	Sets the mode of the data reader. Type 9a is mode 2.	-
2	Header lines to Skip	Number of lines the data reader skips in the connected data file.	-
3	No. of values to read	Should equal number of columns in data file.	-
4	Time interval of data	Time interval used to record the data in the file. Must be an integer multiplier of the simulation time step.	-
5	Interpolate or not?	1 for interpolation, -1 if no interpolation.	-
6	Multiplication factor	Output=value*multiplication factor	-
7	Addition factor	Output=value*multiplication factor + addition factor	-
8	Average or instantaneous value?	Indicates if value in data file is an average over the time step or an instantaneous value.	-
9	Logical unit for input file	Logical unit number assigned by TRNSYS.	-
10	Free format mode	Mode dependent.	-

There are no inputs to Type 9a. The output is the value read from the data file, or values if there are more than one column in the file. The file containing the data must be connected through the External Files tab of the Variable Window by specifying the path to the location of the file.

5.2.3   Type 65 – Plotter

The plotter component is used to register various outputs from the simulation. Multiple plotters may be present in each simulation. Type 65 is used both with and without an external data file. Type 65d provides a graphical presentation of the outputs connected to the plotter. Type 65c provides both a graphical presentation and an external data file. The connected outputs are written to a text file. The path to the data file must be stated in the External Files tab of the variable window of Type 65c. Table 5.23 lists the parameters for Type 65.

Table 5.23: Parameters for Type 65 - Online plotter with or without file. The function of each parameter is described.

	Parameters	Description	Unit
1	No. of left-axis variables	Number of variables plotted to the left Y-axis.	-
2	No. of right-axis variables	Number of variables plotted to the right Y-axis.	-
3	Left axis minimum	Minimum value on left Y-axis.	-
4	Left axis maximum	Maximum value on left Y-axis.	-
5	Right axis minimum	Minimum value on right Y-axis.	-
6	Right axis maximum	Maximum value on right Y-axis.	-
7	Number of plots per simulation	Use -1 for monthly plots.	-
8	X-axis grid points	Number of grid points the X-axis (time) is divided into	-
9	Shut of Online w/o removing	Shut of the online plotter without removing it from project file. -1 = off	-
10	Logical unit for input file	Logical unit number assigned by TRNSYS. Mode-dependent.	-
11	Output file units	Mode-dependent.	-
12	Output file delimiter	Tabs, spaces or commas to delimit columns in output file. Mode dependent.	-

The inputs to Type 65 are the variables or outputs from other components that are to be plotted or written to a file. There are no outputs from Type 65.

6   Simulations and Results

The following chapter presents the various simulations that are conducted during this project. The various simulations are described, and the results are presented. The time step analysis conducted for compressor Type 206 in order to determine the time step of the further simulations is presented first. Results from the standard conditions simulations for compressor Type 206 and 207 are described first, followed by the standard conditions simulations of the variable speed heat pumps Type 216 and 217. Results from a simulation with changed volumetric efficiency of compressor Type 206 is presented next. The next section presents the parametric study of the polytropic exponent in compressor Type 207, followed by simulations of compressor Type 206 and 207 with changed degree of subcooling. The simulations conducted to investigate the effect of changes in evaporating and condensation temperature are described next, along with the results from the simulations. Finally, the results from the simulation of compressor Type 206 with increased overall heat loss coefficient of the building load are presented and the results from simulations of compressor Type 207 with natural refrigerants propane and ammonium. The results are mostly presented graphically and in tables.

6.1   Time step analysis

The TRNSYS system model containing Type 206 compressor, Type 230 evaporator and condenser and Type 240 Building load is simulated at various time steps to analyze the signification of changing the time step of the simulation. The simulation is run at six different time steps, starting at a simulation time step of one hour. The aim is to find the largest simulation time step that provides accurate results, in order to minimize computational time. One hour is the largest simulation time step allowed in TRNSYS simulations. Figure 6.1 shows the TRNSYS system model with compressor Type 206 used for the simulation time step analysis.
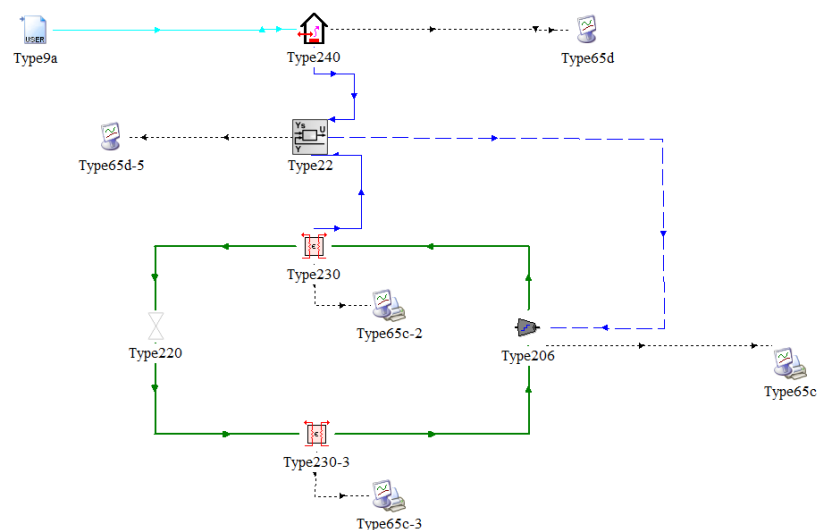
Figure 6.1: *Screenshot from SimulationStudio, TRNSYS model with compressor Type 206.The green lines connect the components of the vapor compression cycle, Type 206, 230 and 220. The dashed black lines are the connections to the online plotters Type 65c registering the output values. The blue lines show the inputs and output to and from the iterative feedback control Type 22. The light blue line is the temperature data input to the building load component Type 240.*

The simulations are executed at standard conditions, except the overall heat loss coefficient of the building which is set to 2.5 W/°C. The values and results of these simulations are not relevant to the other simulations, as the aim is to compare the results at the various simulation time step lengths. The temperature data from Bergen obtained through the Norwegian Meteorological Institute [72] is given on an hourly basis. The data reader interpolates between the data points if the time step of the simulation is shorter than the time step of the data. The data time interval must be an integer multiplier of the simulation time step. Figure 6.2 shows the hourly measured temperatures during 2017 at measuring station Florida in Bergen. The minimum temperature during this year is about −7 °C.
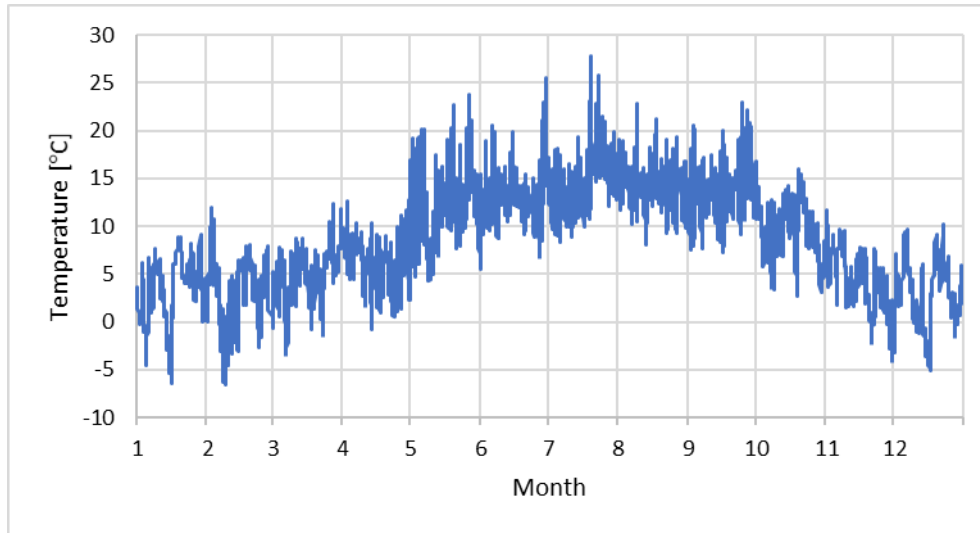
Figure 6.2: *Hourly temperatures measured at Florida, Bergen in 2017. The temperature data is provided by the Norwegian Meteorological Institute [72].The minimum temperature during the year is about -7 °C.*

During simulation the compressor power input is registered at each time step t. The energy consumption for each time step and the total energy consumption $W_{in,total}$ are calculated using Microsoft Excel. The energy consumption of a time step is calculated by Eq. 6-1.

$$W_{in} = \dot{W}_{in} \times t \ \times \frac{1\ kWh}{1000\ Wh} \tag{6-1}$$

where W_{in} is the energy consumption, $\dot{W}_{in}$ is the compressor power input during the time step, and t is the simulation time step. The compressor power input registered by TRNSYS is in watt. The total energy consumption $W_{in,total}$ is calculated by summarizing the energy consumption at each time step.

The highest time step where the total energy consumption is equal to the total energy consumption in the lower time steps is chosen as the time step for all further simulations. Table 6.1 presents the simulation time steps, the calculated total energy consumption over a year, and the difference in the calculated compressor energy consumption $W_{in,total}$ between the timesteps.

Table 6.1: *Time steps, total energy consumption $W_{in,total}$ and difference between time steps of Type 206.*

t [min]	$W_{\text{in,total}}$ [kWh]	Difference between previous and current timestep [kWh]
60	49310	-
30	48912	-398
15	48907	-5
7,5	48917	10
6	48929	11
1	48929	0

The total energy consumption of the system over a year is equal for simulation time step of 6 minutes (0.1 hour) and 1 minute. As the simulations are slower at shorter simulation time steps, the higher time step of 0.1 hour is chosen for further simulations. The remaining simulations are conducted with a higher overall heat loss coefficient of the building. This is further introduced in the following sections. During the time step analysis, the compressor did not at any point operate at maximum frequency, due to the ambient temperature and the chosen value of the overall heat loss coefficient of the building load.

6.2 Standard conditions

The first simulations are carried out for Type 206 and 207 at standard condition parameter values. The standard condition parameter values for all components are listed in Chapter 5.1.1 and 5.1.2. At standard conditions the overall heat loss coefficient in the building is set to match the heating capacity from the condenser at maximum frequency. Refer to Chapter 5.1.6 for further information on estimating the overall heat loss coefficient of the building load. The plotter components (Type 65) registers various variables, the most important being the compressor work input, isentropic efficiency of the compressor, the heat transfer rates of the condenser and evaporator, and the heat demand of the building. The control signal or frequency is also registered. The total energy consumption of the compressor $W_{\text{in,total}}$, the total heat delivered by the condenser $Q_{\text{heat,total}}$, the total heat energy received by the evaporator $Q_{\text{cool,total}}$, and the total heat demand $Q_{\text{dem,total}}$ are calculated as presented for the total energy consumption in the Chapter 6.1.

The seasonal coefficient of performance is calculated for all the standard conditions simulations. The $SCOP$ is compared to the Carnot coefficient of performance in order to check the validity and calculations of the models. The calculated seasonal coefficient of performance cannot be larger than The Carnot coefficient of performance. The Carnot coefficient of performance is calculated by Eq.3-5 as presented in Chapter 3.1.1. Taking the evaporating and condensing temperatures as the temperature of the heat source and sink respectively, the Carnot coefficient of performance is calculated as in Eq. 6-2.

$$COP_{\text{carnot,heat}} = \frac{1}{1 - \frac{(10 + 273)\text{K}}{(55 + 273)\text{K}}} = 7.29 \tag{6-2}$$

The temperature of the heat source must be higher than the evaporating temperature in order to achieve heat transfer and evaporation of the refrigerant in the evaporator. The temperature of the heat sink must be lower than the condensing temperature in order to obtain condensation of the refrigerant in the condenser. When the seasonal coefficient of performance is calculated, the Carnot efficiency η_{Carnot} may be calculated. The Carnot efficiency is introduced and described in Chapter 3.1.

6.2.1 Type 206

The TRNSYS system model with compressor Type 206 is simulated at standard conditions. The TRNSYS system model is the same as in Figure 6.1 presented in Chapter 6.1. The compressor is connected to the evaporator and condenser components (Type 230) and the iterative feedback control (Type 22). The main components of a heat pump, the compressor, condenser, throttling device and evaporator, are connected to form a complete vapor compression cycle. The iterative feedback control is connected to the condenser and the building load, comparing the heat demand of the building to the delivered heat from the condenser. The iterative feedback control is connected to the compressor, providing the frequency required to meet the heat demand of the building.

Figure 6.3 shows the frequency output from the iterative feedback control during the simulated year. The minimum and maximum frequency of 25 Hz and 85 Hz respectively are shown in the graph. The system does not reach the maximum frequency at any point. The highest frequency registered during the simulated year is 83.2 Hz. The maximum and minimum frequency corresponds to 100 % and 30 % load respectively.
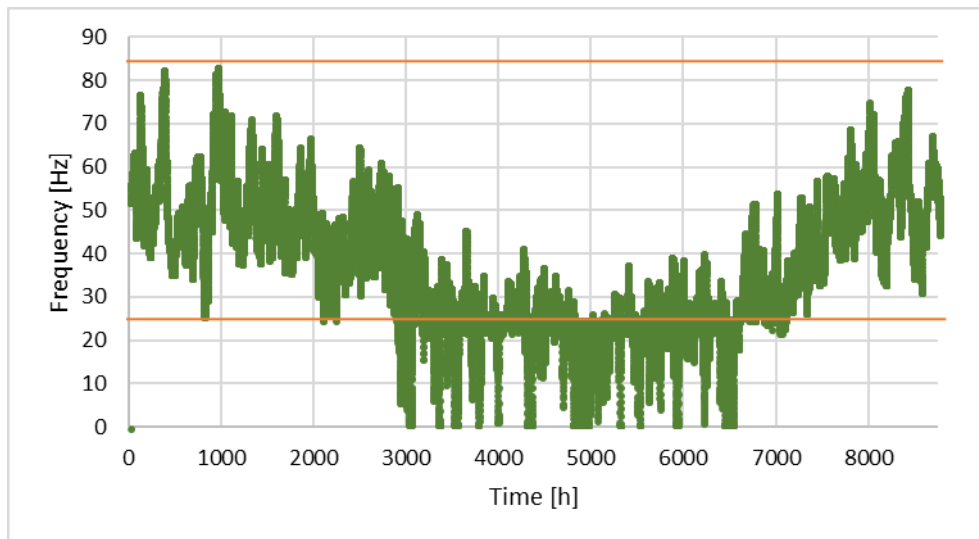
Figure 6.3: Frequency output from the iterative feedback control during the simulated year. The minimum and maximum frequency are marked by lines in the graph. The frequency does not reach the maximum frequency of $f = 85$ Hz at any point.

Figure 6.4 shows the compressor power input during 2017. At frequencies below the minimum frequency of 25 Hz the compressor is off. The compressor is of for about 2300 hours during the simulated year. The maximum power input registered at $f = 83.2$ Hz is $\dot{W}_{in} = 19.80$ kW. The maximum power input registered corresponds well to the compressor power input at maximum frequency $f = 85.0$ Hz of $\dot{W}_{\text{in,max}} = 20.3$ kW provided by Bitzer [59].
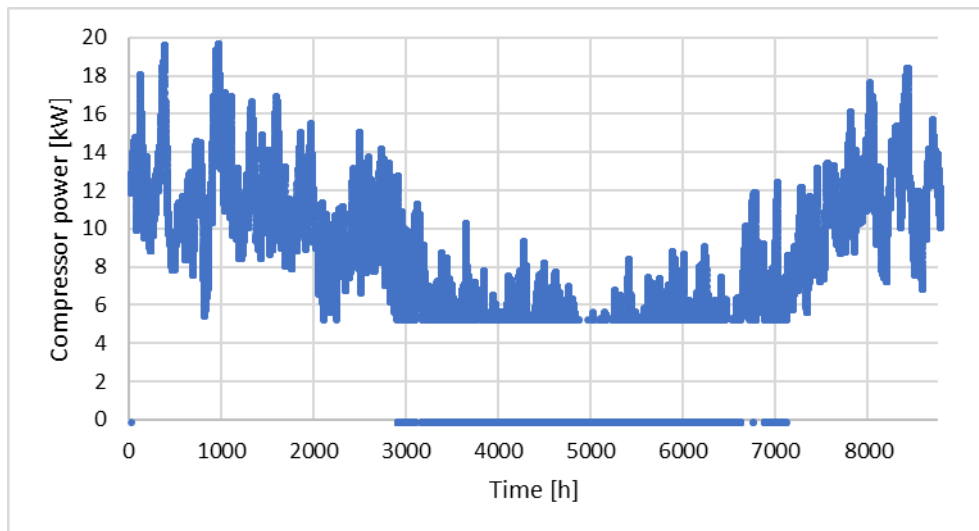
Figure 6.4: *Compressor power input during the simulated year at standard conditions simulation settings and parameters. At frequencies below 25 Hz the compressor is off. The compressor operates on part load all hours of the year.*

Figure 6.5 shows the heating capacity of the heat pump system during 2017. The heat transfer rate corresponds to the heat demand of the building at all times. The exception is during the summer months, when the heat demand is low. The heat pump system switches on and off very often during this period. The maximum heating capacity registered is $\dot{Q}_{heat} = 79.61$ kW. The minimum heating capacity registered is $\dot{Q}_{heat} = 23.34$ kW at the minimum frequency of $f = 25$ Hz.
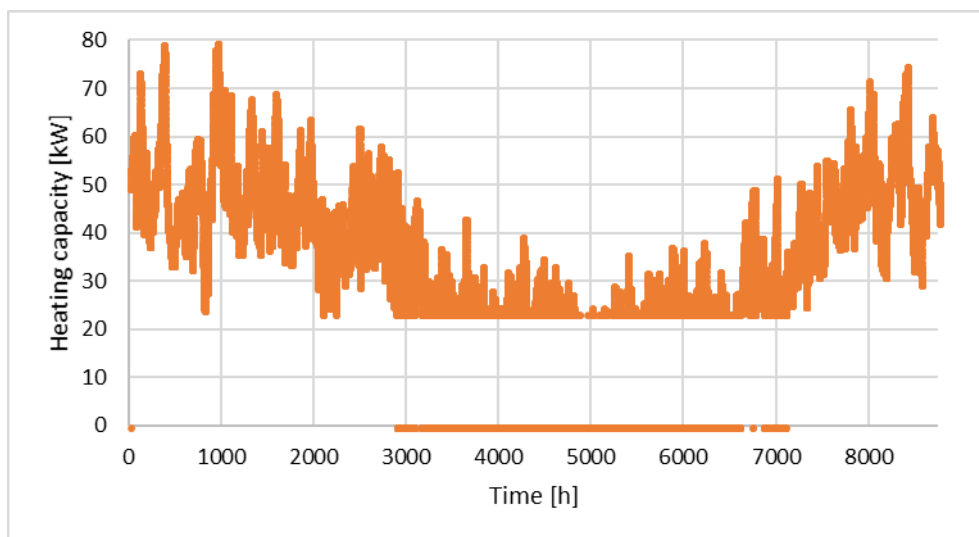
Figure 6.5: *Heating capacity or heat transfer rate of the condenser during the simulated year. The heating capacity corresponds to the heat demand of the building, except for periods with frequency below the minimum frequency due to lower heat demand.*

Figure 6.6 shows the heat demand of the building during 2017. During the summer months the building load is low, and the heat demand is not covered by the heat pump for about 2300 hours during this period due to the low frequency and load of the system. The maximum heat demand registered is $\dot{Q}_{dem} = 79.74$ kW. The horizontal, blue line marks the minimum heating capacity of the system.
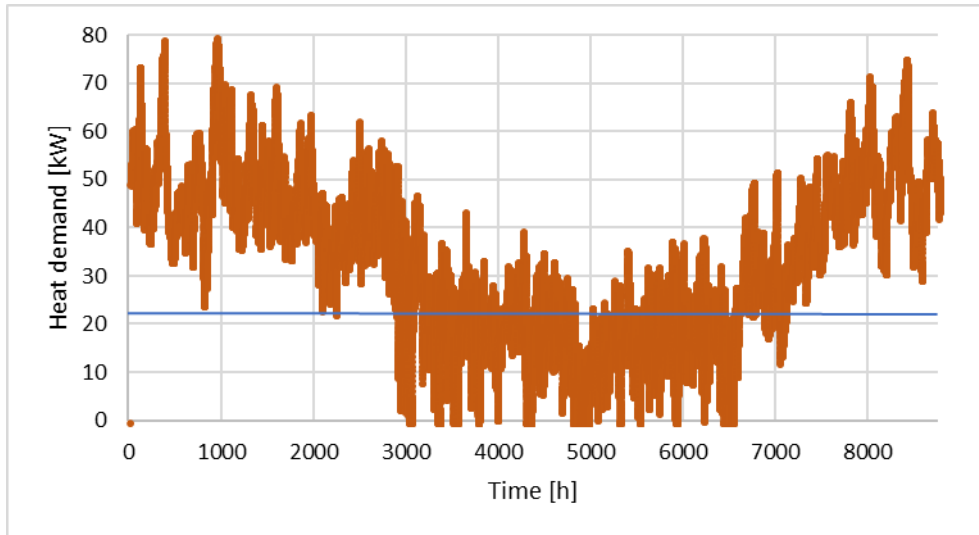
Figure 6.6: *Heat demand of building during the simulated year at standard conditions value of overall heat loss coefficient UA=3 kW/°C. During the summer months the load is too low to be covered by the heat pump system. The minimum heating capacity of the system is marked with a horizontal, blue line in the diagram.*

The total heat demand of the building and the total heat delivered by the condenser are calculated in Microsoft Excel similarly as the total energy consumed by the compressor. Refer to Chapter 6.1 for a description of the calculation of total energy consumption. The total heat delivered by the condenser $Q_{\text{heat,total}}$ equals the sum of the compressor energy consumption $W_{\text{in,total}}$ and the total heat received by the refrigerant in the evaporator $Q_{\text{cool,total}}$. This is as expected for a heat pump, refer to Chapter 3.1 for further information. The calculated total energy values are presented in Table 6.2.

Table 6.2: *Total energy consumption, heat delivered and heat demand during simulation at standard conditions of compressor Type 206.*

Component	Variable	Total heat or work [kWh]
Compressor	$W_{\text{in,total}}$	64581.91
Condenser	$Q_{\text{heat.total}}$	268365.6
Evaporator	$Q_{\text{cool,total}}$	203783.4
Building	$Q_{\text{dem,total}}$	298555.0

The coefficient of performance COP is calculated, refer to Chapter 3.1.4 for detailed information regarding COP. The isentropic efficiency η_{is} is registered as an output from the compressor model. The values of both COP and η_{is} are quite consistent throughout the simulation. This is probably due to the calculation of the discharge specific enthalpy in the compressor model. Refer to Chapter 5.1.1 for a detailed description of the calculations. Table 6.3 shows the minimum and maximum values of the calculated COP and registered η_{is}. The minimum values occur at high compressor loads, and the maximum values occur at lower compressor loads.

Table 6.3: *Minimum and maximum values of COP and η_{is} for the standard conditions simulation of compressor Type 206. The minimum values occur at maximum load, and the maximum values occur at minimum load.*

	COP [−]	η_{is} [%]
Minimum	4.020	57.8 %
Maximum	4.346	63.9 %

Figure 6.7 presents the compressor power input $\dot{W}_{in}$ and the heating capacity of the condenser $\dot{Q}_{heat}$ as functions of the load. *100* % load corresponds to the maximum frequency of *85* Hz. The minimum load is just beneath *30* %, corresponding to the minimum allowed frequency of *25* Hz. The COP is defined as the ratio of the heating capacity to the compressor power input. The values of zero compressor power and heating capacity when the load is below the minimum are not shown in the diagram.
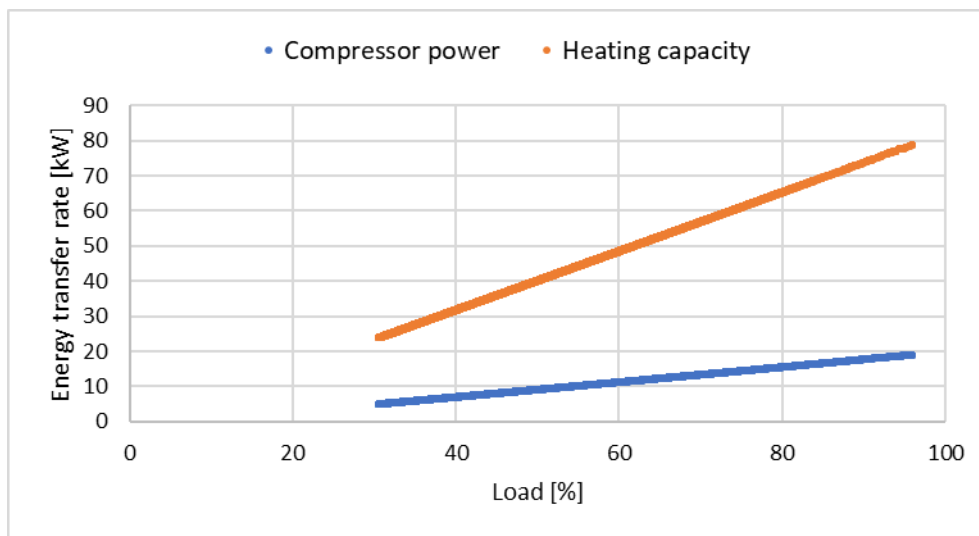
Figure 6.7: *Compressor power input and heating capacity of condenser vs. load for compressror Type 206 at standard conditions. The power input and heating capacity are only presented for loads above the minimum load. At frequencies lower than the minimum frequency, the compressor and heat pump system are off. The values of compressor power and heating capacity when the system is off are not shown.*

The deficit between the heat delivered and heat demand in the building is due to the heat pump system being off during periods with low load. The seasonal coefficient of performance $SCOP$ is calculated from the total energy consumed in the compressor and the total heat delivered by the condenser. Refer to Chapter 3.1.4 for further information regarding the seasonal coefficient of performance. The calculated heat deficit, $SCOP$ and Carnot efficiency η_{carnot} are presented in Table 6.4.

Table 6.4: *Deficit of heat and seasonal coefficient of performance SCOP for TRNSYS system model with compressor Type 206. The deficit of heat is due to periods where the compressor and heat pump system are off due to low load.*

Variable	Value
Deficit of heat [kWh]	30189.41
SCOP [-]	4.155
η_{carnot} **[-]**	0.57

6.2.2 Type 207

The TRNSYS system model with compressor Type 207 is simulated at standard conditions. The polytropic exponent $n = 1.45$ is chosen as a value above the isentropic exponent $\gamma = 1.251$ for refrigerant R134a at 32 °C. Refer to Chapter 5.1.2 for further information regarding the value of the polytropic exponent. The TRNSYS system model is presented in Figure 6.8, showing how the components are connected in TRNSYS Simulation Studio. The set-up in TRNSYS is equal for the two compressor models. The results from this simulation are compared to the results from the simulations of Type 206 at standard conditions. The building load is equal for both simulations.
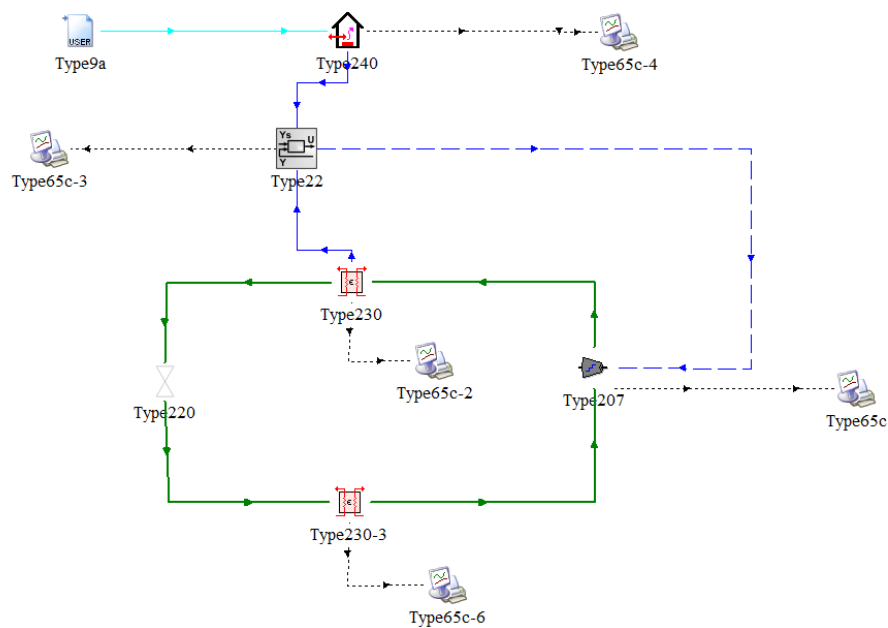
Figure 6.8: *Screenshot of TRNSYS system model with compressor Type 207. The green lines connect the components of the vapor compression cycle, Type 207, 230 and 220. The dashed black lines connect the components to the online printers Type 65c for registering output values. The blue lines connect the inputs and output to and from the iterative feedback control Type 22. The light blue line connects the temperature data to the building load Type 240.*

Figure 6.9 shows the frequency output from the iterative feedback control to the compressor model during the simulation. The highest registered frequency registered is $f = 85$ Hz. The minimum and

maximum frequency values of 25 and 85 Hz respectively are marked by orange, horizontal lines in the diagram.
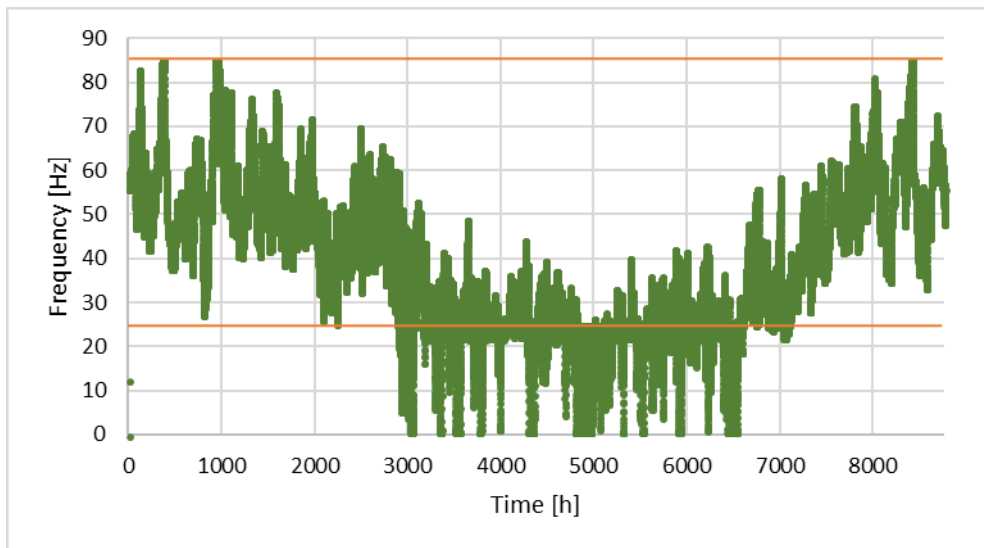
Figure 6.9: Frequency output from iterative feedback controller to compressor Type 207. The minimum and maximum frequency are marked by orange, horizontal lines in the diagram.

Figure 6.10 shows the compressor power consumption for Type 207 through the simulation. The maximum power consumption is $\dot{W}_{in} = 14.1 \text{ kW}$ at the maximum frequency of $f = 85 \text{ Hz}$. The power consumption of Type 207 at standard conditions is lower than the power consumption of Type 206, refer to Figure 6.4. The compressor is off for about 2000 hours during the simulation, when the frequency is lower than the minimum frequency set by the user.
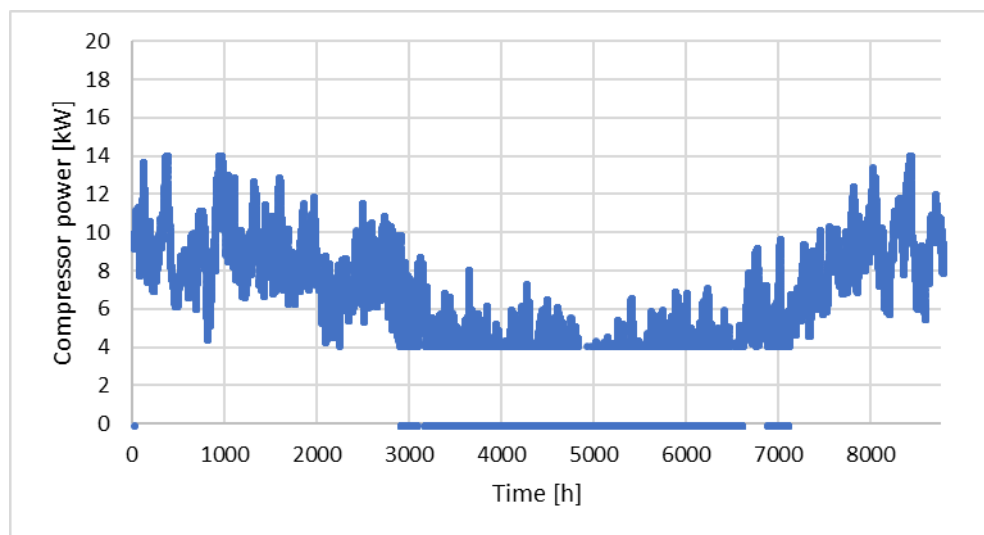
Figure 6.10: The power consumption of compressor Type 207 during simulation at standard conditions. The compressor power is generally lower than the compressor power input to Type 206.

Figure 6.11 presents the heating capacity of the condenser during the simulation. The heating capacity corresponds to the heat demand of the building, refer to Figure 6.6 in the previous section. The heat pump system is off when the frequency provided by the iterative feedback control Type 22 is lower than the minimum allowed frequency. During these periods the system cannot deliver any heat to the

building. The maximum heating capacity registered is $\dot{Q}_{\text{heat}} = 75.2 \text{ kW}$ at the maximum frequency of $f = 85 \text{ Hz}$.
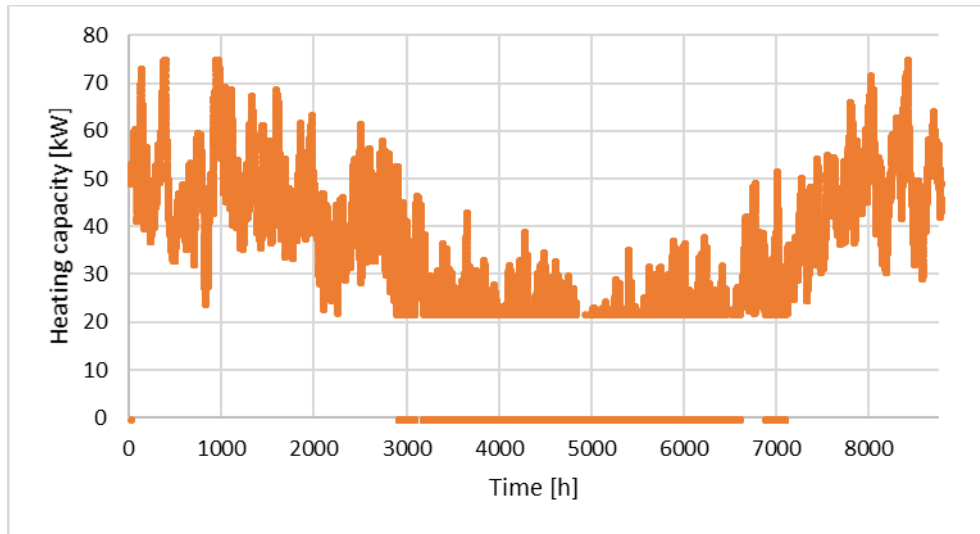
Figure 6.11: The heating capacity of the condenser during standard conditions simulation of TRNYS system model with compressor Type 207. The heating capacity corresponds to the heat demand of the building, except in periods with low load. The heat pump system is off when the frequency is below the minimum allowed value.

The total energy consumed by the compressor $W_{\text{in,total}}$, the total heat delivered by the condenser $Q_{\text{heat,total}}$, the total heat received by the evaporator $Q_{\text{cool,total}}$ and the total heat demand of the building load $Q_{\text{dem,total}}$ are calculated as for Type 206. The results are compared to the results of the standard conditions simulation of Type 206. The compressor energy consumption of Type 207 at standard condition parameter values is about 21 % lower than the required work input to compressor Type 206. The heat delivered by the condenser and the heat demand of the building are approximately equal. The total heat received by the evaporator is higher for the simulation of Type 207. The results are presented in Table 6.5.

Table 6.5: Total energy consumed by the compressor, total heat delivered by the condenser, total heat received in the evaporator and the total heat demand of the building load from simulation of Type 207 at standard conditions. The results are compared to results of Type 206 simulation.

Variable	Type 207	Type 206	% difference 206 results
$W_{\text{in,total}}$ **[kWh]**	51205.1	64581.91	-20.71 %
$Q_{\text{heat,total}}$ **[kWh]**	272587	268365.6	1.573 %
$Q_{\text{cool,total}}$ **[kWh]**	221382	203783.4	8.636 %
$Q_{dem,\text{total}}$ **[kWh]**	298605	298555.0	0.017 %

The sum of the total energy input to the compressor $W_{\text{in,total}}$ and the total heat received by the refrigerant in the evaporator $Q_{\text{cool,total}}$ is equal to the total heat delivered by the condenser $Q_{\text{heat,total}}$. This is consistent with the energy balance for a heat pump system provided in Chapter 3.1.

The coefficient of performance COP is calculated, refer to Chapter 3.1.4 for detailed information regarding COP. The isentropic efficiency η_{is} is registered as an output from the compressor model.

The isentropic efficiency is constant through the entire simulation. This is due to the calculation of the discharge specific enthalpy of the refrigerant in the compressor component. Refer to Chapter 5.1.2 for a detailed description of the calculations. The refrigerant mass flow rate is calculated based on the current frequency and is the only time-dependent variable in the compressor power input calculation. Because of this the ratio between the compressor power input and the mass flow rate is constant, which leads to a constant discharge enthalpy. The COP is almost constant through the simulation. The value for isentropic efficiency and the minimum and maximum value of COP are provided in Table 6.6. The minimum value of COP occurs at maximum load, and the maximum value of COP occurs at the minimum load.

Table 6.6: The minimum and maximum COP and η_{is} calculated for compressor type 207.

	COP [−]	η_{is} [%]
Minimum	5.323	82.7 %
Maximum	5.327	82.7 %

Figure 6.12 presents the compressor power input and the heating capacity of the condenser as functions of the system load. The minimum load is about 30 %, corresponding to the minimum frequency of $f = 25$ Hz. The diagram shows the ratio between the two variables. The ratio equals the coefficient of performance COP for the system, which is close to constant through the simulation. The values of compressor power and heating capacity of zero when the system are off is not shown in the diagram.
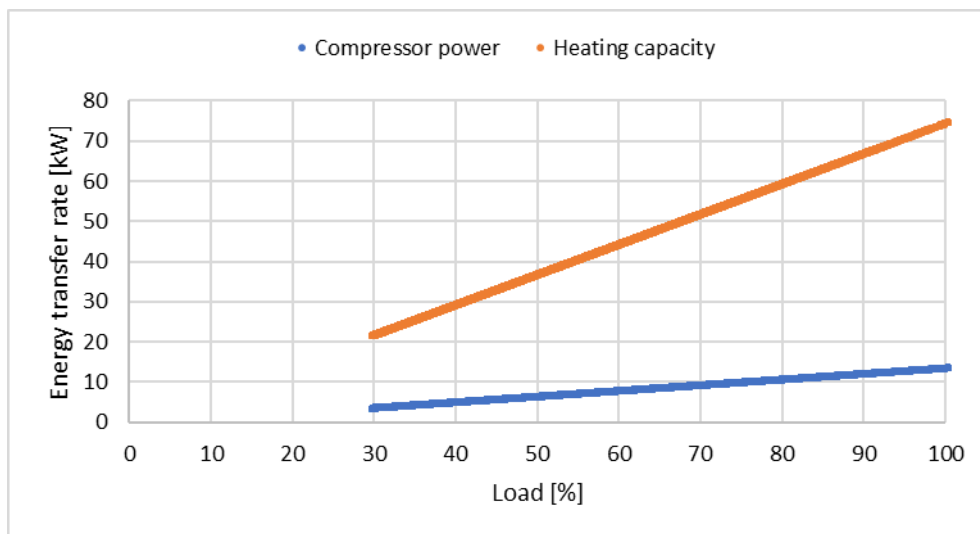
Figure 6.12: Compressor power input and heating capacity of condenser for TRSNYS system simulation of compressor Type 207 at standard conditions. The power input and heating capacity are only presented for loads above the minimum load. At frequencies lower than the minimum frequency, the compressor and heat pump system are off.

The deficit in delivered heat is smaller than for Type 206. The deficit in heat, the seasonal coefficient of performance and the Carnot efficiency are presented in Table 6.7.

Table 6.7: Deficit in heat, the seasonal coefficient of performance and Carnot-efficiency for Type 207 at standard conditions. The deficit in heat is from the periods where the compressor and heat pump system are off. The results are compared to the standard conditions results of Type 206.

Variable	Value	% difference from Type 206
Deficit of heat [kWh]	26017.3	-13.82 %
SCOP [-]	5.323	28.11 %
η_{carnot} [%]	73 %	28.07 %

6.3   Type 216/217 – Variable speed heat pumps

The variable speed heat pump models Type 216 and 217 are based on compressor model Type 206 and 207 respectively. Simulations with standard condition parameter values are conducted for both heat pump models. The standard conditions parameter values for Type 216 and 217 are listed in Chapter 5.1.5. The results are compared to the results from standard condition simulations of Type 206 and 207. The results are expected to be equal to the results from simulations of the compressor components as the heat pump models are a collection of the calculations in Type 220, Type 230, and the respective compressor models.

6.3.1   Type 216

The total energy consumption of the compressor $W_{\text{in,total}}$, the total heat delivered $Q_{\text{heat,total}}$, the total heat received by the refrigerant $Q_{\text{cool,total}}$, and the seasonal coefficient of performance $SCOP$ are calculated from the results of the simulation at standard conditions. The results are presented in Table 6.8 along with the results from the simulation of compressor Type 206 at standard conditions.

Table 6.8: Results of Type 216 at standard conditions compared to results of standard conditions simulation of Type 206 with Type 220 and 230.

	Type 216	Type 206	% difference from 206 results
$W_{\text{in,total}}$ [kWh]	64613.51	64581.91	0.05 %
$Q_{\text{heat,total}}$ [kWh]	268499.5	268365.6	0.05 %
$Q_{\text{cool,total}}$ [kWh]	203886.0	203783.4	0.05 %
$SCOP$ [-]	4.156	4.155	0.02 %

The difference between the results of the standard conditions simulations of Type 206 and 216 are very small, all less than 0.06 %. The results are consistent with the energy balance for a heat pump system presented in Chapter 3.1.

6.3.2   Type 217

The total energy consumption of the compressor $W_{\text{in,total}}$, the total heat delivered $Q_{\text{heat,total}}$, the total heat received by the refrigerant in the evaporator $Q_{\text{cool,total}}$, and the seasonal coefficient of performance $SCOP$ are calculated from the results of the simulation at standard conditions. The

results are presented in Table 6.9 along with the results from the simulation of Type 207 at standard conditions.

Table 6.9: Results of Type 217 simulation at standard conditions compared to results of standard condition simulation of compressor Type 207 with Type 220 and 230.

Variable	Type 217	Type 207	% difference from 207 results
$W_{\mathrm{in,total}}$ [kWh]	51240.7	51205.1	0.07 %
$Q_{\mathrm{heat,total}}$ [kWh]	272776	272587	0.07 %
$Q_{\mathrm{cool,total}}$ [kWh]	221535	221382	0.07 %
$SCOP$ [-]	5.323	5.323	0.00 %

The difference between the results of the standard conditions simulations of Type 207 and 217 are small, all less than 0.08 %. The total heat delivered by the condenser $Q_{\mathrm{heat,total}}$ equals the sum of the total energy input to the compressor $W_{\mathrm{in,total}}$ and the total heat received by the refrigerant in the evaporator $Q_{\mathrm{cool,total}}$ for Type 217. This is as expected based on the energy balance for a heat pump system as presented in Chapter 3.1.

6.4 Effect of the volumetric efficiency of Type 206

A simulation of compressor Type 206 with volumetric efficiency $\eta_{\mathrm{vol}} = 1.0$ is conducted. The standard conditions simulation of compressor Type 206 is conducted with a volumetric efficiency $\eta_{\mathrm{vol}} = 0.85$. This value is chosen based on results from other studies, this is elaborated in Chapter 5.1.1. The effect of the volumetric efficiency on the compressor power input is most likely accounted for in the values of the compressor coefficients. The refrigerant mass flow rate is not calculated by the polynomial and related coefficients according to ARHI Standard 540 [29]. The volumetric efficiency should be included when calculating the refrigerant mass flow rate as in compressor Type 206 and 207. The calculation of mass flow rate is presented in Chapter 5.1.1. The results of the simulation are compared to the standard conditions simulations of compressor Type 206 and 207. The total energy consumption of the compressor is expected to decrease when increasing the volumetric efficiency. The increased volumetric efficiency will result in increased refrigerant mass flow rate, and hence increased heat transfer rate in the condenser and evaporator. The system may then operate at lower load and lower frequencies. The results are presented in Table 6.10 along with results from the standard conditions simulations of compressor Type 206 and 207.

Table 6.10: The total energy consumed by the compressor, the total heat delivered by the condenser, the total heat received by the evaporator and the seasonal coefficient of performance from simulation of compressor Type 206 with volumetric efficiency $\eta_{vol} = 1.0$. The results from the standard conditions simulations of compressor Type 206 and 207 are also presented. The standard conditions value for the volumetric efficiency is $\eta_{vol} = 0.85$.

	$W_{\mathrm{in,total}}$ [kWh]	$Q_{\mathrm{heat,total}}$ [kWh]	$Q_{\mathrm{cool,total}}$ [kWh]	$SCOP$ [-]
$\eta_{\mathrm{vol}} = 1.0$	54134.3	256417	260118	4.737
$\eta_{\mathrm{vol}} = 0.85$	64581.9	268366	203783	4.155
Type 207	51205.1	272587	221382	5.323

The calculation of the compressor power input in compressor Type 206 is independent of the refrigerant mass flow rate and is calculated based on the condensing and evaporating temperatures. The increased volumetric efficiency results in increased refrigerant mass flow rate. The heating and cooling capacity of the system are calculated from the refrigerant mass flow rate, and the heat transfer rate in the evaporator increases. Due to the higher mass flow rate and resulting heat transfer rates in the system the iterative feedback control provides lower frequencies to the compressor. The average frequency of the simulation is $f_{avg} = 34.1$ Hz, while the average frequency during the standard conditions simulation of Type 206 is $f_{avg} = 37.8$ Hz. This decreases the total energy consumption of the compressor, even though the compressor power calculation is independent from the refrigerant mass flow rate. The seasonal coefficient of performance increases because the compressor power input decreases. The lower total heat energy delivered by the condenser is due to the lower frequency and load of the system, resulting in longer periods where the system is off. This also affects the total energy consumed by the compressor. Figure 6.13 shows the power input to compressor Type 206 during the simulation with increased volumetric efficiency.
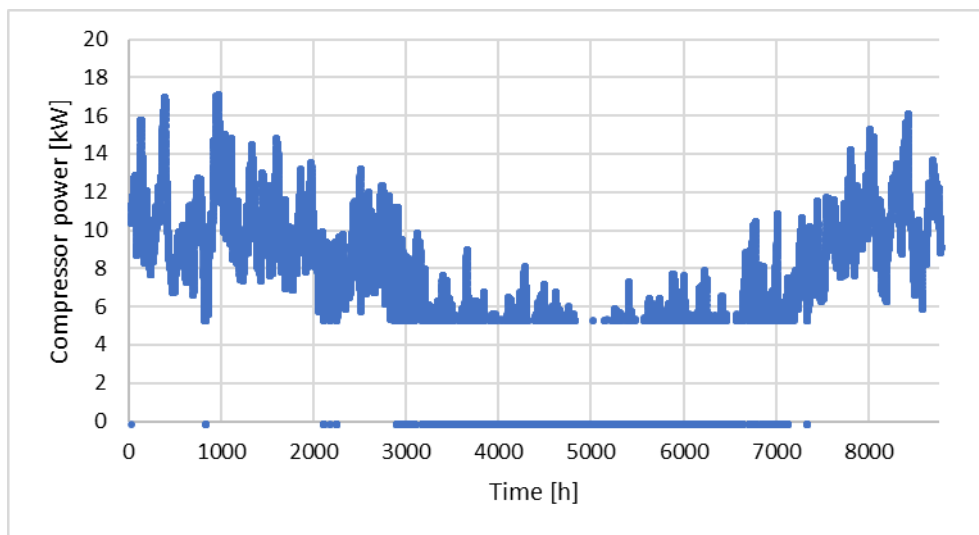
Figure 6.13: *Compressor power input during simulation of Type 206 with increased volumetric efficiency $\eta_{vol} = 1.0$. The compressor power input is generally lower than for the standard conditions simulation.*

6.5 Parametric study of polytropic exponent in Type 207

A parametric study of the polytropic exponent in compressor Type 207 is conducted due to the difference in compressor power input between compressor Type 206 and 207 at standard conditions. The aim is to find the value of the polytropic exponent that provides a compressor power input similar to the power input to compressor Type 206. The standard condition value for the polytropic exponent is 1.45, refer to Chapter 5.1.2 for further information regarding the chosen value. The values of the polytropic exponent investigated in the parametric study are listed in Table 6.11. The values are chosen as a percentage increase or decrease of the standard condition value of the polytropic exponent. The total energy consumption $W_{in,total}$ of the compressor, the total heat $Q_{heat,total}$ delivered from the condenser and the seasonal coefficient of performance $SCOP$ are calculated for each case and compared to the standard condition value. The aim is to investigate if the compressor power input follows the same percentage increase or decrease as the value of the polytropic exponent.

Table 6.11: *Values of polytropic coefficient for parametric study and the percentage change from the standard condition value.*

n [-]	Change from standard condition value [%]
1.305	- 10
1.45	–
1.595	+ 10
1.74	+ 20
1.885	+ 30

The total energy consumption of the compressor $W_{\text{in,total}}$ during the simulation is calculated. The correlation between the polytropic exponent and the total energy consumption of the compressor is calculated in Excel. The correlation is $c = 0.996$, which means the two variables are closely correlated and increasing one variable induces an increase of the other variable. The delivered heat $Q_{\text{heat,total}}$ and the seasonal coefficient of performance $SCOP$ are correlated to the polytropic exponent with correlations of $c = -0.996$ and $c = -0.992$ respectively. An increase of the polytropic exponent induces a decrease in the delivered heat of the condenser and $SCOP$. The total energy consumption $W_{\text{in,total}}$, the total heat delivered $Q_{\text{heat,total}}$ and the seasonal coefficient of performance $SCOP$ are presented in Table 6.12 along with the percentage difference to the results of the simulation with standard conditions value of $n = 1.45$. The total heat delivered $Q_{\text{heat,total}}$ is approximately constant during the simulations in the parametric study of the polytropic exponent. The building load heat demand $Q_{\text{dem,total}}$ is the same for all simulations of compressor Type 207.

Table 6.12: *Values of polytropic exponent n, total compressor energy consumption $W_{in,total}$, the total heat delivered $Q_{in,total}$ and the calculated SCOP for the parametric study of compressor Type 207. The percentage difference from the results of simulation with standard conditions value $n = 1.45$ are also presented.*

n	$W_{\text{in,total}}$ [kWh]	% of standard conditions $W_{\text{in,total}}$	$Q_{\text{heat,total}}$ [kWh]	% of standard conditions $Q_{\text{heat,total}}$	$SCOP$ [-]	% of standard conditions $SCOP$
1.305	49237.2	96.16 %	273449	100.3 %	5.554	104.3 %
1.45	51205.1	100.0 %	272587	100.0 %	5.323	100.0 %
1.595	52911.8	103.3 %	272034	99.80 %	5.141	96.58 %
1.74	54368.6	106.2 %	271502	99.60%	4.994	93.82 %
1.885	55597.4	108.6 %	270864	99.37 %	4.872	91.53 %

The total energy consumption shows a linear behavior with the polytropic exponent. A ten percent increase or decrease in the polytropic exponent gives about three percent increase or decrease in the total energy consumption. The total energy consumption $W_{\text{in,total}}$ is presented graphically as a function of the polytropic exponent in Figure 6.14.
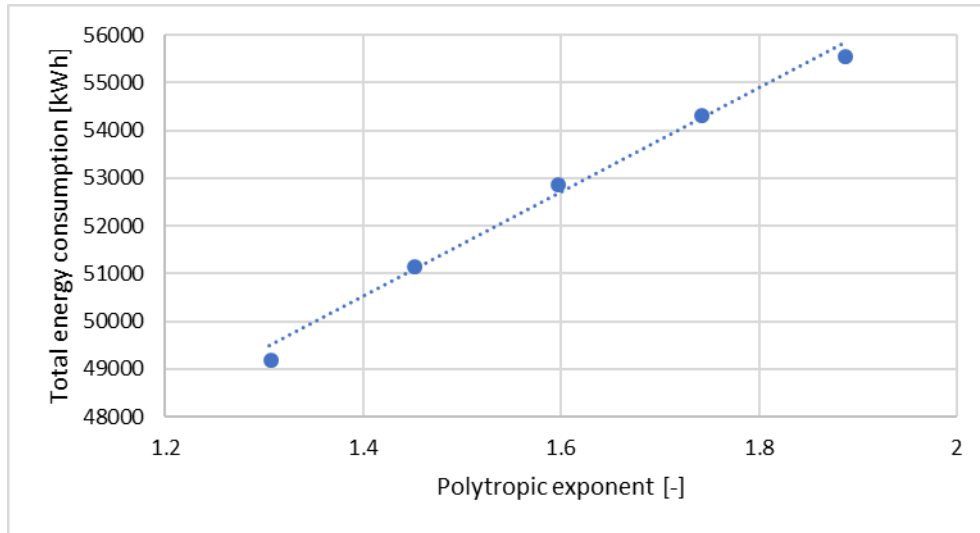
Figure 6.14: Total energy consumption $W_{in,total}$ of compressor Type 207 as a function of the polytropic exponent n. The blue, dashed line is the linear trendline produced by Excel and presented in Eq.6-3. The standard conditions value of the polytropic exponent is n=1.45.

A trend line is added to the diagram and provides a linear function for the total energy consumption as a function of the polytropic exponent. The function is presented in Eq. 6-3.

$$W_{in,total} = (10954n + 35192) \text{ kWh} \qquad (6\text{-}3)$$

The total energy consumption of compressor Type 206 is larger than for Type 207 at any tested polytropic exponent. The linear function presented in Eq. 6-3 may be used to calculate the value of the polytropic exponent required to obtain the same energy consumption for Type 207 as for Type 206 at standard conditions. The required value of the polytropic exponent is calculated by Eq. 6-4.

$$n = \frac{(W_{in,total} - 35192) \text{ kWh}}{10954 \text{ kWh}} \qquad (6\text{-}4)$$

The total energy consumption of compressor Type 206 during the standard conditions simulation is $W_{in,total} = 64581.91$ kWh. Inserting into Eq. 6-4 yields $n = 2.683$. A final simulation is conducted, with polytropic exponent $n = 2.683$. The results are compared to the results of the standard condition simulation of compressor Type 206. The total energy consumed by the compressor, the total heat delivered by the condenser, the total heat received in the evaporator, and the seasonal coefficient of performance are presented in Table 6.13 for both compressor Type 207 with polytropic exponent $n = 2.683$ and compressor Type 206 at standard conditions. The results are expected to be approximately equal.

Table 6.13: *The total energy consumed by the compressor, the total heat delivered by the condenser, the total heat received in the evaporator, and the seasonal coefficient of performance for compressor Type 207 with polytropic exponent $n = 2.683$. The results are compared to the standard conditions results of compressor Type 206. The results are expected to be approximately equal*

Variable	$n = 2.683$	Type 206	% difference of 206 results
$W_{\text{in,total}}$ [kWh]	61029.2	64581.9	5.50 %
$Q_{\text{heat,total}}$ [kWh]	268574	268366	-0.08 %
$Q_{\text{cool,total}}$ [kWh]	207545	203783	-1.85 %
$Q_{\text{dem,total}}$ [kWh]	298605	298555	-0.02 %
$SCOP$ [$-$]	4.401	4.155	-5.92 %

6.6 Effect of subcooling

The coefficients obtained from Bitzer software for compressor Type 206 are valid only for certain values of superheat and subcooling. The standard condition parameter values for degree of superheat T_{SH} and degree of subcooling T_{SC} is 10 °C and 8 °C respectively. The coefficients obtained for the compressor are valid for $T_{\text{SH}} = 10\,°\text{C}$ and $T_{\text{SC}} = 0\,°\text{C}$, corresponding to AHRI Standard 540 test conditions [29]. A simulation of compressor Type 206 is carried out with degree of subcooling $T_{\text{SC}} = 0\,°\text{C}$ and the remaining parameters at standard condition value. The results are compared in terms of the total energy consumption of the compressor, the total heating and cooling energy and the seasonal coefficient of performance $SCOP$. The simulation with degree of subcooling $T_{\text{SC}} = 0\,°\text{C}$ is also conducted with polytropic compressor Type 207. The polytropic compressor component does not have limitations on either saturation temperatures or degree of superheat and subcooling. The results of the two simulations at AHRI conditions are compared to the standard conditions simulations. The aim is to investigate if the two models are equally sensitive to change in the degree of subcooling.

Decreasing the degree of subcooling is assumed to increase the vapor quality at the outlet of the throttling device and inlet of evaporator. The minimum degree of subcooling possible is 0 °C. The cooling capacity, or heat transferred to the refrigerant in the evaporator is expected to decrease [70]. Refer to Chapter 3.1.3 and 5.1.3 for further information regarding vapor quality and the throttling device. The decreased heat transfer rate in the evaporator will decrease the heating capacity of the system if the compressor power input is constant. The result is decreased COP and $SCOP$. The outlet conditions of the evaporator are the same as for the standard conditions simulations, as the heat source is assumed to provide sufficient energy regardless of operational conditions. The results of the simulations of compressor Type 206 and 207 are presented in Table 6.14 and Table 6.15 respectively.

Table 6.14: Comparison of compressor power input, heating capacity, cooling capacity, and SCOP of TRSNYS model with compressor Type 206 at standard conditions with $T_{SC} = 8\ °C$ and the simulation with $T_{SC} = 0\ °C$, corresponding to the AHRI Standard 540 [29] test conditions.

Variable	Component	$T_{SC} = 0°C$	$T_{SC} = 8\ °C$	% difference from standard conditions
$W_{in,total}$ [kWh]	Compressor	70653.6	64581.9	9.4 %
$Q_{heat,total}$ [kWh]	Condenser	273796	268366	2.0 %
$Q_{cool,total}$ [kWh]	Evaporator	203142	203783	-0.31 %
$SCOP$ [−]	−	3.88	4.16	-6.74 %

Table 6.15: Comparison of total energy consumption of compressor, total heat delivered, total heat received and SCOP of compressor Type 207 at standard conditions with $T_{SC} = 8\ °C$ and the simulation with $T_{SC} = 0\ °C$, corresponding to the AHRI Standard 540 [29] test conditions.

Variable	Component	$T_{SC} = 0°C$	$T_{SC} = 8°C$	% difference from standard conditions
$W_{in,total}$ [kWh]	Compressor	56027.3	51205.1	9.42 %
$Q_{heat,total}$ [kWh]	Condenser	277523	272587	1.81 %
$Q_{cool,total}$ [kWh]	Evaporator	221495	221382	0.05 %
$SCOP$ [−]	−	4.953	5.323	-5.81 %

The changes in the total energy consumption of the compressor and the total heat delivered by the condenser are about equal for compressor Type 206 and 207. The change in total heat received in the evaporator is small for both simulations but increases for Type 207 while decreasing for Type 206.

6.7   Effects of changes in evaporating temperature

The effect of changing the evaporating temperature is investigated. The coefficients for compressor Type 206 have a limited valid temperature range and the valid temperatures are better suited for refrigerating than heating purposes. Refer to Chapter 5.1.1 for further information regarding the coefficients and their validity range. Simulations are conducted for five different evaporating temperatures T_e. The temperatures simulated are both within and outside the valid range of $T_e = -25 - 12.5\ °C$. The condensing temperature has a much larger range of valid temperatures and is kept at the standard conditions parameter value of $T_c = 55\ °C$. The aim of these simulations is to investigate how sensitive compressor Type 206 is to changes in evaporating temperatures, and the impact of using temperatures outside the validity range of the coefficients. The simulations are also conducted with compressor Type 207 in order to compare the change in the total energy consumption of the compressor models. The evaporating temperatures used in these simulations are listed in Table 6.16.

Table 6.16: Evaporating temperatures tested in parametric study of compressor Type 206 and 207.

Validity	T_e [°C]
Inside range	0
Inside range	5
Inside range	10
Outside range	15
Outside range	20

The total energy consumption of the compressor is expected to decrease with increasing evaporating temperature. This is due to the lower specific volume of the suction gas at higher temperatures, which results in a lower work input required by the compressor. The seasonal coefficient of performance is expected to increase when increasing the evaporating temperature due to lower compressor work input and higher heat transfer rate in the evaporator. The performance of the system and the compressor efficiency and performance are closely related to the temperature difference between the heat source and sink, and hence the necessary evaporating and condensing temperatures. The remaining parameters are kept at the standard condition values. The results of the parametric study of Type 206 is presented in Table 6.17. The standard conditions simulation results of compressor Type 206 are provided for evaporating temperature $T_e = 10\ °C$.

Table 6.17: The calculated total energy consumption of the compressor, total heat delivered by the condenser, total heat received by the evaporator and the seasonal coefficient of performance for Type 206. The correlations between the variables and the evaporating temperature is also presented.

Evaporating temperature [°C]	$W_{in,total}$ [kWh]	$Q_{heat,total}$ [kWh]	$Q_{cool,total}$ [kWh]	$SCOP$ [-]
0	80625.0	284548	203923	3.530
1	79392.0	284160	204767	3.579
5	73462.1	279535	206072	3.805
10	64581.9	268366	203783	4.155
15	55248.2	254182	198933	4.601
20	45575.3	235708	190133	5.172
Correlation	-0.999	-0.982	-0.862	0.991

The total energy consumption of the compressor shows a linear behavior with the changing evaporating temperature. The correlation between the two variables is calculated to $c = -0.999$, meaning that the compressor energy consumption decreases linearly as the evaporating temperature increases. The total energy consumption $W_{in,total}$ is presented as a function of the evaporating temperature in Figure 6.15. The blue dashed line is a linear trend line produced by Microsoft Excel.
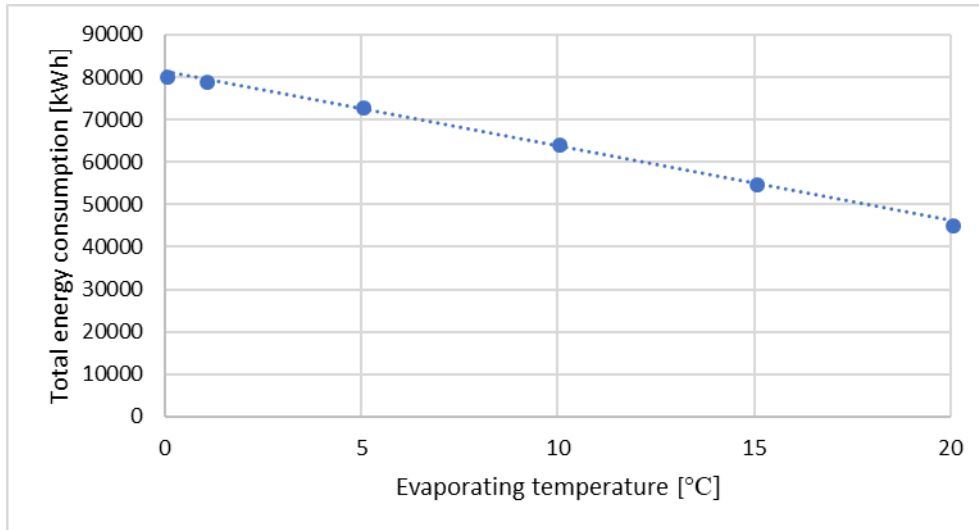
Figure 6.15: The total energy consumption of compressor Type 206 presented as a function of the evaporating temperature. The total energy consumption shows a linear behavior with the evaporating temperature. The dashed blue line is a linear trend line produced by Microsoft Excel.

The total heat received in the evaporator $Q_{cool,total}$ does not show a linear behavior at the lower evaporating temperatures. An extra simulation with evaporating temperature $T_e = 1°C$ is conducted, to investigate if the evaporating temperature of $T_e = 0°C$ is not possible to use in compressor model Type 206. The results from the simulation of evaporating temperature $T_e = 1°C$ is consistent with the previous results, and the evaporating temperature of $T_e = 0°C$ is considered valid. The correlation between $Q_{cool,total}$ and the evaporating temperature is calculated to $c = -0.862$. The total heat received in the evaporator is presented as a function of the evaporating temperature in Figure 6.16. The difference between the highest and lowest total heat $Q_{cool,total}$ is about 7.5 % of the maximum value. The blue, dashed line is the linear trend line produced by Microsoft Excel.
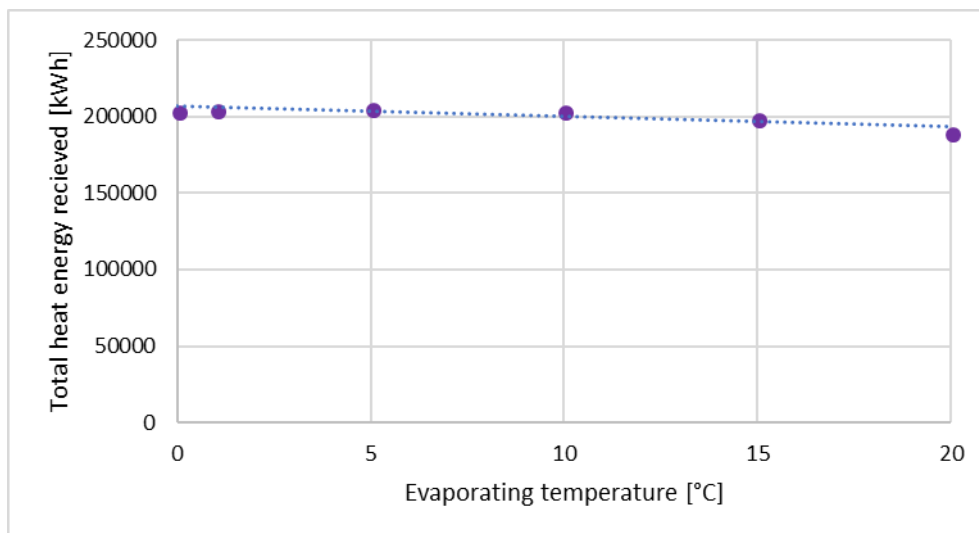
Figure 6.16: The total heat received by the evaporator during simulations of compressor Type 206 as a function of the evaporating temperature. The total heat does not have a linear relationship with the evaporating temperature. The blue, dashed line is the linear trend line produced by Microsoft Excel.

The results from the parametric study of the evaporating temperature simulations of compressor Type 207 are presented in Table 6.18. The standard conditions simulation results of compressor Type 207 are provided for evaporating temperature $T_e = 10\ °C$. The total energy consumed by the compressor is completely correlated to the evaporating temperature. The correlations calculated for the total heat delivered and the seasonal coefficient of performance are approximately equal to the correlations calculated for the results from compressor Type 206. The total heat received in the evaporator is not consistent with the results from simulations of compressor Type 206. The heat received by the evaporator $Q_{cool,total}$ is poorly correlated to the evaporating temperature.

Table 6.18: The calculated total energy consumption of the compressor, total heat delivered by the condenser, total heat received by the evaporator and the seasonal coefficient of performance for Type 207. The correlations between the variables and the evaporating temperature is also presented.

Evaporating temperature [°C]	$W_{in,total}$ [kWh]	$Q_{heat,total}$ [kWh]	$Q_{cool,total}$ [kWh]	$SCOP$ [-]
0	68975.0	283659	214684	4.112
1	67412.8	283933	216520	4.212
5	60555.2	281745	221190	4.653
10	51205.1	272587	221382	5.323
15	42154.4	260276	218122	6.174
20	33392.1	243211	209819	7.283
Correlation	-1.000	-0.967	-0.346	0.991

The total energy consumption of compressor Type 207 is presented as a function of the evaporating temperature in Figure 6.17. The blue, dashed line is the trendline produced by Excel. The total energy consumption $W_{in,total}$ is completely correlated to the evaporating temperature and shows a linear behavior with the changing evaporating temperature.
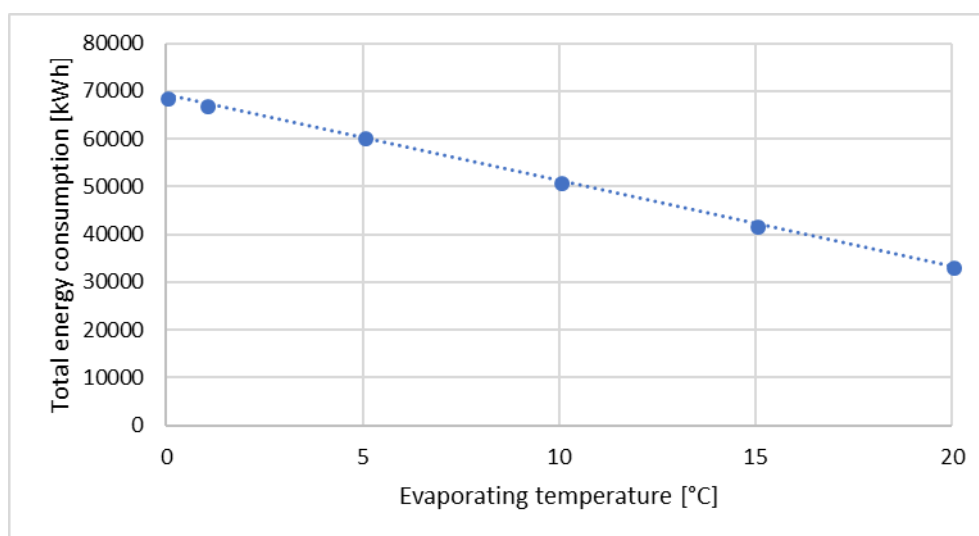
Figure 6.17: The total energy consumption of compressor Type 207 as a function of the evaporating temperature. The blue, dashed line is the trendline produced by Excel. The total energy consumption $W_{in,total}$ shows a linear behavior with the evaporating temperature, and the two variables are completely correlated.

The total heat received in the evaporator $Q_{\text{cool,total}}$ is not closely correlated with the evaporating temperature. Figure 6.18 presents the total heat received in the evaporator as a function of the evaporating temperature. The total heat received by the evaporator does not change much during the various simulations and does not show a linear relationship with the evaporating temperature. The difference between the highest and lowest total heat received $Q_{\text{cool,total}}$ is about 5 % of the maximum value.
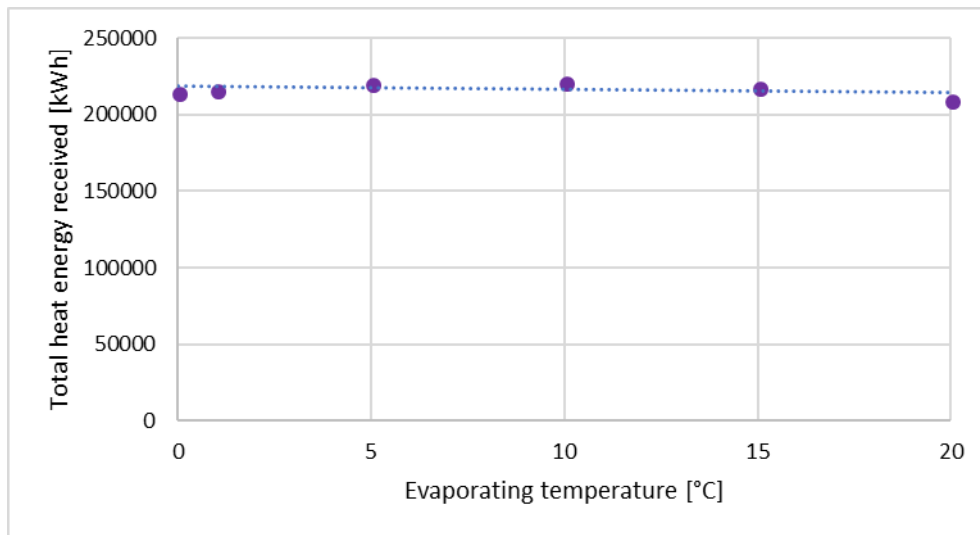
Figure 6.18: The total heat received by the evaporator in the simulations of compressor Type 207 presented as a function of the evaporating temperature. The total heat received in the evaporator is not closely correlated to the evaporating temperature. The difference between the highest and lowest total heat $Q_{cool,total}$ is about 5 % of the maximum value.

6.8   Effects of changes in the condensing temperature

A parametric study of the condensing temperature is conducted. The validity range of the condensing temperature of the compressor coefficients obtained from Bitzer [59] is $T_{\text{c}} = 10\ ℃ - 80\ ℃$. All the tested condensing temperatures are within the valid range. The simulations are conducted for both compressor Type 206 and 207. The aim is to investigate if the two compressor models show the same behavior when changing the condensing temperature. Compressor Type 207 does not have limitations on the condensing temperature, except that the condensation state of the refrigerant must be below the critical point of the refrigerant. Refer to Chapter 3.6 for further information regarding the critical temperature and pressure of refrigerants. The remaining parameters, including the evaporation temperature, are kept at the standard conditions value. Table 6.19 present the condensation temperatures tested in the parametric study of compressor Type 206 and 207.

Table 6.19: Condensation temperatures tested in parametric study of compressor Type 206 and 207. All the temperatures are within the validity range of the compressor coefficients for Type 206.

Validity	T_c [°C]
Inside range	50
Inside range	55
Inside range	60
Inside range	65
Inside range	70

Increasing the condensation temperature is expected to result in a higher compressor power input, as the compressor must deliver a higher discharge pressure in order to obtain the desired saturation temperature in the condenser. The performance of both the compressor and heat pump system is highly dependent on the temperature difference between the evaporator and condenser. Due to the increased compressor power input, the seasonal coefficient of performance is expected to decrease.

Table 6.20 present the results for compressor Type 206 with the various condensing temperature values. The total energy consumed by the compressor, the total heat delivered by the condenser, the total heat received by the evaporator and the seasonal coefficient of performance are all closely correlated to the condensing temperature. The results from the standard conditions simulation is presented for $T_c = 55\ °C$.

Table 6.20: *Results of parametric study of condensing temperature of model with compressor Type 206. The total energy consumed by the compressor, the heat delivered by the condenser and received by the evaporator and the seasonal coefficient of performance are calculated. All variables are closely correlated with the condensing temperature.*

Condensing temperature [°C]	$W_{in,total}$ [kWh]	$Q_{heat,total}$ [kWh]	$Q_{cool,total}$ [kWh]	$SCOP$ [-]
50	58760.8	266389	207628	4.533
55	64581.9	268366	203783	4.155
60	70515.2	270588	200073	3.837
65	76563.6	272957	196394	3.565
70	82734.3	275388	192653	3.329
Correlation	1.000	0.999	-1.000	-0.996

The total heat received by the evaporator during the simulations of Type 206 changes even though the evaporating temperature is the same for all simulations. This is because the specific enthalpy at the inlet of the evaporator is equal to the specific enthalpy at the condenser outlet, which depends on the condensing temperature and degree of subcooling. However, the change in the total heat received by the evaporator is quite small compared to the change in the total energy consumption of the compressor. The energy consumption of compressor Type 206 during the simulations of the varying condensing temperature increases by 40 % from the lowest condensing temperature to the highest. The heat received by the evaporator decreases by about 7 %. The heat delivered by the condenser increases by only 3 %, resulting in a 27 % decrease in the seasonal coefficient of performance from the lowest to the highest condensation temperature.

Table 6.21: *Results of parametric study of condensing temperature of model with compressor Type 207. The total energy consumed by the compressor, the heat delivered by the condenser and received by the evaporator and the seasonal coefficient of performance are calculated. All variables are closely correlated with the condensing temperature.*

Condensing temperature [°C]	$W_{\text{in,total}}$ [kWh]	$Q_{\text{heat,total}}$ [kWh]	$Q_{cool,\text{total}}$ [kWh]	$SCOP$ [-]
50	44158.7	270967	226808	6.136
55	51205.1	272587	221382	5.323
60	58686.8	274164	215477	4.672
65	66673.0	275788	209115	4.136
70	75209.4	277383	202174	3.688
Correlation	0.999	1.000	-0.999	-0.993

The total heat received by the evaporator during the simulations of Type 207 changes even though the evaporating temperature is the same for all simulations. This is because the specific enthalpy at the inlet of the evaporator is equal to the specific enthalpy at the condenser outlet, which depends on the condensing temperature and degree of subcooling. However, the change in the total heat received by the evaporator is quite small compared to the change in the total energy consumption of the compressor. The energy consumption of the compressor increases by about 70 % from the lowest condensing temperature to the highest, while the heat received in the evaporator decreases by about 11 %. The total heat delivered by the condenser only increases by 2 %, resulting in a 40 % decrease of the seasonal coefficient of performance from the lowest to the highest condensing temperature.

6.9 Increased building load

Heat pump systems are usually dimensioned to cover only part of the maximum heat demand of the building. This is due to the maximum load being calculated at the dimensioning ambient temperature DUT. Refer to Chapter 5.1.6 for further information regarding the dimensioning ambient temperature. If the heat pump system is dimensioned to cover about 40 – 60 % of the maximum building load heat demand, the system usually may cover most of the annual heat demand [46]. For these simulations the building load overall heat loss coefficient is increased to obtain a higher heat demand. The aim is to reduce the number of hours the heat pump system operates at part load. In real systems, an additional heating system must cover the top load, for example an electric water heater. The auxiliary heating system must be dimensioned to cover the maximum heat demand of the building. When installing a heat pump system for larger industrial or office buildings there must be a backup system in place, and this may also cover the top load. The maximum heating capacity $\dot{Q}_{\text{heat,max}}$ of the heat pump system is about 80 kW. This should equal 70 % of the maximum heat demand $\dot{Q}_{\text{dem,max}}$ of the building for this simulation. The maximum heat demand of the building is calculated by Eq. 6-5.

$$\dot{Q}_{\text{dem,max}} = \frac{\dot{Q}_{\text{heat,max}}}{0.7} = \frac{80 \text{ kW}}{0.7} = 114.3 \text{ kW} \tag{6-5}$$

Based on the maximum heat demand of the building, the desired room temperature T_{room} and the minimum ambient temperature $T_{amb,min}$ during year 2017 in Bergen (see Chapter 6.1), a new overall heat loss coefficient is calculated. The calculation is presented in Eq. 6-6.

$$UA = \frac{\dot{Q}_{dem,max}}{T_{room} - T_{amb,min}} = \frac{114.3 \text{ kW}}{20 \text{ °C} - (-7 \text{ °C})} = 4.233 \frac{\text{kW}}{\text{°C}} \tag{6-6}$$

Based on the previous calculations the overall heat loss coefficient is set to $UA = 4.3$ kW/°C in the first simulation. The remaining parameters are set to the standard condition values.

Following the same procedure as above, a second simulation is conducted where the building load is further increased. The maximum heating capacity $\dot{Q}_{heat,max}$ of the condenser equals 40 % of the maximum heat demand $\dot{Q}_{dem,max}$ of the building load in this simulation. The overall heat loss coefficient is set to $UA = 7.5$ kW/°C. The remaining parameters are set to the standard condition values. The heat pump system is expected to operate at full load most hours of the simulated year, which should be advantageous for the performance of the system.

The frequency output from Type 22 iterative feedback control for the first simulation with $UA = 4.3$ kW/°C and the second simulation with $UA = 7.5$ kW/°C of Type 206 are presented in Figure 6.19 and Figure 6.20 respectively. The minimum and maximum frequency of 25 Hz and 85 Hz respectively are marked by horizontal lines in the diagram. The maximum frequency registered is the maximum allowed frequency of $f = 85$ Hz.
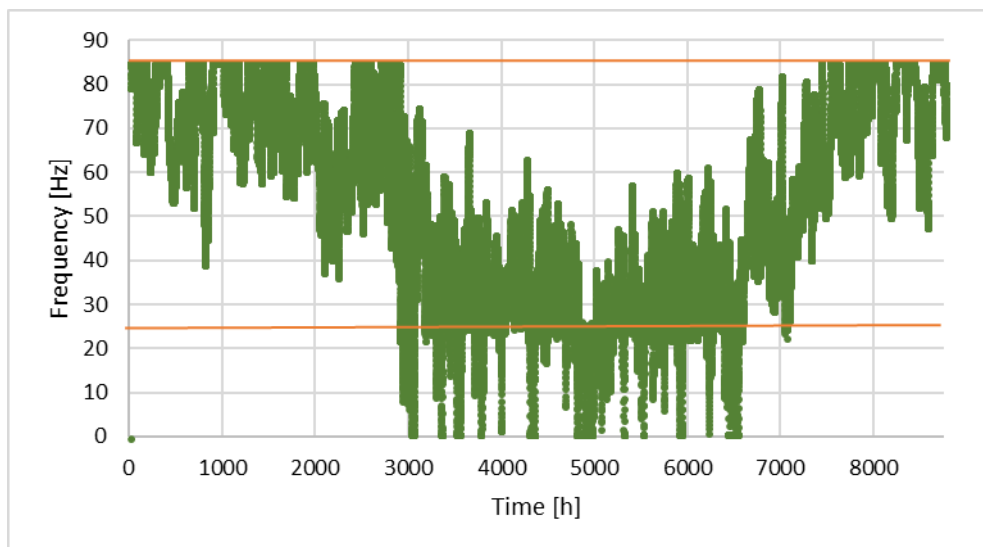
Figure 6.19: Frequency output from Type 22 iterative feedback control during simulation with increased overall heat loss coefficient $UA = 4.3$ kW/°C of building load. The minimum and maximum frequency of 25 Hz and 85 Hz respectively are marked by orange, horizontal lines in the diagram.
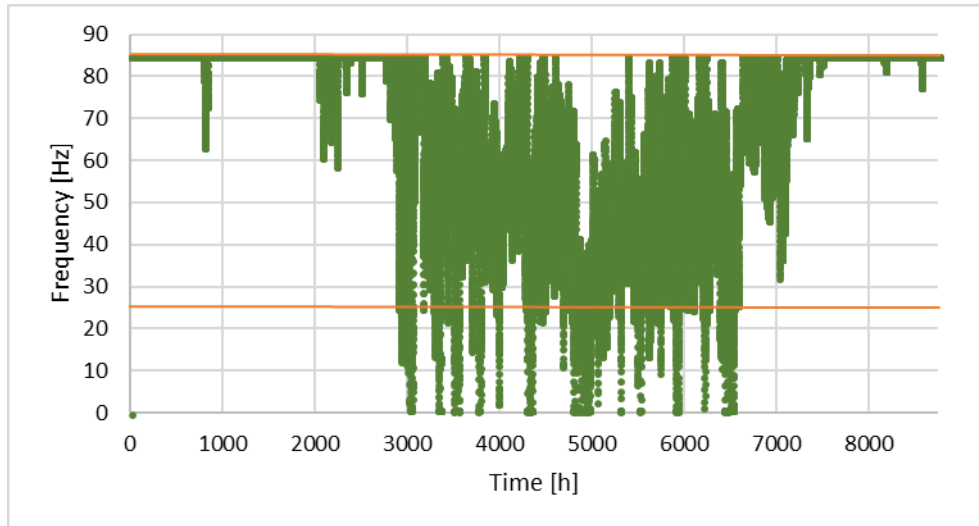
Figure 6.20: *Frequency output from Type 22 iterative feedback control during simulation with increased overall heat loss coefficient $UA = 7.5\ kW/°C$ of the building load. The minimum and maximum frequency of 25 Hz and 85 Hz respectively are marked by orange, horizontal lines in the diagram.*

The compressor power input for the first simulation with $UA = 4.3\ \text{kW}/°C$ and the second simulation with $UA = 7.5\ \text{kW}/°C$ of Type 206 are presented in Figure 6.21 and Figure 6.22 respectively. During the standard conditions simulation the compressor and system does not reach full load at any point. The maximum registered compressor power input during these two simulations is $\dot{W}_{in} = 20.30\ \text{kW}$ at the maximum frequency of $f = 85\ \text{Hz}$. The maximum compressor power input is consistent with the value provided by Bitzer of $\dot{W}_{in,max} = 20.30\ \text{kW}$ [59]. Notice that the axis for power and heating capacity in the following diagrams have higher values than the axis of the previously presented diagrams due to the increased building load and heat demand.
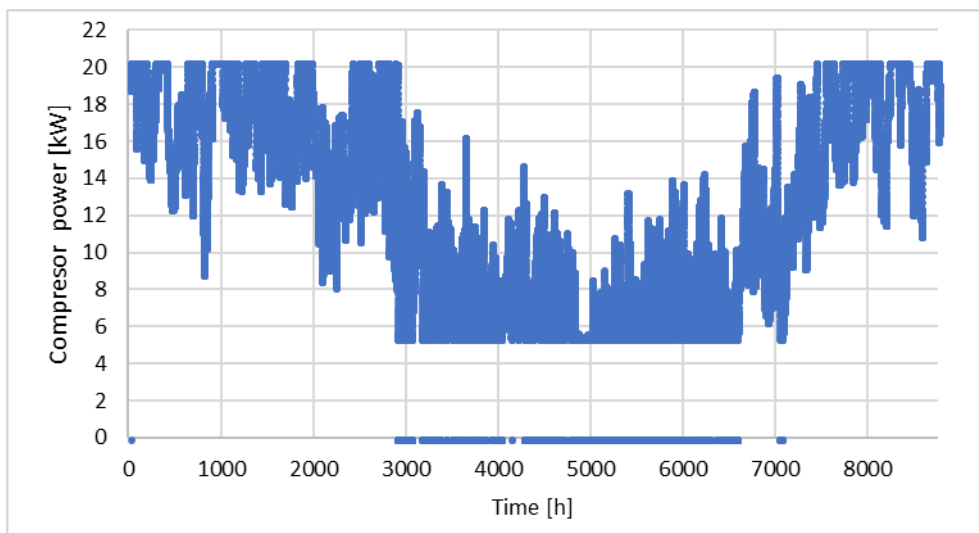
Figure 6.21: *Type 206 compressor power input during simulation with increased overall heat loss coefficient $UA = 4.3\ kW/°C$ of the building load. The compressor operates at full load for about 1150 hours during the simulated year.*

109
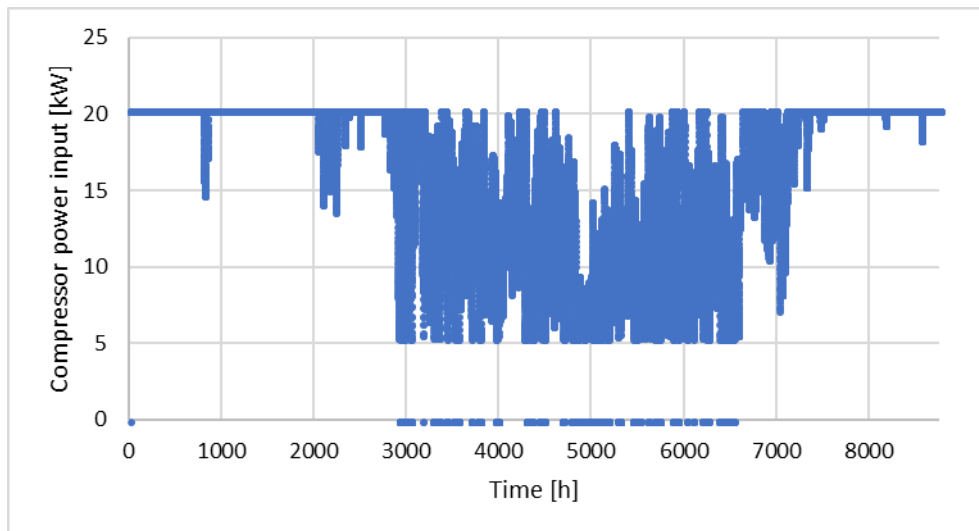
Figure 6.22: *Type 206 compressor power input during simulation with increased overall heat loss coefficient*

$UA = 7.5\ kW/°C$ of the building load. The compressor operates at full load for about 4700 hours during the simulated

year.

The heating capacity of the condenser during the simulation with $UA = 4.3\ \text{kW/°C}$ and $UA = 7.5\ \text{kW/°C}$ are presented in Figure 6.23 and Figure 6.24 respectively. The maximum heating capacity registered during the simulation with $UA = 4.3\ \text{kW/°C}$ is $\dot{Q}_{\text{heat}} = 76.1\ \text{kW}$ at the maximum frequency of $f = 85$ Hz. The maximum heating capacity corresponds to 67 % of the maximum heat demand of the building load, and corresponds well to the value provided by Bitzer of $\dot{Q}_{\text{heat,max}} = 76.9\ \text{kW}$ [59]. At higher heat demand than $\dot{Q}_{\text{dem}} = 76.1\ \text{kW}$ the heat pump system can not cover the full heat demand of the building load. An additional heating system dimensioned to cover the highest building heat demand must be in place to cover the top load. The top load system may also cover the lower heat demand during the summer season when the heat pump system is off due to low frequency.
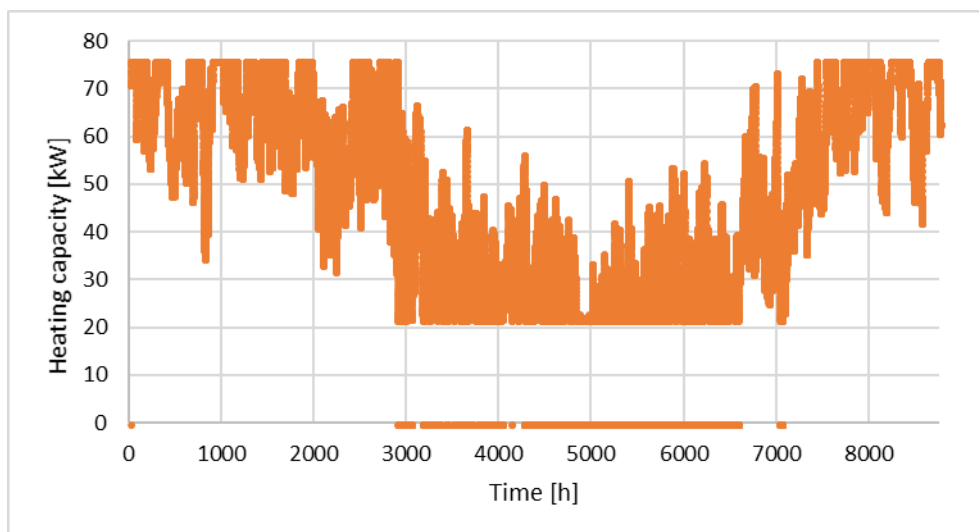
Figure 6.23: *Heating capacity of condenser during simulation with increased overall heat loss coefficient $UA = 4.3\ kW/°C$*

of the building load. The maximum heating capacity registered is $\dot{Q}_{heat} = 76.1\ kW$, corresponding to 67 % of the maximum

building load.

The maximum heating capacity registered during the simulation with $UA = 7.5 \text{ kW/}°\text{C}$ is $\dot{Q}_\text{heat} = 81.3 \text{ kW}$ at the maximum frequency of $f = 85 \text{ Hz}$. The maximum heating capacity corresponds to 41 % of the maximum heat demand of the building load, and corresponds quite well to the value provided by Bitzer of $\dot{Q}_\text{heat,max} = 76.9 \text{ kW}$ [59]. The small difference may be due to the condenser component Type 230. At higher heat demand than $\dot{Q}_\text{dem} = 81.3 \text{ kW}$ the heat pump system can not cover the full heat demand of the building load. An additional heating system dimensioned to cover the highest building load heat demand must be in place to cover the top load. The top load system may also cover the lower heat demand during the summer season when the heat pump system is off due to low frequency.
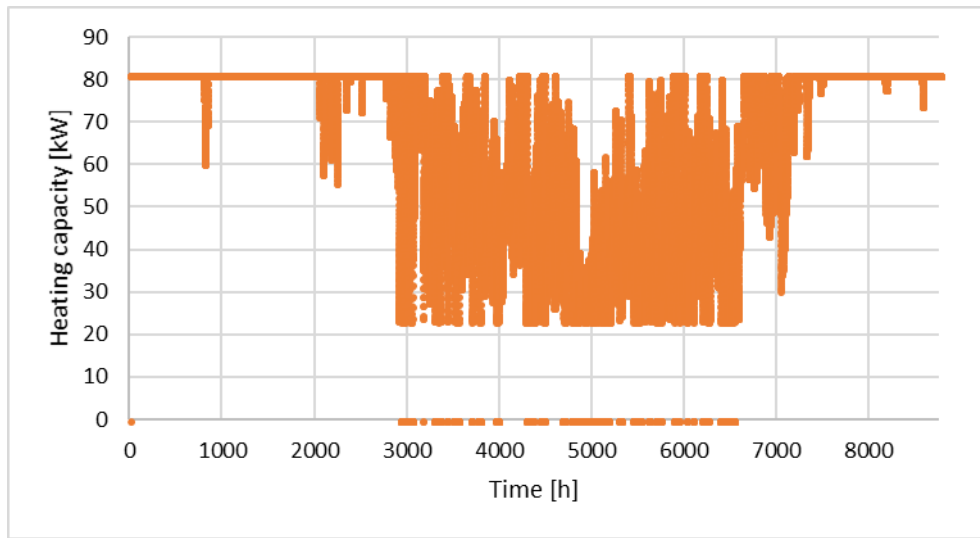
Figure 6.24: Heating capacity of condenser during simulation with increased overall heat loss coefficient $UA = 7.5 \ kW/°\text{C}$ of the building load. The maximum heating capacity registered is $\dot{Q}_{heat} = 81.3 \ kW$, corresponding to 41 % of the maximum building load.

The heat demand of the building load during simulations with $UA = 4.3 \text{ kW/}°\text{C}$ and $UA = 7.5 \text{ kW/}°\text{C}$ are presented in Figure 6.25 and Figure 6.26 respectively. The highest heat demand registered during simulation with $UA = 4.3 \text{ kW/}°\text{C}$ is $\dot{Q}_\text{dem} = 114.3 \text{ kW}$, and the for the simulation with $UA = 7.5 \text{ kW/}°\text{C}$ the highest registered heat demand is $\dot{Q}_\text{dem} = 199.4 \text{ kW}$. When the heat pump system is dimensioned to cover only 40 % of the building load, the heat pump system may only cover a small portion of the heat demand during the colder season with high heat demand. During the summer season the building load is low, and the frequency provided by the iterative feedback control is lower than the minimum frequency $f = 25 \text{ Hz}$. During these hours the heat pump system cannot deliver the required heat. The minimum and maximum heating capacity the heat pump system can deliver is marked by blue, horizontal lines in the diagrams.
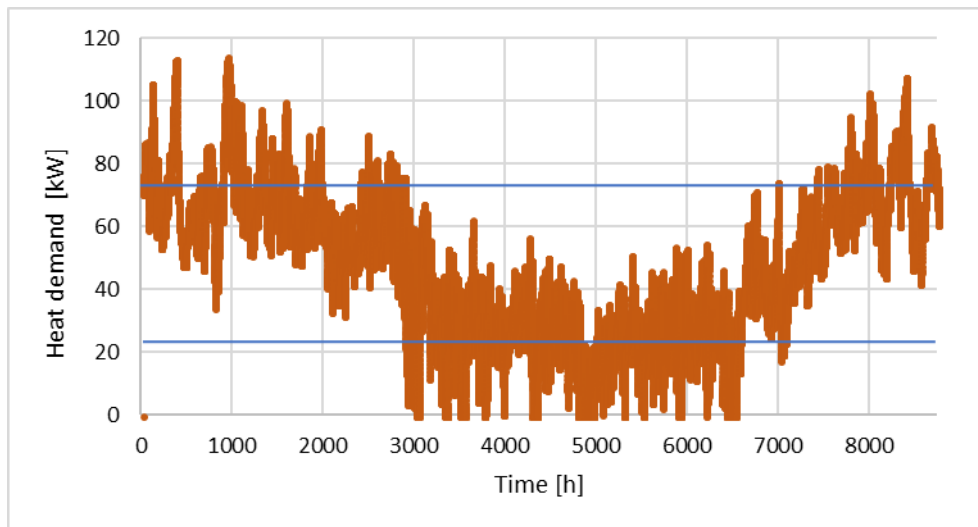
Figure 6.25: *Heat demand of Type 240 building load during simulation with increased overall heat loss coefficient $UA = 4.3\ kW/°C$ of the building load. The highest heat demand registered is $\dot{Q}_{dem} = 114.3\ kW$. The minimum and maximum heating capacity of the system is marked by blue, horizontal lines.*
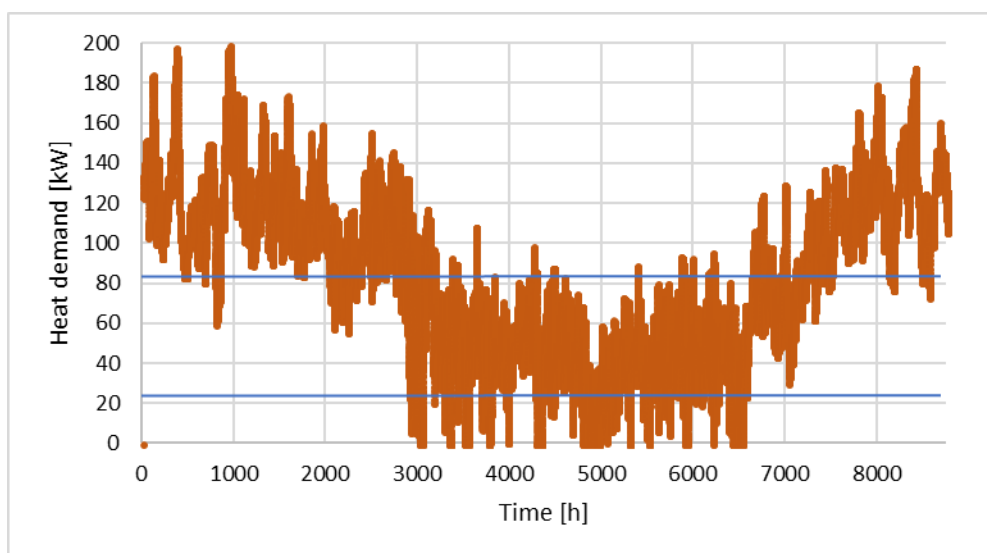
Figure 6.26: *Heat demand of Type 240 building load during simulation with increased overall heat loss coefficient $UA = 7.5\ kW/°C$ of the building load. The highest heat demand registered is $\dot{Q}_{dem} = 199.4\ kW$. The minimum and maximum heating capacity of the system is marked by blue, horizontal lines.*

The minimum and maximum values of the coefficient of performance COP and the isentropic efficiency η_{is} for both simulations are presented in Table 6.22. The maximum COP and η_{is} occurs at higher loads, and the minimum values at lower loads.

Table 6.22: *Minimum and maximum values for coefficient of performance COP and isentropic efficiency η_{is} of compressor Type 206 during simulation with increased overall heat loss coefficient of building load.*

	$UA = 4.3$ kW/°C		$UA = 7.5$ kW/°C	
	COP [–]	η_{is} [%]	COP [–]	η_{is} [%]
Minimum	3.756	58 %	4.014	58 %
Maximum	4.059	64 %	4.346	64 %

The total energy consumption $W_{in,total}$, the total heat delivered by the condenser $Q_{heat,total}$, the total heat received by the evaporator $Q_{cool,total}$ and the total heat demand of the building $Q_{dem,total}$, and the seasonal coefficient of performance $SCOP$ for both simulations are presented in Table 6.23. In addition, the results from the standard conditions simulation with $UA = 3.0$ kW/°C are presented. The total heat delivered equals the sum of the total energy consumption and the total heat received by the evaporator for both simulations, which is consistent with the energy balance of a heat pump as presented in Chapter 3.1. The heat demand of the building is much higher than the heat delivered by the heat pump. This is due to the increased building load, and the periods where the load is too low for the compressor and heat pump system to operate.

Table 6.23: *Total energy consumption of compressor, total heat delivered by the condenser, total heat received by the evaporator and seasonal coefficient of performance for simulation of Type 206 with increased building load.*

UA [kW/°C]	$W_{in,total}$ [kWh]	$Q_{heat.total}$ [kWh]	$Q_{cool,total}$ [kWh]	$Q_{dem,total}$ [kWh]	SCOP [–]
3.0	64581.9	268366	203783	298555	4.155
4.3	106217	405828	299610	427929	3.821
7.5	141439.6	571684	430244	746387	4.042

The number of hours on full load, the number of hours the system is off, the average frequency and the total deficit in heat delivered are presented in Table 6.24. Refer to Chapter 6.2 for results from the standard conditions simulations of compressor Type 206. The deficit in delivered heat must be covered by an alternative heating system. The heat pump system covers about 95 % of the total heat demand of the building load when $UA = 4.3$ kW/°C, and about 90 % for the standard conditions simulation of compressor Type 206. When increasing the building load even further, to make the maximum heating capacity of the heat pump system cover only 40 % of the maximum heat demand, the heat pump system only covers 75 % of the total heat demand.

Table 6.24: *Number of hours on full load and number of hours the compressor is off due to low frequency for simulations of compressor Type 206 at standard conditions and the increased building load of UA=4.3 kW/°C and UA=7.5 kW/°C.*

Overall heat loss coefficient [kW/°C]	Hours on full load	Hours Off	Reduction in hours off from standard conditions	Average frequency [Hz]	Heat deficit [kWh]
3	0	2300	0 %	37.8	30189.4
4.3	1500	1000	57 %	54.3	22298.4
7.5	4700	500	78 %	69.4	174704

6.10 Comparison of refrigerants

Compressor Type 207 is simulated with refrigerant R290 (propane) and refrigerant R717 (ammonium) at standard condition parameter values. The refrigerant is changed by changing the parameter value of the components. Refer to Chapter 5.1.2 for a detailed description of the parameters included in compressor Type 207. The simulation with propane as refrigerant is expected to give results closer to the standard conditions simulation with refrigerant R134a. This is due to the similar properties of the two refrigerants. The properties of refrigerant R717 are quite different from R134a. Refer to Chapter 3.6 for further information about important refrigerant properties and their effect on the heat pump performance. These simulations cannot be conducted with compressor Type 206 due to the coefficients obtained are valid for only refrigerant R134a.

The total energy consumption of the compressor $W_{\mathrm{in,total}}$, the total heat delivered by the condenser $Q_{\mathrm{heat,total}}$, the total heat received in the evaporator $Q_{\mathrm{cool,total}}$ and the seasonal coefficient of performance $SCOP$ are calculated and compared to the results from the standard conditions simulation with refrigerant R134a. The results are presented in Table 6.25 along with the results from the standard conditions simulation of compressor Type 207 with refrigerant R134a.

Table 6.25: Total energy consumption of compressor Type 207, total heat delivered by the condenser, total heat received in evaporator and the seasonal coefficient of performance for refrigerants R134a, R290 propane and R717 ammonium.

Refrigerant	$W_{\mathrm{in,total}}$ [kWh]	$Q_{\mathrm{heat,total}}$ [kWh]	$Q_{cool,\mathrm{total}}$ [kWh]	$SCOP$ [-]
R134a	51205.1	272587	221382	5.323
R290	46607.6	248976	202369	5.342
R717	35097.3	208731	173633	5.947

The total energy consumption of the compressor $W_{\mathrm{in,total}}$ is the highest for synthetic refrigerant R134a. During simulation of natural refrigerants R290 and R717 the frequency did not reach the maximum value of $f = 85$ Hz. The building load heat demand is equal for all three simulations. The low frequency indicates that the heating capacity of the system is higher than for refrigerant R134a, and the system may cover the heat demand of the building at lower loads. The average frequency provided by the iterative feedback control, the deficit in heat delivered and the number of hours the system is off during the simulation is presented in Table 6.26. The results for refrigerant R134a are from the standard conditions simulation of compressor Type 207.

Table 6.26: The average frequency, the deficit in heat delivered and the number of hours the system is off due to low frequency. The lower average frequency and higher deficit in heat and number of hours the system is off indicate that the system is over-dimensioned.

Refrigerant	Average frequency [1/s]	Deficit in heat [kWh]	Number of hours off [h]
R134a	40.11	26017.3	2100
R290	32.54	49628.3	3100
R717	26.07	89873.8	4450

The compressor power input $\dot{W}_{in}$ for refrigerant R290 and R717 is presented in Figure 6.27 and Figure 6.28 respectively. The compressor power input during these simulations are smaller compared to the compressor power input from the standard conditions simulation of compressor Type 207 presented in Figure 6.10. The periods when the system is off due to low frequency from the iterative feedback control are also visibly larger than for compressor Type 207 at standard conditions with refrigerant R134a.
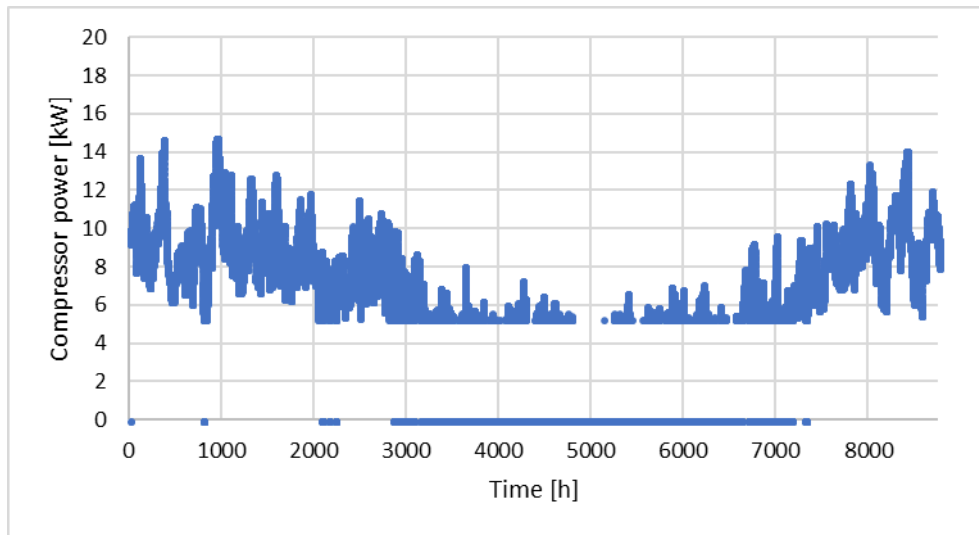
Figure 6.27: Compressor power input $\dot{W}_{in}$ during simulation of compressor Type 207 with refrigerant R290 propane. The compressor power input and total energy consumption is lower than with refrigerant R134a.
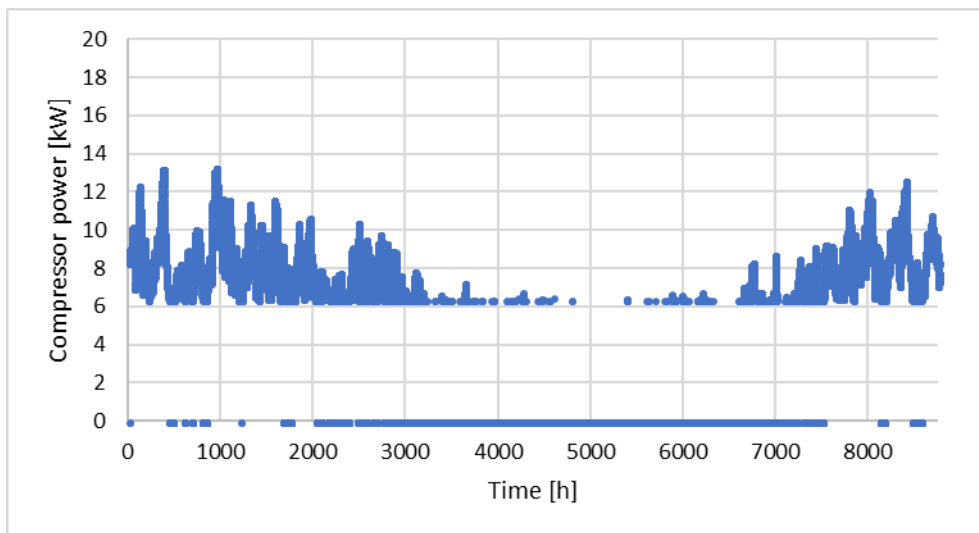
Figure 6.28: Compressor power input $\dot{W}_{in}$ during simulation of compressor Type 207 with refrigerant R717 ammonium. The compressor power input and total energy consumption is lower than with refrigerant R134a and R290 propane.

7   Discussion

In the following chapter, a discussion and evaluation of the various TRNSYS components and simulations are presented. The first section concerns the validity of the results and the components. The second section provides suggestions and recommendations for further development of the components, making the components better resemble the behavior of real heat pumps and systems.

7.1   Results and validation

The main goal of the simulations executed in this project is to show that the compressor and the developed heat pump models work as expected and that the source codes and calculations are correct. The limitations of the various components and suggestions for further developments are discussed. The models do not provide a complete picture of the performance of a heat pump at all operational conditions. The models are created as a basis for further work and development.

All models work as expected based on the energy balance for heat pumps provided in Chapter 3.1. The energy balance states that the heat delivered by the condenser $\dot{Q}_{\mathrm{heat}}$ is equal to the sum of the compressor power input $\dot{W}_{\mathrm{in}}$ and the heat received by the evaporator $\dot{Q}_{\mathrm{cool}}$ [16]. The iterative feedback control Type 22 regulates the compressor speed, and the compressor power input, heating capacity and cooling capacity of the system follows the building load heat demand. The compressor and heat pump components are modelled with a minimum allowed frequency, providing a lower limit for the heating capacity of the system. During periods with low building load, the heat pump system cannot deliver the required heat as the frequency is lower than the minimum frequency. The chosen minimum frequency for the simulations conducted in this project is consistent with the minimum frequency of Bitzer reciprocating compressor 4NE-20.F4Y [59]. The components and models are not validated by own experimental data. This imposes an uncertainty regarding the validity of the components.

7.1.1   Standard conditions simulations

The compressor power input and the heating capacity calculated in the standard conditions simulation of compressor Type 206 are consistent with the data provided by Bitzer [59]. This indicates that the polynomial for compressor power input and the interpolation of the coefficients in the source code works correctly. The calculated coefficient of performance COP varies only slightly during the simulation and has the highest value at minimum load conditions. This is not consistent with either literature stating that the compressor is most efficient at full load conditions [9] or data from Bitzer [59]. The coefficient of performance at minimum load with $f = 25\,\mathrm{Hz}$ provided by Bitzer is $COP = 2.8$, while the coefficient of performance at minimum load during the standard conditions simulation of compressor Type 206 is $COP = 4.35$. This inconsistency is probably due to the condenser and evaporator components Type 230 and the simplified calculations of the heat transfer rates, as the calculated compressor power input at minimum frequency is consistent with data from Bitzer.

The largest limitation of compressor Type 206 is the difficulty of obtaining the compressor coefficients for the calculation of the power input. An alternative approach to obtaining the coefficients is by measured or experimental data of a compressor. If the compressor power input is measured at various frequencies and saturation temperatures, the coefficients may be calculated. According to AHRI Standard 540 [29] the coefficients are determined by the least squares method.

The results from the standard conditions simulations of the two compressor Types 206 and 207 are not consistent. Compressor Type 207 requires less compressor power input than compressor Type 206, resulting in a higher seasonal coefficient of performance $SCOP$. The total energy consumption of Type 207 is about 21 % lower than for Type 206. The calculated values of the seasonal coefficient of performance for both compressor types are outside the range of typical values of $SCOP = 3 - 4$ for medium range condensing temperatures [46]. However, the typical values are provided for evaporating temperature $T_e = 0\,°\text{C}$. The seasonal coefficient of performance calculated for $T_e = 0\,°\text{C}$ during the simulations of Type 206 with varying evaporating temperature is $SCOP = 3.53$, which is consistent with the range of typical values. The seasonal coefficient calculated for Type 206 of 4.155 may be correct as the evaporating temperature in the standard conditions simulation is $T_e = 10\,°\text{C}$. The high $SCOP$ of Type 207 may be due to both the value of the polytropic exponent and the simplified condenser and evaporator components Type 230, where heat exchanger efficiency is not accounted for. The compressor power input calculated for compressor Type 206 is consistent with data obtained from Bitzer [59].

The building load heat demand is approximately equal for the two simulations. This is as expected as the building load parameters and inputs are the same for all simulations except for the increased building load simulations. The very small differences may be explained by small differences in the iterations and convergence in the time steps of the various simulations. The delivered heat from the condenser is about equal for the two simulations, which is expected due to the equal building load heat demand. The compressor speed and performance are controlled by the current deficit or surplus of heating capacity compared to the heat demand of the building load. The tolerance of the iterative feedback control Type 22 is set to a low value (see Chapter 5.2.1) in order to make the system capacity follow the building load heat demand closely.

The standard conditions simulation of variable speed heat pump components Type 216 and 217 gives results that are approximately equal to the simulations of Type 206 and 207 respectively at standard conditions. This is as expected as the variable speed heat pump components are simply a collection of the calculations in compressor Type 206 and 207, and condenser and evaporator component Type 230. The small differences of less than 1 % are probably due to differences in the iterations and convergence during the simulations in TRNSYS. The main difference between the simulation of the compressor and variable speed heat pump components is the total computational time. At the chosen time step of $t = 0.1\,\text{h}$ and the simulation time of $8760\,\text{h}$, the simulations of the compressor models takes about an hour, while the simulations of the variable speed heat pump components takes about 20 minutes. The large difference in computational time is due to the TRNSYS kernel having less components to call when running simulations of the variable speed heat pump components.

The various losses that occur in real systems, for example pressure loss due to fluid friction and heat loss to the ambient in both compressor and heat exchangers, are not accounted for in these simulations. The reason for this is that the values of these parameters should be determined experimentally, which has not been an option during the work of this project. Values for various efficiencies and parameters for heat loss may be found in the literature [2, 22, 39]. The heat exchanger models provided in many studies are more advanced than the simplified condenser and evaporator Type created in this project, and the various parameters utilized in these studies do not apply to Type 230. This is elaborated in Chapter 7.2. The parameters are added to enable future users to account for the effects of heat exchanger efficiency and various pressure and heat losses in the heat pump system.

These parameters will affect the heat transfer rate in both the condenser and the evaporator and would induce changes in the performance of the system. Any pressure loss on the low pressure side of the system will require a higher work input to the compressor [45]. The main focus of this project is the modelling of the compressor, and these parameters where not prioritized for sensitivity analysis. For further development of the components created in this project, the parameters for heat and pressure loss should be investigated. The heat loss and varying efficiencies of a compressor is accounted for by the coefficients of compressor Type 206 as the coefficients are calculated from measurements of a compressor. This could explain the difference between the results from the two compressor models.

The isentropic efficiency is constant through the simulations of compressor Type 207 and varies only slightly during the simulations of compressor Type 206. The values of the calculated isentropic efficiency for compressor Type 206 of $\eta_{is} = 57.8 - 63.9\,\%$ is on the lower side compared to results by Tassou et al. [10] where the isentropic efficiency of the tested reciprocating compressor varies between $60 - 70\,\%$ with variable speed. The isentropic efficiency calculated for compressor Type 207 of $\eta_{is} = 82.7\,\%$ is higher than results from other studies [8, 10]. The inconsistency of the isentropic efficiency compared to literature is probably due to the calculation of the discharge enthalpy, which is based on an energy balance. The discharge enthalpy is assumed to equal the sum of the suction specific enthalpy and the specific work consumed by the compressor. The compressor is assumed to compress the refrigerant vapor to the condensing pressure. In real systems the compressor discharge pressure is higher than the average condensing pressure in order to allow the vapor to flow against the condenser pressure [45]. A higher discharge pressure results in a higher discharge temperature and specific enthalpy of the refrigerant. This would result in a lower isentropic efficiency. For the calculation of compressor power input in Type 207, the mass flow rate is the only time-dependent variable included. Due to this, the ratio of the compressor power input and the mass flow rate remains constant, and hence the discharge enthalpy is constant through the simulation. The compressor power input of Type 206 is calculated independently from the refrigerant mass flow rate. Still, the ratio of the compressor power input and the refrigerant mass flow rate does not change much during the simulation. The highest isentropic efficiency for compressor Type 206 occurs at low load. This is not as expected as compressors are designed to obtain optimal performance full load [9, 12]. The calculation of the discharge enthalpy limits the components in terms of simulation of part load performance. For further development of the components, an evaluation of the calculation of discharge specific enthalpy from the compressor is recommended.

The volumetric efficiency of the compressor should be accounted for in the coefficients, and the value of the volumetric efficiency for compressor Type 206 should be investigated. The value of the volumetric efficiency does not affect the calculation of power input to the compressor directly but would affect the refrigerant mass flow rate and hence the heat transfer rates in the condenser and evaporator. As a consequence of the increased heating and cooling capacity of the system, the system will operate at lower load. The total energy consumption of the compressor is therefore lower than for the standard conditions simulations. The coefficient of performance is also improved. The effect of the volumetric efficiency is further discussed in the next section.

7.1.2 Volumetric efficiency

The results from the simulation of compressor Type 206 with volumetric efficiency $\eta_{vol} = 1.0$ are closer to the results of the standard conditions simulations of Type 207. This is as expected as the

increased volumetric efficiency results in increased refrigerant mass flow rate. The effect of the volumetric efficiency of the compressor is most likely accounted for in the compressor coefficients provided by Bitzer, and the standard conditions parameter value for the volumetric efficiency of Type 206 should have been set to one, or the parameter could be removed from this component. If the components and models are to be further developed, this should be investigated and removing the volumetric efficiency parameter from Type 206 should be considered. Kinab et al. [28] created a TRNSYS component for a heat pump compressor using the polynomial and coefficients as described in AHRI Standard 540 [29]. The refrigerant mass flow rate is calculated by the polynomial and related coefficients for mass flow rate, and the volumetric efficiency is an output from the model. The polynomial is not used for the calculation of the refrigerant mass flow rate in this project due to the difficulty of obtaining coefficients. An alternative compressor model should be developed using the polynomial expression for the refrigerant mass flow rate as well as the compressor power input. The results should be compared to compressor Type 206 results, in order to investigate the validity of the calculation of mass flow rate in Type 206. Alternatively, the calculation of the refrigerant mass flow rate should be validated by experimental data.

In some studies [1, 23, 27] the volumetric efficiency is calculated by an empirical expression based on the suction and discharge pressure and the clearance volume of the compressor. This may be a viable option if the volumetric efficiency is required as an output. The calculated volumetric efficiency may still be used to calculate the refrigerant mass flow in the source code, and the clearance volume must be added as a parameter to the compressor component. The volumetric efficiency mostly changes with the compression ratio of the compressor [13]. The volumetric efficiency is also affected by the clearance volume, leakage, flow characteristics and heating of the compressor [45]. The clearance volume is only relevant for reciprocating compressors, as screw and scroll compressors do not have any clearance volume. The volumetric efficiency of screw and scroll compressors are generally higher than the volumetric efficiency of reciprocating compressors. During these simulations the saturation temperatures and the degree of superheat is constant, and the volumetric efficiency calculated by the empirical expression would be constant. If the models where adapted to fluctuating saturation temperatures and degree of superheat, the compression ratio would change during the simulations. Using the empirical expression would provide a varying volumetric efficiency output.

7.1.3   Polytropic exponent

The total energy consumed by the compressor is lower for Type 207 than Type 206 at standard conditions. The various losses that may occur in a heat pump system and compressor, for example heat loss to the ambient or pressure loss in components and pipes, are not accounted for during the simulations of compressor Type 207. The chosen value of the polytropic exponent is consistent with values found in the literature [22] but the parametric study of the polytropic exponent concludes that the value of the polytropic exponent must be $n = 2.683$ to obtain the same total energy consumption of compressor Type 206. A final simulation is conducted of compressor Type 207 with the polytropic exponent calculated from the linear expression of the compressor energy consumption as a function of the polytropic exponent. The results are consistent with the results from the standard conditions simulation of compressor Type 206. The small deviations are due to the linear function being an approximation, and the total energy consumption of the compressor not being completely correlated to the polytropic exponent.

If the polytropic exponent is calculated to match the total energy consumption of compressor Type 206 from the simulation with volumetric efficiency $\eta_{\mathrm{vol}} = 1.0$, the polytropic exponent should be $n = 1.73$. This is a more sensible value of the polytropic exponent and implies that the volumetric efficiency parameter of compressor Type 206 should be removed.

The polytropic exponent should be determined by experimental data from an actual compressor. This is due to the polytropic exponent being a property of the compressor, and not the refrigerant [45]. However, Rasmussen et al. [21] state that the polytropic exponent is highly dependent on the refrigerant. Byrne et al. [22] determine the polytropic exponent for various refrigerants based on measurements and experimental data of the polytropic exponent of one refrigerant in one study, and uses an experimentally determined polytropic exponent for a scroll compressor in another study [1]. Wenhua Li [27] estimates the polytropic exponent by defining some dimensionless parameters that must be added to the isentropic exponent to obtain the value of the polytropic exponent. The dimensionless parameters are determined by fitting of experimental data. The polytropic exponent may be estimated from experimental data by measuring the compressor power input, the suction and discharge pressure and the refrigerant mass flow rate. Refer to Chapter 3.2 for a description of the relationship between the polytropic exponent and the mentioned variables.

7.1.4   Effect of subcooling

The simulations with degree of subcooling $T_{\mathrm{SC}} = 0\,^{\circ}\mathrm{C}$ are conducted in order to investigate the effect of degree of subcooling on the two compressor models. The compressor coefficients obtained through Bitzer [59] are valid for a specific degree of subcooling and superheat. Compressor Type 207 does not have such limitations on temperatures or other parameter values. In real heat pump systems there are limitations on the degree of subcooling. In order to achieve heat transfer and evaporation in the evaporator, the temperature of the refrigerant must exceed the heat source temperature. Hence, the degree of subcooling must be adapted to the temperature of the heat source. The change in the total energy consumed by the compressor, the total heat delivered by the condenser and the seasonal coefficient of performance are about equal for compressor Type 206 and Type 207. This implies that the use of a degree of subcooling of $T_{\mathrm{SC}} = 8\,^{\circ}\mathrm{C}$ does not highly affect the results of Type 206, or the comparison of the results of the two compressor models.

The cooling capacity $\dot{Q}_{\mathrm{cool}}$ of the evaporator is expected to decrease when the degree of subcooling is decreased from the standard conditions value of $T_{\mathrm{SC}} = 8\,^{\circ}\mathrm{C}$ to $T_{\mathrm{SC}} = 0\,^{\circ}\mathrm{C}$ [45, 70], as the refrigerant enters the evaporator with a higher vapor quality and hence higher specific enthalpy. Refer to Chapter 3.1.3 for further information regarding subcooling of the refrigerant. As the outlet conditions of the evaporator Type 230 are the same regardless of degree of subcooling, the energy received by the refrigerant in the evaporator is expected to be lower than if the refrigerant is subcooled. The difference in the specific enthalpy of the refrigerant at the evaporator inlet with and without subcooling is presented in Figure 7.1. The state of 4 denotes the state of the refrigerant at the inlet of the evaporator with subcooling, and 4' denotes the state of the refrigerant at evaporator inlet without any subcooling.
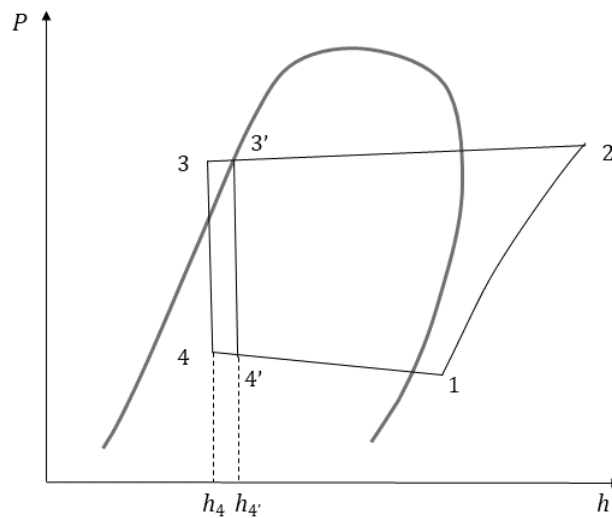
Figure 7.1: Vapor compression cycle with and without subcooling presented on a Pressure-Specific enthalpy diagram. 4 denotes the state of the refrigerant at the evaporator inlet with subcooling, and 4' denotes the state of the refrigerant at the inlet of the evaporator without any subcooling.

The calculated total heat energy received by the refrigerant in the evaporator during these simulations is approximately equal to the results from the standard conditions simulations. The change in $SCOP$ is due to the increase in compressor energy consumption exceeding the increase in heat energy delivered by the condenser. The change in the seasonal coefficient of performance is expected to arise from decreased cooling capacity in the evaporator [70]. A refrigeration cycle with subcooling should obtain higher COP and $SCOP$ without increasing the required compressor power input [45]. The increased compressor power input may be explained by the increased load of the system. The average frequency during simulation of compressor Type 206 is $f = 54.3$ Hz, while the average frequency during the standard conditions simulation is $f = 37.8$ Hz. The average frequency of compressor Type 207 is $f = 42.7$ Hz and $f = 40.1$ Hz during the reduced subcooling and standard conditions simulations respectively. The load is increased because the reduced degree of subcooling results in lower heat transfer rates. The inconsistency with theoretical predictions for the effect of changing the degree of subcooling is most likely due to the simplified calculations within the evaporator and condenser components Type 230. In addition, the iterative feedback control and the method of controlling the system may affect the results when decreasing the degree of subcooling. The iterative feedback control is designed to make the system deliver the necessary heat energy regardless of the operational conditions within the cycle. The exception is when the heat demand of the building is too high or low, resulting in a frequency outside the valid operating range of the compressor. A more thorough sensitivity analysis should be conducted of both the degree of subcooling and superheat for both models if the components and models are to be further developed. The inlet and outlet enthalpy of the evaporator should be added as outputs from the model and may be of assistance when comparing the standard conditions subcooling and the reduced subcooling simulations.

7.1.5 Evaporating and condensing temperature

The total energy consumption of the compressor is closely correlated to the evaporating temperature for both compressor components. Compressor Type 206 has a limited valid range of evaporating temperatures [59]. The temperatures chosen for the parametric study are both within and outside the valid range of evaporating temperatures. As Type 207 shows the same behavior with changing evaporating temperatures as for Type 206, the validity range of the evaporating temperatures may not affect the simulations of Type 206 very much. The linearity of the compressor energy consumption to the evaporating temperature is consistent for all tested temperatures. The standard conditions condensing temperature that is used for all the simulations in the parametric study of the evaporating temperature is within the valid range of condensing temperatures [59]. The two compressor models show the same behavior with the changing evaporating temperature. The exception is the heat received by the refrigerant in the evaporator $Q_{\mathrm{cool,total}}$. The total heat received in the evaporator is not closely correlated to the evaporating temperature and is less correlated to the evaporating temperature in the simulations of compressor Type 207 than Type 206. This is unexpected as the evaporator component Type 230 is the same for all simulations. The total heat in the evaporator does not change very much during the simulations, the difference between the highest and lowest value is about 7.5 % for Type 206 and 5 % for Type 207. The highest total heat received in the evaporator does not occur at the highest evaporating temperature. The negative correlation values calculated indicate that the cooling capacity decreases with increasing evaporating temperature. These effects may be due to the simplifications of the calculations in the condenser and evaporating component Type 230. The component does not consider the properties of the heat source or heat sink, only the refrigerant properties at inlet and outlet state. This puts great limitations on the performance evaluations of the models and should be further investigated.

The total energy consumed by the compressor $W_{\mathrm{in,total}}$, the total heat delivered by the condenser $Q_{\mathrm{heat,total}}$, the total heat received in the evaporator $Q_{\mathrm{cool,total}}$ and the seasonal coefficient of performance $SCOP$ are closely correlated to the condensing temperature for both compressor components Type 206 and 207. All the condensing temperatures investigated in this parametric study are within the valid range of condensing temperatures for compressor Type 206 [59]. The change in total heat energy $Q_{\mathrm{cool,total}}$ received in the evaporator changes even at constant evaporating temperature during the simulations with changing condensing temperature. This is due to the frequency control and the increased load of the system when increasing the condensing temperature. The average frequency increases with the increasing condensing temperature, resulting in increased refrigerant mass flow rate.

The models and components seem to be more sensitive to changes in the evaporating temperature than the condensing temperature. This is due to the compressor being more sensitive to the refrigerant conditions in the evaporator than the condenser. The main factor affecting the compressor performance and capacity is the evaporation temperature [45]. The properties of the refrigerant at suction state varies with the evaporation temperature. The changes in compressor energy consumption with varying evaporation and suction gas temperature is mostly due to the changes in the specific volume or density of the refrigerant. Increased density of the refrigerant at higher temperatures results in higher mass flow rate and hence increased compressor power input. The condensing temperature affects the compressor power input by changing the required discharge pressure the compressor must deliver.

The coefficient of performance and the seasonal coefficient of performance is inversely proportional to the temperature difference between the condensing and evaporating temperature [16, 46]. The calculated $SCOP$ values for these simulations are consistent with the theoretical predictions, increasing when increasing the evaporating temperature and decreasing when increasing the condensing temperature. When the temperature difference between the condensing and evaporating temperature increases, the work energy required by the compressor increases as the difference between the suction and discharge pressure increases.

7.1.6 Increased building load

The simulations with the increased building load are conducted to simulate a heat pump system that is dimensioned to cover only 70 % and 40 % of the maximum building load. Heat pump systems are usually designed to cover only parts of the maximum load. This is done to minimize the part load operation of the system [46]. Especially the compressor experiences lower efficiencies at part load conditions, decreasing the performance of the system [8, 12]. The results show that the heat pump operates on full load for more hours during the simulated year, and the average frequency is increased. The advantages of operating on full load is not clear from the results of these simulations. This is due to the inability of the models to reflect the varying performance of the heat pump system. The heat pump covers 90 % of the total heat demand of the building load when the heat pump is dimensioned to cover 70 % of the maximum heat demand. The heat pump covers only 75 % of the annual heat demand when it is dimensioned to cover only 40 % of the maximum heat demand. During the standard conditions simulation of compressor Type 206 the average frequency is lower, and the heat pump system barely reaches full load. The heat pump system covers about 90 % of the total heat demand of the building during the standard conditions simulations but operates on part load most of the year.

In order to properly simulate a heat pump dimensioned to cover less than the full heat demand of the building load, the compressor parameter values should be altered. Compressor Type 206 requires a new set of coefficients, obtained for a compressor of smaller size and capacity. In addition, the swept volume parameter must be changed. For compressor Type 207 the swept volume parameter value may be changed to simulate a smaller capacity compressor.

To minimize the part load operation of the heat pump and operate at a more even load, especially during the warmer summer season, the heat pump may also be used for heating of domestic water. The demand for domestic hot water is more constant during the year compared to the heat demand of the building, especially for residential buildings [73].

7.1.7 Refrigerants

The components work as expected when changing the refrigerant parameter value. The total energy consumption of the compressor is lower, and the total heat energy delivered by the condenser is higher, resulting in increased seasonal coefficient of performance. The simulations of compressor Type 207 with natural refrigerants propane and ammonium provide results that are consistent with the properties of the refrigerants compared to synthetic refrigerant R134a.

The heat pump system may deliver the necessary heat to the building at much lower frequencies and load for the natural refrigerants propane R290 and ammonium R717. This is due to the heat of evaporation being higher for the natural refrigerants than synthetic refrigerant R134a used in the

standard conditions simulations [53]. The effect of the higher heat of evaporation would be more apparent if the simulations were conducted with an increased building load heat demand. The compressor power input and total energy consumed by the compressor is lower with the natural refrigerants due to the low load and frequency. The main aim of these simulations is to show that the models work with other refrigerants than R134a. The simulations are not executed for compressor Type 206 due to the lack of proper coefficients for these refrigerants. If such coefficients are obtained, the data may be added to the component through the Proforma file or Variable window of the component. Refer to Chapter 5.1.1 and Appendix IV and V for further information regarding the incorporation of coefficients and data files to the components.

The difference in heat of evaporation is clear when considering pressure-specific enthalpy for the three refrigerants. The diagrams for refrigerant R134a, R290 and R717 are presented in Figure 7.2 – 7.4 respectively, along with the refrigeration cycle at the standard condition parameter values provided in Chapter 5.1.2. The compression process is presented with both isentropic efficiency $\eta_{is} = 100\,\%$ and the calculated isentropic efficiency from the respective simulations. The diagrams are obtained from the freely available program CoolPack [54].
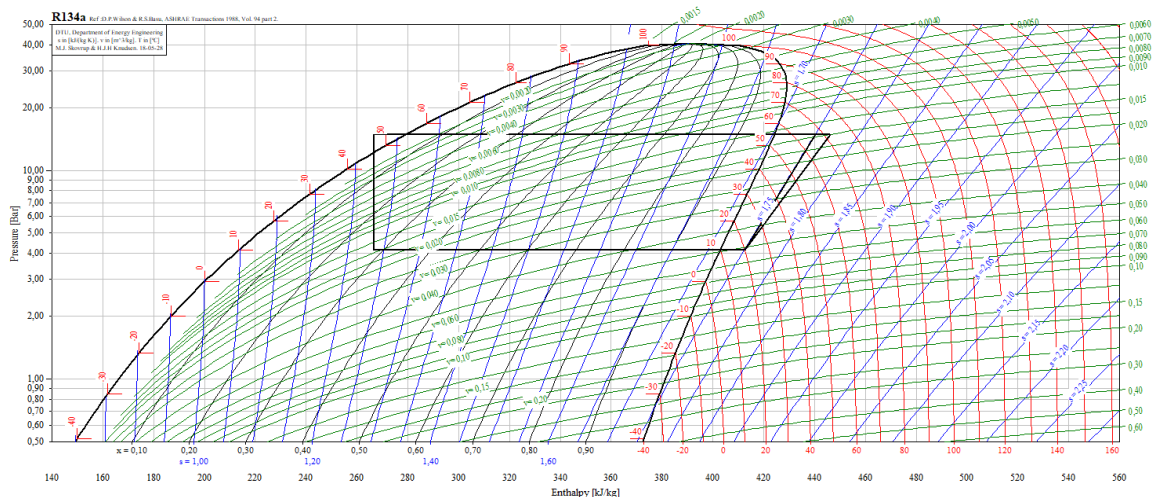
Figure 7.2: Pressure-specific enthalpy diagram for refrigerant R134a, showing the refrigeration cycle at the standard condition parameter values. The compression process is presented with both isentropic efficiency $\eta_{is} = 100\,\%$ and the calculated $\eta_{is} = 83\,\%$ from the standard conditions simulations of compressor Type 207. The Y-axis presents pressure in unit bar and the X-axis presents the specific enthalpy in unit kJ/kg. The diagram is obtained from CoolPack [54].

Comparison of the diagrams for refrigerant R134a and R290 show the similarities of the two refrigerants, for example the critical pressure being approximately $40\,\mathrm{bar}$ or $4\,\mathrm{MPa}$ for both refrigerants. Notice that the scale of the specific enthalpy axis is not equal for the two diagrams. The heat of evaporation at a given saturation pressure or temperature equals the difference in specific enthalpy between the saturated vapor and saturated liquid state. At $T_{sat} = 55\,°\mathrm{C}$, the heat of evaporation for R134a is approximately $h_{fg} = 145$ kJ/kg and for R290 approximately $h_{fg} = 270$ kJ/kg. These values may be obtained from Coolpack [54].
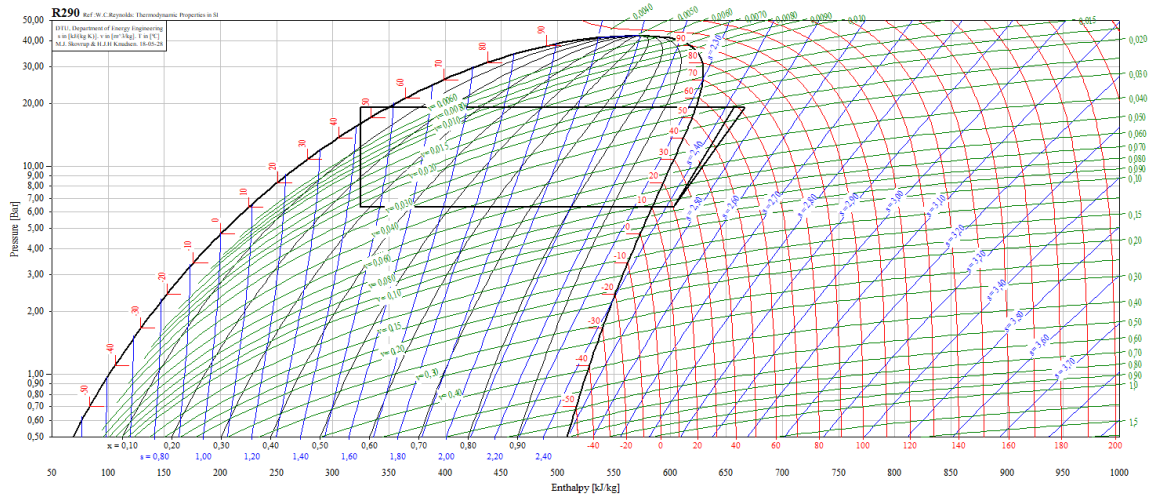
Figure 7.3: *Pressure-specific enthalpy diagram for refrigerant R290, showing the refrigeration cycle at the standard condition parameter values. The compression process is presented with both isentropic efficiency $\eta_{is} = 100\,\%$ and the calculated $\eta_{is} = 85\,\%$ from the simulation of refrigerant R290. The Y-axis presents pressure in unit bar and the X-axis presents the specific enthalpy in unit kJ/kg. The diagram is obtained from CoolPack [54].*

The pressure-enthalpy diagram for refrigerant R717 deviates from the two previous diagrams for refrigerants R134a and R290. The critical pressure of ammonium is higher than for propane and R134a at about $100\,\text{bar}$ or $10\,\text{MPa}$. At saturation temperature $T_{\text{sat}} = 55\,°\text{C}$ the heat of evaporation is approximately $h_{\text{fg}} = 1030\ \text{kJ/kg}$ according to CoolPack [54], making ammonium well suited for large capacity applications [19]. Due to the high critical temperature ammonium may deliver heat at higher temperatures at medium pressures, and standard components and pipes may be used [45]. The discharge temperature of the cycle at standard conditions presented in Figure 7.4 is about $110\,°\text{C}$. The high discharge temperature imposes thermal stress resulting in fatigue and wear of the compressor components [8]. The discharge temperature is reduced by cooling of the compressor and by keeping the degree of superheat at a minimum [45].
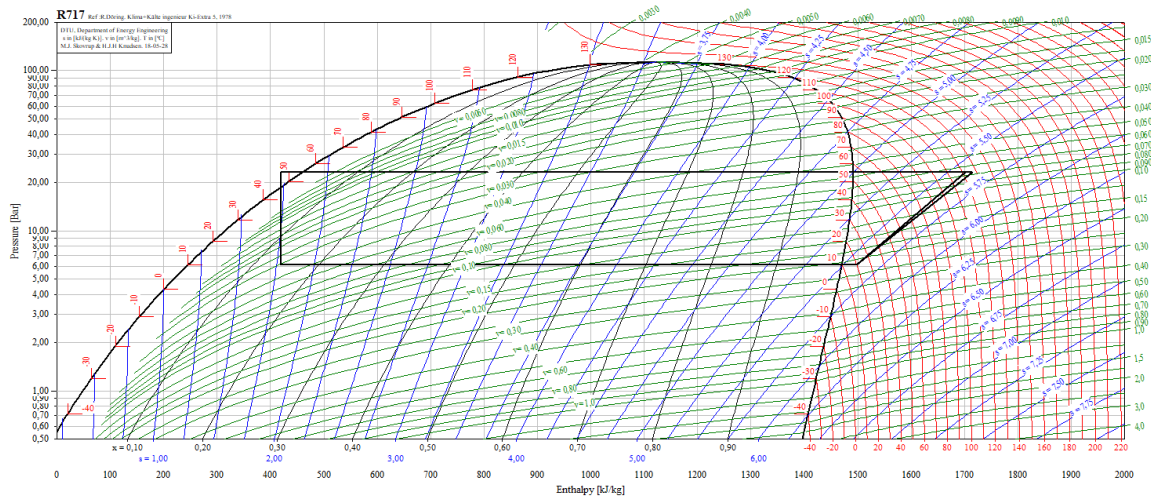
Figure 7.4: *Pressure-specific enthalpy diagram for refrigerant R717, showing the refrigeration cycle at the standard conditions simulations. The compression process is presented with both isentropic efficiency $\eta_{is} = 100$ % and the calculated $\eta_{is} = 94$ % from the simulation of refrigerant R717. The Y-axis presents pressure in unit bar and the X-axis presents the specific enthalpy in unit kJ/kg. The diagram is obtained from CoolPack [54].*

The isentropic efficiency calculated for refrigerants propane and ammonium are $\eta_{is} = 84.8$ % and $\eta_{is} = 93.8$ % respectively. According to literature one of the drawbacks of natural refrigerants are the decreased efficiencies compared to the most used synthetic alternatives [19]. The high isentropic efficiencies obtained from the simulations in this project probably derives from the calculation of the discharge enthalpy of the compressor. This is elaborated in the first subsection of this chapter, regarding the standard conditions simulations. The parameters of the compressor and the other components have not been adapted to fit the natural refrigerants. The various natural refrigerants require system modifications compared to synthetic refrigerant R134a [19]. Propane requires additional safety measures due to the high flammability of the hydrocarbon refrigerant [39, 44]. The safety requirements include limits on the refrigerant charge, which further limits the size and capacity of heat pump and refrigeration systems using propane as refrigerant [44]. Ammonium is corrosive, and hence not compatible with the often-used copper pipes and components. Aluminum may be used as it is more withstanding to the corrosive effect of ammonium. In addition, ammonium is not miscible with lubrication oils, and an oil separator must be added to the discharge line of the compressor [45].

The higher heating capacity of the system when using the natural refrigerants indicate that the system may provide heat to a larger system, or the heat pump system and compressor may be smaller to match the current heat demand of the building. The refrigerant charge may be lower, which is advantageous when considering the flammability of both propane and ammonium and toxicity of ammonium. As for the increased building load simulations, the compressor component may be smaller for the natural refrigerant. This may be accomplished by changing the coefficients and swept volume parameter for compressor Type 206 and changing the swept volume parameter value for compressor Type 207.

7.2   Model improvements

The following section presents suggestions and recommendations for further development of the components created in this project. In addition, suggestions for expansion of the TRNSYS system models are presented. Further development of the components and models should focus on making the models better resemble a real system and represent all operational conditions.

7.2.1   Compressor

This section presents suggestions for further development of the compressor models, and which variables that require further investigation. Some variables and properties of the compressor model apply to both compressor Type 206 and 207. The properties that are specific to one of the compressor models are presented separately in the following subsections.

The compressor models do not account for either heat loss and cooling, effects of lubrication or leakages from the compressor in these simulations. Leakage of refrigerant during compression may affect the volumetric efficiency and result in a varying refrigerant mass flow. The amount of leakage may also vary with load and compression ratio. Leakage through suction and discharge valves of a reciprocating compressor is unavoidable, as the valves cannot open and close instantly [45].

The maximum frequency should have been defined as a parameter instead of being controlled by the iterative feedback control Type 22, similarly to the minimum frequency. This would give an even better visual representation of the heat pump operational range, when all frequencies above the maximum value would appear in the diagram showing the frequency output from the iterative feedback control. Refer to Figure 6.3 in Chapter 6.2 for an example of a frequency-time diagram.

A disadvantage of both compressor models is the disability to reflect the varying isentropic efficiency when the compressor operates at variable speed [10]. This is due to the calculation of the discharge enthalpy and the assumption that the discharge pressure equals the condensing pressure. There are various approaches to this problem found in the literature. Some studies use normalized values to obtain various variables at part load, e.g. as function of the normalized rotational speed [27]. Mendoza-Miranda et al. [8] calculates compressor efficiencies based on dimensionless parameters obtained from experimental data. Nunes et al. [2] calculate efficiency and compressor power input based on the normalized pressure and other dimensionless parameters. Shao et al. [65] calculates refrigerant mass flow rate and compressor power input based on calculated values at the nominal frequency and polynomial functions of the nominal and actual frequency. The same method is applied by Aprea et al. [32]. Byrne et al. [1] uses empirical equations based on experimental data in their TRNSYS compressor component to calculate the isentropic efficiency of the compressor. There are many approaches to the calculation of the isentropic efficiency and the relevant variables necessary, and this should be investigated during any further development of the compressor components.

The compressor models are not concerned with type of compressor. This may be accounted for by the choice of proper parameter values, especially the volumetric efficiency is an important parameter that varies between the various compressor types. The volumetric efficiency may approach unity for screw and scroll compressors, due to the non-existing clearance volume of these compressor types [1, 45]. When simulating compressor Type 206 the coefficients are provided for a certain compressor type and model, and the results are valid only for the relevant compressor. Compressor models found in literature are mostly created for a specific compressor type and specific refrigerant. When models are validated by experimental data the verification process is only valid for the given compressor type,

model and refrigerant used for measurements. Examples of this is the study by Byrne et al. [22] where a scroll compressor for hydrocarbon refrigerants is modelled, and the study of a reciprocating compressor with refrigerant R450A conducted by Mota-Babiloni et al. [74]. One exception is the loss-efficiency compressor model created by Yang et al. [64], which is validated for three types of positive displacement compressors. Generalization of compressor models to fit all types of compressor and refrigerants is difficult due to the varying properties and requirements of both compressor types and refrigerants. The models created in this project are quite generalized, which may impose inaccuracies in the results. The models should be further developed and validated based on experimental data, resulting in a narrower application range for the models.

Type 206

The compressor power input calculated by compressor Type 206 is consistent with data obtained from Bitzer [59] at various frequencies. The calculation of the mass flow rate is more uncertain, as this calculation involves the volumetric efficiency which is assumed to be accounted for in the compressor coefficients. The effect of the volumetric efficiency is discussed in Chapter 7.1. A third compressor model using the 10-coefficient polynomial from AHRI Standard 540 [29] should be developed, in order to compare the calculated mass flow rates and the resulting performance and efficiency to Type 206. When using the polynomial for the calculation of the mass flow rate, the volumetric efficiency may be provided as an output from the model. Other compressor models developed using the polynomial and coefficients for calculation of the compressor power input also use the polynomial and relevant coefficients for the calculation of the refrigerant mass flow rate [28, 39, 63, 65]. Coefficients for reciprocating compressor model 4NE-20.4FY are available from Bitzer software [59]. The compressor coefficients are determined from experimental data measured at standard test conditions according to AHRI Standard 540. The coefficients should account for variations in efficiency and other compressor properties.

The heat loss parameter added to compressor Type 206 may be used to account for either heat loss to the ambient or intentional cooling of the compressor. The heat loss or cooling is accounted for when calculating the discharge enthalpy of the refrigerant, which is consistent with other models available in the literature [64]. The heat loss parameter was not used during the simulations of compressor Type 206 in order to validate the results against the data obtained from Bitzer software [59].

Type 207

The main challenge of compressor Type 207 is the determination of the polytropic exponent's value. The value should be determined experimentally for a given compressor and refrigerant. The standard conditions value of $n = 1.45$ is too low compared to the results and calculated compressor power input of compressor Type 206 which is validated by data from Bitzer. Byrne et al. [22] estimates the polytropic exponent based on the polytropic exponent at nominal conditions and the isentropic exponent at nominal and actual conditions. The isentropic exponent is equal to the specific heat ratio and may be determined for any refrigerant at a given temperature. For compressor Type 207 the mass flow rate is calculated using the volumetric efficiency. If the volumetric efficiency is desired as an output, it may be calculated by an empirical relation as in other models [2, 27]. The polytropic exponent is also used for the calculation of the volumetric efficiency in these models.

Another issue in the polytropic compressor model is the calculation of the isentropic efficiency, which results in a constant isentropic efficiency of the compressor. This is due to the calculation of both the refrigerant mass flow rate and the compressor power input. This is elaborated in Chapter 7.1. The discharge pressure is assumed to be equal to the condensing pressure. The discharge pressure is normally higher than the condensing pressure, resulting in a higher discharge specific enthalpy. Alternative approaches to the calculation of the isentropic efficiency are presented in the previous section.

7.2.2   Condenser and evaporator

The evaporator and condenser component Type 230 is very simplified, considering only the inlet and outlet state of the refrigerant. The heat source and sink properties are not accounted for, and it is assumed that the capacity of the heat source and sink are always sufficient. The results from the various simulations conducted indicate that the simplified evaporator and condenser components impose limitations to the models in terms of system performance evaluation. The calculated coefficient of performance for the simulations is generally high and does not vary much with the changing load and operational conditions. Recommendations and suggestions for further development of Type 230 are presented in the following subsection.

To better reflect a real heat pump system, the heat source and heat sink should be incorporated in the models. The temperature of the heat source and sink should be added to Type 230 as time-dependent inputs, as especially heat sources usually have a varying temperature [46]. The varying heat source and sink temperature will affect the heat transfer rate of the evaporator and condenser, even at constant saturation temperature as the heat transfer rate depends on properties of both the refrigerant and the secondary fluid. Any pressure loss in the evaporator or condenser will affect the saturation temperature, which is a function of the pressure for a saturated liquid-vapor mixture. This is not accounted for in the calculation of the outlet enthalpy of the refrigerant in Type 230.

The heat exchanger component created in this project is a black-box model, concerning only the inlet and outlet conditions of the refrigerant. Common approaches in other models is to divide the heat exchanger in to zones, where the outlet conditions of one zone are the inlet conditions to the next zone. The number of zones determine the level of detail, ranging from only a few zones to the finite element method providing a high accuracy and detail level. In studies by Kinab et al. [28] and Byrne et al. [1] the evaporator is divided into two zones and the condenser into three zones. This is the most utilized method of modelling condenser and evaporator components [23].

There are two dominant approaches to the calculation of heat transfer rates and outlet temperatures in heat exchangers. The LMTD-method (logarithmic mean temperature difference) considers inlet and outlet temperatures of both the refrigerant and the secondary fluid. A heat transfer coefficient for the heat exchanger must be determined. The LMTD-method is often used to determine the necessary size of a heat exchanger when the operational conditions, i.e. temperatures, and the required heat transfer rate is known [75]. The LMTD-method is used in the heat exchanger model developed by Kinab et al. [28] and Byrne et al. [1]. Another approach to heat exchanger modelling is the ε-NTU method (effectiveness-number of transfer units), which is based on the inlet and outlet temperatures and the heat capacity rate of the two fluids. The heat capacity rate is defined as the product of the mass flow rate and the specific heat of the given fluid. The smallest heat capacity rate of the refrigerant and the secondary fluid determines the maximum heat transfer rate of the condenser or evaporator.

Kinab et al. [28] uses the ε-NTU method in combination with the LMTD-method. Both methods are described in textbook Heat and Mass transfer by Çengel and Ghajar [75]. Note that both methods must be adapted to the phase change heat transfer process of the refrigerant. The heat transfer rate and properties of the evaporator and condenser may also be described by the fundamental equations of continuity, energy and momentum as in the study by Koury et al. [23].

7.2.3 Building load

The building load component created in this project is very simplified, and the objective of the component is to simply provide a varying heat demand to the iterative feedback controller Type 22. The building load does not account for effects as internal gains, solar radiation or ventilation air properties. During the warm season of the simulated year, the heat demand of the building load is generally low. This results in the heat pump system switching on and off very often. In a real building, internal gains and solar radiation would result in even lower heat demand during the warmer season. This would decrease the on and off operation of the compressor, and the system would be off for longer periods.

The building load component does not consider the actual distribution system of the building. If a more detailed building load is to be simulated, there are several options in the TRNSYS and TESS component libraries. For a detailed building load, a 3D multizone building may be created in TRNBuild which is a part of the TRNSYS package. This requires extensive information regarding the properties and usage of the building. Other options are the lumped capacitance building component (Type 88), which requires information regarding the ventilation system of the building. The Synthetic building load found in the TESS component library (Type 686) simulates building loads by the maximum heat demand and various sine-functions, allowing the user to define both seasonal and daily variations in heat demand. TRNSYS documentation [56] provides extensive information regarding the numerous components available. The building load component may also include the properties of the heat distribution system of the building, providing a return temperature to the condenser component.

If the TRNSYS system model is adapted to simulate a realistic heat pump system, a more detailed building load component is recommended. The simplified building component created in this project may serve as a basis for further development by the addition of parameters and input variables. The building load component may also be connected to the condenser directly by adapting the condenser component Type 230. The condenser component must be extended to also consider the properties of the heat sink. Recommendations for further development of Type 230 are presented in the previous subsection of this chapter.

7.2.4 TRNSYS models

The TRNSYS system models may be expanded to include several other components and cover more complicated systems than the systems modelled in this project. If the condenser and evaporator component Type 230 is further developed to account for the heat source and heat sink conditions, the heat source and heat sink properties must be included in the TRNSYS system models. The varying temperatures may be provided as an input by for example a forcing function (Type 14) or a data reader component (Type 9).

The models may also be expanded to include the auxiliary top load system, by incorporating an electric water heater in the system. This would require more complicated control functions of the system,

including an on/off control of the electric water heater. The load of the system is not usually controlled by the difference in heating capacity and heat demand as in the models created in this project. If the building load and condenser components were further developed, the system should be controlled by the return temperature to the condenser from the distribution system of the building or by the ambient temperature [46].

8 Conclusions

TRNSYS may be used for simulations of vapor compression cycles and heat pump systems on component level. Components may be created by the user and implemented in TRNSYS models. The compressor and heat pump components created provide results consistent with the energy balance for heat pumps. The components are created as a starting point for the development of more complex compressor and heat pump components.

The polynomial based compressor is validated and found to be consistent with available manufacturer data. However, the two created compressor models do not provide consistent results in their test environment in TRNSYS. The total energy consumption of the polytropic compressor is 21 % lower than the total energy consumption of the polynomial-based compressor. The calculated seasonal coefficient of performance for the polynomial based compressor is consistent with typical values found in the literature. The continuous value of COP and the isentropic efficiency is not as consistent with theoretical predictions or values found in literature. This may be due to both calculations within the compressor models and the simplified evaporator and condenser component. The variable speed heat pump components created based on the two compressor components are consistent with results of the compressor models.

The models and components generally work as expected when changing various parameter values. The polynomial based compressor has certain limitations to parameter values but does not seem more sensitive to parameter value changes than the polytropic compressor.

The created components are considered a good starting point for further development and work in TRNSYS on component level. Future developments should be focused on better reflecting the performance of a real heat pump system under varying operational conditions. The calculation of the discharge enthalpy in both compressor components should be evaluated. A third compressor model should be developed, using the 10-coefficient polynomial for both compressor power input and refrigerant mass flow rate. A thorough sensitivity analysis of all loss parameters is recommended, and other approaches to account for the various losses should be explored. A test bench facility should be established at HVL to enable validation and further development of the components.

The evaporator and condenser component should be expanded to account for the secondary fluid and the properties of the heat source and sink. The TRNSYS models may be expanded to include any auxiliary equipment and a more detailed control function.

9   References

1.   Paul Byrne, Jacques Miriel, and Yves Lénat, *Modelling and simulation of a heat pump for simultaneous heating and cooling.* Building Simulation, 2012. 5: p. 219-233.
2.   Nunes, T.K., et al., *Modeling, simulation and optimization of a vapor compression refrigeration system dynamic and steady state response.* Applied Energy, 2015. 158: p. 540-555.
3.   Europoean Heat Pump Association. *Heat Pump Sales*. 2018 [22.01.2018]; Available from: http://stats.ehpa.org/hp_sales/country_cards/.
4.   Norwegian Ministry of Climate and Environment. *Enova*. 2018; Available from: https://www.enova.no.
5.   United Nations Environment Programme. *The Kigali Amendment*. 2016 [30.01.2018]; Available from: http://ozone.unep.org/en/handbook-montreal-protocol-substances-deplete-ozone-layer/41453.
6.   Umwelt Bundesamt. *EU Regulation concerning fluorinated greenhouse gases*. 2016 [30.01.2018]; Available from: https://www.umweltbundesamt.de/en/topics/economics-consumption/products/fluorinated-greenhouse-gases-fully-halogenated-cfcs/statutes-regulations/eu-regulation-concerning-fluorinated-greenhouse#textpart-1.
7.   Statistisk Sentralbyrå. *Nedgang i klimagassutslippene*. 2017 [22.01.2018]; Available from: http://www.ssb.no/natur-og-miljo/artikler-og-publikasjoner/nedgang-i-klimagassutslippene.
8.   Mendoza-Miranda, J.M., et al., *Comparative evaluation of R1234yf, R1234ze(E) and R450A as alternatives to R134a in a variable speed reciprocating compressor.* Energy, 2016. 114: p. 753-766.
9.   Aprea, C., R. Mastrullo, and C. Renno, *Experimental analysis of the scroll compressor performances varying its speed.* Applied Thermal Engineering, 2006. 26(10): p. 983-992.
10. Tassou, S.A. and T.Q. Qureshi, *Comparative performance evaluation of positive displacement compressors in variable-speed refrigeration applications.* International Journal of Refrigeration, 1998. 21(1): p. 29-41.
11. Sârbu, I. and C. Sebarchievici, *Ground-source heat pumps : fundamentals, experiments and applications*. 2016, Academic Press.
12. Widell, K.N. and T. Eikevik, *Reducing power consumption in multi-compressor refrigeration systems.* International Journal of Refrigeration, 2010. 33(1): p. 88-94.
13. O'Neill, P.A., *Industrial Compressors - Theory and Equipment*. 1993: Butterworth-Heinemann.
14. Adhikari, R.S., et al., *Energy Savings through Variable Speed Compressor Heat Pump Systems.* Energy Procedia, 2012. 14: p. 1337-1342.
15. Cuevas, C. and J. Lebrun, *Testing and modelling of a variable speed scroll compressor.* Applied Thermal Engineering, 2009. 29(2): p. 469-478.
16. Çengel, Y.A. and M.A. Boles, *Thermodynamics - an engineering approach*. 7 ed. 2011: McGraw Hill.
17. Abagnale, C., et al., *Ideal Specific Work of Rotary Compressors: A New Approach.* Energy Procedia, 2016. 101: p. 710-717.
18. Misra, N.N., et al., *Landmarks in the historical development of twenty first century food processing technologies.* Food Research International, 2017. 97: p. 318-339.
19. Calm, J.M., *The next generation of refrigerants – Historical review, considerations, and outlook.* International Journal of Refrigeration, 2008. 31(7): p. 1123-1133.
20. Sanner, B., *Ground Source Heat Pumps - history, development, current status and future prospects.* 12th IEA Heat Pump Conference, 2017.
21. Rasmussen, B.D. and A. Jakobsen, *Review of Compressor Models and Performance Characterizing Variables.* International Compressor Engineering Conference, 2000.
22. Byrne, P., R. Ghoubali, and J. Miriel, *Scroll compressor modelling for heat pumps using hydrocarbons as refrigerants.* International Journal of Refrigeration, 2014. 41: p. 1-13.

23. Koury, R.N.N., L. Machado, and K.A.R. Ismail, *Numerical simulation of a variable speed refrigeration system.* International Journal of Refrigeration, 2001. 24(2): p. 192-200.

24. Balduzzi, F., et al., *Two-dimensional approach for the numerical simulation of large bore reciprocating compressors thermodynamic cycle.* Applied Thermal Engineering, 2018. 129: p. 490-501.

25. Roskosch, D., V. Venzik, and B. Atakan, *Thermodynamic model for reciprocating compressors with the focus on fluid dependent efficiencies.* International Journal of Refrigeration, 2017. 84: p. 104-116.

26. Castaing-Lasvignottes, J. and S. Gibout, *Dynamic simulation of reciprocating refrigeration compressors and experimental validation.* International Journal of Refrigeration, 2010. **33**(2): p. 381-389.

27. Li, W., *Simplified steady-state modeling for variable speed compressor.* Applied Thermal Engineering, 2013. 50(1): p. 318-326.

28. Kinab, E., et al., *Reversible heat pump model for seasonal performance optimization.* Energy and Buildings, 2010. 42(12): p. 2269-2280.

29. Air-conditioning Heating & Refrigeration Institute, *Standard 540: Standard for Performance Rating of Positive Displcement Refrigerant Compressors and Compressor Units*. 2015.

30. Lemmon, E.W., Huber, M.L., McLinden, M.O., *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 9.1, National Institute of Standards and Technology,*. 2013.

31. Safa, A.A., A.S. Fung, and R. Kumar, *Performance of two-stage variable capacity air source heat pump: Field performance results and TRNSYS simulation.* Energy and Buildings, 2015. 94: p. 80-90.

32. Aprea, C., R. Mastrullo, and C. Renno, *Determination of the compressor optimal working conditions.* Applied Thermal Engineering, 2009. 29(10): p. 1991-1997.

33. Wang, Z., et al., *Research of thermal dynamic characteristics for variable load single screw refrigeration compressor with different capacity control mechanism.* Applied Thermal Engineering, 2017. 110: p. 1172-1182.

34. Jakobsen, A., B.D. Rasmussen, and J. Suess, *Impact of Refrigerant Fluid Properties on the Compressor Selection.* International Compressor Engineering Conference, 2000.

35. Brendeng, E., *Reciprocating compressors or screw compressors?* International Journal of Refrigeration, 1979. 2(3): p. 163-170.

36. Zhang, L., et al., *Advances in vapor compression air source heat pump system in cold regions: A review.* Renewable & Sustainable Energy Reviews, 2018. 81: p. 353-365.

37. Ahamed, J.U., R. Saidur, and H.H. Masjuki, *A review on exergy analysis of vapor compression refrigeration system.* Renewable and Sustainable Energy Reviews, 2011. 15(3): p. 1593-1600.

38. Sarkar, J., *Ejector enhanced vapor compression refrigeration and heat pump systems—A review.* Renewable and Sustainable Energy Reviews, 2012. 16(9): p. 6647-6659.

39. Nawaz, K., et al., *R290 (propane) and R600a (isobutane) as natural refrigerants for residential heat pump water heaters.* Applied Thermal Engineering, 2017. 127: p. 870-883.

40. Choudhari, C.S. and S.N. Sapali, *Performance Investigation of Natural Refrigerant R290 as a Substitute to R22 in Refrigeration Systems.* Energy Procedia, 2017. 109: p. 346-352.

41. Venzik, V., D. Roskosch, and B. Atakan, *Propene/isobutane mixtures in heat pumps: An experimental investigation.* International Journal of Refrigeration, 2017. 76: p. 84-96.

42. Cecchinato, L., et al., *Carbon dioxide as refrigerant for tap water heat pumps: A comparison with the traditional solution.* International Journal of Refrigeration, 2005. 28(8): p. 1250-1258.

43. Ma, Y., Z. Liu, and H. Tian, *A review of transcritical carbon dioxide heat pump and refrigeration cycles.* Energy, 2013. 55: p. 156-172.

44. Ciconkov, R., *Refrigerants: There is still no vision for sustainable solutions.* International Journal of Refrigeration, 2018. 86: p. 441-448.

45. Dossat, R.J.H., Thomas J, *Principles of Refrigeration*. 5 ed. 2002.

46. Kompetansebiblioteket, *Prenøk - Prosjektering av energianlegg*. 2016.

47. Steinar Sundberg, UiB/EIA, *Personal communication*. 2018.

48. Ding, G.-l., *Recent developments in simulation techniques for vapour-compression refrigeration systems.* International Journal of Refrigeration, 2007. 30(7): p. 1119-1133.

49. Giampaolo, T., *Compressor Handbook - Principles and Practice*. 2010: The Fairmont Press.

50. Fakhari;, A., *Heat Exchanger Efficiency.* Journal of Heat Transfer, 2007.

51. United Nations Environment Programme. *The Montreal Protocol on Substances that Deplete the Ozone Layer*. 2016 [30.01.2018]; Available from: http://ozone.unep.org/en/treaties-and-decisions/montreal-protocol-substances-deplete-ozone-layer.

52. ASHRAE, *Standard 34: Designation and Safety Classification of Refrigerants*. 2007, American National Standards Institute.

53. Standard Norge, *NS-EN 378-1: Kuldeanlegg og varmepumper - Sikkerhets- og miljøkrav - Del 1: Grunnleggende krav, definisjoner, klassifisering og vurderingskriterier*. 2017.

54. Jakobsen, A., B.D. Rasmussen, and S.E. Andersen, *CoolPack – Simulation tools for refrigeration systems.* Scan Ref, 1999. 28: p. 7-10.

55. Makhnatch, P., A. Mota-Babiloni, and R. Khodabandeh, *Experimental study of R450A drop-in performance in an R134a small capacity refrigeration unit.* International Journal of Refrigeration, 2017. 84: p. 26-35.

56. Thermal Energy Systems Specialists, *TRNSYS Documentation Vol. 1-10*. 2017.

57. Davies, R., A. Rea, and D. Tsaptsinos, *Introduction to Fortran 90 - An introduction for Novice Programmers*, Cardiff London and Belfast HPC T&E Centres.

58. Vikrant Aute, C.M., Reinhard Radermarcher, *A study of methods to represent compressor performance data over an operating envelope based on a finite set of test data*. 2015, Air-conditioning, Heating and Refrigeration Institute.

59. Bitzer. *Bitzer Software, .* [27.04.18]; Available from: https://www.bitzer.de/websoftware/.

60. Shen, B., O. Abdelaziz, and C.K. Rice, *Development of a cold climate heat pump using two-stage compression*, in *24th IIR International Congress of Refrigeration*. 2015: Yokohama, Japan.

61. Cheung, H., O. Sarfraz, and C.K. Bach, *A method to calculate uncertainty of empirical compressor maps with the consideration of extrapolation effect and choice of training data.* Science and Technology for the Built Environment, 2017: p. 1-16.

62. Arjun, R., S. Lukasz, and B. Ian, *Compressor map prediction tool.* IOP Conference Series: Materials Science and Engineering, 2015. 90(1): p. 012042.

63. Park, Y.C., Y.C. Kim, and M.-K. Min, *Performance analysis on a multi-type inverter air conditioner.* Energy Conversion and Management, 2001. 42(13): p. 1607-1621.

64. Yang, L., et al., *Loss-efficiency model of single and variable-speed compressors using neural networks.* International Journal of Refrigeration, 2009. 32(6): p. 1423-1432.

65. Shao, S., et al., *Performance representation of variable-speed compressor for inverter air conditioners based on experimental data.* International Journal of Refrigeration, 2004. 27(8): p. 805-815.

66. Dorin. *Dorin Software*. [27.04.18]; Available from: http://www.dorin.com/en/Software/.

67. Johnson Controls. *Sabroe* [27.04.18]; Available from: http://www.sabroe.com/en/home/.

68. Mayeakawa. *Mayekawa Mycom*. [27.04.18]; Available from: http://www.mayekawa.com/.

69. Bell, I.H.a.W., Jorrit and Quoilin, Sylvain and Lemort, Vincent, *Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp.* Industrial \& Engineering Chemistry Research, 2014. 53(6).

70. Pottker, G. and P. Hrnjak, *Effect of the condenser subcooling on the performance of vapor compression systems.* International Journal of Refrigeration, 2015. 50: p. 156-164.

71. SINTEF Byggforsk, *451.021 Klimadata for teknisk dimensjonering og frostsikring*, in *Byggforskserien*, SINTEF, Editor. 2012, SINTEF.

72.   Norwegian Meteorological Institute. *eklima*. 2018; Available from: http://sharki.oslo.dnmi.no/portal/page?_pageid=73,39035,73_39049&_dad=portal&_schema=PORTAL&6009_BATCHORDER_3197941.

73.   Hanne Midttveit and Elisabeth Vindenes, *DHW consumption in Norway – Consumption patterns, dimensioning and energy consumption*. 2016, Western Norway University of Applied Sciences and Engineering.

74.   Mota-Babiloni, A., et al., *Experimental study of an R1234ze(E)/R134a mixture (R450A) as R134a replacement.* International Journal of Refrigeration, 2015. 51: p. 52-58.

75.   Yunus A. Çengel and Afshin J. Ghajar, *Heat and Mass Transfer: Fundamentals and Applications*. 5 ed. 2014: McGraw-Hill Education.

Appendix I   Source codes for created components

```fortran
SUBROUTINE Type206

!**********************************************************

! This model models a compressor using polynomial from EN12900 for
compressor power.

!

! Y = C1 + C2*Te + C3*Tc + C4*Te^2 + C5*Tc*Te + C6*Tc^2 + C7*Te^3 +
C8*Tc*Te^2 + C9*Tc^2*Te + C10*Tc^3

!

! Where Y = compressor power consumption [W]

!

! and C1 ... C10 coefficients provided by manufacturer.

!

!      Parameters:

!      1. T_e          Evaporating temperature [C]

!      2. T_c          Condensing temperature [C]

!      3. V_sw         Swept volume of compressor [m^3]

!      4. nref         Refrigerant

!      5. T_SH         Degree of superheat [C]

!      6. P_loss    Fractional pressure loss in suction line [-]

!      7. f_min     Minimum frequency [1/s]

!      8. n_vol     Volumetric efficiency [-]

!      9. r_loss    Fractional heat loss/cooling [-]

!      10. LU1         Logical unit number [-]

!      11. LU2         Logical unit number [-]

!      12. LU3         Logical unit number [-]

!

!      Inputs:

!      1. m_suc        Suction mass flow rate [kg/h]

!      2. f            Frequency [1/s]

!

!      Outputs:

!      1. W_in         Compressor power [W]

!      2. n_is         Isentropic efficiency [-]

!      3. rpm          Rotational speed [1/min]

!      4. m_suc        Mass flow rate [kg/h]
```

```fortran
!     5. v_suc         Specific volume [m3/kg]
!     6. h_dis         Discharge enthalpy [kJ/kg]
!     Thermodynamic properties:
!
!     1. h             Enthalpy [kJ/kg]
!     2. p             Pressure [Pa]
!     3. T             Temperature [C]
!     4. qu            Quality    [-]
!     5. v             Specific volume [m3/kg]
!     6. s             Entropy    [kJ/kg]
!
!export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE206
!-------------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-------------------------------------------------------------------------
! Variable declerations
IMPLICIT NONE          ! All variables are explicitly defined
!Parameters:
REAL :: T_e, T_c, V_sw, T_SH, P_loss, f_min, r_loss, n_vol
INTEGER :: nref, LU1, LU2, LU3
!Inputs
REAL :: m_suc, f
!Outputs
REAL :: W_in, n_is, rpm, v_suc, h_dis
!Thermodynamic properties
REAL :: qu, P_e, h_e, v_e, s_e, P_c, T_suc, P_suc, h_suc, s_suc, P_dis,
T_dis, v_dis, h_is
!Others
REAL :: Time, Timestep, X(7), Y(10), U_sw, U_re, C1, C2, C3, C4, C5, C6,
C7, C8, C9, C10, prop(9), w
INTEGER :: iType, iFlagr, ios, CurrentUnit, CurrentType, nY, NX, nIND, LU,
i
```

```fortran
!------------------------------------------------------------------------
!Get the Global Trnsys Simulation Variables
      Time = getSimulationTime()
      Timestep = getSimulationTimeStep()
      CurrentUnit = getCurrentUnit()
      CurrentType = getCurrentType()
!------------------------------------------------------------------------

!------------------------------------------------------------------------
!Set the Version Number for This Type
      If(getIsVersionSigningTime()) Then
          Call SetTypeVersion(17)
          Return
      Endif
!------------------------------------------------------------------------

!------------------------------------------------------------------------
!Do All of the Last Call Manipulations Here
      If(getIsLastCallofSimulation()) Then
          Return
      Endif
!------------------------------------------------------------------------

!------------------------------------------------------------------------
!Perform Any "End of Timestep" Manipulations That May Be Required
      If(getIsEndOfTimestep()) Then
          Return
      Endif
!------------------------------------------------------------------------

!------------------------------------------------------------------------
!Do All of the "Very First Call of the Simulation Manipulations" Here
      If(getIsFirstCallofSimulation()) Then
!  Tell the TRNSYS Engine How This Type Works
    Call SetNumberofParameters(12)
    Call SetNumberofInputs(2)
    Call SetNumberofDerivatives(0)
    Call SetNumberofOutputs(7)
    Call SetIterationMode(1)
    Call SetNumberStoredVariables(0,0)    !(nStatic,nDynamic)
```

```
        Call SetNumberofDiscreteControls(0)


    Return

 EndIf

!-------------------------------------------------------------------------

!-------------------------------------------------------------------------

!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time

 If (getIsStartTime()) Then

! Read in the Values of the Parameters from the Input File

        T_e=(getParameterValue(1))

        T_c=(getParameterValue(2))

        V_sw=(getParameterValue(3))

        nref=(getParameterValue(4))

        T_SH=(getParameterValue(5))

        P_loss=(getParameterValue(6))

        f_min=(getParameterValue(7))

        n_vol=(getParameterValue(8))

        r_loss=(getParameterValue(9))

        LU1=(getParameterValue(10))

        LU2=(getParameterValue(11))

        LU3=(getParameterValue(12))


! check parameters for errors

        m_suc=(getInputValue(1))

        f=(getInputValue(2))

!  Set the Initial Values of the Outputs

     Call setOutputValue(1,0.0)

        Call setOutputValue(2,0.0)

        Call setOutputValue(3,0.0)

        Call setOutputValue(4,getInputValue(1))

        Call setOutputValue(5,0.0)

        Call setOutputValue(6,0.0)

        Call setOutputValue(7,0.0)

        Return

   EndIf
```

```fortran
!-----------------------------------------------------------------------
!ReRead the Parameters if Another Unit of This Type Has Been Called Last
   If(getIsReReadParameters()) Then
!  Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))
      T_c=(getParameterValue(2))
      V_sw=(getParameterValue(3))
      nref=(getParameterValue(4))
      T_SH=(getParameterValue(5))
      P_loss=(getParameterValue(6))
      f_min=(getParameterValue(7))
      n_vol=(getParameterValue(8))
      r_loss=(getParameterValue(9))
      LU1=(getParameterValue(10))
      LU2=(getParameterValue(11))
      LU3=(getParameterValue(12))
   Endif
!-----------------------------------------------------------------------
! Get current input values
      m_suc=(getInputValue(1))
      f=(getInputValue(2))
If (f<f_min) Then
      f=0.0
      rpm=0.0
      W_in=0.0
      m_suc=0.0
      n_is=0.0
ElseIf (f>=f_min)Then
! Check input values
If (m_suc<0) Call FoundBadInput(1,'Fatal','The suction mass flow rate
cannot be negative')
If (f<0) Call FoundBadInput(2,'Fatal','The frequency cannot be negative')
```

```fortran
!     Interpolate coefficients to match frequency
! Use correct data file
      If (nref==134) Then
            LU=LU1
      ElseIf (nref==290) Then
            LU=LU2
      ElseIf (nref==717) Then
            LU=LU3
      EndIf


      nX=7              ! There are 7 values of variable f in the file
      X(1)=f            ! Sets X to local variable f
      nY=10             ! There are 10 coefficients to each value of f
      nIND=1            ! There is 1 independent variable (frequency)


            Call InterpolateData(LU,nIND,nX,nY,X,Y)


            C1 = Y(1)
            C2 = Y(2)
            C3 = Y(3)
            C4 = Y(4)
            C5 = Y(5)
            C6 = Y(6)
            C7 = Y(7)
            C8 = Y(8)
            C9 = Y(9)
            C10 = Y(10)


! Calculate compressor power


W_in =
C1+(C2*T_e)+(C3*T_c)+(C4*T_e**2)+(C5*T_c*T_e)+(C6*T_c**2)+(C7*T_e**3)+(C8*T
_c*T_e**2)+(C9*T_c**2*T_e)+(C10*T_c**3)


! Calculate refrigerant properties
! Thermodynamic properties at evaporator outlet
```

```fortran
      qu = 1                                ! Saturated vapor state

      prop(1)=T_e

      prop(5)=qu

      Call Coolprop_fluid_properties(prop,nref,15,iFlagr)

            If (ErrorFound()) Return

      P_e = prop(2)

      h_e = prop(3)

      s_e = prop(4)

      v_e = prop(6)
!     Thermodynamic properties at suction state
If ((T_SH>0.d0) .or. (P_loss>0.d0)) Then
! Not saturated, P and T may be used

            T_suc=T_e+T_SH

            P_suc=P_e-(P_e*P_loss)

            prop(1)=T_suc

            prop(2)=P_suc

            Call coolprop_fluid_properties(prop,nref,12,iFlagr)

                If (ErrorFound()) Return

            h_suc=prop(3)

            s_suc=prop(4)

            v_suc=prop(6)
ElseIf ((T_SH==0.d0) .and. (P_loss==0.d0)) Then
!Saturated, P and T can not be used in combination as known properties

            P_suc=P_e

            h_suc=h_e

            s_suc=s_e

            v_suc=v_e

      EndIf


! Calculate mass flow rate

      rpm=((f*60.0)/2.0)              ! Assuming 4 poles in electric motor

      w=(rpm/60.0)                    ! Revolutions per second

      m_suc=3600.0*w*V_sw*n_vol/v_suc   ! Mass flow rate in kg/h


! Find condensing pressure

      qu=0                            ! Saturated liquid
```

```fortran
        prop(1)=T_c

        prop(5)=qu

        Call Coolprop_fluid_properties(prop,nref,15,iFlagr)

            If (ErrorFound()) Return

        P_c=prop(2)


! Thermodynamic properties at discharge state

        P_dis=P_c   ! Assuming discharge pressure equals condenser pressure

        h_dis = h_suc+(((W_in/1000.0)*(1-r_loss))/(m_suc/3600.0))
        !m_suc in kg/s, W_in in kW --> h in kJ/kg

        prop(2)=P_dis

        prop(3)=h_dis

        Call Coolprop_fluid_properties(prop,nref,23,iFlagr)

            If (ErrorFound()) Return

        T_dis=prop(1)

        v_dis=prop(6)


! Calculate isentropic efficiency

        prop(2)=P_dis

        prop(4)=s_suc

        Call Coolprop_fluid_properties(prop,nref,24,iFlagr)

            If (ErrorFound()) Return

        h_is=prop(3)

        n_is=((h_is-h_suc)/(h_dis-h_suc))
EndIf


! Set the outputs from this model
        Call SetOutputValue(1,W_in)

        Call SetOutputValue(2,n_is)

        Call SetOutputValue(3,rpm)

        Call SetOutputValue(4,m_suc)

        Call SetOutputValue(5,v_suc)

        Call SetOutputValue(6,h_dis)

        Call setOutputValue(7,P_c)

        Return

End Subroutine Type206
```

```
SUBROUTINE Type207

!*****************************************************************

! This model models a polytropic compressor

!

!     W_in = m*P_e*v_suc(n/(n-1))((P_c/P_e)^((n-1)/n)-1)

!

!     where      W_in is the compressor power [W]

!     and        n is the polytropic exponent

!

!     Parameters:

!     1. T_e              Evaporating temperature [C]

!     2. T_c              Condensing temperature [C]

!     3. V_sw             Swept volume of compressor [m^3]

!     4. nref             Refrigerant

!     5. T_SH             Degree of superheat

!     6. P_loss   Fractional pressure loss in suctionline [-]

!     7. n           Polytropic exponent

!     8. n_vol    Volumetric efficiency [-]

!     9. f_min    Minimum frequency [1/s]

!

!     Inputs:

!     1. f        Frequency [1/s]

!     2. m_suc    Mass flow rate [kg/h]

!

!     Outputs:

!     1. W_in          Compressor power [W]

!     2. n_is          Isentropic efficiency [-]

!     3. m_suc    Refrigerant mass flow rate [kg/h]

!     4. rpm           Rotational speed [1/min]

!     5. h_dis    Discharge enthalpy [kJ/kg]

!     6. T_dis    Discharge temperature [C]

!

!     Thermodynamic properties:

!     1. h        Enthalpy [kJ/kg]

!     2. P        Pressure [kPa]

!     3. T        Temperature [C]
```

```fortran
!     4. qu        Quality [-]
!     5. v         Specific volume [m3/kg]
!     6. s         Entropy [kJ/kg]
!
!export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE207
!-------------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-------------------------------------------------------------------------
! Variable declerations
IMPLICIT NONE     ! All variables are explicitly defined


!Parameters:
REAL :: T_e, T_c, V_sw, T_SH, P_loss, n, n_vol, f_min
INTEGER :: nref


!Inputs:
REAL :: f


!Outputs:
REAL :: W_in, n_is, m_suc, rpm, h_dis, T_dis


!Thermodynamic properties:
REAL :: qu, P_e, P_c, T_suc, P_suc, h_suc, v_suc, s_suc, P_dis, v_dis,
s_dis, h_is


!Others
REAL :: Time, Timestep, prop(9), w
INTEGER :: CurrentUnit, CurrentType, iType, iFlagr


!-------------------------------------------------------------------------
!Get the Global Trnsys Simulation Variables
     Time = getSimulationTime()
     Timestep = getSimulationTimeStep()
```

146

```fortran
        CurrentUnit = getCurrentUnit()

        CurrentType = getCurrentType()

!------------------------------------------------------------------------

!------------------------------------------------------------------------

!Set the Version Number for This Type

        If(getIsVersionSigningTime()) Then

            Call SetTypeVersion(17)

            Return

        Endif

!------------------------------------------------------------------------

!------------------------------------------------------------------------

!Do All of the Last Call Manipulations Here

        If(getIsLastCallofSimulation()) Then

            Return

        Endif

!------------------------------------------------------------------------

!------------------------------------------------------------------------

!Perform Any "End of Timestep" Manipulations That May Be Required

        If(getIsEndOfTimestep()) Then

            Return

        Endif

!------------------------------------------------------------------------

!------------------------------------------------------------------------

!Do All of the "Very First Call of the Simulation Manipulations" Here

        If(getIsFirstCallofSimulation()) Then

!   Tell the TRNSYS Engine How This Type Works

        Call SetNumberofParameters(9)

        Call SetNumberofInputs(2)

        Call SetNumberofDerivatives(0)

        Call SetNumberofOutputs(6)

        Call SetIterationMode(1)

        Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)

        Call SetNumberofDiscreteControls(0)

    Return

 EndIf

!------------------------------------------------------------------------
```

```
!-----------------------------------148-------------------------------------
!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time
 If (getIsStartTime()) Then
! Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))

      T_c=(getParameterValue(2))

      V_sw=(getParameterValue(3))

      nref=(getParameterValue(4))

      T_SH=(getParameterValue(5))

      P_loss=(getParameterValue(6))

      n=(getParameterValue(7))

      n_vol=(getParameterValue(8))

      f_min=(getParameterValue(9))



      f=(getInputValue(1))

      m_suc=(getInputValue(2))


!  Set the Initial Values of the Outputs
     Call setOutputValue(1,0.0)

        Call setOutputValue(2,0.0)

        Call setOutputValue(3,1.0)

        Call setOutputValue(4,0.0)

        Call setOutputValue(5,0.0)

        Call setOutputValue(6,0.0)

        Return

   EndIf

!-------------------------------------------------------------------------
!ReRead the Parameters if Another Unit of This Type Has Been Called Last
   If(getIsReReadParameters()) Then
!  Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))

      T_c=(getParameterValue(2))

      V_sw=(getParameterValue(3))
```

```fortran
        nref=(getParameterValue(4))
        T_SH=(getParameterValue(5))
        P_loss=(getParameterValue(6))
        n=(getParameterValue(7))
        n_vol=(getParameterValue(8))
        f_min=(getParameterValue(9))
    EndIf


!----------------------------------------------------------------------
! Get current input values


        f=(getInputValue(1))
        m_suc=(getInputValue(2))


If (f<f_min) Then
        W_in=0.0
        f=0.0
        rpm=0.0
        m_suc=0.0
        n_is=0.0
ElseIf (f>=f_min) Then


! Calculate condensing and evaporating pressures


        qu=1.0                         ! Saturated vapor state


        prop(1)=T_e
        prop(5)=qu


        Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return


        P_e=prop(2)


        qu=0.0                         ! Saturated liquid state
```

```fortran
      prop(1)=T_c
      prop(5)=qu


      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return


      P_c=prop(2)



! Properties at compressor inlet state (suction)


If ((T_SH>0.0) .or. (P_loss>0.0)) Then
! Not saturated, P and T may be used
      T_suc=T_e+T_SH
      P_suc=P_e-(P_e*P_loss)
      prop(1)=T_suc
      prop(2)=P_suc
      Call coolprop_fluid_properties(prop,nref,12,iFlagr)
            If (ErrorFound()) Return
      h_suc=prop(3)
      s_suc=prop(4)
      v_suc=prop(6)
ElseIf ((T_SH==0.0) .and. (P_loss==0.0)) Then
!Saturated, P and T can not be used in combination as known properties
      qu=1.0
      prop(1)=T_e
      prop(5)=qu
      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return
      P_suc=prop(2)
      h_suc=prop(3)
      s_suc=prop(4)
      v_suc=prop(6)
EndIf


! Calculate new mass flow rate
```

```
    rpm=(f*60)/2                    ! Assuming 4 poles in electric motor

    w=(rpm/60.)                     ! Revolutions per second

    m_suc=w*V_sw*n_vol/v_suc    ! Mass flow rate in kg/s


! Calculate compressor power input

! in ratio P_c/P_suc no correction necessary. To obtain W_in in kW P_suc
must be in kPa in first part of equation.

W_in=(m_suc*(P_suc/1000)*v_suc*(n/(n-1)))*(((P_c/P_suc)**((n-1.0)/n))-1.0)

! W_in in kW


! Properties at discharge statements

    P_dis=P_c

! Assuming discharge pressure is equal to condensing pressure

    h_dis=h_suc+(W_in/m_suc)            ! Heat loss accounted for in n


    prop(2)=P_dis

    prop(3)=h_dis

    Call coolprop_fluid_properties(prop,nref,23,iFlagr)

        If (ErrorFound()) Return

    T_dis=prop(1)

    v_dis=prop(6)


! Calculate isentropic efficiency

    ! Isentropic compression

    P_dis=P_c

! Assuming discharge pressure is equal to condenser pressure

    s_dis=s_suc                         ! Isentropic compression

    prop(2)=P_dis

    prop(4)=s_dis

    Call coolprop_fluid_properties(prop,nref,24,iFlagr)

        If (ErrorFound()) Return

    h_is=prop(3)


    n_is=(h_is-h_suc)/(h_dis-h_suc)
```

151

```
! Unit conversions
      m_suc=m_suc*3600.0              ! Mass flow rate in kg/h
      W_in=W_in*1000.0               ! Compressor power in W
EndIf


! Set the outputs from this model
      Call SetOutputValue(1,W_in)
      Call SetOutputValue(2,n_is)
      Call SetOutputValue(3,m_suc)
      Call SetOutputValue(4,rpm)
      Call SetOutputValue(5,h_dis)
      Call setOutputValue(6,T_dis)


      Return


End Subroutine Type207
```

```fortran
SUBROUTINE Type220

! This type models a throttling device for use in vapor compression cycles.
!     Parameters:
!     1. nref          Refrigerant [-]
!     2. T_e           Evaporating temperature [C]
!     3. T_c           Condensing temperature [C]
!     4. T_SC          Degree of subcooling [C]
!
!     Inputs:
!     1. m        Mass flow rate [kg/h]
!
!     Outputs:
!     1. m        Mass flow rate [kg/h]
!     2. P_e          Evaporating pressure [kPa]
!     3. qu_out   Vapor quality at outlet [-]
!     4. T_out    Temperature at outlet [C]
!     5. h_out    Enthalpy at outlet    [kJ/kg]
!
!export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE220
!-----------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-----------------------------------------------------------------------
!     Variable declerations
IMPLICIT NONE     ! All variables are explicitly defined


!     Parameters:
          INTEGER :: nref
          REAL :: T_e, T_c,T_SC


!     Inputs:
          REAL :: m
```

```fortran
!       Outputs:
            REAL :: P_out              ! Pressure at outlet
            REAL :: qu_out             ! Quality out of expansion valve
            REAL :: T_out              ! Temperature at outlet
            REAL :: h_out              ! Enthalpy at outlet


!       Thermodynamic properties:
            REAL :: h_in, T_in, v_in, v_out, qu, P_c, P_e


!       Others:
            REAL :: Time, timestep, prop(9)
            INTEGER :: CurrentUnit, CurrentType, iFlagr, iType, i


!-------------------------------------------------------------------------
! Get the global TRNSYS Simulation Variables
            Time=getSimulationTime()
            Timestep=getSimulationTimestep()
            CurrentUnit=getCurrentUnit()
            CurrentType=getCurrentType()


!-------------------------------------------------------------------------
!-------------------------------------------------------------------------
!Set the version number for this type
        If (getIsVersionSigningTime()) Then
            Call SetTypeVersion(17)
            Return
        EndIf
!-------------------------------------------------------------------------
!-------------------------------------------------------------------------
!Do All of the Last Call Manipulations Here
        If(getIsLastCallofSimulation()) Then
            Return
        Endif
!-------------------------------------------------------------------------
```

```
!----------------------------------------------------------------------
!Do All of the "Very First Call of the Simulation Manipulations" Here
      If(getIsFirstCallofSimulation()) Then
!  Tell the TRNSYS Engine How This Type Works
            Call SetNumberofParameters(4)

            Call SetNumberofInputs(1)

            Call SetNumberofDerivatives(0)

            Call SetNumberofOutputs(5)

            Call SetIterationMode(1)

            Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)

            Call SetNumberofDiscreteControls(0)

            Return

      EndIf

!----------------------------------------------------------------------

!----------------------------------------------------------------------

!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time
      If (getIsStartTime()) Then
! Read in the Values of the Parameters from the Input File
            nref=(getParameterValue(1))

            T_e=(getParameterValue(2))

            T_c=(getParameterValue(3))

            T_SC=(getParameterValue(4))
If (T_SC<0.d0) Call foundBadParameter(4,'Fatal','The degree of subcooling
cannot be negative')
                If (ErrorFound()) Return


            m=(getInputValue(1))


! Set initial values of the outputs
            Call setOutputValue(1,getInputValue(1))

            Call setOutputValue(2,1.0)

            Call setOutputValue(3,1.0)

            Call setOutputValue(4,1.0)

            Call setOutputValue(5,1.0)

            Return

      EndIf
```

155

```fortran
!----------------------------------------------156-----------------------------------------
!ReRead the Parameters if Another Unit of This Type Has Been Called Last
      If(getIsReReadParameters()) Then
!  Read in the Values of the Parameters from the Input File
            nref=(getParameterValue(1))
            T_e=(getParameterValue(2))
            T_c=(getParameterValue(3))
            T_SC=(getParameterValue(4))
      EndIf


!--------------------------------------------------------------------------
! Get current input values
            m=(getInputValue(1))
! Find properties at condenser outlet/prior to subcooling
      qu=0                        ! Saturated liquid state
      prop(1)=T_c
      prop(5)=qu
      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return
      P_c=prop(2)


! Calculate inlet temperature
      T_in=(T_c-T_SC)
! Find properties at inlet conditions
      If (T_SC>0.d0) Then
            prop(1)=T_in
            prop(2)=P_c
            Call coolprop_fluid_properties(prop,nref,12,iFlagr)
                If (ErrorFound()) Return
            h_in=prop(3)
            v_in=prop(6)
      ElseIf (T_SC==0.d0) Then
            qu=0
            prop(1)=T_in
            prop(5)=qu
```

```fortran
          Call coolprop_fluid_properties(prop,nref,15,iFlagr)
                If (ErrorFound()) Return
          h_in=prop(3)
          v_in=prop(6)
      EndIf


! Find properties at outlet conditions
! Calculate pressure in evaporator
      qu=1          ! Saturated vapor state
      prop(1)=T_e
      prop(5)=qu
      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return
      P_e=prop(2)
!P_out is assumed to be equal to P_e (evaporating pressure)


! Calculate outlet conditions with known pressure and enthalpy
      P_out=P_e
      h_out=h_in         ! Isenthaplpic expansion process in expansion valve
      prop(2)=P_out
      prop(3)=h_out


      Call coolprop_fluid_properties(prop,nref,23,iFlagr)
            If (ErrorFound()) Return
      T_out=prop(1)
      qu_out=prop(5)
!-------------------------------------------------------------------------
! Set the outputs from this model
      Call SetOutputValue(1,m)
      Call SetOutputValue(2,P_out)
      Call SetOutputValue(3,qu_out)
      Call SetOutputValue(4,T_out)
      Call SetOutputValue(5,h_out)
      Return


End Subroutine Type220
```

```fortran
Subroutine Type230
! This Type models a condenser/evaporator
!
! Inputs:
! 1. mrefr            Refrigerant mass flow rate    [kg/h]
! 2. h_in             Refrigerant enthalpy inlet    [kJ/kg]
!
! Parameters:
! 1. Mode             Condenser or evaporator       [-]
! 2. T_sat            Saturation temperature        [C]
! 3. deltaT           Superheat or subcool          [C]
! 4. nref             Refrigerant                   [-]
! 5. eff              Efficiency                    [-]
! 6. P_loss           Fractional pressure loss      [-]
!
! Outputs:
! 1. mrefr            Refrigerant mass flow rate    [kg/h]
! 2. h_out            Refrigerant enthalpy outlet   [kJ/kg]
! 3. Q_eff            Heat transfer rate            [W]



! Export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE230
!-------------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-------------------------------------------------------------------------
! Variable declarations
IMPLICIT NONE
! Inputs
REAL :: mrefr, h_in


! Parameters
REAL :: T_sat, deltaT, eff, P_loss
INTEGER :: nref, mode
```

```fortran
! Outputs
REAL :: h_out, Q_eff, P_sat


! Others
REAL :: Time, Timestep, qu,qu_out, prop(9), T_out, Q_refr, Q_water, Q_max,
P_out
INTEGER :: CurrentUnit, CurrentType, iType, iFlagr


!-----------------------------------------------------------------------
!Get the Global Trnsys Simulation Variables
      Time = getSimulationTime()

      Timestep = getSimulationTimeStep()

      CurrentUnit = getCurrentUnit()

      CurrentType = getCurrentType()
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
!Set the Version Number for This Type
      If(getIsVersionSigningTime()) Then

            Call SetTypeVersion(17)

            Return

      Endif
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
!Do All of the Last Call Manipulations Here
      If(getIsLastCallofSimulation()) Then

            Return

      Endif
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
!Perform Any "End of Timestep" Manipulations That May Be Required
      If(getIsEndOfTimestep()) Then

            Return

      Endif
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
```

```
!Do All of the "Very First Call of the Simulation Manipulations" Here
      If(getIsFirstCallofSimulation()) Then
!  Tell the TRNSYS Engine How This Type Works
      Call SetNumberofParameters(6)
      Call SetNumberofInputs(2)
      Call SetNumberofDerivatives(0)
      Call SetNumberofOutputs(4)
      Call SetIterationMode(1)
      Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)
      Call SetNumberofDiscreteControls(0)
    Return
 EndIf
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time
 If (getIsStartTime()) Then
! Read in the Values of the Parameters from the Input File
      mode=(getParameterValue(1))
      T_sat=(getParameterValue(2))
      deltaT=(getParameterValue(3))
      nref=(getParameterValue(4))
      eff=(getParameterValue(5))
      P_loss=(getParameterValue(6))


! check parameters for errors
If ((eff<0) .or. (eff>1)) Call FoundbadParameter(5,'Fatal','Efficiency must
be between 0 and 1')
      If (ErrorFound()) Return


      mrefr=(getInputValue(1))
      h_in=(getInputValue(2))


!  Set the Initial Values of the Outputs
      Call setOutputValue(1,getInputValue(1))
      Call setOutputValue(2,0.0)
      Call setOutputValue(3,0.0)
```

```fortran
        Call setOutputValue(4,0.0)

        Return

    EndIf

!-------------------------------------------------------------------------

!ReRead the Parameters if Another Unit of This Type Has Been Called Last

    If(getIsReReadParameters()) Then

!   Read in the Values of the Parameters from the Input File

        mode=(getParameterValue(1))

        T_sat=(getParameterValue(2))

        deltaT=(getParameterValue(3))

        nref=(getParameterValue(4))

        eff=(getParameterValue(5))

        P_loss=(getParameterValue(6))


    Endif

!-------------------------------------------------------------------------

! Get current input values

        mrefr=(getInputValue(1))

        h_in=(getInputValue(2))


! Find saturation pressure

        qu=0.5                    ! Saturated

        prop(1)=T_sat

        prop(5)=qu

        Call coolprop_fluid_properties(prop,nref,15,iFlagr)

            If (ErrorFound()) Return

        P_sat=prop(2)


! Find outlet enthalpy of refrigerant

    If ((deltaT/=0.0) .or. (P_loss/=0.0)) Then

      If (mode==1) Then       ! condenser

          T_out=T_sat-deltaT

      ElseIf (mode==2) Then   ! evaporator

          T_out=T_sat+deltaT

      EndIf

        prop(1)=T_out
```

```fortran
        prop(2)=P_sat-(P_sat*P_loss)
        Call coolprop_fluid_properties(prop,nref,12,iFlagr)
            If (ErrorFound()) Return
        h_out=prop(3)
    Else
      If(mode==1) Then        ! condenser
        qu_out=0.0
      ElseIf(mode==2) Then    ! evaporator
        qu_out=1.0
      EndIf
      prop(1)=T_sat
      prop(5)=qu_out
        Call coolprop_fluid_properties(prop,nref,15,iFlagr)
        If (ErrorFound()) Return
        h_out=prop(3)
      EndIf


! Calculate heat transfer to/from refrigerant
      Q_refr=(mrefr/3600.d0)*(ABS(h_in-h_out))        !Q_ref in kW


! Find actual heat transfer rate
If(mode==1)Then
      Q_eff=Q_refr*eff        ! Heating capacity - Condenser
ElseIf(mode==2)Then
      Q_eff=Q_refr*(1/eff)    ! Cooling capacity - Evaporator
EndIf


! Unit conversions
      Q_eff=Q_eff*1000        ! Q_eff in W
      P_sat=P_sat/1000        ! Psat in kPa
```

```fortran
! Set the outputs from this Type

    Call setOutputValue(1,mrefr)

    Call setOutputValue(2,h_out)

    Call setOutputValue(3,Q_eff)

    Call setOutputValue(4,P_sat)

    Return


End Subroutine Type230
```

```fortran
! Set the outputs from this Type

    Call setOutputValue(1,mrefr)

    Call setOutputValue(2,h_out)

    Call setOutputValue(3,Q_eff)
```

```fortran
Subroutine Type240


! Type 240 models a building heat demand based on the ambient temperature,
! desired room temperature and overall heat loss coefficient.
!
! Parameters:
! 1. T_room      Room temperature [C]
! 2. UA               Overall heat loss coefficient [W/C]
!
! Inputs:
! 1. T_amb       Ambient temperature [C]
!
! Outputs:
! 1. Q_dem       Heat demand [W]
!
!! Export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE240
!-------------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-------------------------------------------------------------------------
! Variable declarations
IMPLICIT NONE
! Parameters
REAL :: T_room, UA
! Inputs
REAL :: T_amb
! Outputs
REAL :: Q_dem
! Others
REAL :: Time, Timestep
INTEGER :: CurrentUnit, CurrentType
!-------------------------------------------------------------------------
```

```
!Get the Global Trnsys Simulation Variables
      Time = getSimulationTime()
      Timestep = getSimulationTimeStep()
      CurrentUnit = getCurrentUnit()
      CurrentType = getCurrentType()
!-----------------------------------------------------------------------
!Set the Version Number for This Type
      If(getIsVersionSigningTime()) Then
            Call SetTypeVersion(17)
            Return
      EndIf
!-----------------------------------------------------------------------
!Do All of the Last Call Manipulations Here
      If(getIsLastCallofSimulation()) Then
            Return
      EndIf
!-----------------------------------------------------------------------
!Perform Any "End of Timestep" Manipulations That May Be Required
      If(getIsEndOfTimestep()) Then
            Return
      Endif
!-----------------------------------------------------------------------
!Do All of the "Very First Call of the Simulation Manipulations" Here
      If(getIsFirstCallofSimulation()) Then
!  Tell the TRNSYS Engine How This Type Works
      Call SetNumberofParameters(2)
      Call SetNumberofInputs(1)
      Call SetNumberofDerivatives(0)
      Call SetNumberofOutputs(1)
      Call SetIterationMode(1)
      Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)
      Call SetNumberofDiscreteControls(0)
    Return
 EndIf
!-----------------------------------------------------------------------
```

```fortran
!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time

 If (getIsStartTime()) Then

! Read in the Values of the Parameters from the Input File

      T_room=(getParameterValue(1))

      UA=(getParameterValue(2))


      T_amb=(getInputValue(1))


!  Set the Initial Values of the Outputs

     Call setOutputValue(1,1.0)

      Return

  EndIf
!-----------------------------------------------------------------------

!ReRead the Parameters if Another Unit of This Type Has Been Called Last

  If(getIsReReadParameters()) Then

!  Read in the Values of the Parameters from the Input File

      T_room=(getParameterValue(1))

      UA=(getParameterValue(2))

  Endif
!-----------------------------------------------------------------------

! Get current input values

      T_amb=(getInputValue(1))

! Calculate heat demand

If ((T_room-T_amb)<0.d0) Then

      Q_dem=0.d0

Else

      Q_dem=UA*(T_room-T_amb)

EndIf


! Set outputs

      Call setOutputValue(1,Q_dem)

      Return


End Subroutine Type240
```

```
SUBROUTINE Type216

!*************************************************************

! This type models a heat pump with variable speed drive compressor.

! The compressor power input is calculated by polynomial from EN12900 for
compressor power.

!

! Y = C1 + C2*Te + C3*Tc + C4*Te^2 + C5*Tc*Te + C6*Tc^2 + C7*Te^3 +
C8*Tc*Te^2 + C9*Tc^2*Te + C10*Tc^3

!

! Where Y = compressor power consumption [W]

! and C1 ... C10 coefficients provided by manufacturer.

!

!      Parameters:

!      1. T_e           Evaporating temperature [C]

!      2. T_c           Condensing temperature [C]

!      3. V_sw           Swept volume of compressor [m^3]

!      4. nref           Refrigerant

!      5. T_SH           Degree of superheat [C]

!      6. T_SC           Degree of subcooling [C]

!      7. P_loss    Fractional pressure loss in suction line [-]

!      8. f_min     Minimum frequency [1/s]

!      9. n_vol     Volumetric efficiency [-]

!      10. r_loss   Fractional heat loss/cooling [-]

!      11. eff           Efficiency of evap/cond. [-]

!      12. LU1           Logical unit number [-]

!      13. LU2           Logical unit number [-]

!      14. LU3           Logical unit number [-]

!

!      Inputs:

!      1. f        Frequency [1/s]

!

!      Outputs:

!      1. W_in          Compressor power [W]

!      2. n_is          Isentropic efficiency [-]

!      3. rpm           Rotational speed [1/min]

!      4. m_suc         Mass flow rate [kg/h]

!      5. Q_heat         Heating capacity [W]
```

```fortran
!     6. Q_cool          Cooling capacity [W]
!
!     Thermodynamic properties:
!     1. h          Enthalpy [kJ/kg]
!     2. p          Pressure [kPa]
!     3. T          Temperature [C]
!     4. qu         Quality    [-]
!     5. v          Specific volume [m3/kg]
!     6. s          Entropy    [kJ/kg]
!
!export this subroutine for its use in external DLLs.
!DEC$ATTRIBUTES DLLEXPORT :: TYPE216
!-------------------------------------------------------------------------
! USE statements
 USE TrnsysConstants
 USE TrnsysFunctions
!-------------------------------------------------------------------------
! Variable declerations
IMPLICIT NONE          ! All variables are explicitly defined


!Parameters:
REAL :: T_e, T_c, V_sw, T_SH, T_SC, P_loss, f_min, r_loss, n_vol, eff
INTEGER :: nref, LU1, LU2, LU3


!Inputs
REAL :: f


!Outputs
REAL :: W_in, n_is, rpm, Q_heat, Q_cool, m_suc



!Thermodynamic properties
REAL :: qu, P_e, h_e, v_e, s_e, P_c, T_suc, P_suc, h_suc, s_suc, v_suc,
P_dis, T_dis, v_dis, h_dis, h_is, T_cout, h_cout


!Others
```

```fortran
REAL :: Time, Timestep, X(6), Y(10), U_sw, U_re, C1, C2, C3, C4, C5, C6,
C7, C8, C9, C10, prop(9), w

INTEGER :: iType, iFlagr, ios, CurrentUnit, CurrentType, nY, NX, nIND, LU,
i

!----------------------------------------------------------------------

!Get the Global Trnsys Simulation Variables

      Time = getSimulationTime()

      Timestep = getSimulationTimeStep()

      CurrentUnit = getCurrentUnit()

      CurrentType = getCurrentType()

!----------------------------------------------------------------------

!Set the Version Number for This Type

      If(getIsVersionSigningTime()) Then

            Call SetTypeVersion(17)

            Return

      EndIf

!----------------------------------------------------------------------

!Do All of the Last Call Manipulations Here

      If(getIsLastCallofSimulation()) Then

            Return

      Endif

!----------------------------------------------------------------------

!Perform Any "End of Timestep" Manipulations That May Be Required

      If(getIsEndOfTimestep()) Then

            Return

      Endif

!----------------------------------------------------------------------

!Do All of the "Very First Call of the Simulation Manipulations" Here

      If(getIsFirstCallofSimulation()) Then

!  Tell the TRNSYS Engine How This Type Works

      Call SetNumberofParameters(14)

       Call SetNumberofInputs(1)

      Call SetNumberofDerivatives(0)

      Call SetNumberofOutputs(6)

      Call SetIterationMode(1)

      Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)

      Call SetNumberofDiscreteControls(0)
```

```
    Return

 EndIf

!------------------------------------------------------------------------

!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time

 If (getIsStartTime()) Then

! Read in the Values of the Parameters from the Input File

        T_e=(getParameterValue(1))

        T_c=(getParameterValue(2))

        V_sw=(getParameterValue(3))

        nref=(getParameterValue(4))

        T_SH=(getParameterValue(5))

        T_SC=(getParameterValue(6))

        P_loss=(getParameterValue(7))

        f_min=(getParameterValue(8))

        n_vol=(getParameterValue(9))

        r_loss=(getParameterValue(10))

        eff=(getParameterValue(11))

        LU1=(getParameterValue(12))

        LU2=(getParameterValue(13))

        LU3=(getParameterValue(14))


        f=(getInputValue(1))



!  Set the Initial Values of the Outputs

        Call setOutputValue(1,0.0)

        Call setOutputValue(2,0.0)

        Call setOutputValue(3,0.0)

        Call setOutputValue(4,0.0)

        Call setOutputValue(5,0.0)

        Call setOutputValue(6,0.0)

            Return

   EndIf

!------------------------------------------------------------------------
```

```fortran
!ReRead the Parameters if Another Unit of This Type Has Been Called Last
   If(getIsReReadParameters()) Then
!  Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))

      T_c=(getParameterValue(2))

      V_sw=(getParameterValue(3))

      nref=(getParameterValue(4))

      T_SH=(getParameterValue(5))

      T_SC=(getParameterValue(6))

      P_loss=(getParameterValue(7))

      f_min=(getParameterValue(8))

      n_vol=(getParameterValue(9))

      r_loss=(getParameterValue(10))

      eff=(getParameterValue(11))

      LU1=(getParameterValue(12))

      LU2=(getParameterValue(13))

      LU3=(getParameterValue(14))


   EndIf


!----------------------------------------------------------------------
! Get current input values
      f=(getInputValue(1))


If (f<f_min) Then

      f=0.0

      W_in=0.0

      m_suc=0.0

      n_is=0.0

      rpm=0.0

      Q_cool=0.0

      Q_heat=0.0
ElseIf (f>=f_min) Then


! Calculate condensing and evaporating pressures
```

```fortran
      qu=1.0                        ! Saturated vapor state
      prop(1)=T_e
      prop(5)=q
      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
           If (ErrorFound()) Return
      P_e=prop(2)


      qu=0.0                  ! Saturated liquid state
      prop(1)=T_c
      prop(5)=qu
      Call coolprop_fluid_properties(prop,nref,15,iFlagr)
           If (ErrorFound()) Return
      P_c=prop(2)


!     Interpolate coefficients to match frequency


! Use correct data file
      If (nref==134) Then
           LU=LU1
      ElseIf (nref==290) Then
           LU=LU2
      ElseIf (nref==717) Then
           LU=LU3
      EndIf

      nX=7              ! There are 7 values of variable f in the file
      X(1)=f            ! Sets X to local variable f
      nY=10             ! There are 10 coefficients to each value of f
      nIND=1            ! There is 1 independant variable (frequency)

           Call InterpolateData(LU,nIND,nX,nY,X,Y)
           C1 = Y(1)
           C2 = Y(2)
           C3 = Y(3)
           C4 = Y(4)
           C5 = Y(5)
```

```
            C6 = Y(6)

            C7 = Y(7)

            C8 = Y(8)

            C9 = Y(9)

            C10 = Y(10)


    ! Calculate compressor power
W_in =
C1+(C2*T_e)+(C3*T_c)+(C4*T_e**2)+(C5*T_c*T_e)+(C6*T_c**2)+(C7*T_e**3)+(C8*T
_c*T_e**2)+(C9*T_c**2*T_e)+(C10*T_c**3)


! Calculate refrigerant properties
! Thermodynamic properties at evaporator outlet
        qu = 1                                  ! Saturated vapor state
        prop(1)=T_e
        prop(5)=qu
 Call Coolprop_fluid_properties(prop,nref,15,iFlagr)
        If (ErrorFound()) Return
        P_e=prop(2)
        h_e=prop(3)
        s_e=prop(4)
        v_e=prop(6)


!     Thermodynamic properties at suction state
    If ((T_SH>0.0) .or. (P_loss>0.0)) Then
! Not saturated, P and T may be used
        T_suc=T_e+T_SH
        P_suc=P_e-(P_e*P_loss)
        prop(1)=T_suc
        prop(2)=P_suc
        Call coolprop_fluid_properties(prop,nref,12,iFlagr)
            If (ErrorFound()) Return
        h_suc=prop(3)
        s_suc=prop(4)
        v_suc=prop(6)
    ElseIf ((T_SH==0.d0) .and. (P_loss==0.d0)) Then
!Saturated, P and T can not be used in combination as known properties
```

173

```fortran
        P_suc=P_e

        h_suc=h_e

        s_suc=s_e

        v_suc=v_e

    EndIf


! Calculate mass flow rate
        rpm=((f*60)/2.0)                    ! Assuming 4 poles in motor

        w=(rpm/60.0)                        ! Revolutions per second

        m_suc=3600.0*w*V_sw*n_vol/v_suc    ! Mass flow rate in kg/h



! Thermodynamic properties at condenser outlet
    If (T_SC>0.0) Then

        T_cout=T_c-T_SC

        prop(1)=T_cout

        prop(2)=P_c

        Call Coolprop_fluid_properties(prop,nref,12,iFlagr)

            If (ErrorFound()) Return

        h_cout=prop(3)
    ElseIf (T_SC==0.0) Then

        qu=0.0

        prop(1)=T_c

        prop(5)=qu

        Call coolprop_fluid_properties(prop,nref,15,iFlagr)

            If (ErrorFound()) Return

        h_cout=prop(3)

    EndIf


! Thermodynamic properties at discharge state
        W_in=W_in/1000.0                ! W_in in kW

        m_suc=m_suc/3600.0              ! m_suc in kg/s

        P_dis=P_c
        ! Assuming discharge pressure equals condenser pressure

        h_dis = h_suc+(W_in*(1-r_loss)/m_suc)
        ! m_suc in kg/s, W_in in kW --> h in kJ/kg
```

```fortran
        prop(2)=P_dis
        prop(3)=h_dis
 Call Coolprop_fluid_properties(prop,nref,23,iFlagr)
        If (ErrorFound()) Return
        T_dis=prop(1)
        v_dis=prop(6)


! Calculate isentropic efficieny
        prop(2)=P_dis
        prop(4)=s_suc
 Call Coolprop_fluid_properties(prop,nref,24,iFlagr)
 If (ErrorFound()) Return
        h_is=prop(3)


        n_is=(h_is-h_suc)/(h_dis-h_suc)


! Calculate cooling capacity
        Q_cool=(1/eff)*(m_suc)*(h_suc-h_cout)
! Cooling capacity [kW] equals heat removed from water/source


! Calculate heating capacity
        Q_heat=eff*(m_suc)*(h_dis-h_cout)
! Heating capacity [kW] equals heat recieved by water/sink


! Unit conversions
        Q_cool=Q_cool*1000.0                        ! Q_cool in W
        Q_heat=Q_heat*1000.0                        ! Q_heat in W
        W_in=W_in*1000.0                            ! W_in in W
        m_suc=m_suc*3600.0                              ! m_suc in kg/h
EndIf


! Set the outputs from this model
        Call SetOutputValue(1,W_in)
        Call SetOutputValue(2,n_is)
        Call SetOutputValue(3,rpm)
        Call SetOutputValue(4,m_suc)
```

```
    Call SetOutputValue(5,Q_cool)

    Call SetOutputValue(6,Q_heat)

    Return


End Subroutine Type216
```

```
    Call SetOutputValue(5,Q_cool)

    Call SetOutputValue(6,Q_heat)

    Return
```

```
SUBROUTINE Type217

!*********************************************************************

! This model models a variable speed heat pump with polytropic compressor

!

!     W_in = m*P_e*v_suc(n/(n-1))((P_c/P_e)^((n-1)/n)-1)

!

!     where W_in is the compressor power [W]

!     and         n is the polytropic exponent

!

!     Parameters:

!     1. T_e          Evaporating temperature [C]

!     2. T_c          Condensing temperature [C]

!     3. V_sw         Swept volume of compressor [m^3]

!     4. nref         Refrigerant [-]

!     5. T_SH         Degree of superheat [C]

!     6. T_SC         Degree of subcooling [C]

!     7. P_loss       Fractional pressure loss in suction line [-]

!     8. n            Polytropic exponent [-]

!     9. n_vol        Volumetric efficiency [-]

!     10. f_min       Minimum frequency [1/s]

!     11. eff         Efficiency evap/cond. [-]

!

!     Inputs:

!     1. f            Frequency [1/s]

!

!     Outputs:

!     1. W_in         Compressor power [W]

!     2. n_is         Isentropic efficiency [-]

!     3. m_suc        Refrigerant mass flow rate [kg/h]

!     4. rpm          Rotational speed [1/min]

!     5. Q_cool       Cooling capacity [W]

!     6. Q_heat       Heating capacity [W]

!     7. h_dis        Discharge enthalpy [kJ/kg]

!

!     Thermodynamic properties:

!     1. h        Enthalpy [kJ/kg]
```

```fortran
!     2.  P          Pressure [kPa]

!     3.  T          Temperature [C]

!     4.  qu         Quality [-]

!     5.  v          Specific volume [m3/kg]

!     6.  s          Entropy [kJ/kg]

!

!export this subroutine for its use in external DLLs.

!DEC$ATTRIBUTES DLLEXPORT :: TYPE217

!-------------------------------------------------------------------------

! USE statements

 USE TrnsysConstants

 USE TrnsysFunctions

!-------------------------------------------------------------------------

! Variable declerations

IMPLICIT NONE      ! All variables are explicitly defined


!Parameters:

REAL :: T_e, T_c, V_sw, T_SH, T_SC, P_loss, n, n_vol, f_min, eff

INTEGER :: nref


!Inputs:

REAL :: f


!Outputs:

REAL :: W_in, n_is, m_suc, rpm, Q_cool, Q_heat, h_dis


!Thermodynamic properties:

REAL :: qu, P_e, P_c, T_cout, h_cout, T_suc, P_suc, h_suc, v_suc, s_suc,
T_dis, P_dis, v_dis, s_dis, h_is


!Others

REAL :: Time, Timestep, prop(9), w

INTEGER :: CurrentUnit, CurrentType, iType, iFlagr

!-------------------------------------------------------------------------

!Get the Global Trnsys Simulation Variables

      Time = getSimulationTime()
```

```fortran
    Timestep = getSimulationTimeStep()

    CurrentUnit = getCurrentUnit()

    CurrentType = getCurrentType()
!-----------------------------------------------------------------------
!Set the Version Number for This Type

    If(getIsVersionSigningTime()) Then

        Call SetTypeVersion(17)

        Return

    EndIf
!-----------------------------------------------------------------------
!Do All of the Last Call Manipulations Here

    If(getIsLastCallofSimulation()) Then

        Return

    EndIf
!-----------------------------------------------------------------------
!Perform Any "End of Timestep" Manipulations That May Be Required

    If(getIsEndOfTimestep()) Then

        Return

    EndIf
!-----------------------------------------------------------------------
!Do All of the "Very First Call of the Simulation Manipulations" Here

    If(getIsFirstCallofSimulation()) Then
!  Tell the TRNSYS Engine How This Type Works

    Call SetNumberofParameters(11)

    Call SetNumberofInputs(1)

    Call SetNumberofDerivatives(0)

    Call SetNumberofOutputs(7)

    Call SetIterationMode(1)

    Call SetNumberStoredVariables(0,0)  !(nStatic,nDynamic)

    Call SetNumberofDiscreteControls(0)

    Return

 EndIf
!-----------------------------------------------------------------------
!Do All of the "Start Time" Manipulations Here - There Are No Iterations at
the Intial Time

 If (getIsStartTime()) Then
```

```
! Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))
      T_c=(getParameterValue(2))
      V_sw=(getParameterValue(3))
      nref=(getParameterValue(4))
      T_SH=(getParameterValue(5))
      T_SC=(getParameterValue(6))
      P_loss=(getParameterValue(7))
      n=(getParameterValue(8))
      n_vol=(getParameterValue(9))
      f_min=(getParameterValue(10))
      eff=(getParameterValue(11))


      f=(getInputValue(1))


!  Set the Initial Values of the Outputs
      Call setOutputValue(1,0.0)
      Call setOutputValue(2,0.0)
      Call setOutputValue(3,1.0)
      Call setOutputValue(4,0.0)
      Call setOutputValue(5,0.0)
      Call setOutputValue(6,0.0)
      Call setOutputValue(7,0.0)
      Return

  EndIf
!-----------------------------------------------------------------------
!ReRead the Parameters if Another Unit of This Type Has Been Called Last
  If(getIsReReadParameters()) Then
!  Read in the Values of the Parameters from the Input File
      T_e=(getParameterValue(1))
      T_c=(getParameterValue(2))
      V_sw=(getParameterValue(3))
      nref=(getParameterValue(4))
      T_SH=(getParameterValue(5))
      T_SC=(getParameterValue(6))
      P_loss=(getParameterValue(7))
```

```fortran
        n=(getParameterValue(8))
        n_vol=(getParameterValue(9))
        f_min=(getParameterValue(10))
        eff=(getParameterValue(11))
    EndIf
!------------------------------------------------------------------------
! Get current input values
        f=(getInputValue(1))


If (f<f_min) Then
        W_in=0.0
        rpm=0.0
        f=0.0
        m_suc=0.0
        n_is=0.0
        Q_heat=0.0
        Q_cool=0.0
ElseIf (f>=f_min) Then
! Calculate condensing and evaporating pressures
        qu=1.0                          ! Saturated vapor state


        prop(1)=T_e
        prop(5)=qu
        Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return
        P_e=prop(2)


        qu=0.0                          ! Saturated liquid state


        prop(1)=T_c
        prop(5)=qu
        Call coolprop_fluid_properties(prop,nref,15,iFlagr)
            If (ErrorFound()) Return
        P_c=prop(2)
```

Elisabeth Vindenes

```
! Properties at compressor inlet state (suction)
If ((T_SH>0.0) .or. (P_loss>0.0)) Then
! Not saturated, P and T may be used
     T_suc=T_e+T_SH
     P_suc=P_e-(P_e*P_loss)
     prop(1)=T_suc
     prop(2)=P_suc
     Call coolprop_fluid_properties(prop,nref,12,iFlagr)
          If (ErrorFound()) Return
     h_suc=prop(3)
     s_suc=prop(4)
     v_suc=prop(6)
ElseIf ((T_SH==0.0) .and. (P_loss==0.0)) Then
!Saturated, P and T can not be used in combination as known properties
     qu=1.0
     prop(1)=T_e
     prop(5)=qu
     Call coolprop_fluid_properties(prop,nref,15,iFlagr)
          If (ErrorFound()) Return
     P_suc=prop(2)
     h_suc=prop(3)
     s_suc=prop(4)
     v_suc=prop(6)
EndIf


! Calculate mass flow rate
     rpm=(f*60)/2               ! Assuming 4 poles in electric motor
     w=(rpm/60)                 ! Revolutions per second
     m_suc=w*V_sw*n_vol/v_suc   ! Mass flow rate in kg/s
! Calculate compressor power
! in ratio P_c/P_suc no correction necessary. To obtain W_in in kW P_suc
! must be in kPa in first part of equation.
W_in=(m_suc*(P_suc/1000)*v_suc*(n/(n-1)))*(((P_c/P_suc)**((n-1.0)/n))-1.0)
! W_in in kW
```

```fortran
! Properties at discharge state
    P_dis=P_c        ! Assuming discharge pressure = condensing pressure
    h_dis=h_suc+(W_in/m_suc)            ! Heat loss accounted for in n


    prop(2)=P_dis
    prop(3)=h_dis
    Call coolprop_fluid_properties(prop,nref,23,iFlagr)
        If (ErrorFound()) Return
    T_dis=prop(1)
    v_dis=prop(6)


! Thermodynamic properties at condenser outlet
If (T_SC>0.0) Then         ! Not saturated
    T_cout=T_c-T_SC
    prop(1)=T_cout
    prop(2)=P_c
 Call Coolprop_fluid_properties(prop,nref,12,iFlagr)
    If (ErrorFound()) Return
    h_cout=prop(3)
ElseIf (T_SC==0.0) Then
    qu=0                    ! Saturated liquid
    prop(1)=T_c
    prop(5)=qu
 Call Coolprop_fluid_properties(prop,nref,15,iFlagr)
    If (ErrorFound()) Return
    h_cout=prop(3)
Endif



! Calculate isentropic efficiency
    ! Isentropic compression
    P_dis=P_c        ! Assuming discharge pressure = condenser pressure
    s_dis=s_suc      ! Isentropic compression
    prop(2)=P_dis
    prop(4)=s_dis
```

```fortran
      Call coolprop_fluid_properties(prop,nref,24,iFlagr)
          If (ErrorFound()) Return

      h_is=prop(3)


      n_is=(h_is-h_suc)/(h_dis-h_suc)


! Calculate cooling capacity

      Q_cool=(1/eff)*m_suc*(h_suc-h_cout)
! Cooling capacity [kW] equals heat removed from water/source


! Calculate heating capacity

      Q_heat=eff*m_suc*(h_dis-h_cout)
! Heating capacity [kW] equals heat recieved by water/load


! Unit conversions
      m_suc=m_suc*3600      ! Mass flow rate in kg/h
      W_in=W_in*1000        ! Compressor power in W
      Q_heat=Q_heat*1000    ! Heating capacity in W
      Q_cool=Q_cool*1000    ! Cooling capacity in W


EndIf


! Set the outputs from this model
      Call SetOutputValue(1,W_in)
      Call SetOutputValue(2,n_is)
      Call SetOutputValue(3,m_suc)
      Call SetOutputValue(4,rpm)
      Call SetOutputValue(5,Q_cool)
      Call SetOutputValue(6,Q_heat)
      Call setOutputValue(7,h_dis)
Return


End Subroutine Type217
```

Appendix II Basic Structure of TRNSYS-component source code

TRNSYS contains components for almost any energy system. However, TRNSYS focuses on the system level, and does not have all the components necessary to create a model of a heat pump on component level. Two different compressor components are created, and a model simulating an expansion valve. A component is created to simulate a condenser or evaporator. Finally, a simplified building load Type is created.

The user created Types in this project are written in Notepad++ using the TRNSYS Documentation Vol. 7 Programmer's Guide. The Proforma file was not used to create the source codes, because the author wanted to learn and understand the TRNSYS functions and how the program and simulations work. In the following the general TRNSYS functions are presented. These parts of the source code are similar for each component and must be included for the Type to be used in TRNSYS. The calculations in each of the user-made components are presented separately.

The very first line in the source code states the Type number of the component.

```
SUBROUTINE TYPEn
```

where n is a number between 1 and 9999.

After the first line the source code contains some comments regarding the purpose of the model. These comments are not necessary for the source code itself, but helps users understand how the model works. All lines starting with exclamation mark are comments and are not part of the code. The author has chosen to list all parameters, inputs, and outputs in the section for comments, in order to clarify the meaning of the various variable-names and their units.

Prior to any statements or functions in the code, the program must instruct TRNSYS to create a .DLL (dynamic link library) file in order to use the component in a project or simulation. The .DLL file is created when the source code is compiled in the TypeStudio. TRNSYS places the .DLL file in a proper location for the TRNSYS engine to load it during a simulation. The following line ensures the creation of the .DLL file.

```
!DEC$ATTRIBUTES DLLEXPORT :: TYPEn
```

The exclamation mark at the beginning of this line does not imply that this is a comment. Fortran will recognize the command to create a .DLL file. This command must be placed in the first column of code, without any tab or space prior to the exclamation mark and code.

The next step in the code is to allow the Type to access and use global constants and functions. These constants and functions are contained in various data modules. The data modules may be accessed through a USE statement. When applying this statement to the source code, all variables in the data modules are declared automatically, and the component may use all constants and functions without declaring them specifically. There are various data modules available, not all will be presented here.

```
USE TrnsysConstants

USE TrnsysFunctions
```

The `TrnsysConstants` data module contains various rules for the functions and subroutines in the TRNSYS kernel. The constants set limitations on the simulations, for example in terms of maximum

number of units or the minimal simulation time step. The `TrnsysFunctions` data module contains numerous access functions that may be used by the component. Some of these functions are used by all types, including functions to obtain information about the simulation time step and to set the number of inputs and parameters. Other functions are not mandatory but very useful in some components. Examples are functions to interpolate data from an external file and functions to obtain thermodynamic properties of fluids. These functions will be presented in detail in the following text.

The following step in the source code is the variable declarations. All variables used in the source code (local variables) and arguments for the relevant subroutines must be declared specifically. The global variables are as mentioned above declared through the `USE` statements. The local variables must also be assigned a specific data type. The numerical data types in Fortran 90 are real, integer and complex. In addition, there are the data types character and logical. An example of variable declarations is given below.

```
IMPLICIT NONE

REAL :: variable1, variable2

INTEGER :: variable3, variable4
```

Only real and integer variables are used in the source codes produced in this project. Real variables may have any value, both positive and negative. Integer variables are discrete, whole values only. In Fortran 90 language the variables are assigned data type according to the first letter in the variable name if implicit typing is permitted. The IMPLICIT NONE statement allows the programmer to select variable names freely, and all the variables are explicitly defined. The variable names are carefully chosen to make the code understandable. Variable names must start with a letter. The variable names may be up to 31 characters, containing numbers, letters, and underscore character. In Fortran 90 there is no case sensitivity, meaning that lower and uppercase of the same letter are equivalent.

After the variable declarations the executable part of the program starts. When the program runs each line of code will be executed sequentially. The first step is to obtain the global simulation variables. This is done by access functions accessible through the `TrnsysFunctions` data module.

```
Time = getSimulationTime()

Timestep = getSimulationTimeStep()

CurrentUnit = getCurrentUnit()

CurrentType = getCurrentType()
```

The variables Time (simulation time), Timestep (simulation time step), CurrentUnit and CurrentType are declared as integer variables. The information is received from the TRNSYS kernel through the respective access functions. These functions are placed at the beginning of the code because this information is required at the beginning of each simulation time step and each iteration.

As TRNSYS has developed over the years, the coding standards have changed. At the beginning of a simulation each component must log its TRNSYS coding standard. TRNSYS may run components created in different coding standards. The TRNSYS kernel treats the various coding standards differently and must know which coding standard the component has. This is done by another access function.

```
If(getIsVersionSigningTime()) Then

    Call SetTypeVersion(17)

    Return

EndIf
```

The coding standard has not changed between TRSNYS version 17 and 18. If the Type does not register the coding standard, the TRNSYS kernel will abort the simulation and generate an error message.

The next step in the source code is the Last Call Manipulations. These calls are executed at the end of each simulation. Last call manipulations may be writing a message to the list file, closing any external files, or calculating summary information. For the components created in this project there are no calls at the end of the simulations, and control is returned to the kernel directly. The code is as follows.

```
If(getIsLastCallofSimulation()) Then

    Return

EndIf
```

Similarly, the kernel recalls each Type at the end of each timestep. For some components End of Timestep manipulations are necessary. These may be updating stored variables or resetting values. For the components created in this project there are no recalls at the end of each timestep. Control is returned to the kernel directly.

```
If(getIsEndOfTimestep()) Then

    Return

EndIf
```

At the beginning of each simulation the Types must initialize themselves. In the First Call of Simulation Manipulations, there are some steps that every component must take. These steps tell the TRNSYS kernel how the Type works. These calls occur before the simulation has started.

```
If(getIsFirstCallofSimulation()) Then

    Call SetNumberofParameters(8)

    Call SetNumberofInputs(2)

    Call SetNumberofDerivatives(0)

    Call SetNumberofOutputs(7)

    Call SetIterationMode(1)

    Call SetNumberStoredVariables(0,0)

    Call SetNumberofDiscreteControls(0)

    Return

EndIf
```

The Type first calls the access function `getIsFirstCallofSimulation()`. If this function returns the value "true" the remaining calls are made. These calls tell the TRNSYS kernel how many

187

parameters, inputs, and outputs to expect from this type. The IterationMode usually has the value 1. This means that the kernel calls the Type every iteration, even if the input values have not changed. Variables may be stored between time steps. For the components created in this project there are no stored variables and no discrete controls. During the first call of simulation it is also possible to set the units of the input and output variables. This enables the kernel to check if connected inputs and outputs have the same units. The units may be specified by a three-character code. The unit codes are listed in the Programmers Guide. As the calls to set the units are not mandatory they are not implemented in the components created in this project. Some of the inputs and outputs (e.g. frequency with unit 1/s) used in the components have units that do not have a three-letter code according to the tables in the Programmers Guide. When the first calls of the simulation are completed, the program returns control to the TRNSYS kernel.

The next time the Types are called by the TRNSYS kernel at the simulation start time. During these first calls the parameter values are read from the TRNSYS kernel. The outputs must be assigned initial values. These initial values may be defined as a numerical value directly or be equal to an input value.

```
If (getIsStartTime()) Then

    LocalVariable1=(getParameterValue(1))

    LocalVariable2=(getParameterValue(2))


    Call setOutputValue(1,0.d0)

    Call setOutputValue(2,5.d0)

    Call setOutputValue(3,getInputValue(1))

    Call setOutputValue(4,getInputValue(2))

    Return

EndIf
```

All the parameters, inputs and outputs have a number. The order of the various variables must be consistent in both the Proforma file and the source code. The numbers are not at any point *in the source code* specifically assigned to each variable*, this is controlled by the Proforma file.* It is recommended to keep the same order of the variables in all documents and files, including the comments section, to avoid confusion and mistakes. In addition to reading the parameters, the parameter values may be checked for errors in this section of the source code. This is done by a call to the access function foundBadParameter. An example of this code is shown below. If implemented in the source code it should be placed directly below the getParameterValue functions.

```
If (LocalVariable2 < 0.d0) Call foundBadParameter(2, 'Fatal', &

'The parameter must be positive')

If (ErrorFound()) Return
```

The error messages in the `foundBadParameter` function is written to the projects list file. The `ErrorFound` function returns control to the TRNSYS kernel. The functions to check parameters and inputs (`foundBadInput`) are not implemented in all the components created in this project. Checking

of some parameters and inputs crashed the TRNSYS simulation when the components where used in a simulation. Removing the functions for checking these parameters and inputs enabled the simulations to run properly, and hence the functions were removed from the relevant source codes.

The final special calls that must be made by each component is the Multiple Unit Manipulations. These calls are made only if there are multiple units of the same type in the TRNSYS simulation. This call is made to ensure that the values of the parameters are correct and for the required unit.

```
If(getIsReReadParameters()) Then

    LocalVariable1=(getParameterValue(1))

    LocalVariable2=(getParameterValue(2))

EndIf
```

If the access function `getIsReReadParameters` returns the value "true", meaning that there are more than one unit of this Type in the simulation, the parameter values are reread. Following this section are the calculations and functions that simulate the performance of the component. These calculations are repeated every time step of the simulation. The calculations are different for each component, but some of these steps are the same for all Types. The first step in each time step is to retrieve the current input values to the component.

```
LocalVariable3=(getInputValue(1))

LocalVariable4=(getInputValue(2))
```

Notice there is no access function prior to the functions to obtain the input values. This is because the program executes these functions at each time step of the simulation. The input values may be checked for error similarly as the parameter values. Following the retrieving of input values are all the calculations necessary to simulate the performance of the component. The last step required at each time step it to set the output values. The local variables that are calculated by the program must be set to the TRNSYS global outputs.
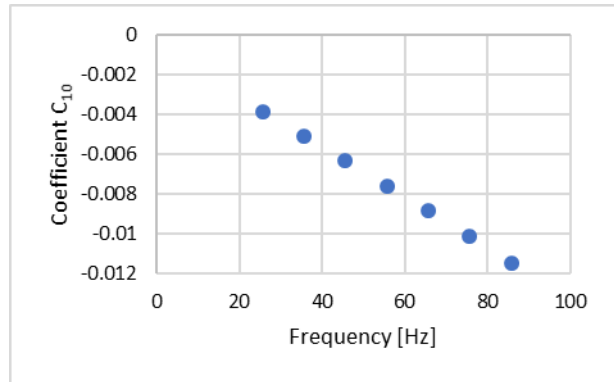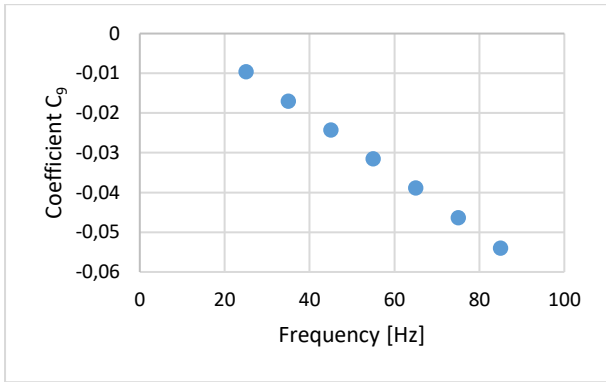
```
Call SetOutputValue(1,LocalVariable5)

Call SetOutputValue(2,LocalVariable6)

Return
```

After the final calls to set the output values the program must return control to the TRNSYS kernel. The very last line of the source code states that the Subroutine has ended.

```
END SUBROUTINE TYPEn
```

Appendix III   Linearity of compressor coefficients
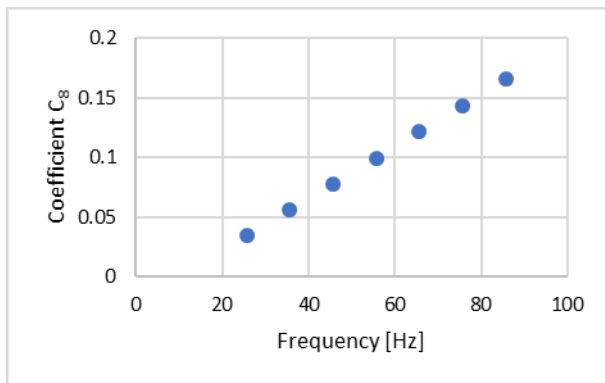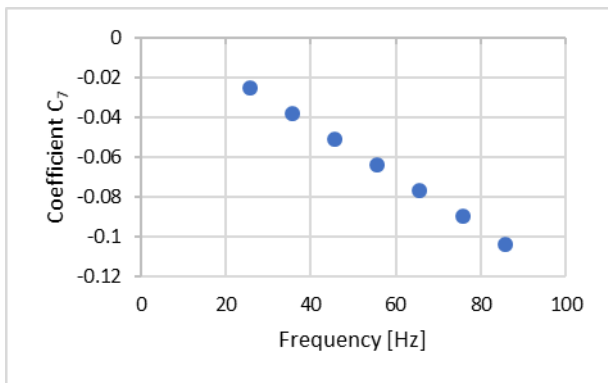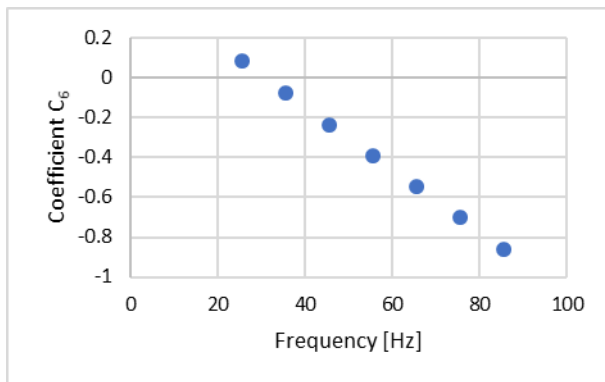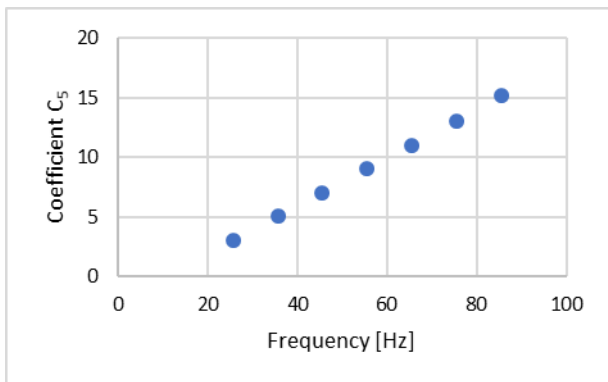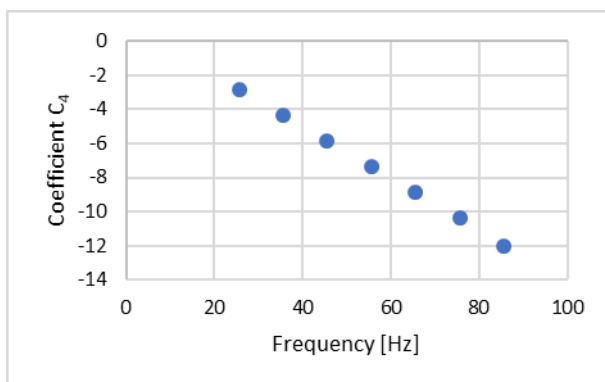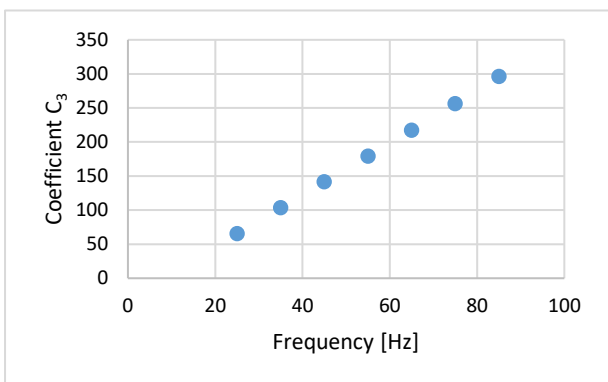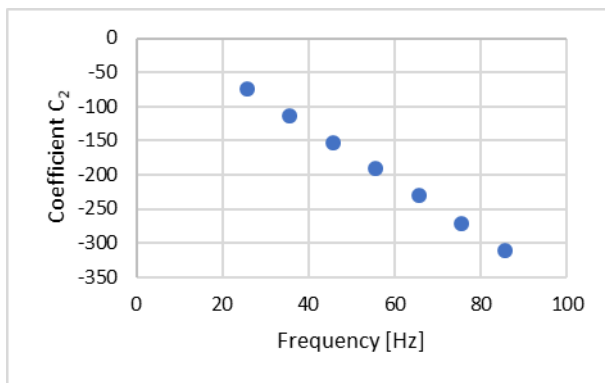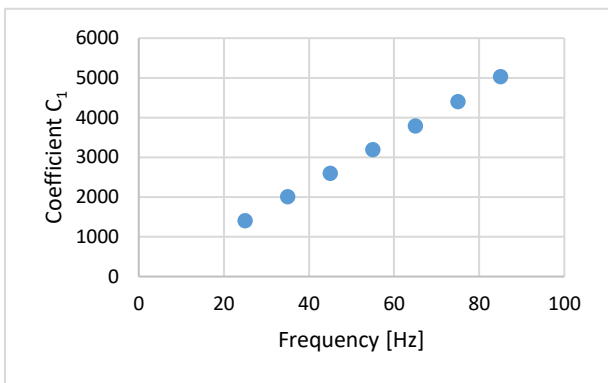
Appendix IV   Data file for InterpolateData subroutine

25.0 35.0 45.0 55.0 65.0 75.0 85.0   ! Values of f

1406.35857	-71.1539835	65.2746588	-2.70247644	3.26997757	0.09746478	-0.02405515	0.03682056	-0.00959297	-0.00374404
2005.86229	-110.421258	103.581768	-4.21657976	5.28263291	-0.06323554	-0.03698693	0.05865876	-0.01697992	-0.00498688
2598.31457	-149.173885	141.372522	-5.7106968	7.267608	-0.22048767	-0.04975137	0.08020125	-0.02425695	-0.00621859
3190.52247	-187.865392	179.088738	-7.20230888	9.24809207	-0.37608953	-0.06249792	0.10169983	-0.03150883	-0.00745374
3789.29307	-226.949305	217.172232	-8.70889732	11.2472743	-0.53183902	-0.07537604	0.12340632	-0.0388203	-0.00870691
4401.43345	-266.87915	256.064824	-10.2479435	13.288344	-0.68953405	-0.08853517	0.14557253	-0.0462761	-0.00999267
5033.75066	-308.108453	296.208329	-11.8369286	15.3944903	-0.85097253	-0.10212477	0.16845029	-0.053961	-0.0113256

Appendix V   Description of data file and implementation in TRNSYS

The first line consists of the values of the independent variable f, frequency. The remaining lines contain the values of the coefficients $C_1 - C_{10}$ for the various frequency values. The number of lines is equal to the number of values of the frequency. The first value on each line is coefficient C_1, the second is coefficient C_2 and so on. The line containing the values of the frequency should be the first line of the data file, and the first line of coefficient values should be the second line of the document. By the use of an exclamation mark, comments may be added at the end of each line.

Additional data files may be added to compressor Type 206 and variable speed heat pump Type 216 through the Files tab of the Proforma window of the component. Figure V.1 shows the files tab of the proforma window of compressor Type 206. The question should describe the contents of the data file. The default answer is the file path. The parameter Logical Unit is chosen from the list.
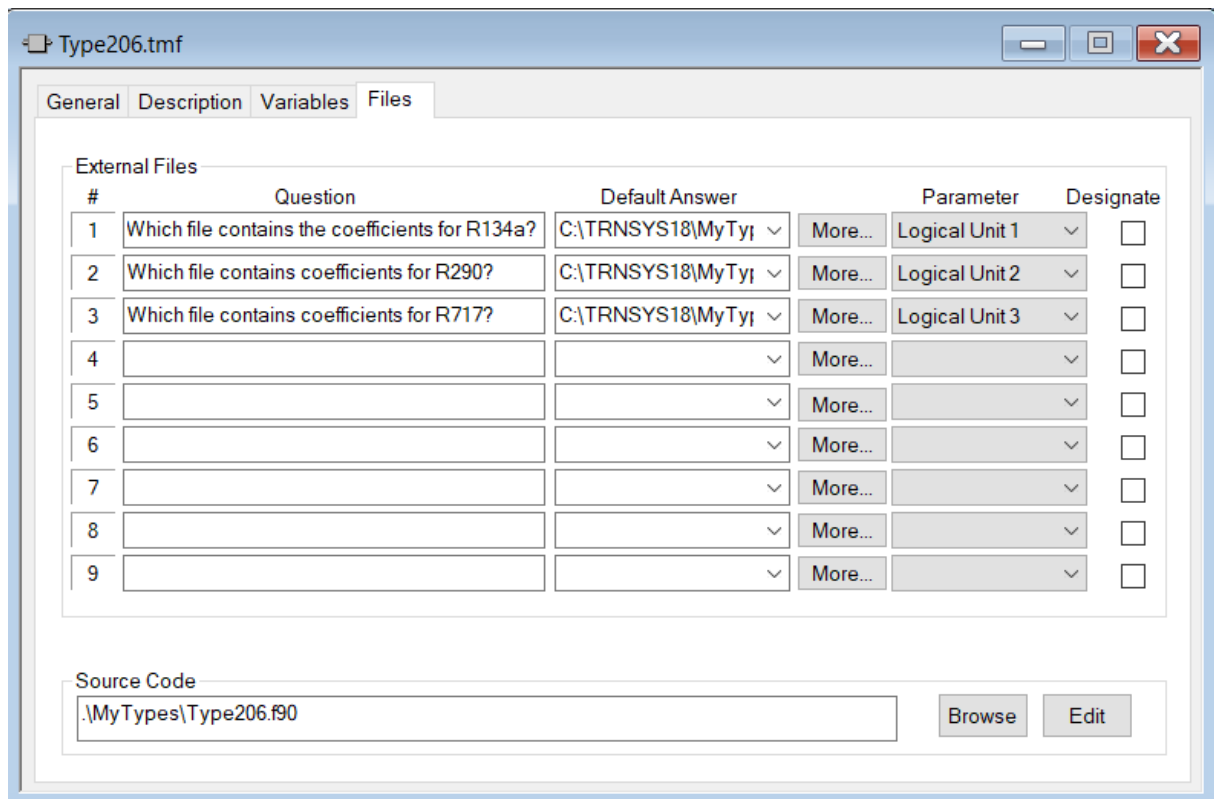

***Figure V.1**: Files tab of the Proforma window of compressor Type 206. Additional data files may be added to the component.*

In order to connect data files to the component additional Logical Unit parameters must be defined. This is done in the Variables tab of the Proforma Window. Figure V.2 shows the Variables tab of the Proforma window of compressor Type 206. The addition of parameters also requires modifications of the source code. This includes variable declaration of the parameter, editing the number of parameters and reading the parameter value from the input file. In the calculations the new refrigerant and Logical Unit number must be added to the code where the correct data file is selected. Refer to TRNSYS documentation Vol. 7 Programmer's Guide for further information on the processing of parameters and parameter values in the source code.
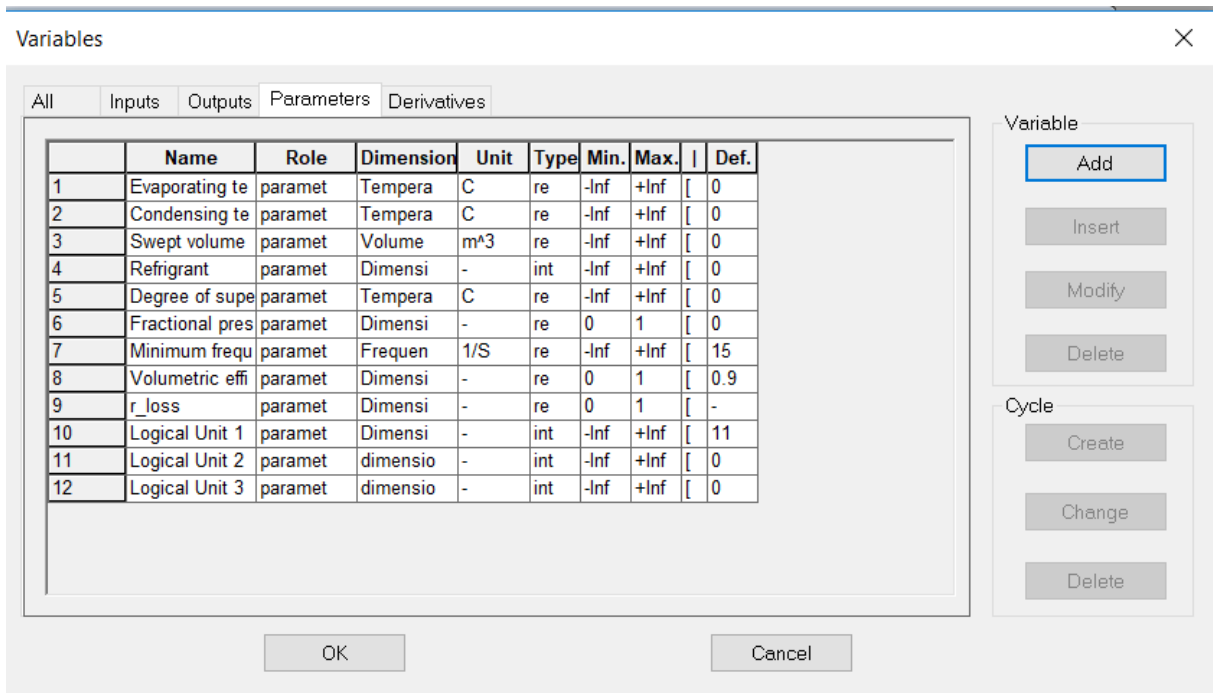
Figure V.2: *The variables tab of the Proforma window of compressor Type 206. New parameters may be added, and existing parameters may be modified.*