Even Kristian Tønsberg

Modeling Approach for a Liquid-Injected NH_3-H_2O Screw Compressor

Modelleringsmetode for en væskeinjesert NH_3-H_2O skruekompressor

Master's thesis in Energy and Environmental Engineering

Supervisor: Trygve Magne Eikevik

June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Even Kristian Tønsberg

Modeling Approach for a Liquid-Injected NH_3-H_2O Screw Compressor

Modelleringsmetode for en væskeinjesert NH_3-H_2O skruekompressor

Master's thesis in Energy and Environmental Engineering
Supervisor: Trygve Magne Eikevik
June 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Norwegian University of Science and Technology

Faculty of Engineering
Department of Energy and Process Engineering

EPT-M-2019-20

MASTER THESIS

for

student Even Kristian Tønsberg

Spring 2020

Modeling approach for a liquid-injected NH_3-H_2O screw compressor

Modelleringsmetode for en væskeinjisert NH_3-H_2O skruekompressor

Background and objective

The heat pump market has so far mainly focused on residential heat pumps for space heating and domestic hot water production. Less focus has been on heat pumps for higher temperature applications and industrial use, due to high initial investment costs, competition with alternative investments, and non-mature or non-existing technologies for the applications. New developments in compact high-pressure components, e.g. compressors, ejectors and heat exchangers for CO_2, ammonia and hydrocarbon heat pump systems, are important drivers to change this situation.

The master work focuses on the development of a modeling approach for a liquid-injected ammonia-water compressor, which is an essential component of a combined absorption-compression heat pump cycle. This includes a comprehensive description of the properties of screw compressors as well as the identification and explanation of existing screw compressor models from the literature. Subsequently, a modeling approach based on the identified screw compressor models will be developed and implemented.

The following tasks are to be considered:

1. Literature review of absorption-compression heat pumps
2. General description of the combined absorption-compression heat pump cycle
3. Definition of working principles, parameters and operating conditions of screw compressors
4. Identification and characterization of existing screw compressor models
5. Development and implementation of a modeling approach
6. Make a scientific paper from the main results of the work
7. Make proposal for further work

-- “ --

Preface

This thesis is submitted to complete the degree of Master of Technology at the Norwegian University of Science and Technology (NTNU). The work was conducted at the Department of Energy and Process Engineering under the supervision of Professor Trygve Magne Eikevik.

The main part of the work concerns computer modeling of screw compressors. Much time and effort have been put into familiarising with possible modeling tools, writing code and optimizing the simulation procedure. I hope the developed modeling approach will be of value in future development and analysis of screw compressors, and that the presented work can act as guidance and inspiration to other modeling researchers.

I would like to express my deep gratitude to Trygve Magne Eikevik and Ignat Tolstorebrov for giving me the opportunity to work with such an interesting topic and their continuous support and encouragement. I would also like to express my heartfelt appreciation to Marcel Ulrich Ahrens for his patient guidance, his enthusiasm for my work and our countless enlightening discussions.

Trondheim, Norway

June 2020

Even Kristian Tønsberg

Summary

Measures to improve energy efficiency is in urgent demand in today's society due to continuously increasing energy consumption and resource depletion. Heat pump technology is a powerful tool to reduce energy consumption related to heating. There has been a lack of development in heat pump technology intended for industrial use, despite the fact that energy intensive industries represent immense energy saving possibilities. Progress has been slow as a result of high investment costs and an absence of mature technology capable of meeting industrial operating requirements. The work presented in this report intends to be a contribution to the much needed development in industrial heat pump technology.

The work concerns a combined absorption-compression heat pump for production of hot water from surplus heat, and the focus is set on the heat pump's compressor. The aim is to gather knowledge related to absorption/compression technology, screw compressors and computer modeling, and to utilize this knowledge to develop a modeling approach for a screw compressor.

The report is divided into eight sections. A brief introduction to the objectives of the work is given in the first section. In the second section of the report, the working principles of basic heat pump cycles are explained, and it is demonstrated how these cycles can be formed into a combined absorption-compression heat pump cycle. Challenges related to combined absorption-compression heat pumps are found through literature review, and it is reasoned that the compressor is a crucial system component. The temperature at the outlet of the compressor is established to be a dominating constraint. High discharge temperatures makes it unfeasible to operate with conventional compressors in the desired heat sink temperature range between 140 °C and 180 °C. Therefore, development of a new compressor capable of operating at high pressure and temperature is needed. An ammonia-water screw compressor with liquid injection is discovered as a promising option.

In the third section of the report, principles regarding design and operation of screw compressors are demonstrated. A screw compressor is a type of positive displacement compressor where vapor is trapped between two helix-shaped rotors. The rotors form a series of working chambers where the volume of each chamber gradually decreases during rotation. The compression process is found to be very complex. Small clearances between the two rotors and the compressor casing cause leakage flows between adjacent chambers. This gives an increase in operating temperature, it lowers the compressor's throughput, and it decreases the system's energy efficiency. Oil or liquid must be injected into the compressor to ensure efficient operation by sealing the clearances and cooling down the compressed vapor.

In the fourth section of the report, the focus is set on modeling and simulation of screw compressors. Numerical modeling is found to be the most suitable approach for the objectives of this work, and two previously developed numerical models by Chamoun et al. (2013) and Tian et al. (2017a) are closely investigated. The models are based on equations for conservation of mass

and energy. Pressure, temperature, enthalpy and other thermodynamic properties are calculated throughout the compression process as a function of the rotational angle of the compressor's male rotor. Furthermore, discussions about simulation environment, numerical procedures and fluid property software are made. The Dymola simulation environment is found practical due to its graphical user interface and the favourable attributes of the Modelica modeling language. The viability of the investigated numerical models is assessed by looking at the simulation results presented by Chamoun et al. (2013) and Tian et al. (2017a). The model developed by Chamoun et al. (2013) offers easy implementation and short computational time, but it lacks accuracy and the output provides little detail. The model developed by Tian et al. (2017a) offers higher complexity and better accuracy.

In the fifth section of the report, a modeling approach for a screw compressor is developed based on knowledge obtained from the literature. The model is written in the Modelica modeling language, and it is implemented and solved in the Dymola simulation environment. The model takes into account the effects of internal leakage flows, heat losses and liquid injection, and it can be used for both steady-state and transient analysis of such phenomena. The model is aimed at efficient investigations of different compressor arrangements and operating conditions, as well as effortless implementation into bigger system models. Simulations are carried out successfully, and a set of results are presented. Among other things, the simulation results show that it is advantageous to inject liquid with a low ammonia mass fraction, and it is shown that the liquid should be injected at the very beginning of the compression phase or during the suction phase.

In the sixth section of the report, a secondary model is developed using the Engineering Equation Solver (EES). This model is designed for detailed thermodynamic analysis of the two-phase process that takes place during compression. The EES model is more limited than the Modelica model, and it is only capable of steady-state analysis. However, opposed to the Modelica model, the EES model is capable of estimating effects of non-equilibrium. Results from two separate calculations are presented, one with complete vapor-liquid equilibrium and one with non-equilibrium conditions. The results show that desorption and absorption of ammonia takes place during compression, and that the rate of desorption/absorption is lower in the case with non-equilibrium.

In the seventh section of the report, the simulation results are discussed and the two modeling approaches are compared against each other. It is argued that the Modelica approach has the biggest potential for utilization and further development. The most challenging aspects of screw compressor modeling are found to be related to internal leakages and liquid evaporation. Moreover, suggestions on how to improve and further develop the models are given. In the eighth and final section of the report, conclusions and proposals for further work are presented. The developed screw compressor model is to be validated against experimental results from a combined absorption-compression heat pump test rig, and the model can be integrated into a bigger system model to simulate the complete heat pump cycle.

Sammendrag

Det er et stort behov for tiltak som kan fremme energieffektivitet i dagens samfunn grunnet stadig økende forbruk av energi og naturressurser. Varmepumpeteknologi er et kraftfullt verktøy for å redusere energiforbruk knyttet til oppvarming. Det har vært mangelfull utvikling i varmepumpeteknologi rettet mot industriell bruk, til tross for at energiintensive industrier representerer enorme energibesparingsmuligheter. Framgangen har vært treg på grunn av høye investeringskostnader og et fravær av moden teknologi som er i stand til å oppfylle industrielle driftskrav. Arbeidet presentert i denne rapporten har til hensikt å være et bidrag til den sterkt trengte utviklingen i industriell varmepumpeteknologi.

Arbeidet omfatter en kombinert absorpsjons-kompresjons-varmepumpe for produksjon av varmtvann fra overskuddsvarme, og fokuset er satt på varmepumpens kompressor. Målet er å tilegne kunnskap relatert til absorpsjons-/kompresjons-teknologi, skruekompressorer og datamodellering, og å utnytte denne kunnskapen til å utvikle en modelleringsmetode for en skruekompressor.

Rapporten er delt inn i åtte deler. En kort introduksjon til oppgavens formål blir gitt i den første delen. I den andre delen av rapporten blir virkemåten til grunnleggende varmepumpesykluser forklart, og det demonstreres hvordan disse syklusene kan formes til en kombinert absorpsjons-kompresjons-varmepumpesyklus. Utfordringer knyttet til kombinert absorpsjons-/kompresjons-varmepumper blir funnet gjennom litteraturgransking, og det argumenteres for at kompressoren er en kritisk systemkomponent. Temperaturen ved utgangen av kompressoren blir påvist som en dominerende begrensning. Høye utløpstemperaturer gjør at det ikke er praktisk gjennomførbart å operere med konvensjonelle kompressorer i det ønskede temperaturområdet for varmesluket mellom 140 °C og 180 °C. Derfor er det behov for utvikling av en ny kompressor som evner å operere ved høyt trykk og høy temperatur. En ammoniakk-vann-skruekompressor med væskeinjeksjon blir avdekket som et lovende alternativ.

I den tredje delen av rapporten blir prinsipper angående utforming og drift av skruekompressorer demonstrert. En skruekompressor er en type fortregningskompressor hvor gass stenges inne mellom to heliksformede rotorer. Rotorene danner en serie med arbeidskamre hvor volumet til hvert kammer gradvis blir mindre under rotasjon. Kompresjonsprosessen påvises å være veldig kompleks. Små klaringer mellom de to rotorene og kompressorskallet forårsaker lekasjestrømninger mellom tilstøtende arbeidskamre. Dette gir økt driftstemperatur, det senker kompressorens gjennomstrømning, og det reduserer systemets energieffektivitet. Olje eller væske må sprøytes inn i kompressoren for å sikre effektiv drift ved å forsegle klaringene og kjøle ned den komprimerte gassen.

I den fjerde delen av rapporten settes fokuset på modellering og simulering av skruekompressorer. Numerisk modellering blir fastslått som den mest egnede metoden for oppgavens formål, og to ulike numeriske modeller utviklet av Chamoun et al. (2013) og Tian et al. (2017a)

studeres grundig. Modellene er basert på ligninger for bevaring av masse og energi. Trykk, temperatur, entalpi og andre termodynamiske egenskaper beregnes gjennom hele kompresjonsprosessen som en funksjon av rotasjonsvinkelen til kompressorens hovedrotor. Videre diskuteres simuleringsplattformer, numeriske prosedyrer og programvare for beregning av termodynamiske egenskaper. Simuleringsplattformen Dymola blir ansett som gunstig på grunn av dens grafiske brukergrensesnitt og de fordelaktige egenskapene til modelleringsspråket Modelica. Kvaliteten på de utforskede numeriske modellene blir vurdert ved å se på simuleringsresultatene presentert av Chamoun et al. (2013) og Tian et al. (2017a). Modellen utviklet av Chamoun et al. (2013) tilbyr enkel implementering og kort beregningstid, med den mangler presisjon og utdataen er lite detaljert. Modellen utviklet av Tian et al. (2017a) har høyere kompleksitet og bedre presisjon.

I den femte delen av rapporten blir en modelleringsmetode for en skruekompressor utviklet basert på kunnskap tilegnet fra litteraturen. Modellen er skrevet i modelleringsspråket Modelica, og den blir implementert og løst i simuleringsplattformen Dymola. Modellen iberegner virkningene av indre lekasjestrømninger, varmetap og væskeinjeksjon, og den kan brukes til både stabil tilstand og transient analyse av slike fenomener. Modellen er tilsiktet effektive undersøkelser av ulike kompressorkonfigurasjoner og driftsforhold, så vel som enkel implementering i større systemmodeller. Vellykkede simuleringer foretas, og et sett med resultater presenteres. Simuleringsresultatene viser blant annet at det er fordelaktig å injesere væske med en lav massefraksjon av ammoniakk, og det blir vist at væsken bør injeseres helt i starten av kompresjonsfasen eller under sugefasen.

I den sjette delen av rapporten blir en sekundær modell utviklet ved bruk av Engineering Equation Solver (EES). Denne modellen er designet for detaljert termodynamisk analyse av to-fase prosessen som foregår under kompresjon. EES-modellen er mer begrenset enn Modelica-modellen, og den er kun kapabel til stabil tilstand analyse. Til gjengjeld er EES-modellen, i motsetning til Modelica-modellen, kapabel til å estimere virkninger av ikke-likevekt. Resultater fra to separate beregninger presenteres, én med fullstendig damp-væske-likevekt og én med ikke-likevekt. Resultatene viser at desorpsjon og absorpsjon av ammoniakk utspiller seg under kompresjon, og at desorpsjons-/absorpsjons-raten er lavere i tilfellet med ikke-likevekt.

I den syvende delen av rapporten blir simuleringsresultatene diskutert og de to modelleringsmetodene sammenlignes med hverandre. Det argumenteres for at Modelica-metoden har størst potensiale for anvendelse og videre utvikling. De mest utfordrene aspektene ved skruekompressormodellering anses å være relatert til indre lekasjestrømninger og fordampning av væske. Videre gis forslag til hvordan modellene kan forbedres og videreutvikles. I den åttende og siste delen av rapporten presenteres konklusjoner og forslag til videre arbeid. Den utviklede skruekompressormodellen skal valideres mot eksperimentelle resultater fra en testrigg for en kombinert absorpsjons-kompresjons-varmepumpe, og modellen kan integreres inn i en større systemmodell for å simulere den fullstendige varmepumpesyklusen.

Contents

List of Figures

Nomenclature

Latin Letters

A	Cross-sectional area	m^2
a	Speed of sound	m/s
B	Dynamic bearing load	N
C	Empirical coefficient	-
c	Specific heat capacity	J/(kg·K)
d	Diameter	m
E	Total energy	J
f	Operational frequency of the compressor	Hz
g	Gravitational acceleration	$\mathrm{m/s}^2$
h	Specific enthalpy	J/kg
m	Mass	kg
$\dot{m}$	Mass flow rate	kg/s
N	Number of	-
n	Polytropic index	-
P	Power	W
p	Pressure	Pa
Q	Thermal energy	J
$\dot{Q}$	Heat transfer rate	W
q	Vapor quality	-
S	Segment number for liquid injection	-
s	Specific entropy	J/(kg·K)
T	Temperature	K
u	Specific internal energy	J/kg

V	Volume	m^3
$\dot{V}$	Volume flow rate	m^3/s
v	Velocity	m/s
W	Mechanical work	J
$\dot{W}$	Mechanical power	W
x	Ammonia mass fraction	-
z	Height	m

Greek Letters

α	Heat transfer coefficient	W/m^2K
γ	Latent heat of vaporization	J/kg
ζ	Leakage flow resistance	-
η	Efficiency	-
θ	Rotational angle of the male rotor	rad
κ	Isentropic exponent	-
λ	Volumetric efficiency	-
μ	Friction coefficient	-
ν	Specific volume	m^3/kg
ν_i	Built-in volume ratio	-
π_i	Built-in pressure ratio	-
ρ	Density	kg/m^3
φ	Coefficient in the Lin equation	-
ω	Angular velocity of the male rotor	rad/s

Subscripts

bear	Bearings
bod	Compressor body

c	Constituent of the working fluid
cha	Compression chamber
$comp$	Compression
cv	Control volume
dis	Discharge
eff	Effective
eq	Equilibrium
f	Saturated liquid
fla	Vapor caused by flash evaporation
g	Saturated vapor
h	At constant enthalpy
$high$	High-pressure chamber
in	Inlet
ind	Indicated
inj	Liquid injection
$isen$	Isentropic
j	j^{th} step in compression cycle / j^{th} control volume
$leak$	Leakage path
$leak\text{-}in$	Leakage flow into the control volume
$leak\text{-}out$	Leakage flow out from the control volume
liq	Liquid
lob	Lobes on the male rotor
low	Low-pressure chamber
mec	Mechanical
out	Outlet

$outflow$	If the flow direction is out of the component
p	At constant pressure
$port$	Fluid port from the TIL Modelica library
$prev$	Previous step in compression cycle
$seal$	Shaft seals
seg	Segment
sha	Compressor shaft
suc	Suction
T	At constant temperature
$theo$	Theoretical value
tot	Total
v	At constant specific volume
vap	Vapor

Abbreviations

CACC	Combined Absorption-Compression Cycle
CACHP	Combined Absorption-Compression Heat Pump
CFD	Computational Fluid Dynamics
COP	Coefficient of Performance
EES	Engineering Equation Solver
NTNU	Norwegian University of Science and Technology
VAC	Vapor Absorption Cycle
VCC	Vapor Compression Cycle
VCHP	Vapor Compression Heat Pump
VLE	Vapor-Liquid Equilibrium
VSD	Variable Speed Drive

1 Introduction

Increasing energy consumption and environmental pollution are of the most important challenges in today's society. According to IEA (2019), energy consumption worldwide grew by 2.3 % in 2018, nearly twice the average rate of growth since 2010. Looking at the United Nations' Sustainable Development Goals, goal number 12 ("Responsible Consumption and Production") aims at "doing more and better with less" (UN, 2019). Hence improving the efficiency of energy systems in order to decrease the consumption of resources is of great interest.

Using heat pumps instead of traditional boilers is an effective measure to reduce energy consumption related to heating. The heat pump market has so far mainly focused on residential heat pumps for space heating and domestic hot water production. Less focus has been on heat pumps for higher temperature applications and industrial use. The industrial sector accounts for the largest share of energy consumption of any end-use sector (EIA, 2017). Energy-intensive industries with high operating temperatures represent possibilities to save substantial amounts of energy through utilization of industrial heat pumps. Arpagaus et al. (2018) recognized large application potentials, especially in food, paper, metal and chemical industries. In their analysis of the European heat pump market, they found a technical potential of 113 PJ for process heat between 100 °C and 150 °C. The lack of focus on heat pumps capable of meeting this demand is due to high initial investment costs, competition with alternative investments and non-mature or non-existing technologies for the applications. New developments related to compact high-pressure components, e.g., compressors, ejectors and heat exchangers for CO_2, ammonia and hydrocarbon heat pump systems, are important drivers to change this situation. The work presented in this report aspires to be a contribution to the development of these components.

The work will concentrate on modeling and simulation of a screw compressor to be utilized in a high-temperature heat pump for production of hot water from surplus heat. A test rig for a combined absorption-compression heat pump is currently being built in a laboratory at the Department of Energy and Process Engineering, NTNU. Among other things, the test rig will be used examine different compressor arrangements. Using computer modeling to optimize the compressor configuration, and later being able to compare experimental data from the test rig with theoretical data from simulations, is of great value.

1.1 Objectives

The main objective of this study is to develop a modeling approach for a liquid-injected screw compressor that uses a mixture of ammonia and water as the working fluid. The model's purpose is to be utilized in research and development of screw compressors, and this study is aimed at development of a screw compressor for a combined absorption-compression heat pump.

The following tasks are to be considered:

1. Literature review of absorption-compression heat pumps.
2. General description of the combined absorption-compression heat pump cycle.
3. Definition of working principles, parameters and operating conditions of screw compressors.
4. Identification and characterization of existing screw compressor models.
5. Development and implementation of a modeling approach.
6. Make a scientific paper from the main results of the work.
7. Make proposal for further work.

2 The Heat Pump Process

2.1 The Vapor Compression and Absorption Cycles

In this section, based on refrigeration fundamentals presented by Tabatabaian and Rajput (2017) and CIBSE (2016), a brief introduction to the vapor compression cycle (VCC) and the vapor absorption cycle (VAC) is given.

The most commonly used residential heat pumps are based on the vapor compression cycle. A basic VCC is illustrated in Figure 2-1. There are four main components in this cycle: a compressor, a condenser, an expansion valve and an evaporator. The chosen working fluid, e.g. ammonia, circulates between the four components in a closed circuit. Saturated vapor at low pressure (state 1) enters the compressor, where it is compressed to a higher pressure, resulting in a higher temperature and a so-called superheated vapor (state 2). The superheated vapor enters the condenser, where it is cooled down and condensed. The change of state from vapor to liquid releases large amounts of heat that is absorbed by a counter-flow of air or water, transferring heat to the high temperature heat sink. The working fluid leaves the condenser as saturated liquid (state 3), before it enters the expansion valve. In the expansion valve the working fluid is throttled down to a lower pressure. This causes partial flash evaporation (a phase change phenomenon observed when a liquid is exposed to a sudden pressure drop) of the working fluid (Mansour and Müller, 2019, p.146). Hence, a mixture of liquid and vapor enters the evaporator (state 4). In the

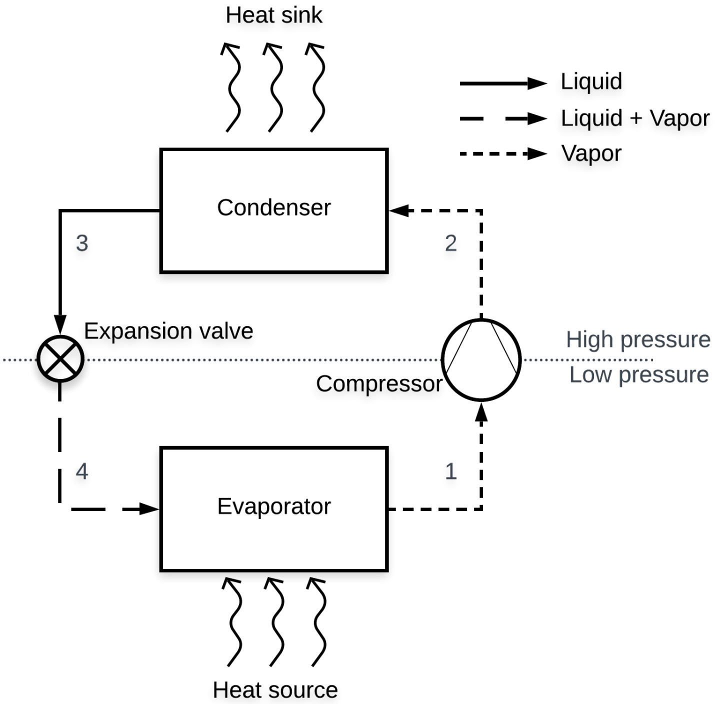


Figure 2-1: Schematic diagram of a basic vapor compression cycle. Based on Tabatabaian and Rajput (2017, p.739), and CIBSE (2016, p.84).

evaporator, the working fluid extracts heat from the low temperature heat source and vaporizes to a low-pressure vapor (state 1). Thus the cycle is complete, and in this manner heat can be transferred from a low-temperature heat source to a high-temperature heat sink.

An alternative to the VCC is the VAC (Vapor Absorption Cycle). A basic VAC is illustrated in Figure 2-2. The two cycles are similar, except that the mechanical compressor in the VCC is replaced by four components creating a sub-cycle in the VAC: an absorber, a pump, a generator and a second expansion valve. This sub-cycle (state $5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 5$ in Figure 2-2) is often referred to as a thermal compressor (Staedter and Garimella, 2018, p.28). A zeotropic mixture circulates in the thermal compressor. A zeotropic, or non-azeotropic, mixture is a mixture of multiple substances with different evaporation temperatures (Gaspar and da Silva, 2015, p.244), e.g. a mixture of ammonia and water. With an ammonia-water mixture, the ammonia acts as the refrigerant and the water acts as the absorbent. The thermal compressor arrangement takes advantage of the fact that the solubility of ammonia in water decreases with increasing temperature and pressure.

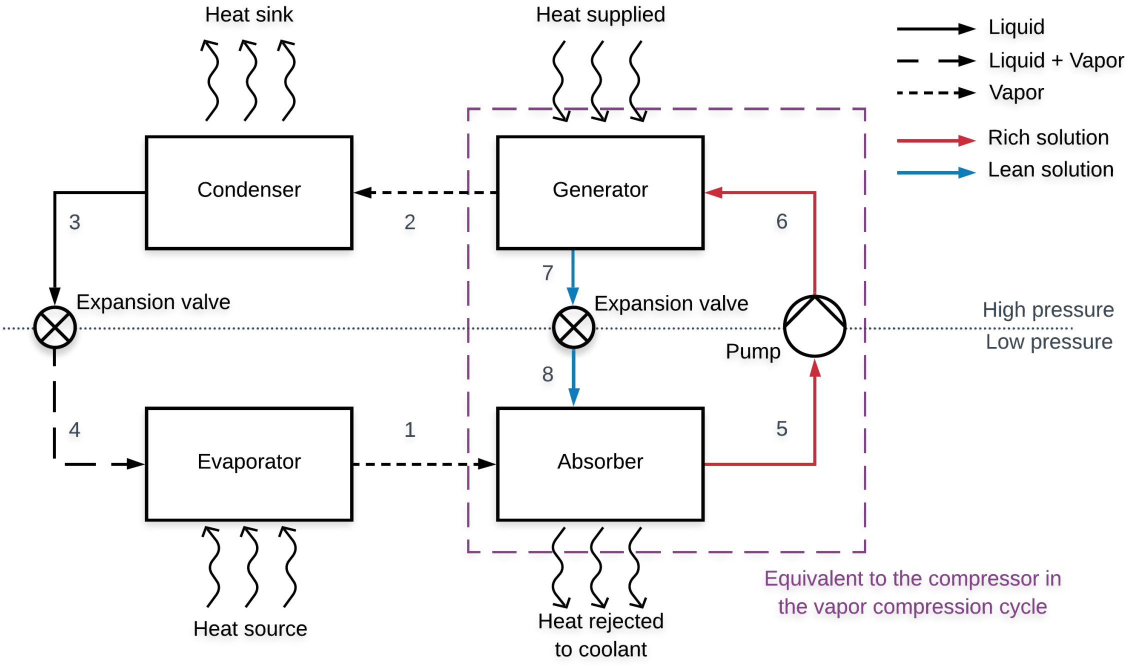


Figure 2-2: Schematic diagram of a basic vapor absorption cycle. Based on Tabatabaian and Rajput (2017, p.755), and CIBSE (2016, p.93).

States 1, 2, 3 and 4 in Figure 2-2 are equal to the states described for the VCC in Figure 2-1. The refrigerant rejects heat through condensation in the condenser, and absorbs heat through evaporation in the evaporator. In the VAC, the saturated vapor leaving the evaporator (state 1) enters the absorber, where it is absorbed by the low-pressure absorbant. Thus creating what we refer to as the rich solution (state 5), i.e., a solution with a high refrigerant-to-absorbent ratio. The change of state from vapor to liquid during the absorption process releases heat. To decrease the temperature of the solution and thereby increase the solubility, heat is rejected to a coolant passing through the absorber. The rich solution is then pumped to a higher pressure (state 6) before it enters the generator. In the generator, heat is supplied in order to raise the temperature of the solution. Higher temperature and pressure causes the solubility to go down and thus separation of high-pressure refrigerant vapor occurs. The high pressure vapor is passed on towards the condenser (state 2), while the lean solution (solution with lower refrigerant concentration) returns to the absorber through an expansion valve (state 7 and 8). The expansion valve throttles the pressure of the solution down to its initial pressure. Thus the cycle is complete.

The main difference between the VCC and the VAC is the type and the amount of energy input. The VAC requires less electric power, since pumping a liquid requires less energy than compressing a vapor between the same pressures. On the other hand, the generator in the VAC requires an additional heat input, which there is no need for in the VCC.

Regarding the illustrations of the VCC and VAC in Figures 2-1 and 2-2, an important matter to keep in mind is that these are simplified cycles. They are made to illustrate the basic working principles, but they do not include all the elements that are present in a practical application. Elements such as receiver tanks, economizers and others are neglected.

2.2 The Combined Absorption-Compression Cycle

The two absorption and compression schemes can be merged into one single cycle. This merged cycle is referred to as the combined absorption-compression cycle (CACC). The most basic type of CACC is the Osenbrück cycle, named after its inventor (Osenbrück, 1895). The Osenbrück cycle is illustrated in Figure 2-3. In what follows, the working principle of this cycle is explained, based on theory presented by Nordtvedt (2005) and Jensen et al. (2014).

In the CACC, the evaporator and the condenser is replaced by a desorber and an absorber. Contrary to the VAC, the absorber is placed on the high-pressure side. The desorber works in the same manner as the generator in the VAC, but in the CACC it is placed on the low-pressure side. The rich solution at low pressure (state 8) enters the desorber where heat from the source is added. This causes the temperature to rise and the solubility to decrease. Hence vapor is generated. The low-pressure vapor and the lean solution are separated before they are drawn towards the compressor and the pump respectively. The compressor increases the pressure and temperature of the vapor (state $1 \rightarrow 2$), while the pump elevates the the pressure of the lean solution correspondingly (state $3 \rightarrow 4$). The temperature rise over the pump is minor. Therefore, to improve the performance of the cycle, a heat exchanger is added to the solution sub-cycle. Heat is exchanged between the lean and rich solutions, causing a temperature rise in the lean solution (state $4 \rightarrow 5$) and a temperature drop in the rich solution (state $6 \rightarrow 7$). The high-pressure lean solution (state 5) is mixed with the high-pressure vapor (state 2) at the entrance of the absorber. Just like in the VAC, the absorption process releases heat, but here at a much higher temperature (since the absorber is located on the high-pressure side). This heat is transferred to the heat sink.

The absorption process results in a rich solution that eventually passes through the solution heat exchanger before it is throttled down to the desorber pressure (state 6 → 7 → 8). Thus the cycle is complete.

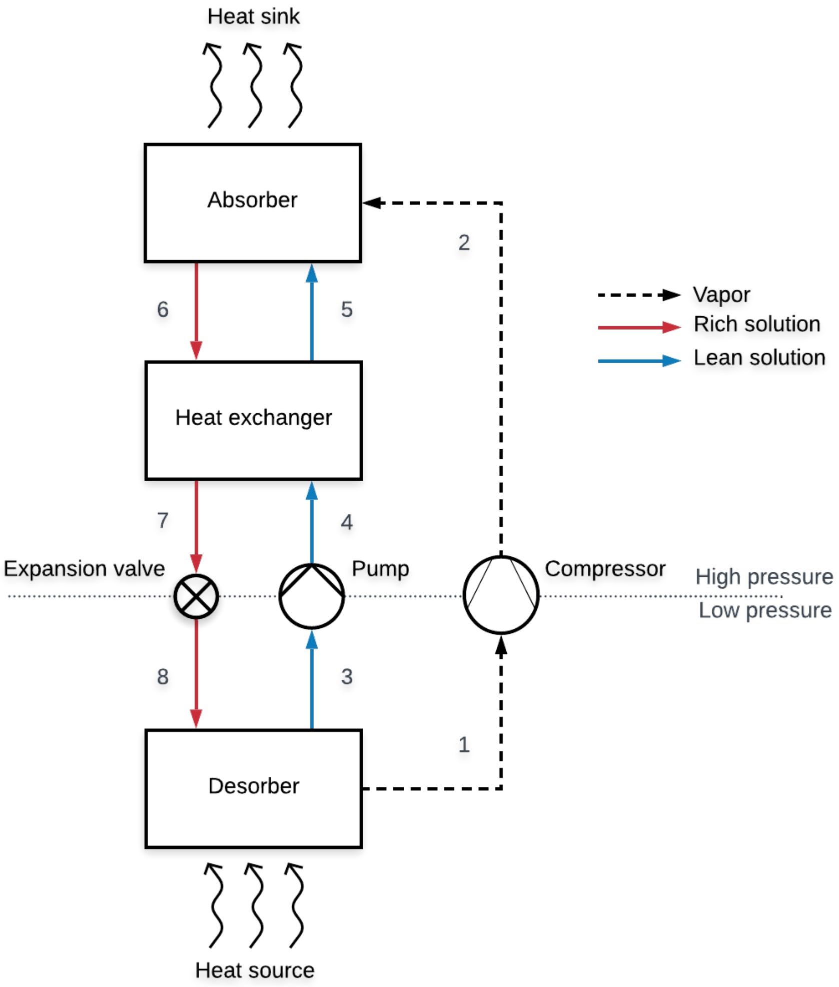


Figure 2-3: Schematic diagram of a basic absorption-compression cycle, the Osenbrück cycle. Adopted from Nordtvedt (2005, p.19).

Nordtvedt (2005) and Jensen et al. (2014) point out several advantages with utilization of combined absorption-compression heat pumps (CACHP). Compared to a conventional vapor compression heat pump (VCHP) using a single working fluid, the working fluid pair in a CACHP offers much more design flexibility. By altering the composition of the working pair, i.e., the refrigerant-to-absorbent ratio, the properties of the working fluid can be adapted to the heat source and heat sink properties, as well as the properties of each heat pump component, to obtain optimal working conditions. Industrial heat supply and recovery of surplus heat requires large heat source and heat sink temperature glides. Contrary to the isothermal processes of evaporation and condensation in a VCHP, the absorption and desorption processes in a CACHP are non-isothermal processes. By matching the temperature glides of the working fluid with the temperature glides of the heat source and heat sink fluids, the thermal losses are reduced, resulting in a higher COP. This makes the CACHP suitable for many industrial applications. Another suitable feature of the CACHP is that using a zeotropic mixture reduces the vapor pressure. The sink temperature from a conventional VCHP is limited because the condenser pressure becomes too high, whereas a CACHP can achieve much higher sink temperatures due to the reduced vapor pressure. In addition, the zeotropic mixture adds an extra degree of control. The heating capacity of the heat pump can be adjusted without changing the rotational speed of the compressor, simply by altering the concentration of refrigerant in the circulating mixture.

2.3 The Challenge of Developing a Suitable Compressor

The groundwork for development of the combined absorption-compression heat pump started in 1895 with the Osenbrück cycle shown in Figure 2-3. The first theoretical study on the CACHP was performed by Altenkirch (1950). For a long time, until the last part of the 20th century, there was little interest and no momentous development regarding the subject. However, in recent years the interest has grown substantially and numerous studies on the CACHP have been carried out.

Itard (1998) carried out a theoretical and experimental study on a wet-compression type of CACHP. She reasoned that wet compression (compression of working fluid in both liquid and vapor phase) may be the only way to reach heat sink temperatures that are high enough for industrial applications, since it limits the temperature of the superheated vapor. If not limited, superheating can cause compressor failure and refrigerant or oil decomposition. In her recommendations for further research, she advised that studies on design of oil-free wet compressors should be initiated in order to find suitable ways of mixing liquid and vapor during compression. Zaytsev and Ferreira (2002) designed and tested an oil-free compressor to be utilized in a CACHP. They recognized the fact that the compressor should be tolerant to liquid carry-over, as the working fluid exiting the desorber is a mixture of liquid and gas, and thus they chose to design a twin-screw compressor. Their argument for oil-free operation was that separation of oil from the liquid portion of the working fluid is challenging. Contrary to dry gas compressors, oil cannot be separated by the use of gravity when liquid refrigerant is present. Additional complex and expensive equipment is needed for separation. An alternative solution would be to not separate the oil and let it circulate in the whole system. Zaytsev and Ferreira excluded this option, as they claimed that the presence of oil in the heat exchangers reduces the heat transfer and the overall performance of the heat pump. Instead, they suited the compressor with injection of liquid ammonia-water from the solution circuit in the CACHP. With the absence of oil, the injection liquid is essential to seal gaps and reduce pressure losses in the compressor. The experimental results from their study confirmed that the compressor was capable of oil-free operation with ammonia-water injection.

Nordtvedt (2005) developed a steady-state computer model and an experimental test rig for a CACHP. The test rig was equipped with a two-stage reciprocating compressor. The compressor was oil-lubricated, had water-cooled cylinder heads and a design pressure of 26 bar. The maximum discharge temperature was set to 160 °C. Experimental results showed that the necessary cooling of the cylinder heads to keep the discharge temperature below this limit was larger than expected. In addition, the results demonstrated that the amount of dissolved oil in the discharge vapor is a function of discharge temperature. High temperatures lead to high amounts of dissolved oil. The effect of an oil film on overall heat transfer coefficient in the absorber and the desorber was calculated, proving that an increase in the amount of oil significantly diminishes the overall performance of the heat pump. With a heat sink inlet temperature of 50 °C, a maximum outlet temperature of 110 °C was obtained in simulations, corresponding to a maximum temperature glide of 60 K. In experiments, a heat sink outlet temperature just below 100 °C was obtained.

Ferreira et al. (2006) continued the research on screw compressors with liquid injection. Both theoretically and experimentally, they investigated the influence of the injection port location on the performance of the compressor. Their results show that injection during the compression process, rather than injection into the suction plenum, gives major performance improvement. With a single injection point, the ideal injection location is found to be during the start phase of compression. Chamoun et al. (2013) developed a mathematical model for a screw compressor to be used in high-temperature heat pumps with water as refrigerant. Their discretized compressor model includes the issue of liquid injection, and accounts for phenomena such as internal leakage flows and heat losses to the surroundings. This model will be discussed further in section 4.3. The mathematical model was solved using Dymola, and the results proclaimed that liquid injection is essential to avoid compressor failure and to improve the efficiency of the compressor.

Under investigation of the technical and economic working domains of an ammonia-water CACHP, Jensen (2015) concludes with the fact that the compressor discharge temperature is a dominating constraint. With constraints on the high pressure and the discharge temperature set to 50 bar and 170 °C respectively, his calculations show a maximum heat supply temperature of 129 °C for a one-stage CACHP and 145 °C for a two-stage CACHP. During his discussion regarding feasible working domains, he mentions that use of a cooled screw compressor or an oil-free compressor could relax the constraint on the discharge temperature and allow heat supply temperatures above 150 °C. Moreover, Ahrens et al. (2019) investigated the availability of compressors, to be utilized in the CACHP test rig at NTNU, that are capable of operating at high pressures to achieve heat sink temperatures of 140 °C to 180 °C. They identified the various requirements for such a compressor and discussed the suitability of different compressor types (Further details are given in section 3.5). In their conclusion, they state that there is no commercially available compressor capable of handling the required operating conditions. Hence, there is a need for development and further research on promising compressor solutions.

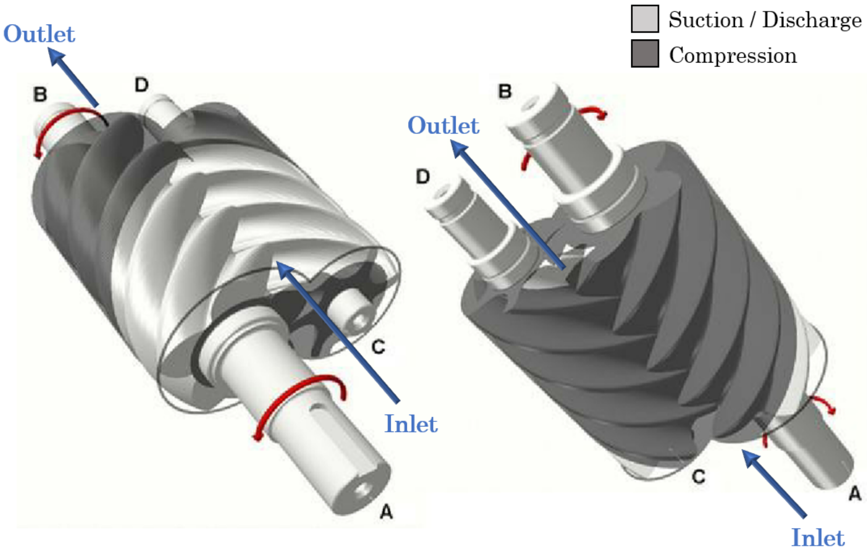
As presented here, much effort has been made to develop a suitable compressor for an ammonia-water CACHP, but there are still unresolved issues related to it. The compressor is a crucial component, and tools that can cause better understanding of it is therefore of great value. This is motivation for the development of a simulation model that can give detailed insight into operation of ammonia compressors at high pressure and temperature. Going forward, the main focus will be on a twin-screw ammonia compressor with injection of liquid ammonia-water.

3 Design and Operation of Screw Compressors

3.1 Working Principle

Compressors can be grouped into two main categories: dynamic compressors and positive displacement compressors. A dynamic compressor induces pressure change in the fluid passing through it by changing the momentum of the fluid. The compressor accelerates the fluid to high velocity, before it is restricted again, reducing the velocity. Consequently kinetic energy is transformed into static pressure, and the pressure of the fluid rises. A positive displacement compressor creates pressure change by trapping the fluid in a closed chamber, before reducing the volume of the chamber. The volume reduction causes the pressure of the fluid inside the chamber to increase. A widely used type of positive displacement compressor is the screw compressor.

A twin-screw compressor consist of two helix-shaped rotors meshed into each other, confined within a casing. The two rotors are often distinguished as the male and the female rotor, where the male rotor has convex lobes that meshes into the concave cavities of the female rotor. An example of such a rotor pair is shown in Figure 3-1. The working principle indicated in this figure is described by Stosic et al. (2005c), and is as follows: Spaces between the lobes on each rotor form a series of working chambers where vapor is contained. Starting at the front and top of the the rotors, looking at Figure 3-1a, there is an initial point where the trapped volume in a chamber is zero (at the point where the male-rotor lobe points towards the axis of rotation of the female rotor). As the rotors continue to rotate in the direction indicated by the red arrows, the line of contact between the two rotors will advance towards the rear of the compressor and the volume of the chamber increases. After one revolution, i.e., when the male rotor has rotated 360°,



(a) View from top and front. (b) View from bottom and rear.

Figure 3-1: Twin-screw compressor rotors (Stosic et al., 2005c, p.5).

the volume of the chamber is at its maximum. Continuing to rotate beyond 360° leads to a new line of contact between the male and female rotor occurring at the front, and advancing towards the rear. Thus the volume of the chamber starts to decrease. After a second revolution of the male rotor, i.e., when it has rotated a total of 720°, the volume of the chamber is back at its initial zero. The dark and light shaded regions in Figure 3-1 show where suction, compression and discharge occurs. The dark shaded regions are where the rotors are surrounded by the casing and the vapor is trapped, while the light shaded regions are where the rotors are exposed to external pressure. The light shaded region in Figure 3-1a corresponds to the suction port of the compressor, while the light shaded region between shaft ends B and D in Figure 3-1b corresponds to the discharge port. Vapor from the suction line fills in the expanding chambers at the top and front of the compressor, before it is trapped, compressed, and eventually discharged at a higher pressure through the discharge port at the bottom and rear of the compressor.

3.2 Types of Screw Compressors

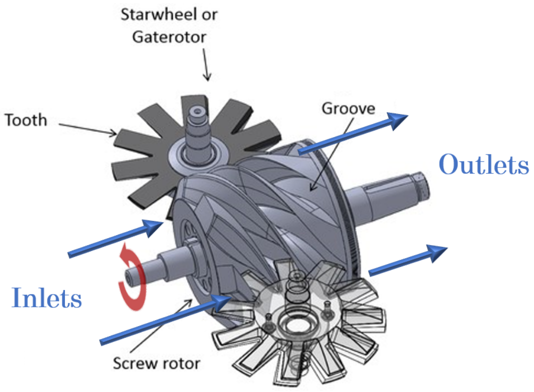
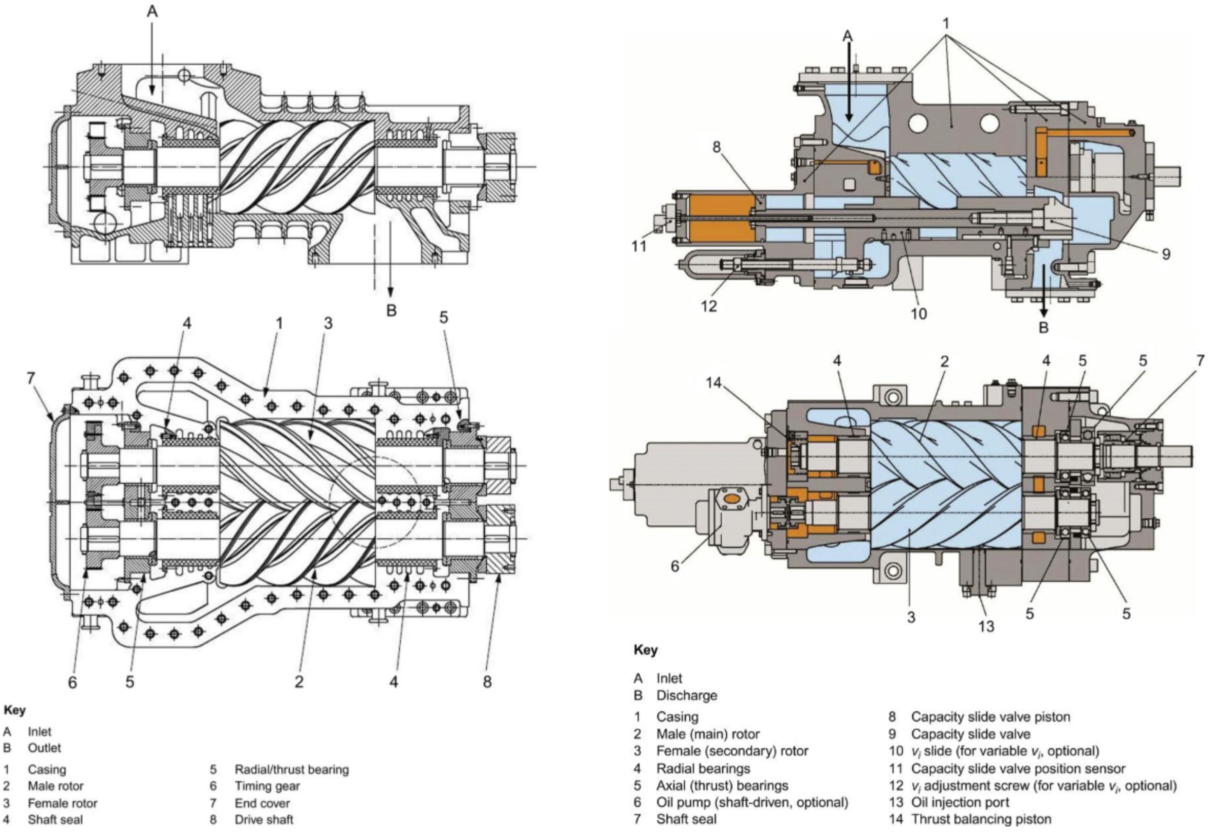


Figure 3-2: Single-screw compressor rotors (Ziviani et al., 2014, p.2).

There are two main types of screw compressors: single screw and twin screw. The working principle of a single-screw compressor is much alike twin-screw compressors, except that the compressor only has one spiral-shaped rotor. To create the volume-changing chambers, the rotor is meshed together with two starwheels on either side, as illustrated in Figure 3-2. The starwheels establish a separation between chambers that are located on the top and the bottom halves of the compressor. In these chambers the vapor is sucked in, trapped, compressed and eventually discharged as it continually rotates. One advantage of the single-screw design compared to the twin screw, is the balance of internal forces. As compression occurs simultaneously at the top and bottom halves in a single-screw compressor, the net radial force on the bearings of the screw becomes negligible. This is not the case for twin-screw compressors, where strong bearings is required to withstand substantial radial forces. On the other hand, single-screw compressors have not been used for as long as twin-screw compressors and the technology is therefore not as mature. Rapid wear down of the starwheels as they are in contact with the screw rotor has been, and still is, a critical issue for single-screw compressors (Wang et al., 2019). Such wear breaks up the sealings between the different chambers in the compressor, and strongly reduces its performance. In the remaining parts of this report, single-screw compressors will not be a topic, and the term screw compressor will be used to refer to twin-screw types of compressors.

Conventional screw compressors can be split into two design categories, depending on whether they operate with or without oil-injection. The design principles of an oil-free and an oil-injected screw compressor are illustrated in Figure 3-3. In an oil-free compressor, contact between the two rotors is avoided by use of timing gears. The timing gears are located at the end of the rotor shafts as shown in Figure 3-3a, and synchronizes the movement of the male and female rotors. There is no oil present in the compression chambers, but the shaft bearings and timing gears are lubricated by oil to reduce friction. Hence shaft seals are needed to prevent oil from entering the oil-free chambers. In oil-injected compressors, as indicated by the name, oil is injected into the compression chambers through an injection port, shown in Figure 3-3b. The oil is eventually discharged together with the compressed vapor, and must be collected in an oil separator downstream of the compressor, before being injected back into the compressor again. Brun and Kurz (2018) state that the oil serves four different purposes:

1. It fills in the internal clearances in the compressor. Smalls gaps between the male and the female rotors, as well as between the rotors and the compressor casing, are sealed by the oil. This reduces the amount of vapor flowing between the different compression chambers, and for this reason, oil-injected compressors have greater volumetric efficiency (defined in equation 3.3) than oil-free compressors.


A
B
1
2
3
4
5
6
7
8
Key
A Inlet
B Outlet
1 Casing
2 Male rotor
3 Female rotor
4 Shaft seal
5 Radial/thrust bearing
6 Timing gear
7 End cover
8 Drive shaft
9
10
11
12
13
14
Key
A Inlet
B Discharge
1 Casing
2 Male (main) rotor
3 Female (secondary) rotor
4 Radial bearings
5 Axial (thrust) bearings
6 Oil pump (shaft-driven, optional)
7 Shaft seal
8 Capacity slide valve piston
9 Capacity slide valve
10 v_i slide (for variable v_i, optional)
11 Capacity slide valve position sensor
12 v_i adjustment screw (for variable v_i, optional)
13 Oil injection port
14 Thrust balancing piston

(a) Oil-free compressor with timing gears.

(b) Oil-injected compressor with capacity slide valve.

Figure 3-3: Oil-free and oil-injected screw compressors (Brun and Kurz, 2018).

2. It creates a thin lubricating film between the male and the female rotor. There are no timing gears in an oil-injected compressor. Instead, one rotor pushes the other rotor around through contact forces between them. Wear and large friction losses are avoided through injection of oil.
3. It removes heat and reduces the discharge temperature. Higher pressure ratios are achievable with oil-injected compressors, because the heat generated by compression is absorbed by the oil. To maximize the cooling effect, the oil is atomized and injected as a spray of fine droplets, so that the contact surface between the oil and vapor being compressed is enhanced.
4. It flushes away contamination. Any unwanted matter that may enter the compressor is removed by the oil.

Contrary to oil-free compressors, there is no need for internal seals in oil-injected compressors. The only seal required is at the drive shaft, to avoid leakage of oil and working fluid to the ambient.

Along with an oil injection port, some screw compressors are equipped with a so-called economizer port. This is an injection port where vapor can be injected at a higher pressure than the suction pressure of the compressor. Utilization of a compressor with an economizer port in a

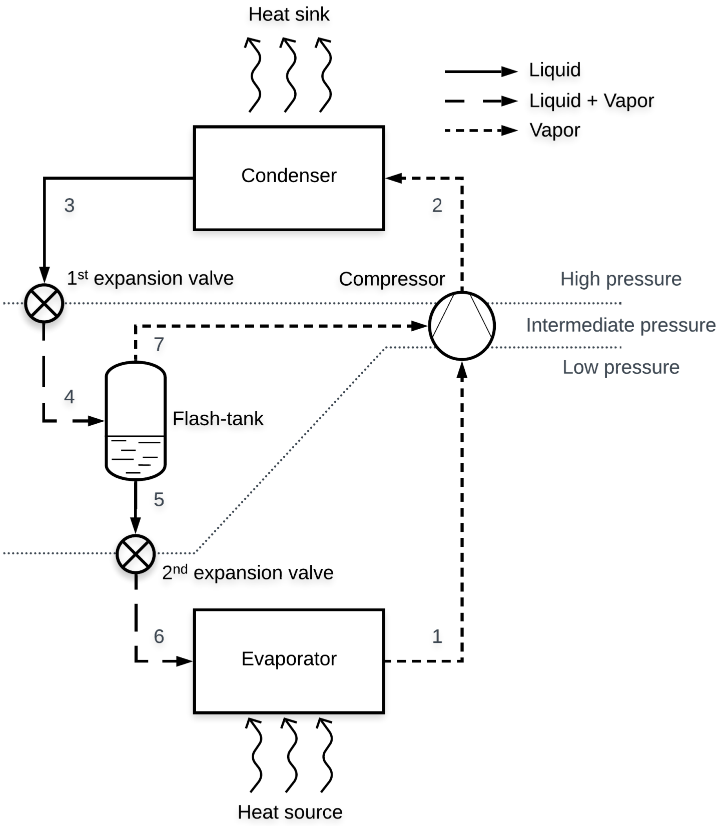


Figure 3-4: Schematic diagram of a vapor compression cycle with economizer flash-tank. Adopted from Lambers (2008, p.1).

vapor compression cycle is illustrated in Figure 3-4. As mentioned in section 2.1, throttling of the pressure from state 3 to state 4 causes partial flash evaporation of the working fluid. Instead of bringing the flashed vapor through the evaporator, it is separated from the liquid in a flash tank and supplied straight to the compressor's economizer port (state 7). This reduces the amount of vapor entering the evaporator, and thereby increases the heat transfer from the heat source and reduces the necessary evaporator size. In addition, the temperature of the vapor injected at intermediate pressure to the compressor (state 7) will be lower than the temperature of the vapor from the suction port that has been compressed to the same pressure level. This gives a cooling effect in the compressor and reduces the temperature of the discharged vapor. Overall, it increases the heat pump's COP and gives potentially greater economic savings. Ergo the name economizer port.

In addition to oil-free and oil-injected screw compressors, a third and less conventional design category can be established: liquid-injected screw compressor. These are compressors with injection of liquid refrigerant rather than oil. In the case of an ammonia-water CACHP, the lean or rich ammonia-water solutions can be injected during compression of the ammonia vapor. The injected liquid refrigerant will perform the same duties as oil, by lubricating the rotors and sealing the gaps. Furthermore, the cooling effect is enhanced as the refrigerant evaporates during the compression process (Stosic et al., 2005a). Liquid-injected screw compressors may or may not have timing gears, depending on the type and the amount of liquid injection.

3.3 Design Parameters

Contrary to reciprocating compressors, there is no pressure-controlled discharge valve in a screw compressor. The ratio between the maximum chamber volume during suction, $V_{suc,theo}$, and the volume of the chamber at beginning of discharge, V_{dis}, is given by the geometry of the compressor casing, especially the size and position of the discharge port. Hence, screw compressors have a built-in volume ratio, v_i, defined as:

$$v_i = \frac{V_{suc,theo}}{V_{dis}} \tag{3.1}$$

The volume ratio is usually a fixed value for a certain compressor, but some screw compressors are equipped with v_i-control and thereby able to adjust the ratio. v_i-control is explained further in section 3.4.

The compressor's pressure ratio, π_i, is given by the built-in volume ratio, v_i, and the isentropic exponent, κ:

$$\pi_i = \frac{p_{dis}}{p_{suc}} = {v_i}^{\kappa} \tag{3.2}$$

where p_{suc} is the suction pressure and p_{dis} is the discharge pressure. The isentropic exponent, κ, is defined as the ratio between the working fluid's specific heat capacity at constant pressure and specific heat capacity at constant volume, c_p/c_V. For a given compressor, π_i is not a fixed value, as the value of κ depends on the working fluid being compressed.

When analyzing a screw compressor it is important to consider internal losses. During operation, the different compression chambers will be at different pressure levels. Pressure differences generate leakage flows, i.e., vapor flowing from high-pressure chambers to chambers with lower pressure. These flows reduce the degree of utilization of the compression chambers, and the amount of vapor flowing into the suction of the compressor is lowered. The effect is quantified by the volumetric efficiency, λ, defined as the ratio between actual suction mass flow, $\dot{m}_{suc}$, and the theoretical suction mass flow that would occur if the entire chamber volume was utilized, $\dot{m}_{suc,theo}$:

$$\lambda = \frac{\dot{m}_{suc}}{\dot{m}_{suc,theo}} = \frac{\dot{m}_{suc}}{\frac{V_{cha} \cdot N_{cha} \cdot f}{v_{suc}}} \tag{3.3}$$

It is shown in equation 3.3 that the theoretical suction mass flow, $\dot{m}_{suc,theo}$, can be calculated from the maximum volume of each compression chamber, V_{cha}, the number of compression chambers, N_{cha}, the operational frequency of the compressor, f, and the specific volume of the vapor at the compressor inlet, v_{suc} (Lambers, 2008).

In addition to leakage flow losses, there will be heat and friction losses in the compressor. The impact of these losses is quantified by the isentropic efficiency, η_{isen}, and can be calculated from the ratio between the theoretical power consumption of compression from suction pressure to discharge pressure, $P_{comp,theo}$, and the actual power consumption of the compressor at the same mass flow, P_{comp}:

$$\eta_{isen} = \frac{P_{comp,theo}}{P_{comp}} = \frac{h_{dis,isen} - h_{suc}}{h_{dis} - h_{suc}} \tag{3.4}$$

Equation 3.4 shows that the ratio between theoretical and actual power consumption can be calculated by looking at the ratio between specific enthalpy change at isentropic compression, $h_{dis,isen} - h_{suc}$, and the actual change in specific enthalpy, $h_{dis} - h_{suc}$.

An important design feature that determines the compressor's capacity and affects the magnitude of internal losses, is the choice of rotor profiles. The possibilities for rotor design are endless, but some examples of common rotor profiles are shown in Figure 3-5. To obtain an efficient screw compressor, Stosic et al. (2005c) give three main criteria for rotor profiles:

1. A large cross-sectional flow area. Larger cross-sectional area results in a higher mass flow rate for the same rotor size and speed.
2. A short sealing line, i.e., a short line of contact between the male and female rotors. Either direct contact or close enough to create a sealing. Shorter sealing lines means smaller leakage flows between the two rotors.
3. A small blow-hole area, i.e a minimal clearance between the rotor tips and the compressor casing. A smaller blow-hole area reduces the amount of leakage flowing along the compressor casing between consecutive compression chambers.

Choosing the optimal rotor profile is not easy, because a larger cross-sectional flow area often results in higher leakage flows. Which type of profile that is considered the best depends on the intended operating conditions for the compressor. At low pressure levels the leakage flows are relatively small. Hence, increasing the cross-sectional flow area in a compressor working at low pressure will presumably give a larger benefit than for a compressor working at high pressure. The rotational speed of the compressor must also be taken into consideration, since lower rotor tip speeds result in more severe blow-hole leakages (Singh and Bowman, 1988). A similar type of consideration must be made when choosing the number of lobes on each rotor, as fewer lobes result in larger flow area, but also higher pressure difference between each compression chamber. Looking at Figure 3-5, the "Compair profile" represents a 4/5 configuration (4 lobes on the male rotor and 5 lobes on the female rotor), while the "FuSheng profile" represents a 5/6 configuration. Most likely, the latter profile has lower leakage flows, but simultaneously a smaller cross-sectional flow area.

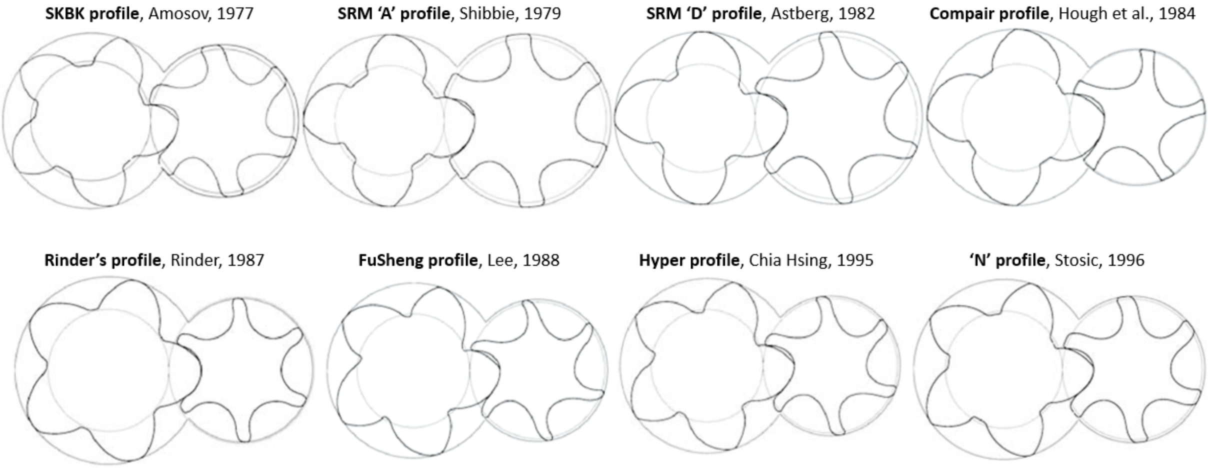

SKBK profile, Amosov, 1977
SRM 'A' profile, Shibbie, 1979
SRM 'D' profile, Astberg, 1982
Compair profile, Hough et al., 1984
Rinder's profile, Rinder, 1987
FuSheng profile, Lee, 1988
Hyper profile, Chia Hsing, 1995
'N' profile, Stosic, 1996

Figure 3-5: Examples of common screw compressor rotor profiles (Stosic et al., 2005c, p.16).

3.4 Operation and Control

The working process in a screw compressor can be split into three phases: suction, compression and discharge. An idealized process in a pressure-volume diagram is illustrated in Figure 3-6. Equation 3.2 shows that the ratio between the outlet and inlet pressures in a screw compressor depends on the built-in volume ratio, v_i. For the compression process in Figure 3-6, the compressor's v_i is well suited because the discharge pressure of the compressor matches the pressure in the discharge line. If there is a mismatch between the compressor's pressure ratio and pressure ratio of the suction and discharge lines, the result is either under-compression or over-compression. These effects are displayed as idealized pressure-volume diagrams in Figure 3-7. The amount of work done by a compressor is equal to the area under the curve in a pressure-volume diagram. Comparing Figure 3-7 with Figure 3-6, it is clearly demonstrated

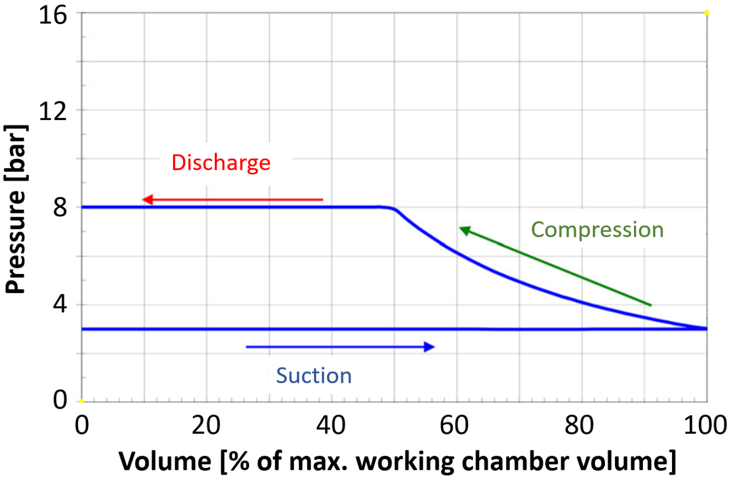
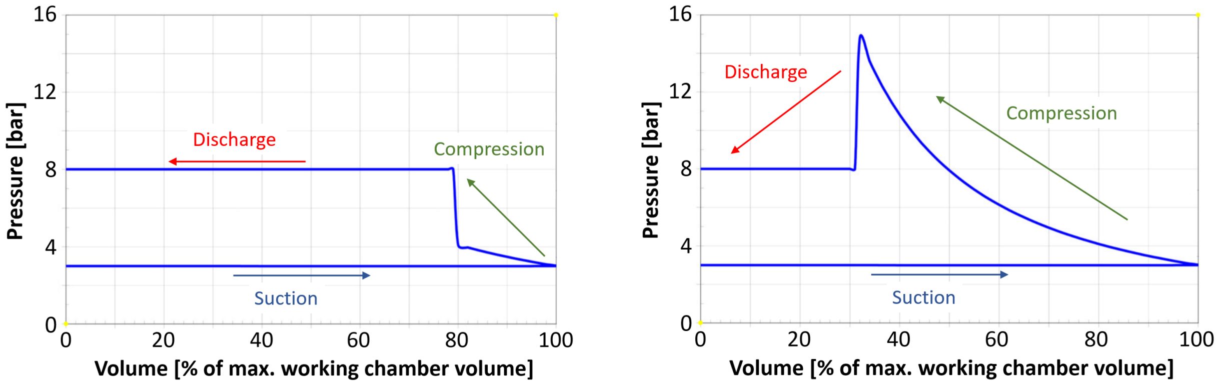


Figure 3-6: Idealized pressure-volume diagram for a screw compressor with well suited built-in volume ratio (Wennemar, 2009, p.151).

that the amount of work done by the compressor is higher if under- or over-compression occurs, since the area under the curve is smaller in Figure 3-6. As explained by Wennemar (2009), under-compression occurs when the compression chamber is opened to the discharge while the pressure is still lower than the discharge pressure (shown in Figure 3-7a). This causes vapor from the discharge line to flow back into the compression chamber until the pressures equalize. The volume of the chamber is reduced further as the rotors continue to rotate, and the vapor is eventually pushed back into the discharge line. This reduces the compressor efficiency and causes an undesirable pulsating flow in the discharge line. Over-compression occurs when the pressure inside the compression chamber surpasses the pressure in the discharge line before the chamber opens up (shown in Figure 3-7b). There is a sudden drop in internal pressure when the compressed vapor flows rapidly into the discharge line. A superfluous amount of work is done by the compressor, as the vapor is compressed and straight afterwards expanded. The rapid



(a) Under-compression.

(b) Over-compression.

Figure 3-7: Idealized pressure-volume diagram for a screw compressor with under- or over-compression (Wennemar, 2009, p.151).

outlet flow can also cause flow pulsations in the discharge line. Over-compression can lead to hazardous overheating of the vapor inside the compression chambers, even when the external pressure ratio between the suction and discharge is small.

To avoid under- or over-compression for a screw compressor operation in varying external conditions, it is possible to equip the compressor with v_i-control, i.e., a way of altering the built-in volume ratio. This can be achieved by installing a v_i-slide valve that controls the geometry of the discharge port (Sjöholm, 1986). If the v_i-slide is adjusted such that the compression chambers are opened to the discharge at an earlier stage, the built-in volume ratio is reduced and so is the pressure ratio. In this manner, the compressor adapts to the external pressure levels, and the compressor efficiency is enhanced.

In the oil-injected screw compressor shown in Figure 3-3b, v_i-control is combined with a second type of control element: a capacity slide valve. The compressor capacity can by reduced by letting vapor at intermediate pressure by-pass back to the suction side of the compressor. Such a by-pass reduces the effective length of the rotors, i.e., the rotors must rotate even further before the compression phase initiates (the dark shaded region in Figure 3-1 is moved forwards and shortened). Hence the utilized compression-chamber volume is reduced, giving lower mass flow and reduced compressor capacity.

A more energy efficient way of adjusting a screw compressor's throughput and capacity, is by using variable speed drive (VSD). This means controlling the amount of vapor being compressed by altering the rotational speed of the compressor. Motors with VSD are more expensive than fixed-speed motors, but if the compressor load is intermittent, the additional investment cost is quickly covered by the elevated compressor efficiency.

3.5 Requirements for High-Temperature Ammonia Compressors

In their search for available compressors to be utilized in an ammonia-water CACHP, Ahrens et al. (2019) list various requirements for the compressor:

- Material compatibility
  - Ammonia can cause corrosion in contact with materials such as copper, copper-based alloys and zinc. It is of high importance to choose ammonia-compatible materials in all parts of the compressor that are in contact with the working fluid. Stainless steel and aluminium are fully compatible with ammonia, and therefore among the most widely used materials in CACHP systems.
- Discharge temperature
  - Due to the relatively low density and specific heat capacity of ammonia, severe temperatures can arise during compression. Jensen (2015) identified the discharge temperature as the dominating constraint in an ammonia-water CACHP. Excessive

temperatures can cause decomposition of the working fluid, breakdown of seals and compressor failure. It is therefore crucial to select a compressor capable of handling or curtailing high discharge temperatures.

- Liquid resistance
  - Because of the need for separation of ammonia vapor from liquid ammonia-water upstream of the compressor in a CACHP, it is possible that the suction vapor will contain droplets of water. Hence, it is important that the compressor is resistant to liquid carry-over. Choosing a liquid-resistant compressor also gives much more operational flexibility, e.g., the possibility of operating closer to the two-phase region and using liquid injection.
- Oil-free operation
  - It is desirable to have a compressor capable of oil-free operation, since oil has been proven to diminish the performance of the rest of the CACHP system (Zaytsev and Ferreira, 2002; Nordtvedt, 2005). Additionally, costs related to oil separation are avoided.

A screw compressor with injection of liquid ammonia-water should be capable of meeting all the above-mentioned requirements, but as no such compressor is commercially available yet, there is lot of uncertainty regarding the design and the operation of it. Analyzing the compressor through use of mathematical models can give better understanding and reduce uncertainty. Screw compressor modeling is therefore the subject in the remaining parts of this report.

4 Screw Compressor Models in the Literature

4.1 Common Screw Compressor Modeling Techniques

When selecting a screw compressor to be implemented in a heat pump system, or developing a new screw compressor design, being able to predict its performance is crucial. Without reliable predictions one may end up with a costly and inefficient system. Constructing and testing prototypes is highly expensive and time consuming, and thus preferably avoided. Making reliable predictions without prototype testing can be achieved through computer modeling and simulation. Numerous modeling methods can be used. Heiyanthuduwage et al. (2011) discuss different modeling techniques used for performance prediction of screw compressor in both industry and academia, namely empirical, analytical and numerical methods.

In empirical screw compressor models, data obtained from testing a particular compressor, or a range of compressors, at different operating conditions is used to predict the performance of a compressor under consideration. An empirical model can be generated by curve fitting the test data or by employing more sophisticated statistical techniques. Correction or correlation factors are derived to be able to predict the performance of a compressor that is designed differently and/or operates under different conditions than the tested compressors. The majority of commercial software developed for screw compressor selection makes use of empirical models someway or another. Empirical models provide fast computations and can be reliable if they are based on substantial amounts of test data. However, it is challenging and expensive to provide sets of test data that are detailed and large enough. Calculations become unreliable as soon as the simulated operating conditions deviate from the tested operating range, making empirical models unsuitable for development of new compressors operating in atypical conditions.

In analytical screw compressor models, pressure-volume diagrams (as the ones shown in Figures 3-6 and 3-7) are used to define the compression process. The area under the curve in the pressure-volume diagram is used to calculate the theoretical power consumption of the compressor, and this power is corrected by considering mechanical losses to obtain the actual power consumption. Pressure-volume diagrams can be represented by simple analytical curves such as polytropic processes, or they can be developed from actual pressure measurements in a chosen compressor, as performed by Huagen et al. (2004) in their experimental study. Analytical models can provide accurate results when pressure-volume diagrams for a range of operating conditions are available for the compressor under consideration. However, the process of developing these diagrams is complicated, and it is impractical to develop numerous diagrams to cover a wide range of operating conditions. Therefore, it is infeasible to use analytical models in compressor selection software, but such models may be of use in research and development projects.

When it comes to numerical screw compressor models, two main types are commonly used: quasi-one-dimensional and three-dimensional. quasi-one-dimensional models are more frequently referred to as chamber models, since the compressor is divided into different chambers or control volumes. These models typically use three chambers (suction chamber, working/compression chamber and discharge chamber), or five chambers (by also considering the suction pipe and the discharge pipe as chambers). The volume of each chamber at any instant is determined as a function of time or rotational angle, and changes in mass or energy in each chamber is computed by use of conservation equations. These equations are discussed further in section 4.3.2. Flows of fluid through various ports, e.g., suction, discharge, oil-/liquid-injection and economizer ports, are represented by a separate set of differential equations. All together, it forms a set of dependent non-linear differential equations. In order to obtain the state of the variables of interest at each point in time, the system of equations must be solved numerically with well-qualified boundary and initial conditions. Three-dimensional numerical models are based on computational fluid dynamics (CFD). Instead of a small number of chambers, the control volume is divided into countless finite sized elements. Conservation of mass, energy and momentum is applied to each element, and the system of equations is solved numerically. If executed correctly, CFD can provide accurate and extremely detailed results. However, the computer resources and the time needed to solve these models are immense. Numerical models, both quasi-one-dimensional and three-dimensional, are widely used in academia to study the performance of screw compressors.

4.2 Criteria for Evaluation of Modeling Approaches

Numerous different approaches can be taken to model and simulate a screw compressor. Since the possibilities are so many, an assessment of previously used methods in literature should be carried out. To be able to determine the optimal approach for the objectives of this work, a set of criteria for the modeling approach is formulated. The criteria are as follows:

1. Physical validity
   - The model should agree with the laws of physics to the greatest practical extent. Simplifications are often necessary to establish a mathematical model of a physical phenomenon. One should make sure that these simplifications do not conflict with the physical validity of the model. Comparison with experimental test results is an effective means for model validation.
2. Accuracy
   - Accuracy of the simulation results is of high priority. The practicality of a model is low if the model has bad accuracy. To be able to gain insight into the complex compression process, and perform detailed performance predictions, high accuracy is required.

3. Computational time
    - The duration of a simulation affects the model's usability. Fewer simulations can be carried out if each simulation requires long computational time and/or large computing power. With faster simulations, a greater number of configurations and operating conditions can be investigated, giving more data and better understanding. This criterion comes in conflict the accuracy criterion, as high accuracy often requires heavy calculations.
4. Complexity
    - The compression process itself is highly complex, and a relatively complex mathematical model is therefore required. However, high complexity can cause confusion and a lack of understanding what gives rise to a certain simulation result. Lower complexity makes the model easier to implement and errors easier to detect. Additionally, high complexity is often closely related to long computational time.
5. Input data
    - It is desirable with as little required input data as possible. There are often many uncertainties in a research and development project, and it is therefore unfavorable to have to determine multiple physical parameters beforehand. Values to be set in advance, such as flow coefficients and correction factors, are very difficult to estimate without experimental test results. Moreover, use of performance data provided by the manufacturer limits the model to a reduced margin of parameters, since the given data cannot be used to predict performance outside the tested operating range.
6. Output data
    - Detailed output data is desired, because it gives better understanding and makes it easier to establish the reason behind each performance prediction. That is, presuming that the accuracy of the data is sufficient. If not, details can mislead and cause confusion. Furthermore, detailed output makes it easier to detect errors and inconsistencies.
7. Stability
    - In numerical modeling, iterative solutions are often required. Simulations may diverge if iteration procedures are not implemented with care, and incorrect solutions may be found. Robust algorithms must be chosen to ensure model reliability.
8. Reusability
    - The practicality of a model is high if the model can be reused and applied in other models without having to make big structural changes to it. It is desirable that the model is easy to use, and that it offers flexibility. If adaptions can be made quick and easy, a great number of configurations can be investigated.

4.3 Description and Evaluation of Modeling Approaches

Through research in literature on screw compressor modeling, and by considering the approach criteria defined in section 4.2, two previously studied approaches were found to be especially well suited for the objectives of this work. Chamoun et al. (2013) describes a relatively simple numerical approach for modeling a screw compressor with water as the working fluid, where an analogy to the chambers in a piston compressor is used. The model includes injection of liquid water in the suction chamber, and accounts for internal leakage flows and heat losses to the ambient. Following, Tian et al. (2017a) describes a model with a similar chamber approach, but of higher complexity related to liquid injection and two-phase flows. The model concerns a screw compressor with water as the working fluid, but contrary to Chamoun et al. (2013), injection of liquid water occurs during the first part of the compression phase. In the following sections, the approaches used by Chamoun et al. (2013) and Tian et al. (2017a) will be discussed, together with alternate solutions used by other researchers.

4.3.1 Assumptions

In order to reduce model complexity and simplify calculations, a number of assumptions are made. The following assumptions are stated by Chamoun et al. (2013):

1. During the suction phase, the inlet velocity is constant and there is no pressure drop.
2. During the compression and discharge phases, pressure and enthalpy are homogeneous throughout the working space at any instant.
3. Pressure pulsations during suction and discharge are neglected.
4. A fixed mechanical efficiency of 90 % is used to calculate mechanical losses.

The following assumptions are stated by Tian et al. (2017a):

1. Changes in kinetic and gravitational energy of the working fluid are negligible.
2. In the control volume, the pressure and the temperature of the vapor are homogeneous at any instant, and there is no temperature or pressure difference between the liquid and the vapor.
3. Heat transfer between the working fluid and the compressor casing is neglected. Cooling effect is mainly a result of heat absorption by injected liquid.
4. Pressure pulsations in the suction phase is neglected.
5. Water is treated as an incompressible fluid.
6. Flows of liquid and vapor in the leakage paths are adiabatic.
7. The amount of flash gas generated by liquid leakage flows is calculated by assuming isenthalpic flow.

8. A fixed mechanical efficiency of 94 % is used.

Pressure pulsations are mentioned in both sets of assumptions. In a screw compressor, connection between the working chambers and the suction or discharge chambers occurs intermittently. This creates unsteady flows and thereby pressure pulsations (Koai and Soedel, 1990). Pressure pulsations can cause vibrations, noise and damage, but at moderate levels it does not affect the overall energy performance of a screw compressor. Thus pressure pulsations can be neglected here. If vibrations and noise are to be studied, accurate predictions can be obtained by using a CFD model as presented by Mujic et al. (2009).

A large number of assumptions can decrease the credibility of a model. Often, a trade-off between accuracy and computational time must be made. Fewer assumptions underpin physical validity, but it also increases the computational time and resources needed to solve the model. A good example here is the assumption that the temperature and pressure is homogeneous in each working chamber at any instant, made by both Chamoun et al. (2013) and Tian et al. (2017a). If this assumption is to be avoided, the working chamber must be divided into many small parts with separate state variables, just like a CFD model. The number of calculations to be made is increased drastically, causing a corresponding increase in required computing power.

4.3.2 The Thermodynamic Basis

Numerical screw compressor models are based on equations for conservation of mass and energy. Since there are leakage flows, and the working chambers are periodically connected to suction and discharge, the control volume is treated as an open system. The first law of thermodynamics for an open system can be formulated as (Bejan, 2016):

$$\frac{dE}{dt} = \dot{Q} - \dot{W} + \sum \dot{m}_{in}(h_{in} + \frac{1}{2}v_{in}^2 + gz_{in}) - \sum \dot{m}_{out}(h_{out} + \frac{1}{2}v_{out}^2 + gz_{out}) \tag{4.1}$$

where E represents the system's total energy. Its change per unit of time depends on the rate of heat input to the system, $\dot{Q}$, the rate of mechanical work done by the system on its surroundings, $\dot{W}$, together with the enthalpy, h, the velocity, v, and the relative height, z, of the flows going in or out of the system. The change of total mass in the system is calculated from the mass conservation equation:

$$\frac{dm}{dt} = \sum \dot{m}_{in} - \sum \dot{m}_{out} \tag{4.2}$$

where $\dot{m}_{in}$ and $\dot{m}_{out}$ are the inlet and outlet mass flow rates, respectively.

Looking back at the first assumption made by Tian et al. (2017a), changes in kinetic and gravitational energy within the working chambers are considered negligible. This assumption is frequently used in literature (Stosic et al., 2005c; Heiyanthuduwage et al., 2011). Clearly, gravitational energy is insignificant, as the height difference between the top and the bottom of a screw compressor is relatively small. Kinetic energy, on the other hand, can be of significance as flow velocities become high at large pressure differences. Nevertheless, kinetic energy is often neglected due to the fact that the internal energy is significantly larger than the kinetic energy in positive displacement compressors (Kovacevic et al., 2016). An example of a scenario where kinetic energy cannot be neglected, is during discharge when substantial under- or over-compression occurs. When the compression chamber opens to the discharge, large pressure difference will cause a rapid flow either going in or out of the chamber, constituting a considerable amount of kinetic energy.

The duration of a compression cycle, i.e., the time it takes from the beginning of the suction phase to the end of the discharge phase, depends on the rotational speed of the compressor. Therefore it is often more convenient to express the equations in terms of rotational angle of the male rotor rather than time. Neglecting kinetic and gravitational energy, and expressing the equations based on rotational angle, gives the conservation equations used by both Chamoun et al. (2013) and Tian et al. (2017a). Equation 4.1 becomes:

$$\frac{d(mu)}{d\theta} = \frac{dQ}{d\theta} - \frac{dW}{d\theta} + \sum \frac{dm_{in}}{d\theta} h_{in} - \sum \frac{dm_{out}}{d\theta} h_{out} \tag{4.3}$$

where u is the specific internal energy and θ is the rotational angle of the male rotor. Likewise, equation 4.2 becomes:

$$\frac{dm}{d\theta} = \sum \frac{dm_{in}}{d\theta} - \sum \frac{dm_{out}}{d\theta} \tag{4.4}$$

Chamoun et al. (2013) modifies the conservation equations one step further by using the angular velocity, $\omega = d\theta/dt$, and the fact that an infinitesimal work transfer equals the pressure times the infinitesimal volume change, $\delta W = pdV$ (Bejan, 2016). As a result, equation 4.3 becomes:

$$\omega \frac{d(mu)}{d\theta} = \dot{Q} - \omega p \frac{dV}{d\theta} + \sum \dot{m}_{in} h_{in} - \sum \dot{m}_{out} h_{out} \tag{4.5}$$

Correspondingly, equation 4.4 becomes:

$$\omega \frac{dm}{d\theta} = \sum \dot{m}_{in} - \sum \dot{m}_{out} \tag{4.6}$$

If a CFD model is to be developed, a third equation of conservation must be used, namely conservation of momentum. Rane et al. (2018) gives a detailed explanation of such a model, used for comprehensive analysis of water evaporation in screw compressors.

4.3.3 The Control Volume

Chamoun et al. (2013) and Tian et al. (2017a) defines the control volume as the volume between a pair of male and female rotor lobes. When illustrating the control volume, an analogy to the chamber volume in a piston compressor is used. This is shown in Figure 4-1. The control volume is an open system with three inlet paths (suction, leak-in and liquid injection) and two outlet paths (discharge and leak-out) connected to it.

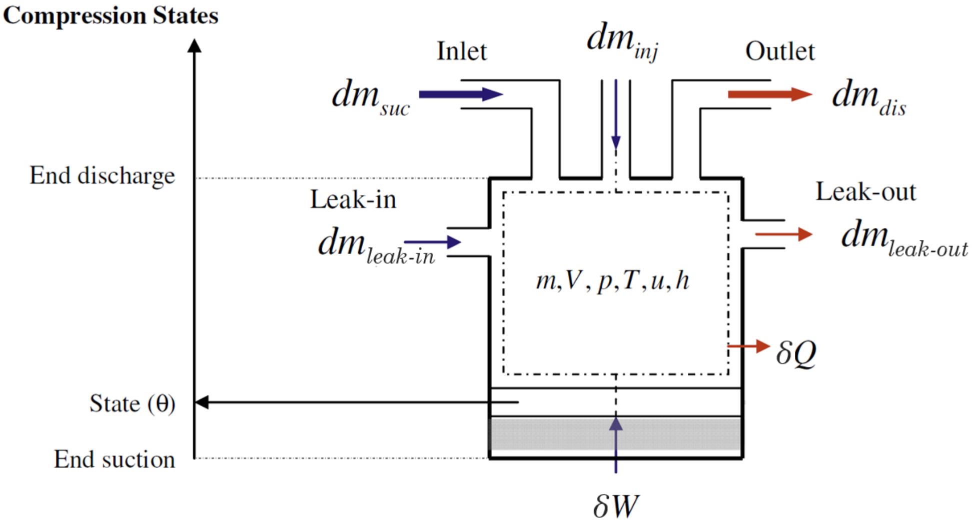


Figure 4-1: Control volume for the compression cycle. Here, m_{suc} is the mass inlet during suction, m_{dis} is the mass outlet during discharge, m_{inj} is the mass from liquid injection, while $m_{leak\text{-}in}$ and $m_{leak\text{-}out}$ is the leakage flows going in and out of the control volume, respectively. W is the work performed by the compressor on the control volume, and Q is the heat loss to the compressor casing and ambient. The compression state is given by θ, the rotational angle of the male rotor. Adopted from Chamoun et al. (2013, p.483).

4.3.4 Leakage Flows

Internal leakage flows have a substantial impact on the overall performance of a screw compressor (Fleming and Tang, 1995). Hence it is of high importance to model and calculate these flows correctly. If leakage flows are excluded or underestimated, the predicted compressor efficiency will be much greater than in reality. Five leakage paths have been identified: (1) through the contact line between the male and the female rotors, (2) between the rotor tips and the compressor casing, (3) through blow holes formed at the tips of the male and female rotors, (4) at the end of the rotor on the suction side, and (5) at the end of the rotor on the discharge side (Fleming and Tang, 1995; Huagen et al., 2004). These are all shown in Figure 4-2. For a given control volume, in order to calculate the mass flow rate through each leakage path, the state of the adjacent control volumes (i.e., the volumes between the neighbouring pairs of rotor lobes) must be known. It that case, according to Chamoun et al. (2013), the inlet and outlet leakage flow rates can be calculated from equations 4.7 and 4.8, respectively.

$$\dot{m}_{leak\text{-}in} = C_{leak} A_{leak} \sqrt{2\rho_{\theta+\frac{2\pi}{N_{lob}}} \left(p_{\theta+\frac{2\pi}{N_{lob}}} - p_{\theta} \right)} \tag{4.7}$$

$$\dot{m}_{leak\text{-}out} = C_{leak} A_{leak} \sqrt{2\rho_{\theta} \left(p_{\theta} - p_{\theta-\frac{2\pi}{N_{lob}}} \right)} \tag{4.8}$$

Here, θ is the rotational angle of the control volume under consideration, while $\theta + \frac{2\pi}{N_{lob}}$ and $\theta - \frac{2\pi}{N_{lob}}$ are the angles of the adjacent control volumes at the same point in time. N_{lob} is the number of lobes on the male rotor, which determines the angular displacement between the control volumes. C_{leak} is a flow coefficient found empirically, and A_{leak} is the cross-sectional area of the leakage path. Establishing the exact cross-sectional area of each leakage path in a screw compressor is impractical. Instead, Fleming and Tang (1995) suggest that average values should be used. The average values can be found either through geometrical calculations or through experimental investigations on the actual compressor (Heiyanthuduwage et al., 2011). As indicated by equation 4.7, the mass that leaks in to the control volume comes from the adjacent control volume that has rotated an additional $\frac{2\pi}{N_{lob}}$ radians. This chamber has come farther in the compression cycle, and thus the pressure, $p_{\theta+\frac{2\pi}{N_{lob}}}$, is higher than the control volume pressure, p_θ. The mass that leaks in to the control volume has density equal to the mass in the adjacent control volume, $\rho_{\theta+\frac{2\pi}{N_{lob}}}$, while any leakage that flows out of the control volume has density equal to the control volume itself, ρ_θ.

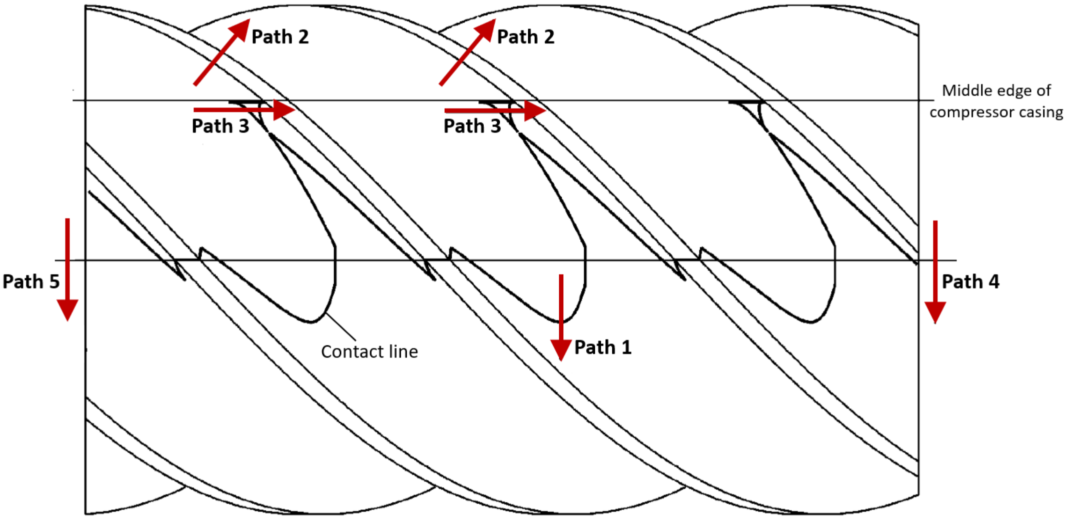


Figure 4-2: Location of leakage paths illustrated on the female rotor. There are five paths: (1) through the contact line between the male and female rotors, (2) between the rotor tip and compressor casing, (3) through blow holes, (4) at the suction end, and (5) at the discharge end. Adopted from Huagen et al. (2004, p.334).

Tian et al. (2017a) goes into more detail by considering the leakage flows as two-phase laminar flows, due to liquid injection. The liquid and the vapor are assumed to be homogeneous, and the leakage flow rates are calculated according to equations 4.9–4.12.

$$\dot{m}_{leak,tot} = \frac{C_{leak} A_{leak} \sqrt{2\rho_{liq}\Delta p}}{(1-q)\varphi + q\sqrt{\rho_{liq}/\rho_{vap}}} \tag{4.9}$$

$$\dot{m}_{leak,vap} = q\dot{m}_{leak,tot} \tag{4.10}$$

$$\dot{m}_{leak,liq} = (1-q)\dot{m}_{leak,tot} \tag{4.11}$$

where $\dot{m}_{leak,tot}$, $\dot{m}_{leak,vap}$ and $\dot{m}_{leak,liq}$ is the total leakage, the vapor leakage and the liquid leakage mass flow rates, respectively. Δp is the pressure difference between the evaluated and the adjacent control volumes, ρ_{liq} is the density of the liquid, ρ_{vap} is the density of the vapor, and q is the vapor quality. Lastly, φ is the coefficient in the Lin equation (Huagen et al., 2004), calculated as:

$$\begin{aligned}\varphi = 1.48625 - 9.26541\left(\frac{\rho_{liq}}{\rho_{vap}}\right) + 44.6954\left(\frac{\rho_{liq}}{\rho_{vap}}\right)^2 - 60.615\left(\frac{\rho_{liq}}{\rho_{vap}}\right)^3 \\ - 5.12966\left(\frac{\rho_{liq}}{\rho_{vap}}\right)^4 - 26.5743\left(\frac{\rho_{liq}}{\rho_{vap}}\right)^5\end{aligned} \tag{4.12}$$

When liquid flows from a high-pressure chamber to low-pressure chamber, the pressure drop will cause flash evaporation. Tian et al. (2017a) assume that the liquid leakage flow is isenthalpic (constant enthalpy), and the amount vapor caused by flash evaporation is calculated with equations 4.13 and 4.14.

$$h_{f,high} = q_{fla}h_{g,low} + (1 - q_{fla})h_{f,low} \tag{4.13}$$

$$\dot{m}_{leak,fla} = q_{fla}\dot{m}_{leak,liq} \tag{4.14}$$

where $h_{f,high}$ is the specific enthalpy of saturated liquid in the high-pressure chamber, $h_{g,low}$ is the specific enthalpy of saturated vapor in the low-pressure chamber, and $h_{f,low}$ is the specific enthalpy of saturated liquid in the low pressure chamber. q_{fla} is the fraction of the liquid leakage flow that turns into vapor by flashing.

Since Tian et al. (2017a) is dealing with compression of water vapor and injection of liquid water, the two-phase flow analysis described above is quite straightforward and manageable. However, in the CACHP studied in this report, the vapor being compressed and the liquid being injected are not equal mediums. Ammonia vapor is compressed, while a liquid mixture of ammonia and water is injected. This makes the calculations more complex, considering that changes in ammonia concentration cause changes in the fluid properties. Further details regarding the properties of ammonia-water mixtures is given in section 4.3.12.

Different procedures for calculating the leakage mass flow rate is used in literature. Stosic et al. (2005c) and Tian et al. (2017b) use equation 4.15.

$$\dot{m}_{leak} = \sqrt{\frac{p_{high}^2 - p_{low}^2}{a^2\left(\zeta + 2\ln\frac{p_{high}}{p_{low}}\right)}} \tag{4.15}$$

where p_{high} and p_{low} are the pressures in the high-pressure and low-pressure chamber, respectively. a is the speed of sound, while ζ is the leakage flow resistance. This resistance can be evaluated for each leakage path in the compressor as a function of its shape, dimensions and flow characteristics. In a complete analysis, evaluation of ζ requires that the model includes friction and drag coefficients in terms of Reynolds and Mach numbers. For simplicity, Stosic et al. (2005c) suggests that ζ is given as an input parameter.

4.3.5 Heat Losses and Heat Transfer

In a compressor used for industrial heat pump applications, the temperature of the working fluid is substantially higher than the temperature of the ambient air surrounding the compressor. Therefore, heat losses to the compressor casing and the ambient should be assessed. The temperature of the vapor in the working chamber rises during compression (see Figure 4-6b). Towards the end of the compression cycle, there is a very large temperature difference between the vapor and the ambient air, which indicates that large heat losses can occur. On the other hand, the volume of the compression chamber becomes smaller and smaller towards the end of compression, and the area available for heat transfer to the ambient diminishes. These contradictions cause uncertainty about whether heat loss to the ambient has a significant impact or not. Since the heat transfer coefficient between the vapor and the compressor body is small, it is often argued in the literature that heat loss to the casing and ambient can be neglected (Stosic et al., 2005b).

Seeing that the working fluid temperature is high, Chamoun et al. (2013) choose to take heat loss to the surroundings into consideration. In the same manner as Huagen et al. (2004), equation 4.16 is used to calculate heat transfer from the working fluid to the compressor body.

$$\dot{Q}_{bod} = \alpha V_{cv}^{2/3} \left(T_{vap} - T_{bod}\right) \tag{4.16}$$

where α is the heat transfer coefficient between the working fluid and compressor body, and the area available for heat transfer is calculated from the chamber volume by $V_{cv}^{2/3}$. If needed, a more exact term for the heat transfer area can be developed through evaluation of the geometrical properties of a screw compressor. T_{vap} and T_{bod} represent the temperature of the vapor in the control volume and the temperature of the compressor body, respectively. Determining the temperature of the compressor body is not an easy task. The thermal conductivity of the body is relatively high, since it is made of metal, and a uniform temperature distribution can be assumed. However, the temperature of the working fluid varies greatly throughout the compressor, and many different factors can influence the body temperature. In some cases, a linear correlation between the discharge temperature and the body temperature is used, while other solutions involve non-linear correlations with the pressure ratio, the mechanical efficiency and multiple other operating conditions (Goossens et al., 2017).

Since the cooling effect from the injected liquid is much more significant than the cooling effect from heat losses to the ambient, Tian et al. (2017a) choose to neglect heat transfer between the working fluid and the compressor body. Instead, cooling is calculated entirely as a result of heat absorption by the injected liquid. If the liquid is able to cool the compressed vapor down to saturation, the amount of heat transferred from the vapor to the liquid is:

$$Q_{liq} = m_{vap} \left(h_{vap} - h_g\right) \tag{4.17}$$

where m_{vap} is the mass of vapor in the control volume, h_{vap} is the specific enthalpy of the vapor, and h_g is the specific enthalpy of saturated vapor at the same pressure. The heat transfer will cause an increase in the temperature of the liquid, and when the liquid reaches its saturation temperature, it will begin to evaporate. The mass of evaporated liquid, m_{evap}, is calculated using equation 4.18.

$$Q_{liq} = m_{liq}c_{p,liq}\left(T_f - T_{liq,prev}\right) + m_{evap}\gamma \tag{4.18}$$

where m_{liq} is the mass of liquid in the control volume, and $c_{p,liq}$ is the specific heat capacity of the liquid at constant pressure. T_f is the saturation temperature of the liquid, while $T_{liq,prev}$ is the temperature of the liquid in the previous step in the compression cycle (discretization of the compression cycle is discussed in section 4.3.9). γ is the latent heat of vaporization.

4.3.6 Power and Efficiencies

In their assumptions, Chamoun et al. (2013) and Tian et al. (2017a) prescribe a fixed mechanical efficiency of 90 % and 94 %, respectively. The theoretical power of the compressor, usually referred to as the indicated power, is calculated through integration:

$$P_{ind} = -N_{lob}f \oint pdV \tag{4.19}$$

By integrating the pressure, p, in the control volume with respect to the volume, V, over the entire compression cycle, the amount of work done by the control volume on its surroundings is obtained. The indicated power of the compressor can then be obtained by taking into account the number of lobes on the male rotor, N_{lob}, and the operational frequency of the compressor, f. Afterwards, the input shaft power to the compressor is calculated as:

$$P_{sha} = \frac{P_{ind}}{\eta_{mec}} \tag{4.20}$$

where η_{mec} is the mechanical efficiency of the compressor.

The isentropic efficiency (defined in equation 3.4) is an effective measure used to evaluate screw compressor performance. Chamoun et al. (2013) calculate the isentropic efficiency, η_{isen} using equations 4.21–4.23.

$$\dot{m} = N_{lob}fm_{suc} \tag{4.21}$$

$$P_{isen} = \dot{m}\left(h_{dis,isen} - h_{suc}\right) \tag{4.22}$$

$$\eta_{isen} = \frac{P_{isen}}{P_{ind}} \tag{4.23}$$

where m_{suc} is the mass sucked in to the control volume at the beginning of the compression cycle, and $\dot{m}$ is the total mass flow rate delivered by all the chambers in the compressor. P_{isen} is the isentropic power of the compressor. h_{suc} and $h_{dis,isen}$ are the specific enthalpies in the suction chamber and the discharge chamber after isentropic compression, respectively.

The volumetric efficiency (defined in equation 3.3) is another important performance measure. Tian et al. (2017a) calculate the volumetric efficiency, λ, using:

$$\lambda = \frac{m_{dis} v_{suc}}{V_{theo}} \tag{4.24}$$

where m_{dis} is the mass in the compression chamber when it opens to the discharge, v_{suc} is the specific volume of the suction vapor, and V_{theo} is the theoretical suction volume.

Many different approaches for calculation of efficiencies are used in literature, especially regarding mechanical efficiency. Contrary to using a fixed mechanical efficiency as shown above, Gudjonsdottir et al. (2019) calculate mechanical losses as a function the compressor's operational frequency, f. It is assumed that there are three main contributors to the mechanical losses: the bearings, the seals and the gears. For the screw compressor studied by Gudjonsdottir et al. (2019), the bearing losses, $\dot{W}_{bear}$, and the sealing losses, $\dot{W}_{seal}$, are estimated with equations 4.25 and 4.26.

$$\dot{W}_{bear} = 3.15 \cdot 10^{-4} \mu B d_{bear} f \tag{4.25}$$

$$\dot{W}_{seal} = 1.75 \cdot 10^{-4} d_{seal} f^{4/3} \tag{4.26}$$

where μ is the friction coefficient and B is the dynamic bearing load, both given by the compressor manufacturer. d_{bear} and d_{seal} are the diameters of the bearings and the seals, respectively. f is the operational frequency of the compressor. In addition, mechanical losses caused be gear meshing and synchronization is set to a constant value of 4 %, i.e., η_{gear} = 96 %. To be able to make these types of loss predictions, detailed manufacturer data is required. The actual value of the losses are highly dependent on the design of the bearings, the seals and the gears, as well as the operating conditions of the compressor. During research and development of new screw compressor, such manufacturer data is usually not available. Thus a fixed mechanical efficiency is used in most cases. Furthermore, it is difficult to predict to what degree the mechanical losses will affect the working fluid itself. Gudjonsdottir et al. (2019) assume that only the sealing losses have an effect. The calculated sealing losses are added as heat input to the the working fluid at the beginning of the suction phase.

4.3.7 Liquid Injection

Chamoun et al. (2013) demonstrate a simple approach to liquid injection. A constant mass flow rate of water is injected into the control volume during the suction phase. There is no pressure change, and the homogeneous specific enthalpy of the working fluid at the end of the suction phase is calculated as a weighted average between of the specific enthalpy of the suction vapor and the injected liquid (see equation 4.31). Tian et al. (2017a) demonstrate a different approach, where liquid is injected during the beginning of the compression phase. Whit a given volume flow rate in the injection line, the mass of injected liquid is calculated from:

$$dm_{inj} = \frac{\rho_{liq}\dot{V}_{inj}}{\omega}d\theta \tag{4.27}$$

where ρ_{liq} is the density of the liquid, $\dot{V}_{inj}$ is the volume flow rate in the injection line, ω is the angular velocity of the male rotor, and θ is the rotational angle of the male rotor. The mass of liquid in each chamber is tracked throughout the entire compression cycle by evaluating the liquid leakage flows (equation 4.11), the flash evaporation (equation 4.14) and the heat transfer evaporation (equation 4.18).

Liquid injection, two-phase flows and phase change occurring inside the compressor makes it very challenging to accurately predict the real compression process. To gain better understanding of the two-phase process, a more detailed and complex approach must be taken, either through CFD analysis or through further development of a quasi-one-dimensional numerical approach. Tian et al. (2017b) describe a numerical model used for analysis of mass and heat transfer in an ammonia screw compressor. The compressor is used in an ammonia refinement system, and a liquid ammonia-water mixture is injected into the suction chamber. Their study involve liquid droplet dynamics. The movement of each droplet throughout the compression cycle is estimated, assuming that the droplets are static and uniformly distributed in the compression chamber at the beginning of the compression phase. Mass diffusion is taken into account, and the impact of the mean diameter of the injected droplets is evaluated. In addition, the effect of adding a second injection nozzle after the beginning of the compression phase is investigated. Developing a model that is capable of making these types of investigations is desirable. However, the model complexity is relatively high, the number of assumptions made are many, and the computational requirements are substantial.

4.3.8 Suction Phase

Now, looking at the model developed by Chamoun et al. (2013), where liquid injection occurs during suction. At the end of the suction phase, the volume of the compression chamber is at its maximum level, here referred to as the compressor's theoretical suction volume, $V_{suc,theo}$. The total mass in the control volume at the end of the suction phase is given by the theoretical suction volume and the density of the working fluid, ρ, as:

$$m_{cv} = \rho V_{suc,theo} \tag{4.28}$$

The volume of the vapor that is actually sucked in to the compression chamber will be smaller than the theoretical suction volume due to the injected liquid and the leakage flows. Hence, the total mass in the control volume at the end of the suction phase is equal to:

$$m_{cv} = m_{suc} + m_{inj} + m_{leak\text{-}in} \tag{4.29}$$

where m_{suc} is the mass of suction vapor, m_{inj} is the mass of injected liquid, and $m_{leak\text{-}in}$ is the mass of leakage flow from the adjacent chamber. $m_{leak\text{-}in}$ is calculated from:

$$m_{leak\text{-}in} = \dot{m}_{leak\text{-}in} t_{suc} \tag{4.30}$$

where the mass flow rate $\dot{m}_{leak\text{-}in}$ is taken from equation 4.7, and t_{suc} is the time span of the suction phase. When the mass distribution is known, the specific enthalpy in the control volume at the end of the suction phase can be calculated from:

$$m_{cv} h_{cv} = m_{suc} h_{suc} + m_{inj} h_{inj} + m_{leak\text{-}in} h_{leak\text{-}in} - Q_{bod} \tag{4.31}$$

where h_{suc}, h_{inj} and $h_{leak\text{-}in}$ is the specific enthalpy of the suction vapor, the injected liquid and the leakage flow, respectively. Q_{bod} is the heat transfer from the working fluid to the compressor body, calculated as $Q_{bod} = \dot{Q}_{bod} t_{suc}$, where $\dot{Q}_{bod}$ is taken from equation 4.16. Keep in mind here that the temperature of the compressor body most likely is higher than the the temperature of the working fluid during suction, and thus $\dot{Q}_{bod} < 0$.

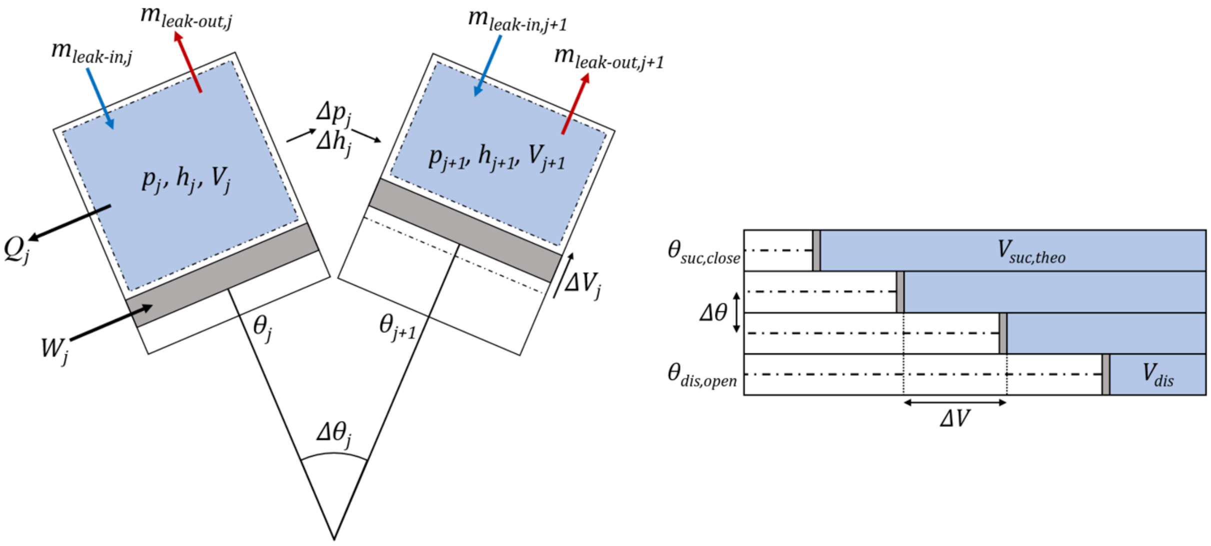


Figure 4-3: Discretization of the compression process. Increasing the rotational angle of the male rotor by $\Delta\theta$ gives a corresponding chamber volume reduction of ΔV. Based on Chamoun et al. (2013, p.484) and Tian et al. (2017a, p.78).

4.3.9 Compression Phase

To be able to numerically calculate values such as pressure, temperature, mass flow and heat transfer throughout the compression process, the process itself must be discretized, i.e., the process must be divided into a series of discrete steps. The chosen step size can be based on different variables. Tian et al. (2017b) use a constant pressure step, Δp. Chamoun et al. (2013) and Tian et al. (2017a) use a more conventional approach with a constant step in the male rotor's rotational angle, $\Delta\theta$. Through use of the piston compressor analogy, the discrete changes in chamber volume during the compression process can illustrated as shown in Figure 4-3. When the control volume reaches the beginning of the compression phase, at a rotational angle denoted by $\theta_{suc,close}$, the connection to the suction line is cut off. Since there is no connection to either the suction port or the discharge port, the only flows of mass in and out of the control volume during the compression phase are leakages to adjacent chambers. If liquid injection occurs during compression, a second mass inlet, $m_{inj,j}$, has to be added in Figure 4-3.

The equations for conservation of energy and mass presented in section 4.3.2 are used together with the illustrated discretization technique to calculate the evolution of the thermodynamic parameters throughout the compression process. By using the definition of specific enthalpy (Bejan, 2016):

$$h = u + pv \tag{4.32}$$

and rearranging equation 4.5, Chamoun et al. (2013) implement conservation of energy in the following manner:

$$\beta_1 \frac{\Delta p}{\Delta \theta} + \beta_2 \frac{\Delta h}{\Delta \theta} + \beta_3 \frac{\Delta V}{\Delta \theta} = h_{inj}\dot{m}_{inj} + h_{leak\text{-}in}\dot{m}_{leak\text{-}in} - h_{leak\text{-}out}\dot{m}_{leak\text{-}out} - \dot{Q}_{bod} \tag{4.33}$$

The coefficients β_1, β_2 and β_3 are defined as:

$$\beta_1 = \omega V \left[\left(\frac{\partial \rho}{\partial p} \right)_h - 1 \right] \tag{4.34}$$

$$\beta_2 = \omega V \left[\rho + h \left(\frac{\partial \rho}{\partial h} \right)_p \right] \tag{4.35}$$

$$\beta_3 = \omega \rho h \tag{4.36}$$

Likewise, conservation of mass is implemented using:

$$\beta_4 \frac{\Delta p}{\Delta \theta} + \beta_5 \frac{\Delta h}{\Delta \theta} + \beta_6 \frac{\Delta V}{\Delta \theta} = \dot{m}_{inj} + \dot{m}_{leak\text{-}in} - \dot{m}_{leak\text{-}out} \tag{4.37}$$

where the coefficients β_4, β_5 and β_6 are defined as:

$$\beta_4 = \omega V \left(\frac{\partial \rho}{\partial p} \right)_h \tag{4.38}$$

$$\beta_5 = \omega V \left(\frac{\partial \rho}{\partial h} \right)_p \tag{4.39}$$

$$\beta_6 = \omega \rho \tag{4.40}$$

Tian et al. (2017a) use the same principles, but the arrangement of the conservation equations is slightly different. In their two-phase analysis, the equations are sorted to calculate the rate of change in vapor pressure and temperature. Thus, equations 4.41 and 4.42 are implemented.

$$\frac{dp}{d\theta} = \frac{\frac{1}{v}\left[\left(\frac{\partial h}{\partial v}\right)_T - \frac{(\partial h/\partial T)_v(\partial p/\partial v)_T}{(\partial p/\partial T)_v}\right]\frac{dv}{d\theta} - \frac{1}{V}\left[\sum \frac{dm_{in}}{d\theta}(h_{in}-h) - \frac{dQ_{liq}}{d\theta}\right]}{1 - \frac{1}{v}\frac{(\partial h/\partial T)_v}{(\partial p/\partial T)_v}} \tag{4.41}$$

$$\frac{dT}{d\theta} = \frac{\left[\frac{1}{v}\left(\frac{\partial h}{\partial v}\right)_T - \left(\frac{\partial p}{\partial v}\right)_T\right]\frac{dv}{d\theta} - \frac{1}{V}\left[\sum \frac{dm_{in}}{d\theta}(h_{in}-h) - \frac{dQ_{liq}}{d\theta}\right]}{\left(\frac{\partial p}{\partial T}\right)_v - \frac{1}{v}\left(\frac{\partial h}{\partial T}\right)_v} \tag{4.42}$$

These equations are developed and described in detail by Zaytsev (2003).

The magnitude of the numerical errors caused by discretization depends on the step size used during calculations. Smaller steps reduce the error. However, smaller steps also increase the number of steps that has to be taken to reach the end of the compression process. This means an increase in the number of calculations, and thereby an increases in computational time. Depending on the available computing power, a suitable balance between discretization errors and computational time must be found. The step size used by Tian et al. (2017a) in their simulations is $\Delta\theta = 1°$. The total rotation during a complete compression cycle, i.e., from the start of suction to the end of discharge, is determined by the length and pitch of the rotors. For the compressor investigated by Tian et al. (2017a), the compression cycle takes a total of 805°. This means that more than 800 calculations steps are taken in one simulation.

4.3.10 Discharge Phase

When the compression chamber opens to the discharge, at a rotational angle denoted by $\theta_{dis,open}$, the discharge temperature and the outlet mass flow rate must be calculated. The pressure level in the discharge line is imposed by the condenser downstream of the compressor (or the absorber in a CACHP), and under-compression or over-compression occurs if the pressure in the control volume does not match the condenser pressure at the end of the compression phase. Chamoun et al. (2013) use a very simple approach, where the compressed vapor is discharged at constant pressure equal to the pressure in the discharge line. If under-compression occurs, the vapor is re-compressed during the first part of the discharge phase, until the discharge pressure level is reached. If over-compression occurs, the vapor goes through an isenthalpic expansion when the compression chamber opens to the discharge.

Tian et al. (2017b) use a more realistic approach, where the vapor is not discharged at constant pressure. Instead, the discharge flow is treated in the exact same way as the leakage flows. With a given discharge port area, the discharge mass flow of vapor and liquid is calculated according to equations 4.9–4.12. If liquid is still present during discharge, heat transfer between the liquid and the vapor is continually calculated throughout the discharge phase as well.

4.3.11 Simulation Environment and Numerical Procedure

Many different tools and numerical procedures can be used to solve the described mathematical models. Chamoun et al. (2013) use the Dymola environment for their simulations. Dymola is based on the Modelica modeling language (Modelica Association, 2017). It supports object-oriented programming, and it allows equations to be written in an acausal manner (such as equations 4.33 and 4.37). Tian et al. (2017a) use C++, another object-oriented programming language. Unlike Modelica, equations must be written in a causal manner (such as equations 4.41 and 4.42). One of the main advantages of using object-oriented programming is that reuse and extension of the model is made simple. When reusing a model, instead of having to copy and adapt procedural code, the already created objects can be used, functionality can be inherited between different objects, and code duplication can be avoided. To give an example: as explained above, Tian et al. (2017a) treat the discharge flow and the leakage flows equally. Therefore, a general path object should be created, before more specific discharge path and leakage path objects are created. The specific paths can simply inherit functionality from the general path, and adaptions to each path can be made without writing the entire path code all over again.

An important difference between Modelica and C++ is that Modelica/Dymola offers a more graphical approach to programming. In Dymola, objects can be represented graphically and physical connections between objects can be created by drawing lines between them. A graphical interface can often make the code easier to understand and the flow of data easier to follow.

To solve the differential equations numerically, Tian et al. (2017a) use the fourth-order Runge-Kutta method. Choosing a suitable numerical procedure is important. A method that is too simple can cause inaccurate results, while a method that is too elaborate can cause extensive computational time. Different methods are used in literature. Gudjonsdottir et al. (2019) use the Heun method to solve their numerical screw compressor model. The Heun method is similar to a second-order Runge-Kutta method (Gautschi, 2012), meaning that the calculations carried out by Gudjonsdottir et al. (2019) probably are less time consuming than Tian et al. (2017a), but the risk of significant numerical errors is higher.

To initialize the simulations, suitable initial conditions must be given as input to the model. Starting off with numerical values that are highly unrealistic and too far away from the correct numerical solution can trigger simulations to give false results or no result at all. Tian et al. (2017a) initialize the simulations by first calculating the pressure and temperature in an ideal adiabatic compression process at the given suction and discharge conditions, and then use the results as initial conditions for calculation of the real process. This gives a starting point that is not too far from the real process, and thus the simulations quickly converge towards the correct solution. Gudjonsdottir et al. (2019) use a similar approach, where the first calculations are simplified and ideal conditions are considered. At first, the model is run without any leakage flow. Afterwards, the results are used to initialize simulations with leakage flow.

Finding the exact numerical solution through iterative calculations is in most cases overly time consuming and not necessary. Therefore, a convergence tolerance is set. Whenever the difference between consecutive calculations becomes lower than the chosen tolerance, the calculations stop and the simulation results are displayed. Which parameter the convergence tolerance is based on depends on the iteration procedure and the desired precision. Tian et al. (2017b) set the tolerance to 0.1 % change in vapor pressure, while Gudjonsdottir et al. (2019) set the tolerance to 0.1 % change in isentropic efficiency.

4.3.12 Properties of the Working Fluid

In order to obtain valid simulation results, various thermodynamic properties of the working fluid must be calculated correctly throughout the entire compression cycle. Multiple procedures, databases and software are designed to achieve this. In the simulations performed by Tian et al. (2017a), the properties of liquid water and water vapor are determined by the REFPROP software (Lemmon et al., 2018). REFPROP is a widely used commercial software for calculation of transport and thermodynamic properties of pure fluids and mixtures. KBC Infochem Multiflash is another example of a fluid property software that is frequently used in industry and academia (Heiyanthuduwage et al., 2011; KBC, 2019).

Chamoun et al. (2013) and Tian et al. (2017b) investigate pure water compression. Contrary to their investigations, the CACHP under consideration here uses a mixture of ammonia and water. Since ammonia and water have different evaporation temperatures, it is a so-called zeotropic mixture. For a given pressure and a pure fluid, either water or ammonia, evaporation takes place at a constant temperature. However, when the two fluid are mixed together, evaporation takes place over a temperature range. This is illustrated by the bubble point and dew point curves in Figure 4-4. The bubble point represent the temperature where the first bubble of vapor is formed when heating the fluid, while the dew point represents the temperature where the first drop of liquid (dew) is formed when cooling the fluid.

The fact that the ammonia-water mixture evaporates over a temperature range, not a constant temperature, complicates the calculations. E.g., looking back at the flash evaporation calculations, the specific enthalpies of saturated vapor, h_g, and saturated liquid, h_f, are used in equation 4.13. For a pure fluid, these enthalpy values are known when the pressure is known. This is not the case for a zeotropic mixture like ammonia-water, where the saturation enthalpies are dependent on the ammonia concentration as well. In other words, at least one additional parameter must be known before the calculations can be performed. Additionally, the ammonia concentration in each control volume varies dependent on liquid injection and leakage flow rates. Thus the temperature range between the bubble point and dew point is constantly changing, making the calculations very complex.

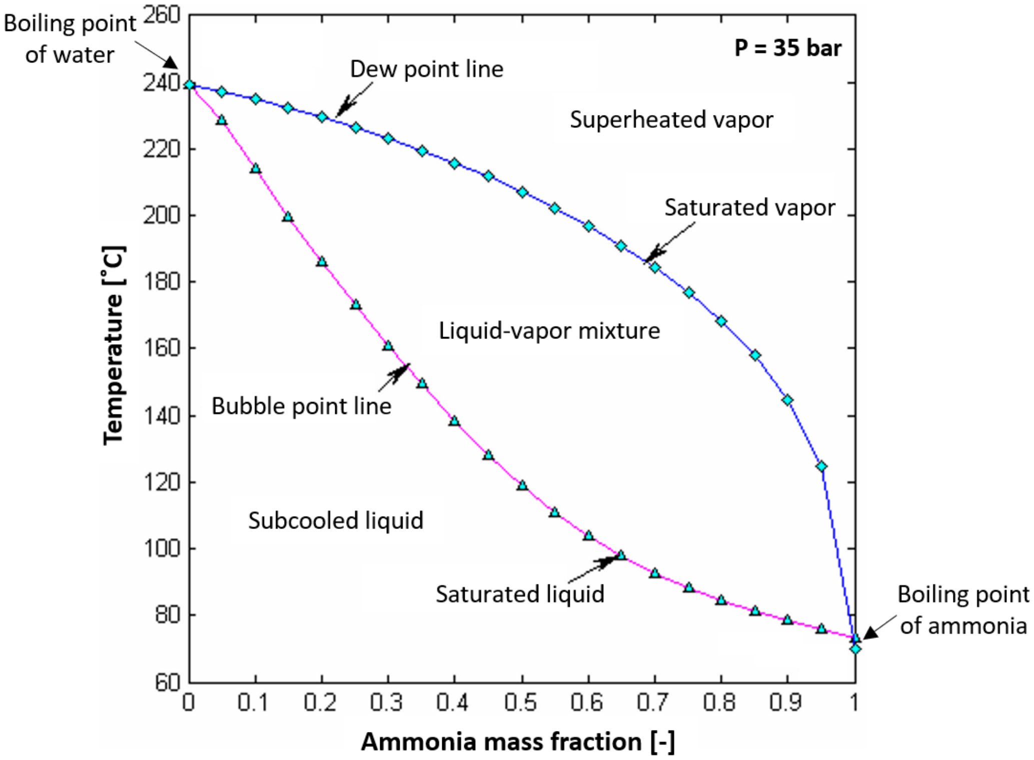


Figure 4-4: Bubble point and dew point curves for ammonia-water mixture at a constant pressure of 35 bar (3500 kPa). Adopted from Ganesh and Srinivas (2017, p.115).

To be able to perform the complex calculations, a fast and accurate way of obtaining the fluid properties at each state point is essential. In their numerical study of an ammonia compressor with injection of liquid ammonia-water, Tian et al. (2017b) take thermodynamic properties such as the bubble point and the dew point from literature by MCE (2006). In addition, enthalpy, entropy and specific volume is calculated through the Gibbs energy method described by Ganesh and Srinivas (2011). To validate their calculations, the REFPROP database is used. In a recent study by Rattner and Garimella (2016), a new routine for fast and stable computation of thermodynamic properties of ammonia-water mixtures is developed. The results in their study indicate a significant reduction in computational time compared to ammonia-water calculations using the REFPROP software. However, in the latest version of REFPROP (version 10.0 from 2018) a new and improved mixture model for ammonia-water is introduced (NIST, 2019). Thus the performance of the Rattner and Garimella (2016) routine relative to the latest REFPROP version has not been established. Rattner and Garimella (2016) demonstrated the practicality of their property routine by developing a transient ammonia-water absorber model. Additionally, the routine is used by Gudjonsdottir et al. (2019) in their screw compressor study.

4.3.13 Validation

In order to prove that a numerical screw compressor model gives valid performance predictions, the model should be compared with results from experimental testing of an actual compressor. Taking this into consideration makes the study by Chamoun et al. (2013) questionable. Many simulation results are presented and discussed in their study, but none of them are validated against physical test results. In fact, their simulation results are known to be somewhat inaccurate due to the simplifications made in the model, e.g., it is unrealistic that the compressed vapor is discharged at a constant pressure equal to the pressure in the discharge line.

Tian et al. (2017a) on the other hand, validate their simulation results by comparison with experimental pressure recordings from a test rig. The comparison is illustrated in a pressure-volume diagram, as shown in Figure 4-5. The ideal adiabatic compression cycle used to initiate the simulation is also shown in this figure. At suction pressure of 59 kPa and a discharge pressure of 110 kPa, the compressor works at over-compression, as indicated by both the simulated result an the experimental recording. Furthermore, due to the cooling effect of liquid injection, the simulated and the experimental processes both deviate from the adiabatic process. Tian et al. (2017a) suggest a reason for the difference between the simulated results and the experimental recording by means of a delay in liquid evaporation. In the model, the amount of evaporated liquid is calculated according to equation 4.18 and it is assumed that vapor is cooled to saturation at each compression step. However, in the first part of the compression process the temperature difference between the vapor and the injected liquid is low, and the heat transfer rate is therefore lower than assumed. Instead, evaporation of the injected liquid occurs later in the compression process, when the vapor temperature is high. This is why the recorded vapor pressure is lower than simulated at first, before it increases rapidly and overshoots the simulation when evaporation later occurs.

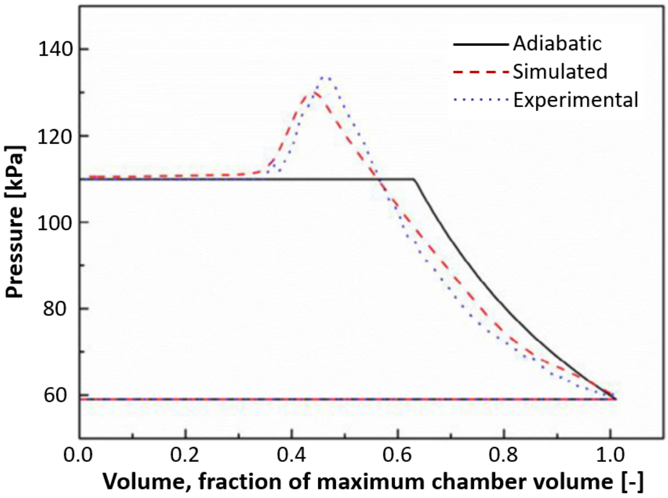
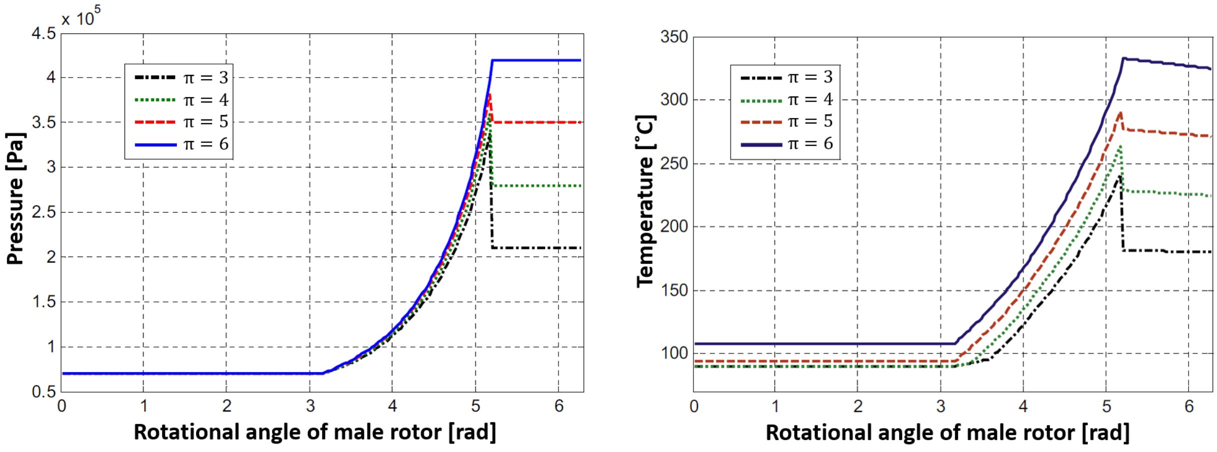


Figure 4-5: Pressure-volume diagram, comparison between simulated result and experimental recording (Tian et al., 2017a, p.81).

4.3.14 Simulation Results

To show some of the simulation results presented by Chamoun et al. (2013), the evolution of pressure and temperature throughout the compression cycle is illustrated in Figure 4-6. Liquid water at 90 °C is injected into the suction chamber, and different external pressure ratios are evaluated. Over-compression occurs with pressure ratios of 3, 4 and 5, while under-compression occurs with a pressure ratio of 6. There is a strong correlation between pressure and temperature, and thus the shape of the evolution curves are very similar. As shown in Figure 4-6b, the temperature is higher throughout the entire compression cycle when the pressure ratio is higher. This is mainly due to increased leakage flows at higher pressure. High-temperature vapor leaks to the low-pressure chambers and thereby cause a temperature increase.



(a) Pressure evolution.

(b) Temperature evolution.

Figure 4-6: Evolution of pressure and temperature throughout the compression cycle for different external pressure ratios, π (Chamoun et al., 2013, p.486).

Furthermore, Chamoun et al. (2013) investigate the effect of different liquid injection mass flow rates. At a constant pressure ratio, injection rates from 0 to 0.03 kg/s are simulated. The results, shown in Figure 4-7, clearly demonstrate the need for liquid injection. Without liquid injection the temperature rises above 300 °C at the end of compression. Such operating conditions will cause a great deal of damage to the compressor in a short amount of time and must be avoided. To keep the temperature below 200 °C, more than 0.02 kg of liquid must be injected every second.

The two-phase analysis by Tian et al. (2017a) goes into further detail about the effect of liquid injection mass flow rate. Here, the liquid is injected during the first part of compression, and the amount of liquid mass is tracked throughout the compression cycle. The results from simulations with injection rates from 0.0022 to 0.0042 kg/s is shown in Figure 4-8a. These mass flow rates are much lower than those investigated by Chamoun et al. (2013) due to a much lower external pressure ratio and thus a lower cooling need. With a liquid injection mass flow rate of 0.0022 kg/s, all the liquid evaporates during compression, and the discharge vapor is superheated.

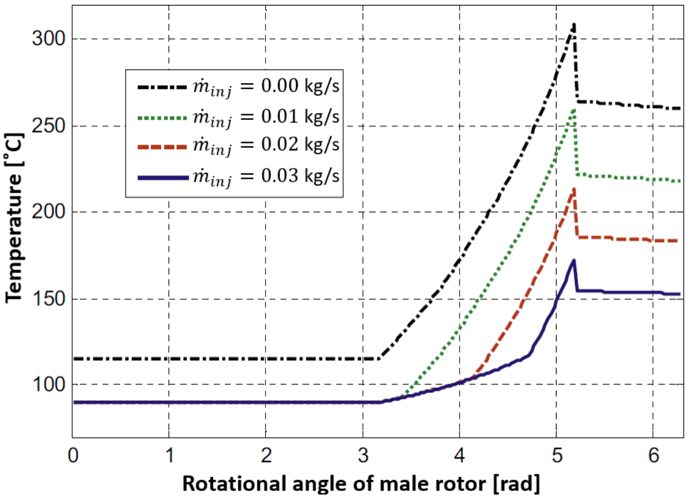
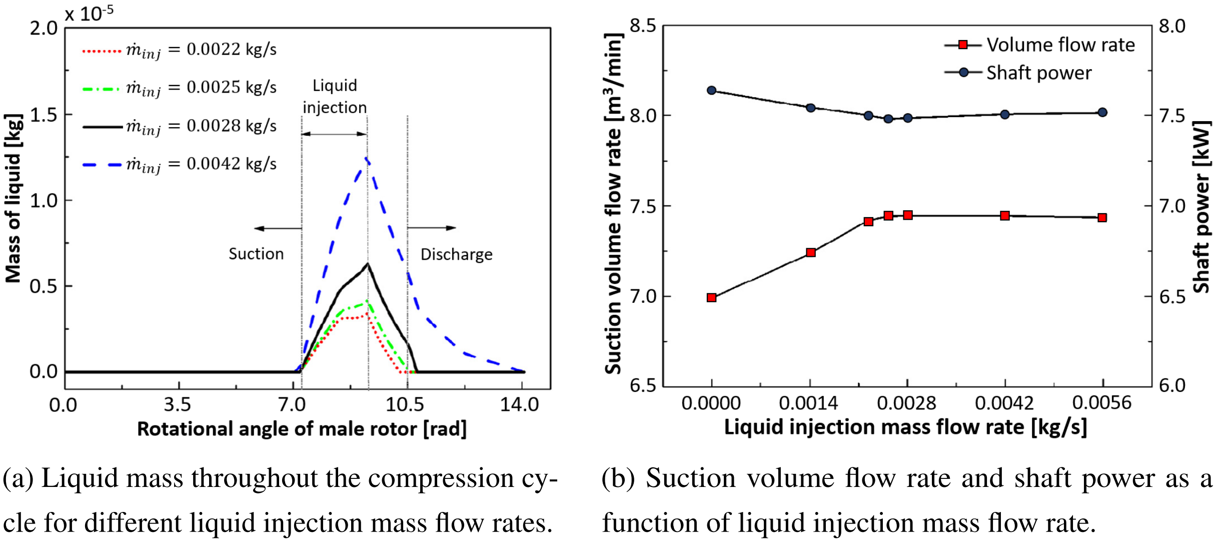


Figure 4-7: Evolution of temperature throughout the compression cycle for different liquid injection mass flow rates, $\dot{m}_{inj}$ (Chamoun et al., 2013, p.487).

With a liquid injection mass flow rate of 0.0042 kg/s, saturated vapor and a significant amount of liquid is discharged. Under the operating conditions investigated here, if discharge with minimal superheat and no liquid is desired, the injected flow rate should be set to 0.0025 kg/s.

Moreover, Tian et al. (2017a) demonstrate the effect of liquid injection on the suction volume flow rate and the shaft power, as shown in Figure 4-8b. When the injection flow rate is increased in the range 0.0000–0.0025 kg/s, there is a significant increase in the suction flow of the compressor. This is due to the liquid's sealing effect. At liquid injection mass flow rates below 0.0025 kg/s, liquid is not present throughout the entire compression cycle, and thus there are large leakages towards the suction side of the compressor. When the liquid injection mass flow rate is increased in the range 0.0025–0.0056, there is a slight decay in the suction flow of the compressor. This



(a) Liquid mass throughout the compression cycle for different liquid injection mass flow rates.

(b) Suction volume flow rate and shaft power as a function of liquid injection mass flow rate.

Figure 4-8: Investigations on the effect of liquid injection flow rate (Tian et al., 2017a, p.82).

is due to the additional liquid that never evaporates and occupies a small part of the chamber volume. Similarly, there is rapid decrease in shaft power at first, and a slow increase in shaft power further on. The shaft power decrease is due to the cooling effect of the injected liquid, while the minor shaft power increase is due to power consumption of liquid transportation.

4.4 Discussion

Compared to conventional prototype testing, computer modeling is a time-saving and cost-effective means to analyse and design energy efficient screw compressors. However, the challenges related to development of such models are many. The screw compressor process is very complex, which makes it difficult to describe the process solely through mathematical equations. Due to great variety in possible modeling approaches, a set of criteria for an optimal approach was defined in section 4.2. By considering these criteria we can argue the following:

An empirical modeling approach is not suitable for the objectives of this work. Empirical models are dependent on experimental testing and overly restricted by the tested operating range. This is not sufficient, since the aim here is to develop a new type of screw compressor. Similarly, analytical screw compressor models are considered impractical due to dependency on physical testing. On the other hand, analytical models can prove to be useful in terms of comparison and model validation.

A numerical approach is the best modeling option for the objectives of this work. Three-dimensional numerical models are however regarded to be too comprehensive and time consuming. Such models lack flexibility and they are not suitable for integration into larger CACHP system models. Thus, quasi-one-dimensional models are considered to be the most promising course going forward. Two different quasi-one-dimensional approaches have been investigated by looking at the work presented by Chamoun et al. (2013) and Tian et al. (2017a). The model developed by Chamoun et al. (2013) is easy to implement and reuse. It has low complexity, short computational time and relatively little required input data. However, due to rather large simplifications, the model lacks accuracy and the output data provides little detail. The model developed by Tian et al. (2017a) is found to have higher complexity and heavier calculations, providing a more detailed output than Chamoun et al. (2013). The model was validated against experimental pressure recordings, proving its physical validity and relatively high accuracy. The most notable inaccuracy is believed to be related to the calculated rate of liquid evaporation during compression.

Furthermore, Chamoun et al. (2013) and Tian et al. (2017a) both use object-oriented modeling languages, which is advantageous in regard to reusability and flexibility. Modelica/Dymola offers a more graphical approach than C++ and can thus make a model easier to understand, adapt and utilize. Modelica is considered the most favourable modeling language for the screw compressor model to be developed here. Not only due to the graphical Dymola environment, but also due to Modelica's capabilities regarding fluid property calculations, the many built-in numerical procedures in Dymola, and the suitable capabilities of the Modelica libraries named TIL and TILMedia. These features will be explained and discussed further in section 5.

5 Modelica Screw Compressor Model

A screw compressor model is developed based on knowledge obtained from the above-presented literature. The aim of the modeling approach is efficient and reliable simulation of a screw compressor with liquid injection and a mixture of ammonia and water as the working fluid. The model must be able to accurately predict the thermodynamic states within the compressor at different operating conditions. It is desired that the model easily can be modified in order to investigate different compressor configurations, especially with regards to the position of the injection port and the composition of the injected liquid. It is also desired that the model can be implemented into a bigger CACHP system model with ease.

The model is written in the Modelica modeling language, and it is implemented and solved in the Dymola simulation environment. Two commercial Modelica libraries are utilized, namely TIL and TILMedia. TIL is a library of components for steady-state and transient simulation of thermodynamic systems, while TILMedia is a library of methods for fluid property calculations. Both libraries are developed by TLK-Thermo GmbH (2020). TILMedia offers thermodynamic property calculations for a wide range of fluids, including the mixture of ammonia and water. The calculation procedures are based on an assumption that the fluid is in a vapor-liquid equilibrium state at all times. Thermophysical data required for each calculation is taken from external databases such as REFPROP, as well as data developed by TLK-Thermo GmbH. A vapor-liquid equilibrium state is a state where the pressure, the temperature and the partial Gibbs energies of the vapor and the liquid portions of the fluid are equal (Bahadori, 2017). In what follows it is demonstrated how these Modelica libraries are utilized in order to build a screw compressor simulation model.

5.1 Model Description

The screw compressor model is made up of multiple sub-models, which is hereinafter referred to as components. The different components are described in sections 5.1.1 – 5.1.4, and in section 5.1.5 it is shown how all the components are combined into a complete screw compressor model.

5.1.1 Component 1 – The Control Volume

The most fundamental model component is the control volume. Each control volume represents the volume of one rotor cavity, equivalent to the approach taken by both Chamoun et al. (2013) and Tian et al. (2017a). A rotor cavity is illustrated in Figure 5-1. The number of control volume components in the screw compressor model depends on the rotor profile and the geometry of the compressor under consideration. The compressor that is currently under consideration for the CACHP test rig is a MYCOM i125L screw compressor (Mayekawa, 2015). The rotor profile in

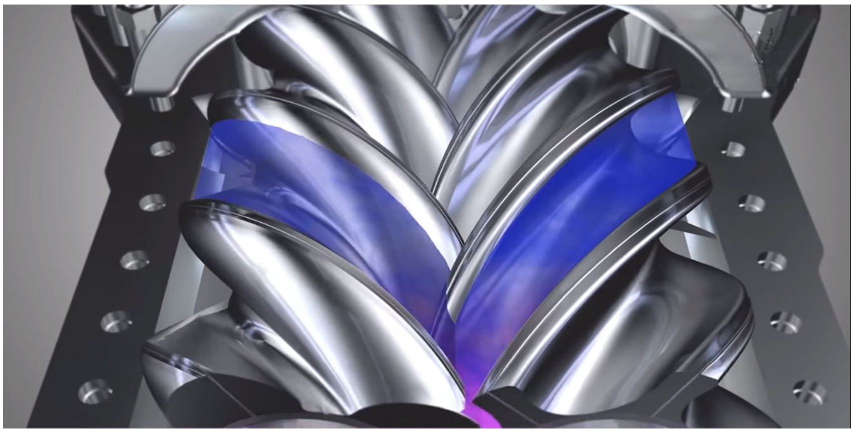
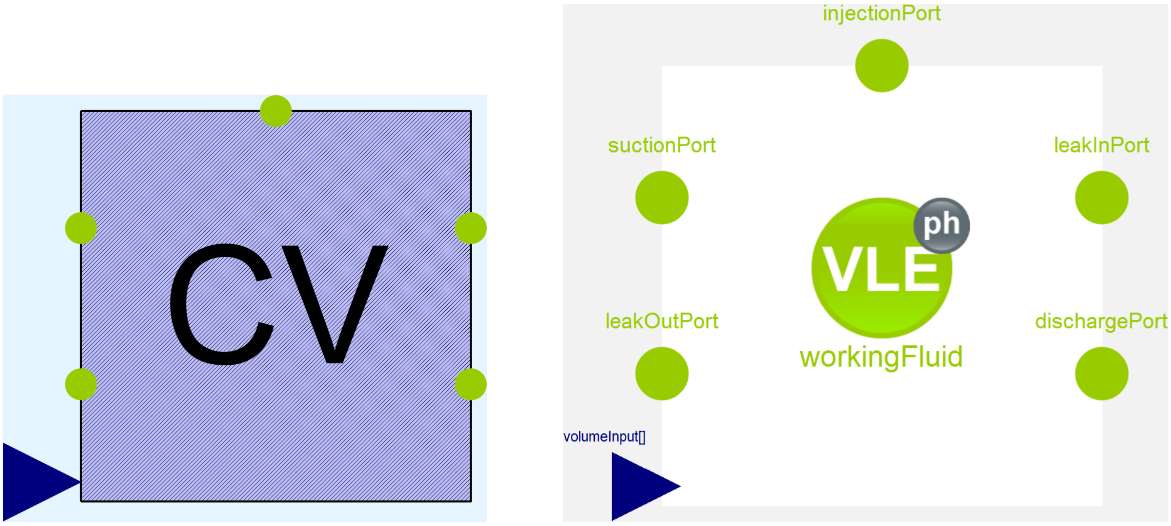


Figure 5-1: Highlight of one rotor cavity, defining the outline of a control volume (ACA, 2018).

this compressor is a 4/6 configuration, i.e., there are four lobes on the male rotor and six lobes on the female rotor. By considering that there are four lobes on the male rotor, and one cavity along each lobe, one might assume that there should be four control volumes in a model for this compressor. However, the number of lobes does not equal the number of control volumes in the model. This is because each rotor cavity is split into two at the point where the male and the female rotors mesh together. The number of times such a split occurs for each cavity depends on the length of the rotors, as well as the pitch of the helix shape for each rotor. For the screw compressor model being developed and implemented here, it is assumed that each cavity along the male rotor is split once, and thus the number of control volumes in the model is set to eight. Nevertheless, the number of control volumes can easily be altered to fit different rotor configurations. In the Dymola simulation environment this type of alteration is done through a simple drag-and-drop procedure. Where and how to place each control volume component in the screw compressor model will be demonstrated in section 5.1.5.

The icon that is displayed on the screen when a control volume component is incorporated into the model is shown in Figure 5-2a. Connections between the different model components are made by drawing lines between the connection points at each component. These connection points are marked by green dots on the component icon. The model diagram, displayed in Figure 5-2b, shows what lies beneath the model icon. The five green connection points represent five different ports where the working fluid can flow in and out of the control volume: the suction port, the discharge port, the injection port, the leak-in port and the leak-out port. These port elements are taken from TIL library and there are four variables associated with each port element, namely p, $\dot{m}$, $h_{outflow}$ and $x_{outflow}$. Here p represents the pressure at the location of the port, while $\dot{m}$ represents the mass flow through the port. The positive flow direction is defined as flow going into the component, and thus cases where fluid flows out of the control volume through the given port gives $\dot{m} < 0$. Furthermore, $h_{outflow}$ represents the specific enthalpy of the mass flow if the flow direction is out of the component. In cases where the flow direction is into the component, the specific enthalpy of the flow is given by the port at the other end of



(a) Component icon.

(b) Model diagram.

Figure 5-2: The component icon and the model diagram for the control volume component. The model diagram (b) shows the structure and the elements that are part of the sub-model represented by the component icon (a).

the connection line, where the fluid comes from. Likewise, $x_{outflow}$ represents the mass fraction of each constituent in the flow if the flow direction through the port is out of the component. $x_{outflow}$ represents the mass fractions of ammonia and water in the investigations presented here.

Another type of element that is shown in the model diagram in Figure 5-2b is the working fluid element, marked by a green circle and the letters VLE. VLE stands for vapor-liquid equilibrium. This element is taken from the TILMedia library, and it provides a fast and efficient method for calculating the thermodynamic state of the working fluid within the control volume. The *ph* marking in the upper-right corner of the element symbol indicates that this property calculation method requires the pressure and the specific enthalpy of the working fluid as input. TILMedia offers calculation procedures with different inputs, such as ρT (density – temperature), *ps* (pressure – specific entropy) and pT (pressure – temperature). *ph* is considered to be the most suitable procedure due the involvement of specific enthalpy in the equations for conservation of energy discussed in section 4.3.2. In calculations where the working fluid is a mixture of multiple constituents, such as ammonia-water, the mass fractions are also required as an input to all these procedures. It is assumed that the thermodynamic conditions within each control volume are homogeneous, i.e., the pressure, the temperature and the mass fractions are equal throughout the entire cavity, and thus there is only need for one working fluid element within each control volume. If heterogeneous conditions are to be taken into consideration, separate calculations must be made for each condition, requiring multiple TILMedia working fluid elements.

The third and final element of the control volume component is the volume input, represented by a blue triangle in Figure 5-2. The blue triangle marks a connection point for the component, similar to the fluid ports marked by green dots. However, this connection point is not associated with flow of fluid. It is a connection point where information about the geometry and the position of the control volume is to be provided to the component. The volume input is a vector containing three variables: V, $\frac{dV}{dt}$ and θ. V represents the volume of the cavity, while $\frac{dV}{dt}$ represents the time derivative of the volume. θ represents the rotational angle of the control volume, i.e., at what stage in the compression cycle the control volume is. These three parameters are all calculated by a sub-model, referred to as a volume function. The icon used to represent a volume function is shown in Figure 5-3. There is a white triangle at the top edge of the icon. This triangle is the volume function's connection point, and a line is drawn between this triangle and the blue triangle on a control volume in order to connect the two. The angle that is displayed in the center of the icon indicates the starting angle for the associated control volume, i.e., the angle of rotation the control volume is located at when the simulation begins at $t = 0$. The total rotation a cavity goes through during a complete compression cycle, from the beginning of the suction phase to the end of the discharge phase, depends on the length and the pitch of the rotors. In the screw compressor model being implemented here, it is assumed that the total rotation during one compression cycle is 720°. Under the assumption that the male rotor spins at frequency of 50 Hz, it takes 40 milliseconds to complete one cycle. The continuously changing cavity volume calculated by the volume function throughout the compression process is illustrated in Figure 5-4. This pattern is repeated every 40 milliseconds. Moreover, there is a constant phase shift between each control volume. When a control volume begins its compression cycle at an angle of 0°, the preceding control volume has already gone through a certain amount of rotation. With 8 control volumes and a total rotation of 720°, the phase shift between each control volume is 90°. This 90° phase shift is demonstrated by the two curves in Figure 5-4. The shape of the curves show how the cavity volume increases during the suction phase, from 0° to 360°, before it decreases again during the compression and discharge phases, from 360° to 720°. The volume function presented here does not represent the exact volume evolution in the actual compressor under

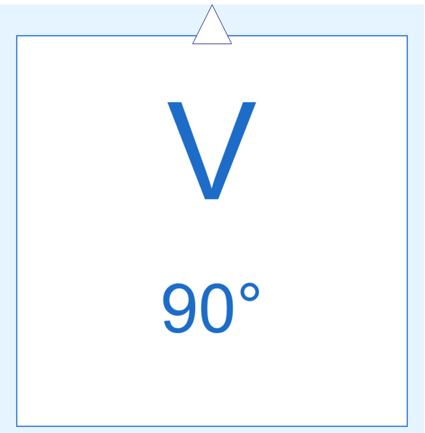


Figure 5-3: Volume function icon. The angle indicates that this function belongs to a control volume starting at a rotational angle of 90°.

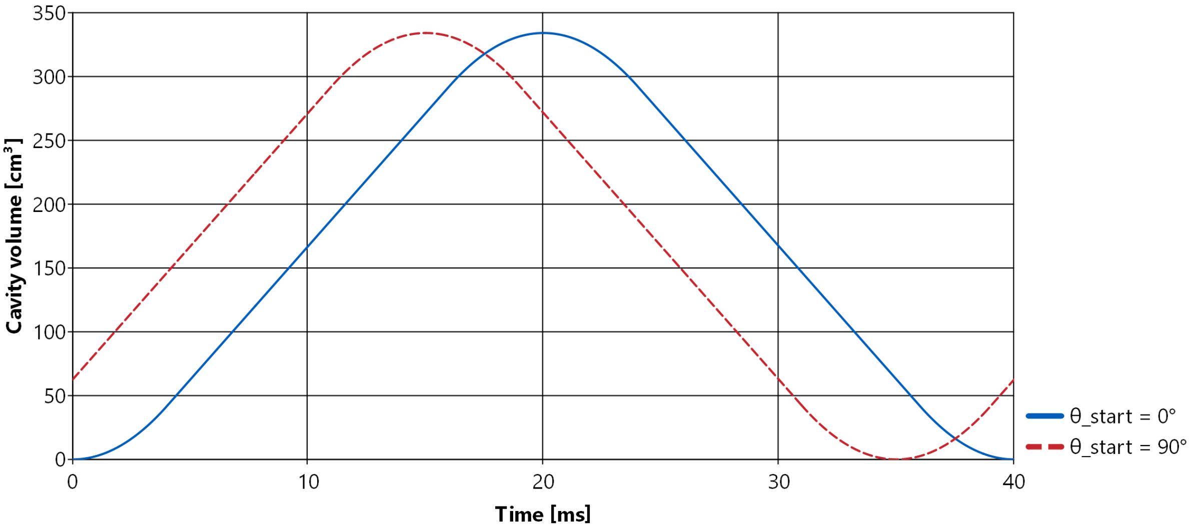


Figure 5-4: Cavity volume as a function of time for a control volume starting at a rotational angle of 0° and the preceding control volume starting at a rotational angle of 90°.

consideration, due to lack of geometrical data for the compressor. However, the fundamental structure of the volume curves are based on a geometry model for a screw compressor developed by Bommel (2016). The Modelica code for the volume function is given in appendix A.1. This code can efficiently be adapted when geometrical data on the screw compressor is obtained, without having to make changes to any other parts of the screw compressor model.

A set of equations for each control volume component has to be solved during simulations. First off, the mass balance equation:

$$\mathrm{der}(m) = \sum \dot{m}_{port} \tag{5.1}$$

Here, der() represents the built-in time derivative operator in Modelica, and it is used to define how the fluid mass within the control volume, m, changes with time. $\sum \dot{m}_{port}$ represents the sum of the mass flows through each of the fluid ports displayed in Figure 5-2. Keep in mind that $\dot{m} < 0$ if the flow direction through a port is out of the component. Additionally, if the working fluid is a mixture, there must be a mass balance for each constituent:

$$\mathrm{der}(mx_c) = \sum \dot{m}_{port} x_{c,port} \tag{5.2}$$

where x_c is the mass fraction of constituent c within the control volume, and $x_{c,port}$ is the mass fraction of the flow through each port. The third and final balance equation is the energy balance:

$$\mathrm{der}(mu) = \dot{W}_{comp} - \dot{Q}_{loss} + \sum \dot{m}_{port} h_{port} \tag{5.3}$$

where u is the specific internal energy within the control volume, $\dot{W}_{comp}$ is the compression power, $\dot{Q}_{loss}$ is the heat transfer rate from the working fluid to the compressor body, and h_{port} is the specific enthalpy of the flow through each port. The compression power is calculated from:

$$\dot{W}_{comp} = -p\frac{dV}{dt} \tag{5.4}$$

where p is the pressure in the control volume and $\frac{dV}{dt}$ is the time derivative of the cavity volume. The required time derivative in equation 5.4 is the reason why the volume input supplied to the control volume contains the derivative variable in addition to the volume variable. The Dymola solver is capable of calculating the time derivative of the volume by itself, but this additional calculation is not required when the derivative is supplied as an input value. Thus the model's computational time is reduced and possible numerical errors are avoided. Furthermore, the heat loss rate is calculated with the same approach as Chamoun et al. (2013):

$$\dot{Q}_{loss} = \alpha V^{2/3} (T - T_{bod}) \tag{5.5}$$

where α is the heat transfer coefficient between the working fluid and the compressor body, V is the cavity volume, T is the temperature of the working fluid, and T_{bod} is the temperature of the compressor body. Here $V^{2/3}$ is denoted the area available for heat transfer, but this expression may be altered for investigations where more detailed geometrical data on the compressor is available. In order to clarify how the different flows of mass and energy are connected to each control volume in the screw compressor model, a principle sketch is shown in Figure 5-5. Notice here that the leak-out flow for a given control volume corresponds to the leak-in flow for the subsequent control volume.

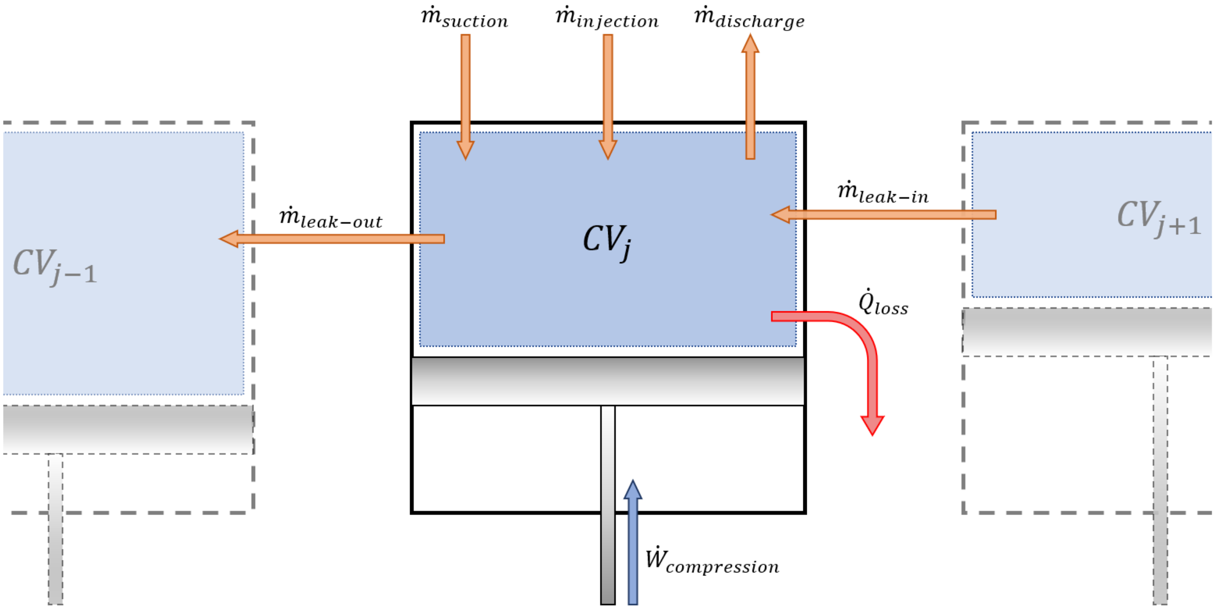


Figure 5-5: Schematic showing the mass and energy flows associated with each control volume.

By using the specific internal energy at each point in time, calculated from equation 5.3, the specific enthalpy in the control volume can be calculated from its definition:

$$h = u + \frac{p}{\rho} \tag{5.6}$$

where ρ is the density of the working fluid. The density is obtained from the TILMedia property calculation procedure. This density value is also utilized in calculation of the total fluid mass within the control volume:

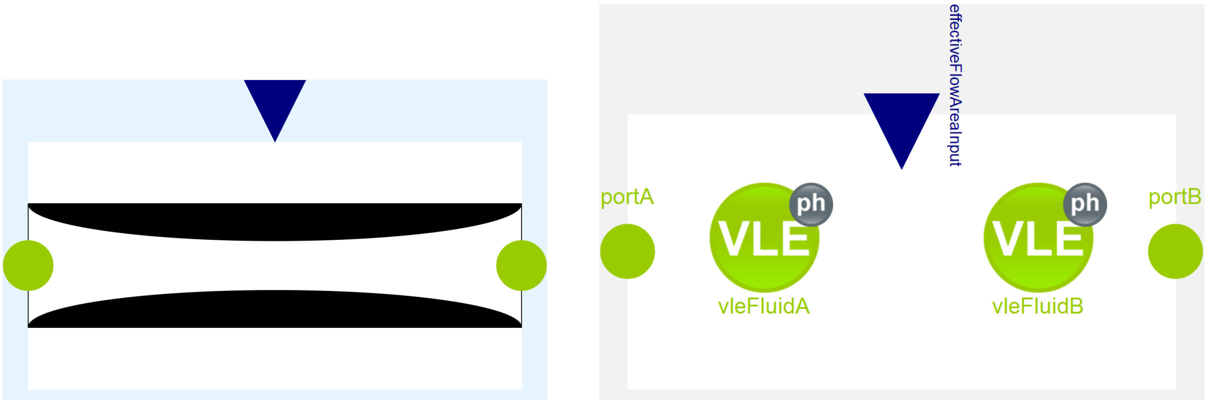
$$m = \rho V \tag{5.7}$$

Based on the pressure, the specific enthalpy and the mass fractions calculated through equations 5.1 – 5.7, the TILMedia procedure calculates multiple other thermodynamic properties that are valuable for analysis of the simulation results, e.g., the temperature, the specific entropy and the vapor quality of the working fluid throughout the compression process.

At the end of a compression cycle, when the cavity volume reaches zero and all the fluid has been discharged from the control volume, the properties of the control volume are reinitialized and the cycle repeats itself. This is done with the built-in $\mathrm{reinit}()$ operator in Modelica. Essentially, the $\mathrm{reinit}()$ operator pauses the simulation and resets a variable to a given value, before the simulation proceeds. The variables that are reinitialized at the end of each cycle are p, h and x_c. When reinitializing, the pressure in the control volume is set equal to the pressure in the suction line connected to the compressor, and the specific enthalpy and mass fractions are set equal to that of the fluid in the suction line. This creates a cycle that repeats itself indefinitely, resulting in a continuous flow of fluid through the screw compressor model. The Modelica code for the control volume component is given in appendix A.2.

5.1.2 Component 2 – The Flow Restrictor

The second component in the screw compressor model is the flow restrictor. This component is used to represent the different paths the working fluid can flow through. These paths involve the leakage paths, the suction paths and the discharge paths. The icon that is shown when this component is implemented in a model is shown in Figure 5-6a. Moreover, the model elements that the component is made up of are shown in Figure 5-6b. Once again the port elements from the TIL library are utilized, marked by green dots on the component icon. These ports are the connection points to the flow restrictor, where the working fluid can flow in or out of the



(a) Component icon.

(b) Model diagram.

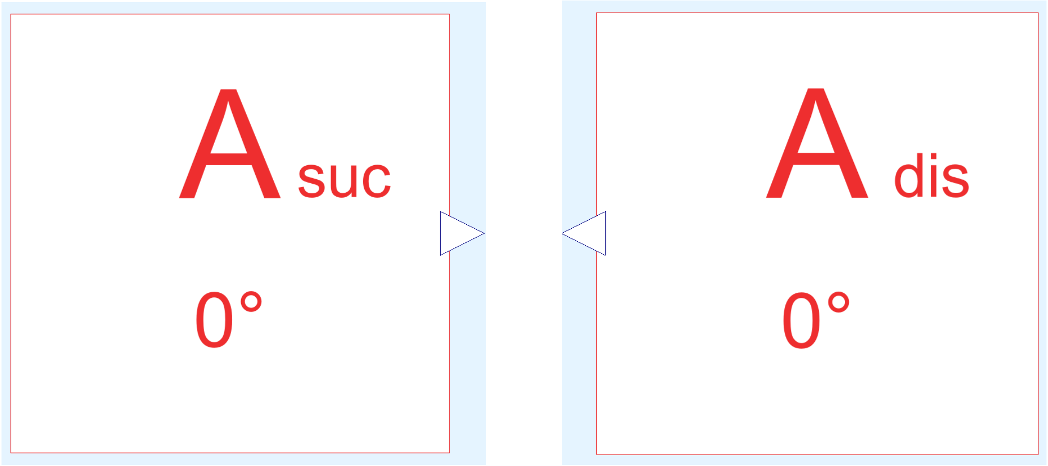
Figure 5-6: The component icon and the model diagram for the flow restrictor component.

component. Similar to the equations Chamoun et al. (2013) used to calculate leakage flow rates, the mass flow rate through a flow restrictor component is calculated from:

$$\dot{m} = A_{eff}\sqrt{2\rho\left(p_{high} - p_{low}\right)} \tag{5.8}$$

where A_{eff} is the effective cross-sectional area of the flow path, ρ is the density of the working fluid upstream of the component, p_{high} is the pressure on the high-pressure side, and p_{low} is the pressure on the low-pressure side. The value for the effective flow area is supplied to the flow restrictor component by an area function that is connected to component's input connection point, marked by a blue triangle in Figure 5-6. It is not given which side of the component that represents the high-pressure side. In fact, the high-pressure side can switch back and forth during a simulation, meaning that the working fluid can flow in both directions through the flow restrictor. Because of this, two separate VLE fluid elements from the TILMedia library is required in the component. Fluid element A calculates the density of the working fluid upstream of the component if the flow direction is into port A, while fluid element B calculates the upstream density if the flow direction is into port B. It is assumed that the flow through each flow restrictor is isenthalpic, i.e., the specific enthalpies for ports A and B are equal. Likewise, there is no change in mass flow rate or fluid composition through the component. The Modelica code for the flow restrictor is given in appendix A.3.

One area of application for the flow restrictor component is to model the suction and the discharge flows for each control volume. A suction flow path to a control volume is implemented by connecting the control volume's suction port to one of the ports on a flow restrictor, and connecting the flow restrictor's other port to the screw compressor's suction port. (This intricate structure is illustrated more clearly section 5.1.5.) The same structure is applied when connecting a control volume to the screw compressors discharge port. Depending on the type of flow



(a) Suction area function.

(b) Discharge area function.

Figure 5-7: Model icons representing the suction area and discharge area functions. The angles indicate that both functions belong to a control volume starting at a rotational angle of 0°.

restrictor, whether it represents a suction path or a discharge path, two different sub-models for calculation of the effective flow area can be connected to the component's input connection point. These sub-models are denoted the suction area function and the discharge area function, and their respective function icons are displayed in Figure 5-7. The output from the suction area function is zero during the compression and the discharge phases, i.e., when the rotational angle of the control volume it is designated to is between 360° and 720°. Thus the mass flow rate through the connected flow restrictor is zero regardless of the pressure difference across the restrictor (shown by inserting $A_{eff} = 0$ in equation 5.8). On the contrary, the output from the suction area function is greater than zero during the suction phase from 0° to 360°. The characteristics of the effective flow area during the suction phase depends on the geometry of the compressor's suction port and the rotor profile. In the screw compressor model implemented here an approximation to the suction port's flow area is used, based on the geometry model developed by Bommel (2016). The output from the suction area function is presented in Figure 5-8. The output from the discharge area function is also presented in the same figure. The discharge port opens at the end of the compression cycle, and the discharge area is substantially smaller than the suction area. The required area is smaller because the density of the working fluid is much higher after it has been compressed. The Modelica codes for the suction area function and the discharge area function are given in appendices A.4 and A.5 respectively.

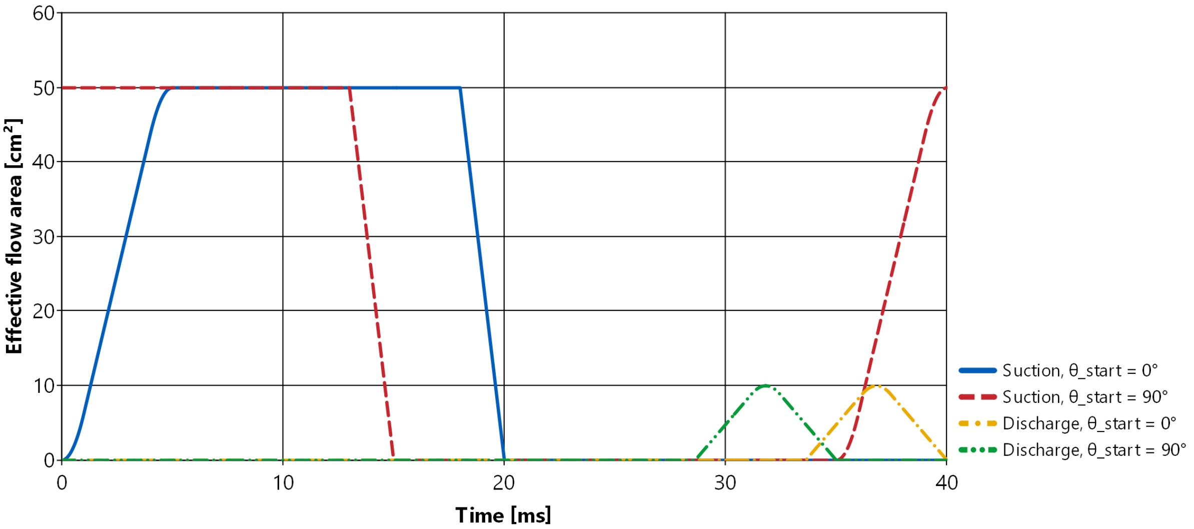


Figure 5-8: Effective flow area of the suction and discharge paths as a function of time for a control volume starting at a rotational angle of 0° and the preceding control volume starting at a rotational angle of 90°. The increasing suction area for the preceding control volume after 35 milliseconds shows that the cycle repeats itself directly after the end of the discharge phase.

The second area of application for the flow restrictor component is to model leakage flows between adjacent control volumes. This is done by connecting one of the flow restrictor's ports to the leak-out port on one of the control volumes, and connecting the flow restrictor's second port to the leak-in port on the other control volume. The sub-model that must be connected to such a

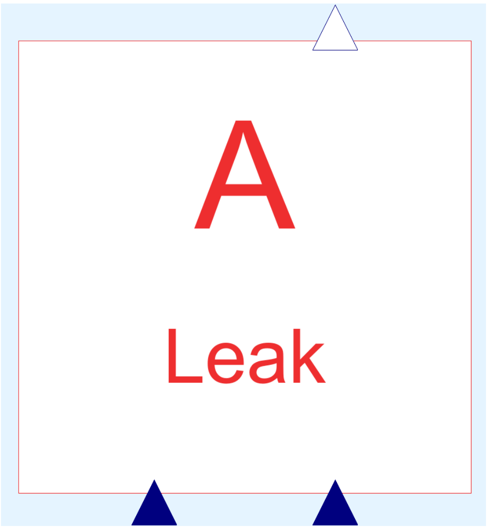


Figure 5-9: Leakage area function icon.

flow restrictor's input connection point is referred to as a leakage area function. The icon used to represent a leakage area function is displayed in Figure 5-9. This function differs from the other functions, as it requires two inputs in order to calculate the output. The input connection points are marked by blue triangles on the function icon. These inputs are required because the geometry of a leakage path depends on the position and the geometry of the two cavities the leakage flows between. Thus the two input connection points are to be connected to the output from the volume functions belonging to the respective control volumes. The volume functions supply information about the position and the volume of each control volume, and the leakage area function calculates the effective flow area of the leakage path based on that information. As discussed in section 4.3.4, multiple leakage paths exist within a screw compressor. The different paths vary greatly in size and impact. Some leakage paths contribute to a significant reduction in compressor efficiency, while other paths are negligible (Fleming and Tang, 1995). In the screw compressor model implemented here, only one of the five leakage paths (shown in Figure 4-2) described by Huagen et al. (2004) is included. The included leakage path is the gap between the rotor tips and the compressor casing, since this is the path which had the most severe impact on the isentropic efficiency of the compressor in the investigations by Fleming and Tang (1995). The effective flow area of the leakage path depends on the length of the sealing line between the rotor tips and the compressor casing. It is assumed here that this length is directly proportional cavity volume, e.g., that a 10 % decrease in cavity volume results in 10 % decrease in sealing line length. Additionally it is assumed that the length of the sealing line between two cavities is determined by the cavity that is smallest in size at each point in time. Thus the effective leakage flow area is calculated by the leakage area function as:

$$A_{eff} = C_{leak} \cdot \min\left(V_{trailing}, V_{leading}\right) \tag{5.9}$$

where C_{leak} represents the leakage flow coefficient, i.e, the proportionality constant. $V_{trailing}$ and $V_{leading}$ represent the volume of the cavity behind the sealing line and the volume of the cavity in front of the sealing line respectively. However, in order to simplify the equation set and decrease the screw compressor model's computational time, the effective flow area of a leakage path is set to zero during periods of time when the two connected control volumes are both in their suction phase or in their discharge phase. By looking back at Figure 5-8 we can see that there is period of time when the suction path is open for both control volumes (t = 0 – 15 ms), as well as a short period of time when the discharge path is open for both control volumes (t = 34 – 35 ms). The suction and discharge paths have a larger effective flow area than the leakage path. Hence a connecting path with much less flow resistance than the leakage path already exists between the two control volumes during these periods of time and there is no need for the leakage path. Furthermore, there is short period of time where the compression cycle of the leading control volume has started over and the trailing control volume still is in its discharge phase (t = 35 – 40 ms). During this period the control volumes are located at opposite sides of the compressor and there is no leakage connection between them. This means that equation 5.9 is only applied in the period from t = 15 ms to t = 34 ms. The output from the leakage area function with the leakage flow coefficient set to 0.05 m^{-1} is shown in Figure 5-10. The characteristics of this leakage area curve is very similar to characteristics obtained by Bommel (2016) for the same leakage path, which indicates that the assumed proportionality between the cavity volume and the sealing line length is valid. The Modelica code for the leakage area function is given in appendix A.6.

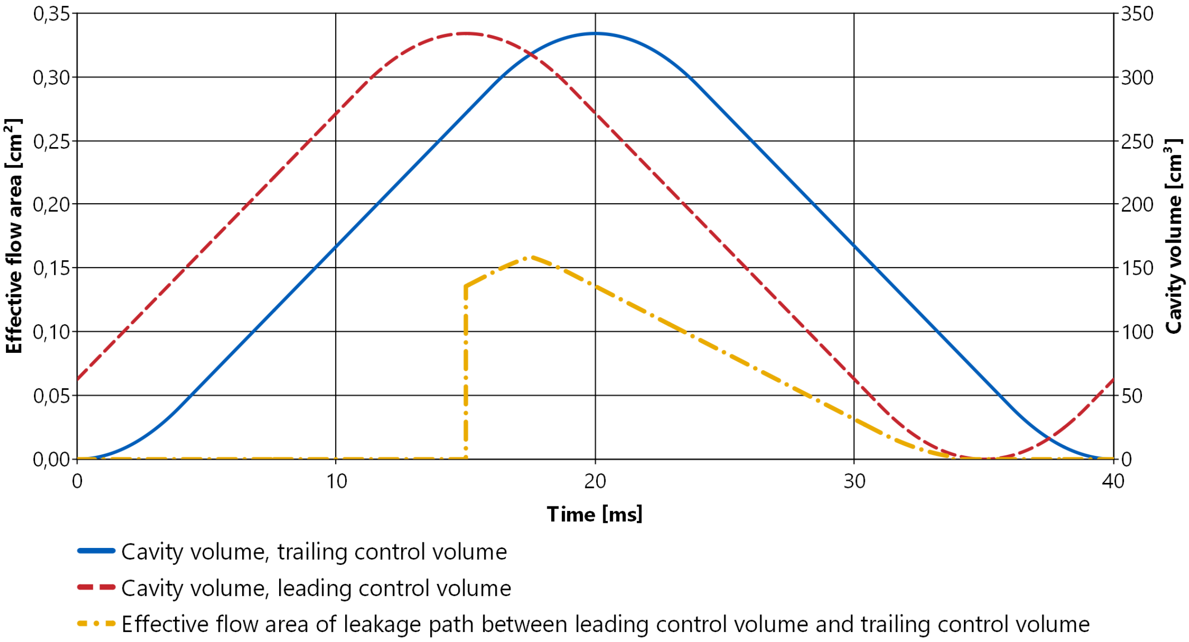
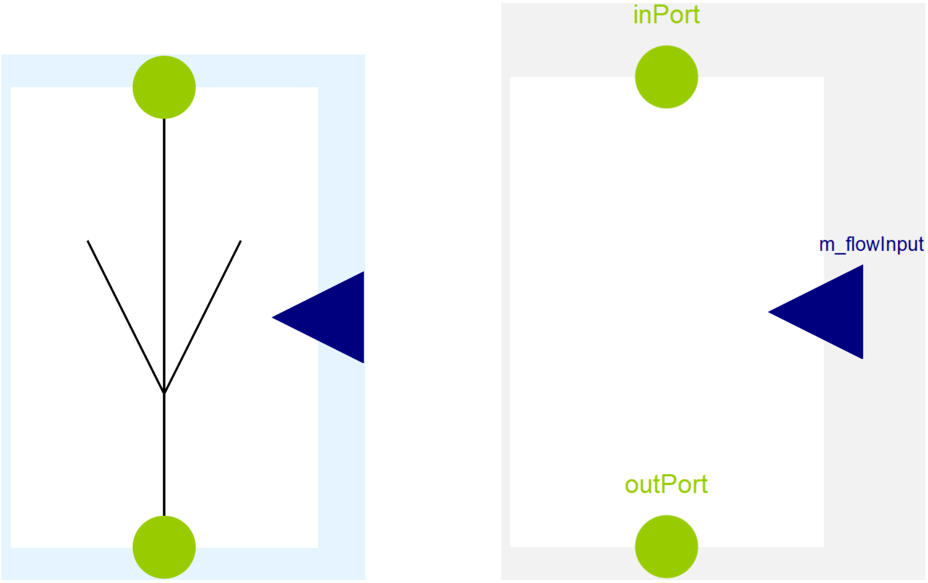


Figure 5-10: Effective flow area of leakage path as a function of time. Leakage through the gap between the rotor tips an the compressor casing.

5.1.3 Component 3 – The Injector

The third component of the screw compressor model is the injector. The injector is a simple component used to control the flow rate of liquid injected into each control volume. The component's icon and model diagram are displayed in Figure 5-11. As shown in the model diagram, the component is built up of only three elements: a fluid port where the working fluid flows into the component, a fluid port where the working fluid flows out of the component, and an input port where the mass flow rate is supplied as an input signal to the component. The outlet port on each injector must be connected to the injection port on the control volume it belongs to. The component offers a simplified way of controlling the flow rate. Contrary to the flow restrictor component, the pressure difference across an injector component does not influence the flow rate through it. No fluid property calculations are required, as the fluid composition and the specific enthalpy of the fluid are assumed to remain constant through the component. The Modelica code for the injector is given in appendix A.7.



(a) Component icon.

(b) Model diagram.

Figure 5-11: The component icon and the model diagram for the injector component.

The sub-model that must be connected to an injector's input connection point is referred to as the injection flow function. The icon used to represent an injection flow function is shown in Figure 5-12. This function calculates the injection flow rate at each point in time based on the position of the associated control volume. The position of the screw compressor's injection port must be predefined by the user of the model, and the injection flow to a control volume is activated as soon as the control volume reaches this position. If a control volume reaches the injection port location and injection begins at a rotational angle of θ_{inj}, the injection flow for that control volume will be deactivated at a rotational angle of $\theta + 90°$, when the trailing control volume reaches the injection port location. In order to avoid discontinuities or infinitely

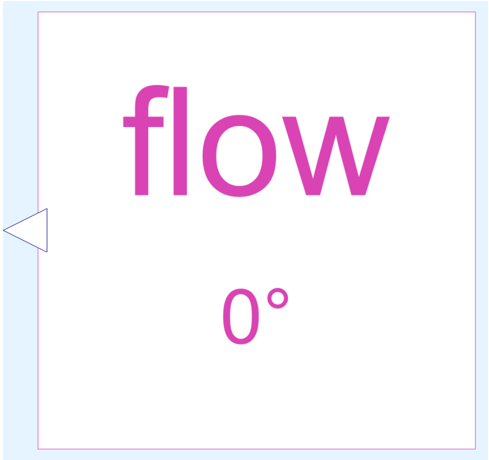


Figure 5-12: Injection flow function icon. The angle indicates that this function belongs to a control volume starting at a rotational angle of 0°.

large derivatives when an injection flow to a control volume is activated or deactivated, the transitions are defined to span over a short period of time instead of occurring instantly. In the model presented here, the transition time corresponds to the time it takes for a control volume to rotate about 2°. The output from the injection flow function, if the injection rate is set to 0.01 kg/s and θ_{inj} is set 360°, is shown in Figure 5-13. The Modelica code for the function is given in appendix A.8.

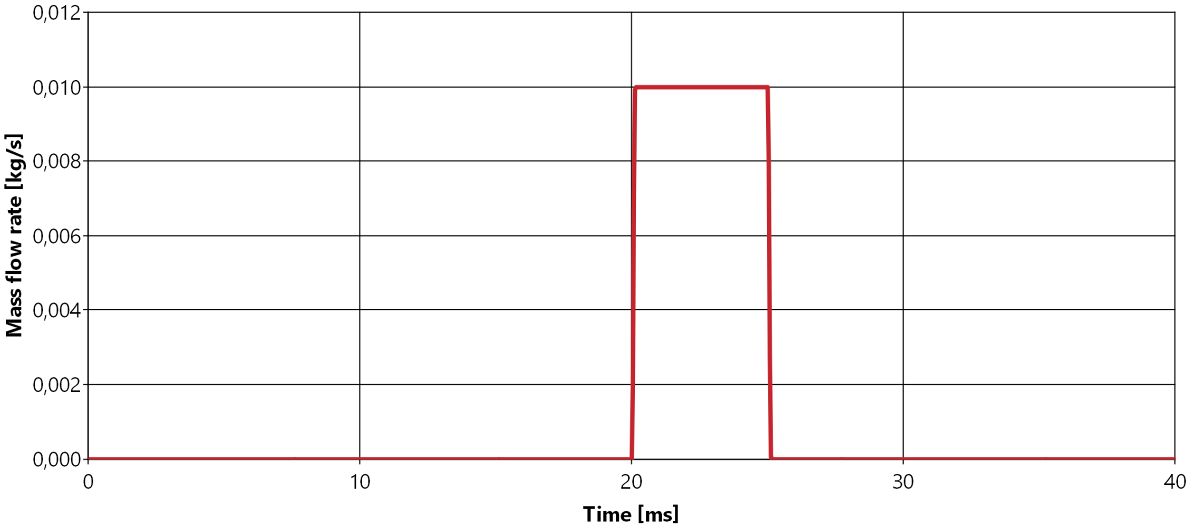
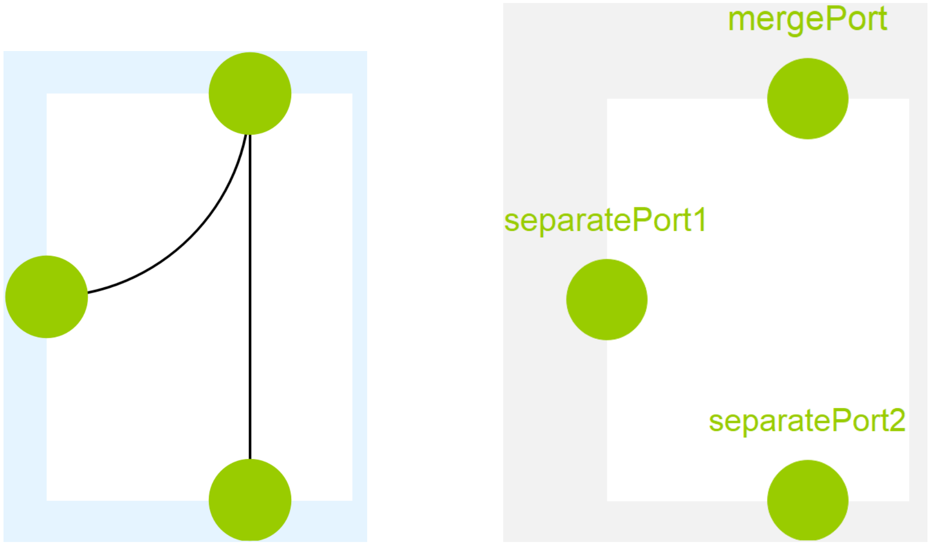


Figure 5-13: Injection mass flow rate as a function of time for a control volume starting at a rotational angle of 0° and the injection port located at a rotational angle of 360°.

5.1.4 Component 4 – The Junction

The last component of the screw compressor model is the junction. Junction components are placed in the model in order to merge two fluid flow paths into one, or to separate one fluid flow path into two. Each of the eight control volumes in the model will periodically go through a discharge phase where the working fluid flows out of the control volumes through their respective discharge paths. This means that eight different fluid flow paths must be connected to the screw compressor's single discharge port, and thus it is required to be able to merge the different paths into one single flow path. The junction component must be placed at each point where two flow paths are merged together, as the equations imposed by the component ensures conservation of mass and energy when the two flows are mixed together. Similarly, the junction component must be placed at each point where a single flow path separates into two paths. This type of separation is required at the suction end of the compressor, where the suction flow must be separated and guided to eight different control volumes. For the same reason it is also required to separate the compressor's injection flow into eight different injection paths. The component icon and the model diagram for the junction component is illustrated in Figure 5-14. The component is built up of three fluid port elements, representing the three points where the working fluid can enter or exit the component. The total fluid mass within the component is assumed to be constant, and the value is set to $m = 1 \cdot 10^{-8}$ kg. Since the mass is constant, the mass balance equation for the component is:



(a) Component icon.

(b) Model diagram.

Figure 5-14: The component icon and the model diagram for the junction component.

$$\sum \dot{m}_{port} = 0 \tag{5.10}$$

where $\dot{m}_{port}$ is the mass flow rate through each port. On the other hand, the mass fractions within the component can vary, depending on the composition of the fluid entering through each port. Thus the second mass balance equation is:

$$m\,\mathrm{der}(x_c) = \sum \dot{m}_{port} x_{c,port} \tag{5.11}$$

where x_c is the mass fraction of fluid constituent c within the junction, and $x_{c,port}$ is the mass fraction of the constituent in each inlet or outlet flow. Remember that $\dot{m}_{port} < 0$ if the flow direction through the given port is out of the component. Lastly, conservation of energy is applied through:

$$m\,\mathrm{der}(h) = \sum \dot{m}_{port} h_{port} \tag{5.12}$$

where h is the specific enthalpy within the junction, and h_{port} is the specific enthalpy of each inlet or outlet flow. It is assumed that there is no pressure drop across the component. Hence the pressure, p, is equal for the three ports. The Modelica code for the junction component is given in appendix A.9.

5.1.5 The Complete Model

Four different model components have been described above, namely the control volume, the flow restrictor, the injector and the junction. Moreover, multiple functions associated with the different components have been described, namely the volume function, the suction area function, the discharge area function, the leakage area function and the injection flow function. The model diagram for the screw compressor model, shown in Figure 5-15, illustrates how all these sub-models are connected in order to represent a complete screw compressor. Since there is a large number of components in the diagram, and structure of the model is rather complex, Figure 5-15 may be hard to read. For convenience, close-ups of the model diagram are provided in appendix B.

From the model diagram in Figure 5-15 (or Figures B-1 and B-2 in appendix B) we can see that the screw compressor has three connection points. The first connection point is the fluid port located at the left edge of the diagram. This represents the suction port of the compressor. The second connection point is the fluid port located at the right edge of the diagram, which represents the discharge port of the compressor. The third and final connection point is the fluid port located at the upper edge of the diagram, representing the compressor's injection port. The green lines in the model diagram represent the different flow paths for the working fluid, while the blue, red and pink lines represent input signals to the different components. In the middle of the model diagram, eight control volumes are placed side by side. Each control volume is connected to a blue volume function that calculates the control volume's position and cavity volume at each point in time. The 90° phase shifts between the control volumes are imposed by the defined starting angle for each volume function, which is set to 0°, 90°, 180° and so forth. Adjacent control volumes are connected through a flow restrictor placed between them. These flow restrictors represent the leakage paths between the control volumes, and they are all connected to a red leakage area function. Each leakage area function receives two input signals through its connections to the two volume functions for the associated control volumes, and calculates the effective leakage flow area based on those inputs. Furthermore, each control volume is connected to the compressor's suction port. The working fluid's path from the suction port to a control volume goes through one or multiple junctions, as well as a flow restrictor. The

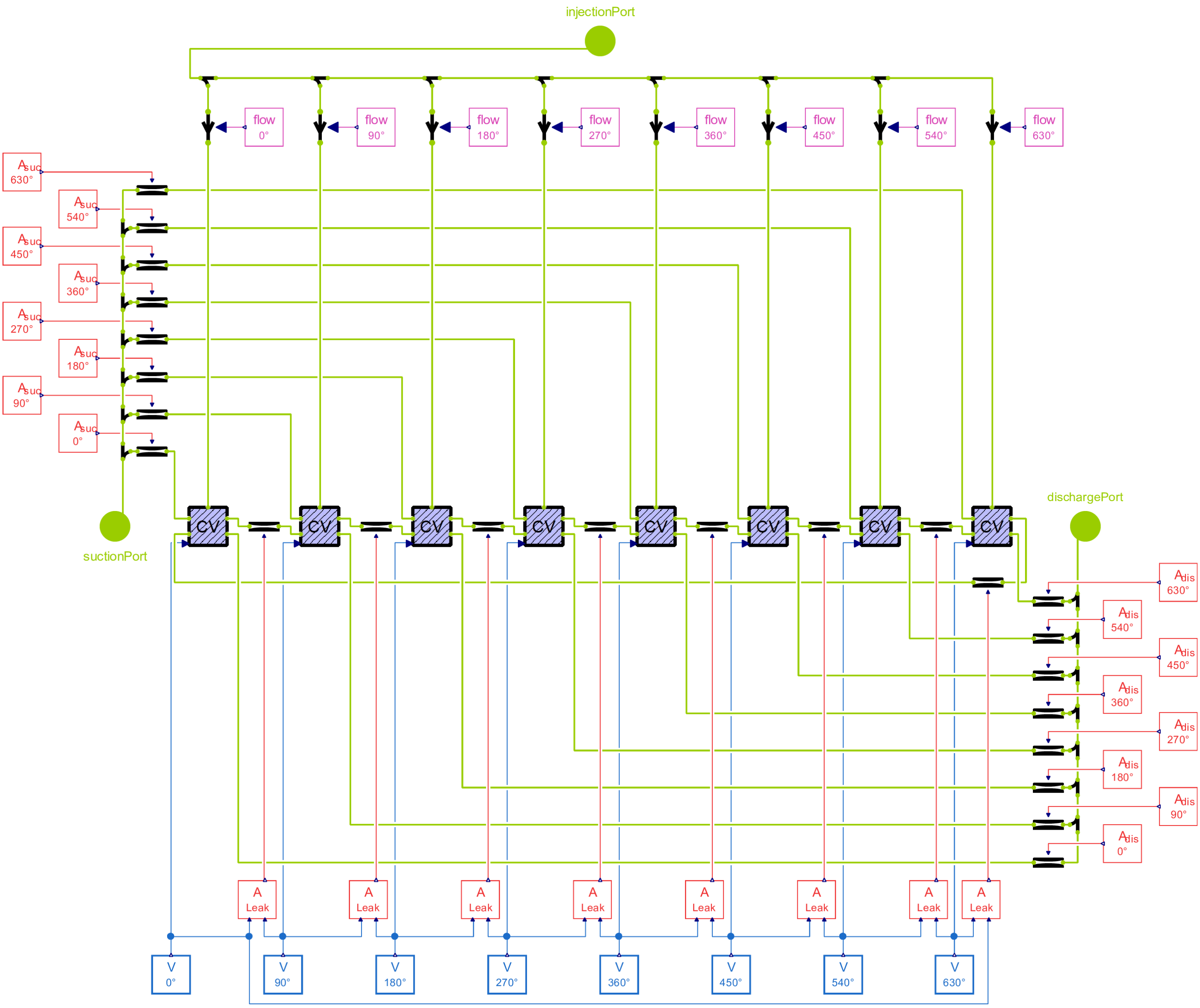


Figure 5-15: Model diagram for the screw compressor model. (Close-ups given in appendix B.)

flow restrictor for each suction path is connected to a red suction area function, which calculates the effective flow area of the suction path based on the position of the associated control volume. In order to obtain the correct position, the starting angle for each suction area function is set to the same value as the starting angle for the corresponding volume function. A similar model structure is shown on the right side of model diagram, where each control volume is connected to the compressor's discharge port through a flow restrictor and one or multiple junctions. The flow restrictor for each discharge path is connected to a red discharge area function, which calculates the effective flow area for the discharge path based on the position of the associated control volume. Lastly, each control volume is connected to the compressor's discharge port through an injector and one or multiple junctions. The injector for each injection path is connected to a pink injection flow function, which calculates the injection flow based on the position of the associated control volume.

When creating bigger system models, e.g., a model for simulation of the entire CACHP system, the screw compressor model can be implemented with a simple drag-and-drop procedure. The icon used to represent the screw compressor model is shown in Figure 5-16. The screw compressor's three connection points are shown on the icon, and the screw compressor can be connected to the rest of the system by drawing lines from these three points to the respective points for suction, discharge and injection in the system model.

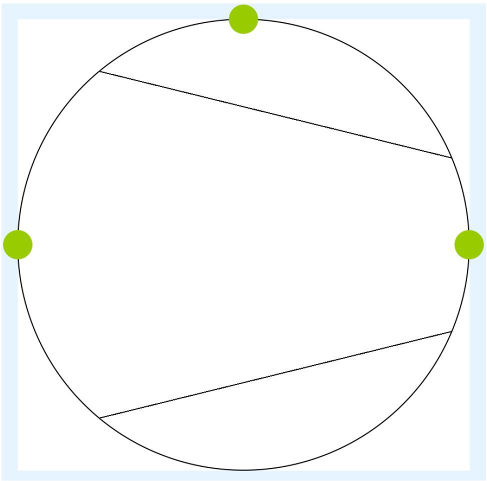


Figure 5-16: Screw compressor model icon.

The Modelica code for the screw compressor model is given in appendix A.10. The code is mainly a series of commands to create connections between the different model components, thus creating the complex model structure shown in Figure 5-15. However, some additional calculations for analysis purposes are included in the code. The throughput of the compressor, i.e., how much working fluid that is sucked into the compressor over a certain period of time, is calculated based on the mass flow rate at the compressor's suction port. The throughput, $m_{suc,tot}$, is equal to the integral of the mass flow rate, $\dot{m}_{suc}$. As integration is not a built-in operator in Dymola, but derivation is, integration is carried out be setting the time derivative of the throughput equal to the mass flow rate:

$$\mathrm{der}\left(m_{suc,tot}\right) = \dot{m}_{suc} \tag{5.13}$$

In a similar fashion, the total compression work, $W_{comp,tot}$, is calculated by setting the time derivative equal to the sum of the compression work rates for all the control volumes:

$$\mathrm{der}\left(W_{comp,tot}\right) = \sum_{cv=1}^{8} \dot{W}_{comp,cv} \tag{5.14}$$

where $\dot{W}_{comp,cv}$ is the instantaneous compression power for control volume cv, calculated from equation 5.4. Friction losses are not included in the calculations. $W_{comp,tot}$ represents the indicated compression work, i.e., the work load for the compressor with a mechanical efficiency of 100 %.

5.1.6 Input Parameters

Before running a simulation with the screw compressor model, values must be given for each of the models input parameters. The different input parameters are described below.

f: The first input parameter is the operational frequency of the compressor, i.e., the number of 360° rotations the male rotor completes during one second. The frequency determines the angular velocity, ω, for each control volume.

θ_{cycle}: The total rotation during one compression cycle, i.e., the rotational displacement from the beginning of the suction phase to the end of the discharge phase for each control volume. Together with the angular velocity, this gives the duration of each compression cycle, $t_{cycle} = \frac{\theta_{cycle}}{\omega}$.

V_{max}: The maximum cavity volume. This input parameter defines the peak value for the volume curves displayed in Figure 5-4, and it represents the cavity volume at the beginning the compression phase.

v_i: The compressor's built-in volume ratio. This ratio is utilized to calculate at which point each control volume's discharge path must open. The compression phase ends and discharge path opens when a control volume's cavity volume reaches a value equal to V_{max}/v_i.

C_{leak}: The leakage flow coefficient. The proportionality constant used in equation 5.9 to calculate the effective flow area for each leakage path.

A_{suc}: The maximum effective flow area for each suction path. This parameter is the peak value for the suction area curves shown in Figure 5-8.

A_{dis}: The maximum effective flow area for each discharge path. This parameter is the peak value for the discharge area curves shown in Figure 5-8.

θ_{inj}: The injection angle, i.e., the rotational angle where injection begins for each control volume. Fluid is injected into a control volume when its rotational angle is between θ_{inj} and $\theta_{inj} + 90°$.

$\dot{m}_{inj}$: The injection rate, i.e., the mass flow rate in each injection path when the path is active. This input parameter defines the peak value for the injection flow curve shown in Figure 5-13.

α: The coefficient of heat transfer between the working fluid and the compressor. This coefficient is required in equation 5.5 to calculate the heat loss rate.

T_{bod}: The average temperature of the compressor body, required in equation 5.5 to calculate the heat loss rate.

5.1.7 Initialization

In addition to the input parameters described above, a set of start values must be predefined in order to initialize a simulation properly. At each simulation time step, the thermodynamic properties of the working fluid within each control volume is calculated with the TILMedia fluid property procedure. The procedure requires values for the pressure, the specific enthalpy and the mass fractions as input to calculate the thermodynamic state. At the very first simulation step, the Dymola solver is not capable of guessing appropriate values for these variables by itself. Without any guidance towards the correct initial thermodynamic state within each chamber, the solver is unable to find a valid state and a simulation error occurs straight away. Therefore, the thermodynamic state at $t = 0$ must established in advance for each control volume. Three start values must be given for each control volume, namely $p_{start,cv}$, $h_{start,cv}$ and $x_{start,cv}$. As indicated by the model diagram in Figure 5-15, four of the eight control volumes are in the suction phase at the beginning of a simulation, with starting rotational angles set to 0°, 90°, 180° and 270°, while the compression phase begins at 360°. The thermodynamic state within these four control volumes are given by the state of the fluid in the suction line connected to the compressor, and the state at $t = 0$ is equal in all four control volumes. The start values for these control volumes are denoted $p_{start,suc}$, $h_{start,suc}$ and $x_{start,suc}$. On the contrary, the initial state of the four remaining control volumes with starting rotational angles of 360°, 450°, 540° and 630° are not given by the state of the suction fluid. Therefore, an estimate must be made for the start values. The estimate can be based on the fact that the pressure and the specific enthalpy increases during compression, which means that $p_{start,cv}$ and $h_{start,cv}$ should be set to higher values for a control volume starting at a higher rotational angle. Furthermore, estimates for the mass fractions, $x_{start,cv}$, can be made based on the fact that the fluid composition will remain nearly constant under normal operating conditions where the injection mass flow rate is much lower than the suction mass flow rate. When setting up a simulation, it is not essential that the estimates are accurate. This is because a preliminary simulation can be run to calculate more accurate start values. By considering that the thermodynamic process repeats itself every t_{cycle} milliseconds, a preliminary simulation of $N \cdot t_{cycle}$ milliseconds, where N is an integer, should be carried out. After $N \cdot t_{cycle}$ milliseconds, the control volumes will be in the exact same position as they were when they started. Hence the thermodynamic state within each control volume at the end of a preliminary simulation will represent the correct start values for the following simulation. Here $N = 1$ is sufficient if the start value estimates are relatively accurate, while a preliminary simulation of multiple cycles may be required if the estimates are poor.

In addition to the start values for each control volume component, a set of start values is required for the junction components. The time derivatives of the mass fractions and the specific enthalpy in each junction are obtained through equations 5.11 and 5.12. These equation are not resolvable without given initial values for x_c and h. Start values for the junctions that are located on the suction side of the compressor are given by the state of the fluid in the connected suction line, $x_{start,suc}$ and $h_{start,suc}$. Start values for the junctions that are connected to the compressor's injection port are given by the state of the fluid in the injection line, $x_{start,inj}$ and $h_{start,inj}$. Start values for the junctions located at the discharge side of the compressor, $x_{start,dis}$ and $h_{start,dis}$, must be set based on an estimated discharge state. The estimation does not have to be accurate, as the thermodynamic variables adjusts rapidly due to the low fluid mass ($m = 1 \cdot 10^{-8}$ kg) within each junction.

5.1.8 Simulation Setup

The developed screw compressor model can be utilized as a stand-alone model, or it can be incorporated into a bigger system model. In order to demonstrate some of the basic analysis capabilities of the developed model, simulations with a stand-alone setup are to be carried out. The simulation setup is illustrated in 5-17. The compressor is connected to three so-called fluid boundaries. The boundaries represents sources and sinks, where an unlimited amount of working fluid can flow out of a source or into a sink. Each boundary must be given three parameters, namely p, T and x. p represents the pressure at the point of the boundary. T represents the temperature of any working fluid that flows out of the boundary. x represents the ammonia mass fraction of any working fluid that flows out of the boundary, as a mixture of ammonia and water is the working fluid under consideration here.

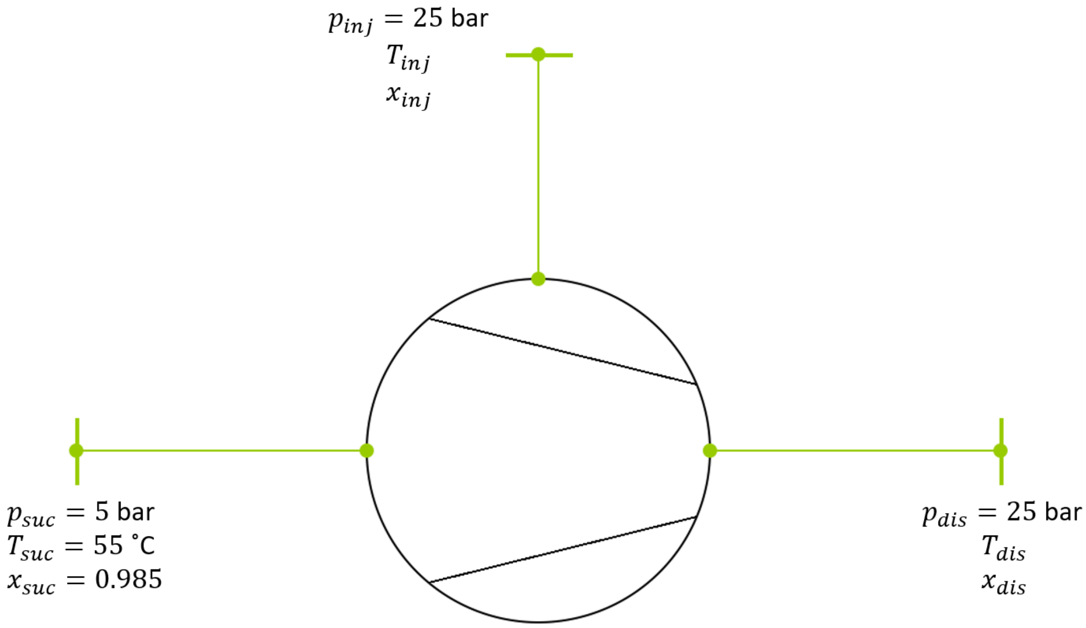


Figure 5-17: Simulation setup.

The simulated operating conditions are aimed towards the expected operating conditions for the screw compressor in the CACHP test rig. It is assumed that the desorber and the absorber pressures are 5 bar and 25 bar respectively. This corresponds to the compressor's suction and discharge pressures, giving an external pressure ratio of $\pi = 5$. In the current CACHP configuration, the ammonia mass fraction at the inlet of the desorber is set to 0.7, i.e., a mixture of 70 % ammonia and 30 % water. The vapor and the liquid at the outlet of the desorber is assumed to have a temperature of 55 °C. In a vapor-liquid equilibrium state at a pressure of 5 bar and a temperature of 55 °C, the ammonia mass fraction of the vapor phase is 0.985, and the ammonia mass fraction of the liquid phase is 0.4. The vapor and the liquid are separated after the desorber, and the vapor is drawn towards the compressor, while the liquid is drawn towards the solution pump. The state of the vapor at the outlet of the desorber is applied to the suction boundary in the simulation model, as illustrated in Figure 5-17. Moreover, the pressure for the injection boundary is set to 25 bar, equal to the pressure on the high-pressure side of the solution pump. The two other injection boundary parameters, T_{inj} and x_{inj}, are not given. Simulations with different injection temperatures and mass fractions are to be carried out. Lastly, the pressure for the discharge boundary is set to 25 bar. In most cases the values for T_{dis} and x_{dis} are trivial, since the working fluid only flows into the discharge boundary. However, T_{dis} and x_{dis} are of significance in cases where under-compression occurs. Under-compression causes a temporarily reversed flow direction, where the working fluid flows out of the discharge boundary and into the compressor's discharge port. In such cases preliminary simulations are carried out in order to determine the compressor's discharge temperature and adjust the boundary temperature accordingly.

A series of simulation results are to be presented in the section below. Some of the screw compressor model's input parameters are to be varied, in order to analyse the effects of injection flow rates, leakage flow rates, heat loss rates and more. However, the input parameters that relates to the general geometry of the compressor will not be varied. The rotational speed of the compressor will also be kept constant during all simulations. The values applied to the unvaried input parameters are given in Table 1. These values do not represent the exact geometry of the compressor under consideration, but the values are established based on Mayekawa (2015) and Bommel (2016).

Table 1: Input parameter values.

Input parameter	f	θ_{cycle}	V_{max}	v_i	A_{suc}	A_{dis}
Value	50 Hz	720°	335 cm^3	3.65	50 cm^2	10 cm^2

5.2 Simulation Results

5.2.1 Integration Method

When summing up the equations for every single element of the simulation model, it comprises a set of 5343 equations. The differential equations that are part of this equation set must be solved through numerical integration. The simulation result can be affected be the choice of integration method, integration step size and error tolerance. A simulation can produce invalid or highly inaccurate results if the chosen integration settings are unsuitable for the model. Dymola offers a wide range of built-in integration methods, from the basic Euler method to more advanced purpose-built methods. Simulating with different integration methods is a fast and easy way to run a quality check on a model. A simulation result is more likely to be valid if it same result can be produced with different integration methods. For that reason, test simulations for the developed screw compressor model with various integration methods are carried out. The standard integration method in Dymola is named DASSL. This method is relatively flexible and thus suitable for a wide range of model types. Nevertheless, the equations in the screw compressor model represent a rather stiff equation set, i.e., a set of differential equations that is challenging to solve numerically due to rapidly changing derivatives or stability issues. Dymola has multiple built-in integration methods that are especially suited for stiff models. One of these methods is named ESDIRK23A. The pressure curves obtained from two separate simulations, one solved with DASSL and one solved with ESDIRK23A, are shown in Figure 5-18. The results are nearly identical with DASSL and ESDIRK23A. Figure 5-18 only shows the pressure, but identical results were obtained for other variables such as temperature, mass flow rate and compression work as well. Besides, identical results were obtained with several other integration methods that are not presented here. Being able to obtain uniform results with different integration methods indicates that the developed screw compressor model is sound and stable.

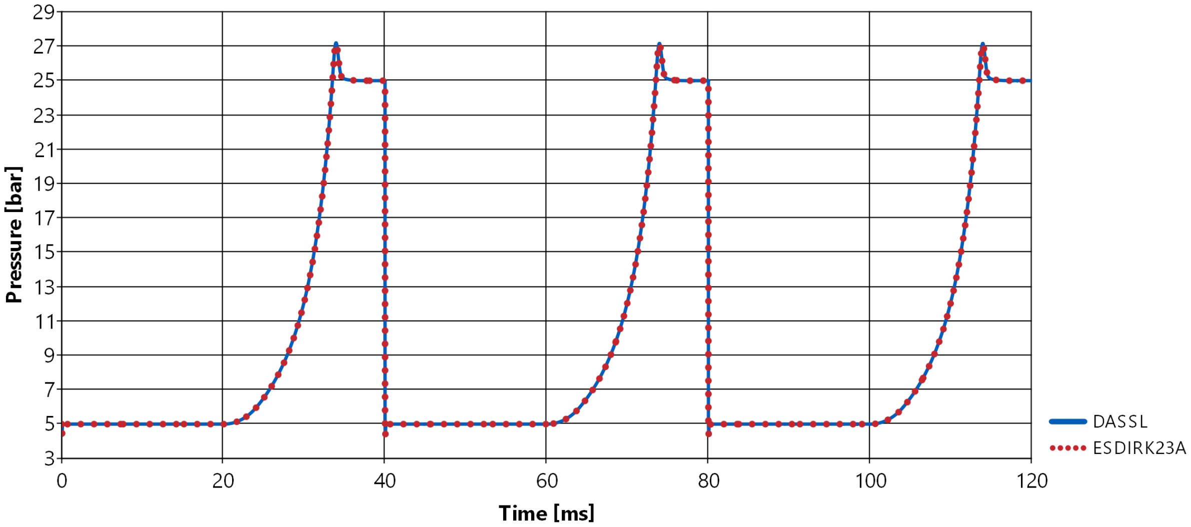


Figure 5-18: Pressure as a function of time for control volume starting at a rotational angle of 0°. Solved with two different integration methods, DASSL and ESDIRK23A.

Simulations are slower when the model is solved with ESDIRK23A compared to DASSL, giving nearly a 50 % increase in computational time. However, ESDIRK23A proves to be a more robust method than DASSL, as the simulation results are less affected by factors such as integration step size and error tolerance. ESDIRK23A is therefore considered to be the most suitable integration method. All simulation results presented further are obtained using

ESDIRK23A. The number of integration steps are set to 1000 per compression cycle. With $t_{cycle} = 40$ ms, this gives an average time step of 0.04 ms. The step size is not constant, as the ESDIRK23A solver adjusts the step size continuously depending on the stiffness of the equation set in each step.

The pressure curve in Figure 5-18 illustrates how the compression cycle repeats itself for every control volume. When a control volume reaches the end of the discharge phase, the pressure is reset to 5 bar and the cycle starts over. There is a small pressure drop during the very first millisecond in each cycle. This occurs because the cavity volume begins to expand before the suction path is fully open. The time it takes for the suction path to open is demonstrated by the suction flow area curves in Figure 5-8. Since the pressure only drops about 0.5 bar and it instantly rises again, it has a negligible impact on the performance of the compressor. The pressure drop would likely be even smaller if every type of leakage path was implemented in the model, such as leakage along the suction end of the rotors (path 4 in Figure 4-2). The suction phase ends after 20 ms in each cycle, and the suction flow rate into the control volume is zero during the compression and discharge phase. However, the suction flow rate into the compressor itself never becomes zero due to the phase shifts between the eight control volumes in the model. Four of the eight control volumes are in the suction phase at all times, and this creates a relatively steady suction flow rate into the compressor. This also applies to the compressor's discharge flow rate, but the discharge flow varies more due to the fact that a maximum of two control volumes are in the discharge phase at the same time. Moreover, relatively large pulsations in the discharge flow occur if there is under- or over-compression. With the cavity volume defined as shown in Figure 5-4 and the built-in volume ratio set to 3.65, the discharge path must open when a control volume reaches a rotational angle of 605°. The short period of time where two control volumes are in the discharge phase at once is shown by the discharge flow area curves in Figure 5-8.

5.2.2 Heat Loss

As discussed in section 4.3.5, heat loss to the compressor casing and the ambient is often considered negligible in the literature. In order to investigate if this is reasonable, a simple heat loss calculation has been implemented in the screw compressor model through equation 5.5. This calculation requires a heat transfer coefficient, α, and an average compressor body temperature, T_{bod}. Investigations on the effect of heat transfer coefficient are carried out by assuming an average compressor body temperature of 80 °C. An ideal case without internal leakage is studied (C_{leak} set to zero), as well as no liquid injection ($\dot{m}_{inj}$ set to zero). The simulation results in terms of working fluid temperature are shown in Figure 5-19.

The results show that the effect of heat transfer between the working fluid and the compressor casing is minimal, even with an unreasonably high value for α such as 5000 W/m^{2}K. The working fluid is heated up slightly towards the end of the suction phase, and there is a slight

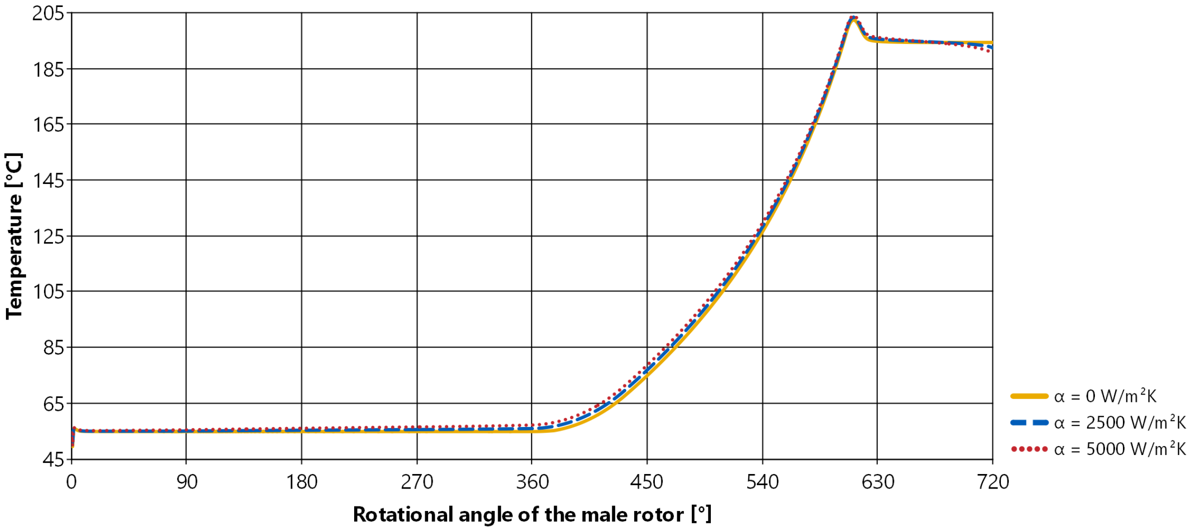


Figure 5-19: Temperature of the working fluid. Simulations with different values for the heat transfer coefficient, α.

cooling effect on the last portion of the working fluid during discharge, but apart from that, the temperature curves are nearly indistinguishable. Equal results are also obtained regarding pressure and compression work. Furthermore, the temperature profile in a real screw compressor is not uniform during operation. The temperature of the compressor body is higher at the discharge end of the compressor compared to the suction end (Stosic et al., 2005b). This means that the temperature difference between the working fluid and the compressor body will be smaller. Thus, there will be less heat transfer than what is simulated with an average body temperature here. This strongly suggests that heat loss to the compressor casing can be neglected. In the simulations presented below, α is set to zero and heat loss is not included.

5.2.3 Leakage Flow Coefficient

Internal leakage flows have been proven to strongly affect the performance of screw compressors (Fleming and Tang, 1995; Zaytsev, 2003). It is therefore of high importance to apply a suitable value for the leakage flow coefficient, C_{leak}. Simulations are carried out with different values for C_{leak}, and the results are presented in Figures 5-20, 5-21 and 5-22. These investigations does not include liquid injection, i.e., $\dot{m}_{inj}$ is set to zero.

The temperature evolution is shown in Figure 5-20. In the ideal case with zero leakage, the temperature of the working fluid is 203 °C at the point where the discharge port opens. It is shown that larger leakage gaps cause an increase in the discharge temperature. With the leakage flow coefficient set to 0.1 the temperature reaches 243 °C at its maximum. The temperature increase is generated by hot vapor leaking back towards the suction end.

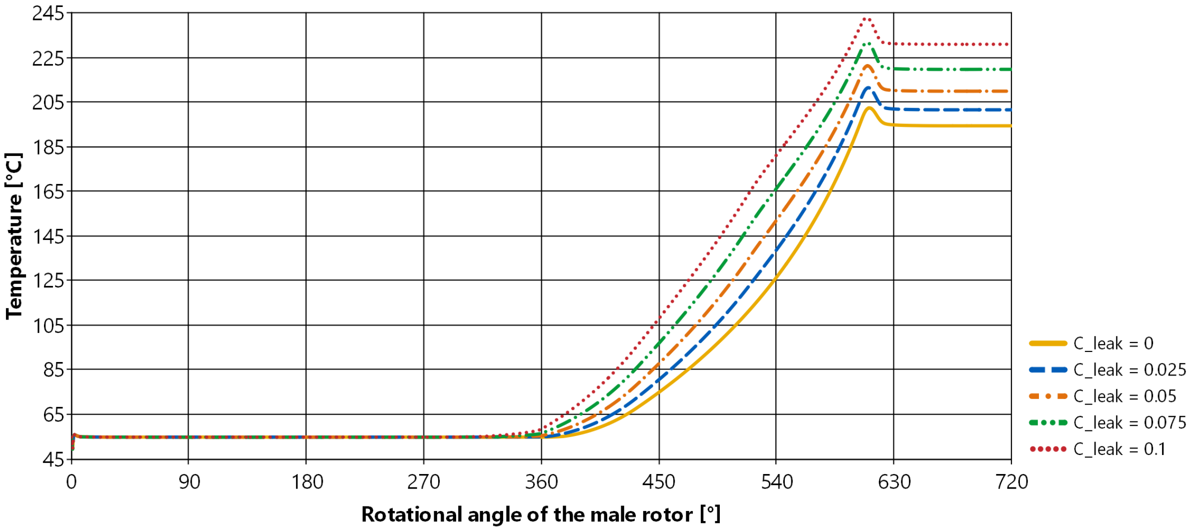


Figure 5-20: Temperature evolution with different leakage flow coefficients, C_{leak}.

A pressure-volume diagram is displayed in Figure 5-21. The diagram shows that internal leakage flows accelerate the pressure evolution. E.g., at a cavity volume of 200 cm^3 the pressure is 9.6 bar in the ideal case, while the pressure has already reached 12.3 bar in the case with $C_{leak} = 0.1$. Additionally, it leads to a higher degree of over-compression. In the ideal case the maximum pressure is 27.1 bar, while the maximum pressure is 28.1 bar with $C_{leak} = 0.1$. Keeping in mind that the amount of compression work equals the area under the curve in the pressure-volume diagram, it is demonstrated that leakage flows increase the work load. With $C_{leak} = 0$ the compressor's work load per compression cycle is calculated to be 2598 J, and with $C_{leak} = 0.1$ the work load per compression cycle is 3019 J. At an operational frequency of 50 Hz this corresponds to indicated powers of 65 kW and 75.5 kW respectively.

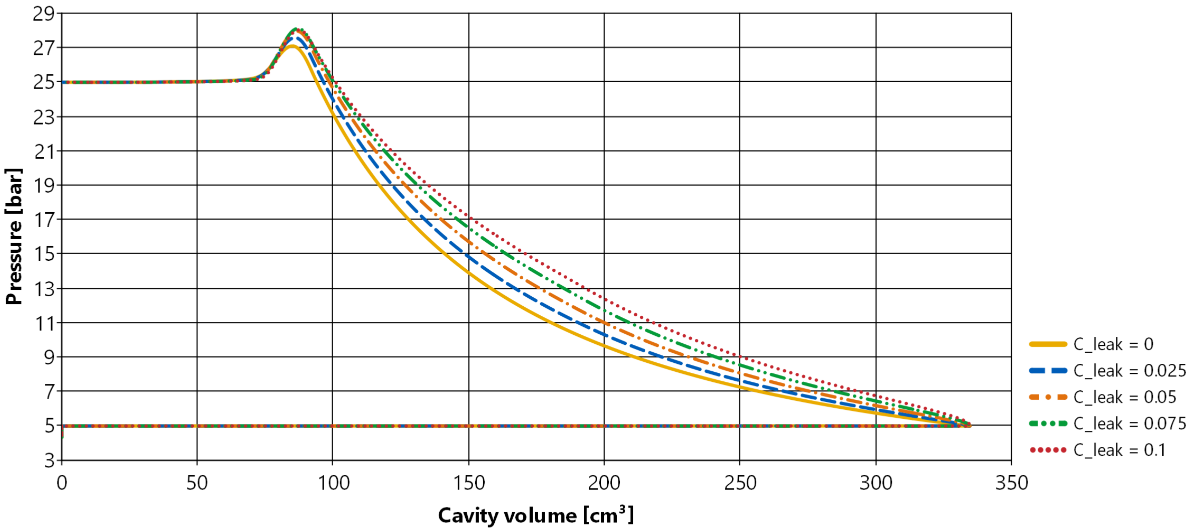


Figure 5-21: Pressure-volume diagram with different leakage flow coefficients, C_{leak}.

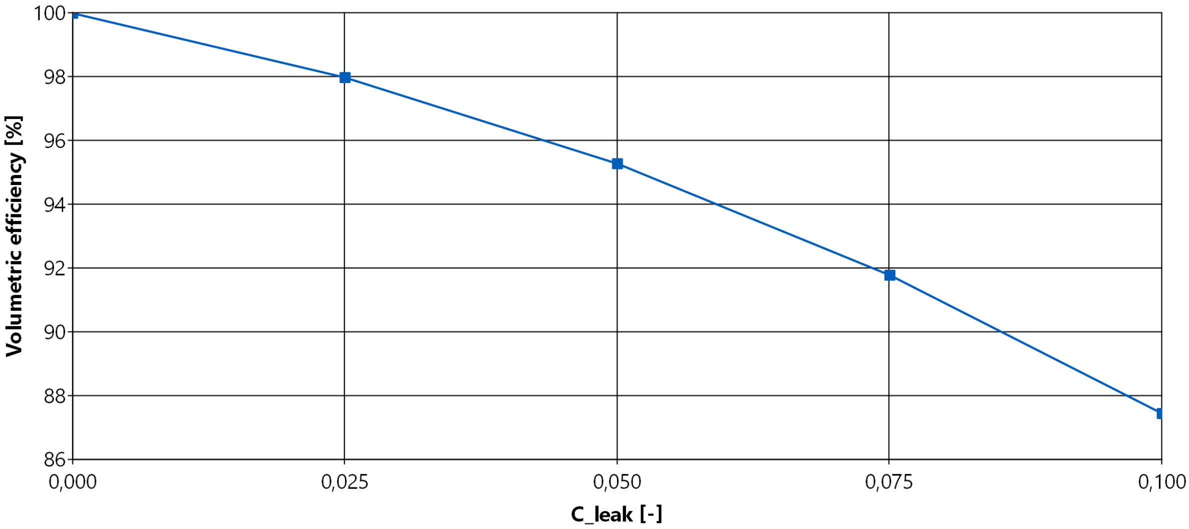


Figure 5-22: Volumetric efficiency as a function of the leakage flow coefficient, C_{leak}.

The vapor that leaks into a control volume during its suction phase occupies a part of the volume and thereby reduces the amount of vapor that is sucked into the control volume. The effect the leakage flow coefficient has on the compressor's volumetric efficiency is investigated by comparing the throughput of the compressor at different values for C_{leak} with the throughput of the compressor in the ideal case. The results are shown in Figure 5-22. In order to determine a suitable value for C_{leak}, the results are evaluated against the results of Fleming and Tang (1995). In their study on different leakage paths in screw compressors, they found that the leakage path between the rotor tips and the compressor casing contributed to a 2 – 3 % decrease in volumetric efficiency compared to a zero-leakage scenario. A matching result can be obtained with the screw compressor developed here by giving C_{leak} a value between 0.025 and 0.035. However, to compensate for the additional loss in volumetric efficiency that would be caused by other leakage paths than the one included in the model, it is considered suitable to apply a value of 0.05 for the leakage flow coefficient. This gives a volumetric efficiency of about 95 % relative to the ideal case with zero leakage. C_{leak} is set to 0.05 in the investigations on liquid injection presented below.

5.2.4 Composition of the Injection Liquid

In a CACHP the composition of the working fluid is not constant throughout the heat pump cycle. This means there are multiple possibilities in regard to the composition of the liquid that is to be injected into the compressor. The compressor's liquid injection line can be connected to the liquid solution circuit at a point where the solution is lean, to apply a relatively low ammonia mass fraction, or at a point where the solution is rich, to apply a relatively high ammonia mass fraction. Moreover, in theory any ammonia mass fraction between 0 and 1 can be obtained by adding additional components for separation of ammonia and water to the heat pump cycle. In order to evaluate which liquid composition that is most favourable, simulations are carried out with different values for the injection boundary's ammonia mass fraction, x_{inj}. The temperature for the injection boundary, T_{inj}, is set to 55 °C, equal to the assumed desorber outlet temperature. In a similar fashion as Tian et al. (2017a), the liquid is injected at the very beginning of the compression phase. The injection angle, θ_{inj}, is set to 360°. The injection mass flow rate, $\dot{m}_{inj}$, is set to 0.01 kg/s, which corresponds to about 5 % of the compressor's suction mass flow rate. The simulation results are shown in Figures 5-23, 5-24 and 5-25.

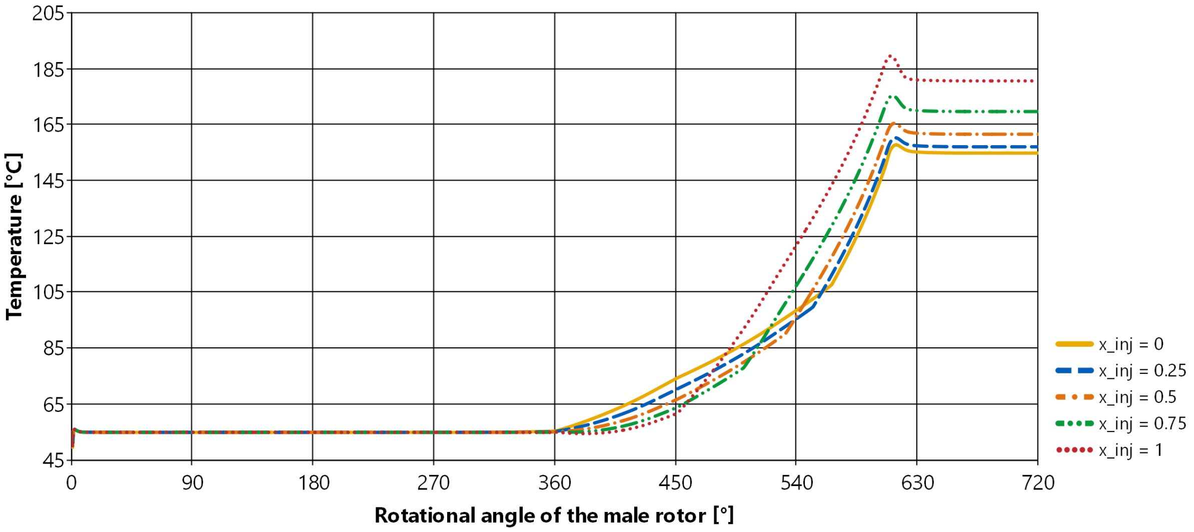


Figure 5-23: Temperature evolution with different injection liquid compositions. x_{inj} denotes the ammonia mass fraction of the injection liquid.

The temperature evolution curves in Figure 5-23 show that the lowest discharge temperature is obtained when the injection liquid is pure water ($x_{inj} = 0$), while the highest discharge temperature is obtained with pure ammonia ($x_{inj} = 1$). There are similarities in the characteristics of the different temperature curves. During the first part of the compression phase, there is a relatively slow temperature increase. At a certain point the gradient of the curve suddenly changes and the temperature shoots upwards. This gradient change occurs at an earlier stage in cases where the ammonia mass fraction of the injection liquid is high, and this causes a higher discharge temperature. The temperature gradients can be seen in relation to the vapor quality at each point in time. The vapor quality evolution is presented in Figure 5-24. The vapor quality drops into the two-phase region when liquid is injected into the the control volume, but it immediately rises again when the fluid is compressed further. With injection of pure water the fluid stays in the two-phase region for a longer period of time. The temperature gradient increase occurs as soon as the fluid reaches the superheated vapor region. Furthermore, it is shown in Figure 5-23 that the temperature during the first part of the compression phase is higher with injection of pure water compared to injection of pure ammonia. The reason for this is that pure water has a much higher saturation temperature than pure ammonia at the same pressure. E.g., at a pressure of 10 bar, pure water has a saturation temperature of 180 °C, while pure ammonia has a saturation temperature of 25 °C. Thus, a higher water fraction in the injection liquid leads to a higher temperature when the fluid is in the two-phase region.

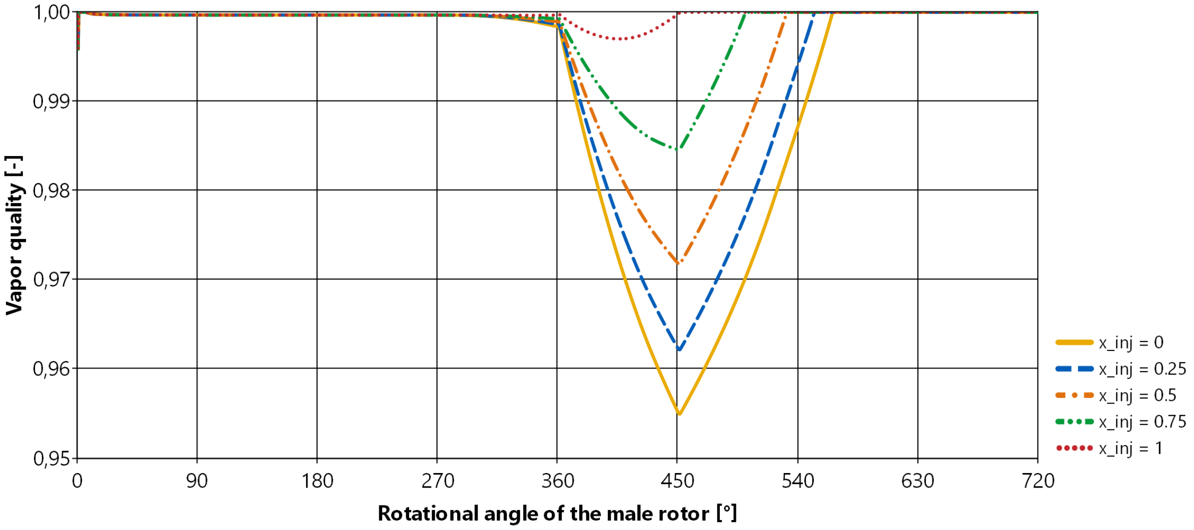


Figure 5-24: Vapor quality evolution with different injection liquid compositions. x_{inj} denotes the ammonia mass fraction of the injection liquid.

The pressure-volume diagram for different compositions of the injection liquid is shown in Figure 5-25. A relatively high degree of over-compression is achieved with injection of pure ammonia. A more favourable result, with very little over-compression, is achieved when the ammonia mass fraction of the injection liquid is set to a value in the range 0 to 0.5. Based on the area under the pressure-volume curves, the compression work load is at its lowest when the injection liquid's ammonia mass fraction is low.

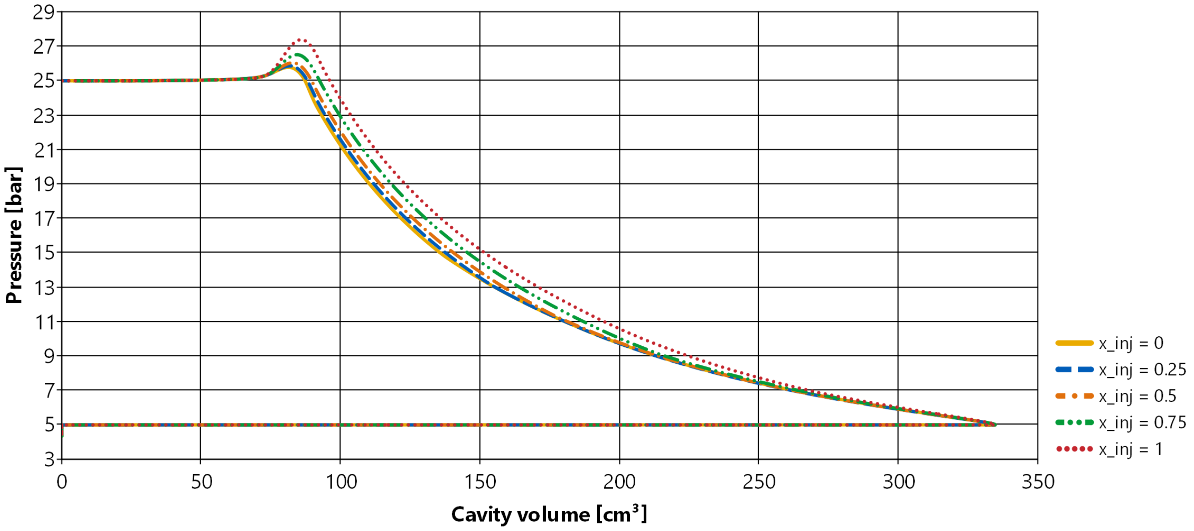


Figure 5-25: Pressure-volume diagram with different injection liquid compositions. x_{inj} denotes the ammonia mass fraction of the injection liquid.

5.2.5 Temperature of the Injection Liquid

Simulations are also carried out to investigate if the temperature of the injection liquid has a significant impact on the compressor's performance. Since injection of pure water is found to be most favourable option in the previous section, the ammonia mass fraction for the injection boundary, x_{inj}, is set to 0. The injection angle, θ_{inj}, is set to 360°. The temperature at the injection boundary, T_{inj}, is varied from 40 °C to 80 °C. Simulation results regarding temperature evolution are shown in Figure 5-26. The results are close to identical for the different injection temperatures. The magnitude of the cooling effect is mainly related to the latent heat of the injected water, i.e., the amount of heat required to evaporate the water. Decreasing the temperature of the water only results in a slight reduction in sensible heat. It does not affect the amount of latent heat. The simulation results show that an increase in injection temperature of 10 °C generates an increase in discharge temperature of only 1.2 °C. Furthermore, results concerning pressure evolution or compression work load are close to unaffected by changes in injection temperature.

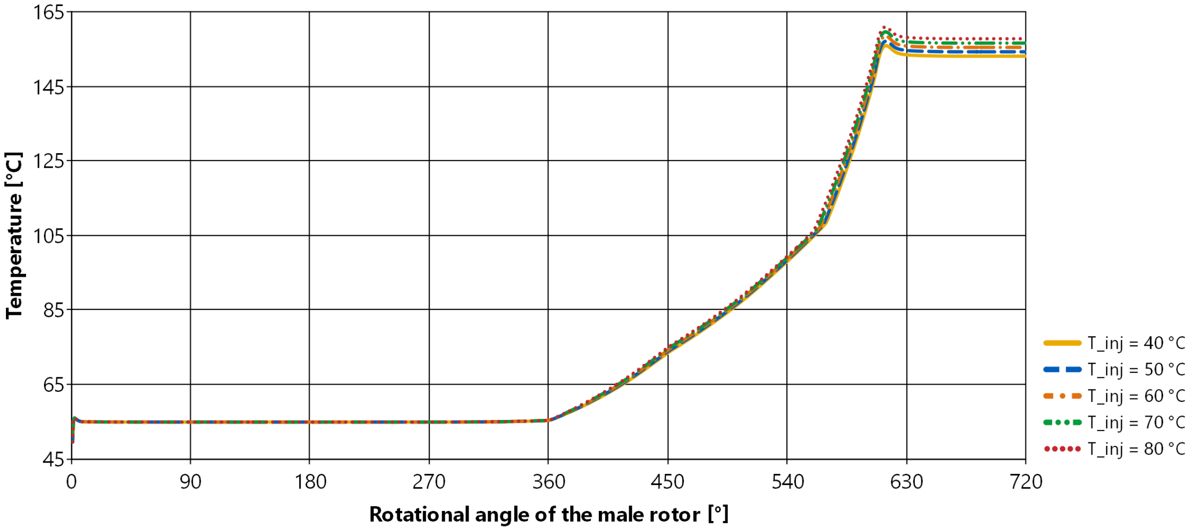


Figure 5-26: Temperature evolution with different temperatures of the injection liquid, T_{inj}.

5.2.6 Injection Flow Rate

The results presented in section 5.2.4 demonstrate that it is beneficial to inject liquid with a relatively low ammonia mass fraction. For the CACHP cycle, this means that the compressor's injection line should be connected to the liquid solution circuit at a point where the solution is lean. Looking back at the CACHP schematic in Figure 2-3, the lean solution is represented by states 3, 4 and 5. To be able to inject fluid into the compressor, the pressure in the injection line must be higher than the pressure in the compressor at the location of the injection port. Thus state 3 (the low-pressure side of the solution pump) can be ruled out as an option. The injection line must either be connected to state 4 or state 5. Due to the internal heat exchanger in the solution circuit, the temperature is higher in state 5 than in state 4. The results presented in section 5.2.5 demonstrate that it is slightly beneficial with a lower temperature. Based on this, state 4 (the high-pressure side of the solution pump) is considered the most suitable connection point for the compressor's injection line in the CACHP system.

Under the assumption that the temperature at the outlet of the desorber is 55 °C, the ammonia mass fraction of the lean solution is 0.4. This value is applied to the injection boundary's mass fraction, x_{inj}. Moreover, it is assumed that there is a small temperature rise across the solution pump, and thus the temperature for the injection boundary, T_{inj}, is set to 60 °C. Simulations are carried out to find the optimal injection mass flow rate. The results are presented in Figures 5-27, 5-28, 5-29 and 5-30. The injection angle, θ_{inj}, is set to 360° in these simulations.

The temperature evolution with injection flow rates ranging from 0 kg/s to 0.02 kg/s is shown in Figure 5-27. The results demonstrate that the injection rate strongly affects the discharge temperature. With zero injection the discharge temperature is 210 °C. This temperature is reduced all the way down to 125 °C with an injection rate of 0.02 kg/s. At a discharge temperature of 125 °C, the working fluid is in the two-phase region during discharge. This is shown by the vapor quality evolution in Figure 5-28. The heating capacity of the absorber in the CACHP system is based on the amount of latent heat that is released when vapor is absorbed into liquid. It is therefore of importance to keep the vapor quality of the compressor's discharge fluid at high level. With an injection rate of 0.015 kg/s, the working fluid reaches the superheated vapor region at the very end of the compression phase, and the discharge temperature is 140 °C.

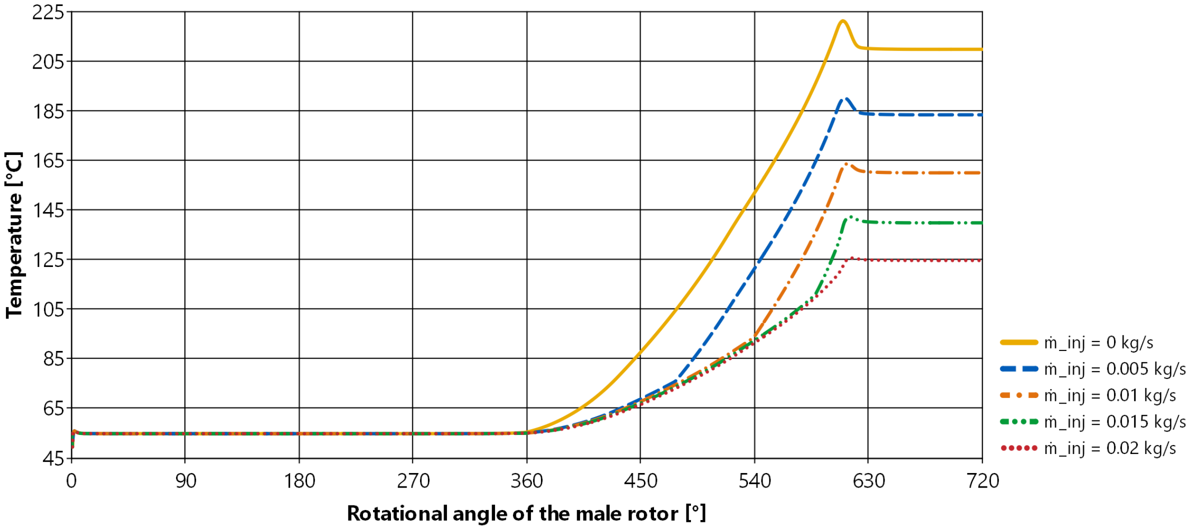


Figure 5-27: Temperature evolution with different injection flow rates, $\dot{m}_{inj}$.

The pressure-volume diagram for different injection flow rates is presented in Figure 5-29. Here it is shown that a higher injection flow rate results in lower pressure and less compression work. Over-compression occurs in the two cases where the injection rate is set to 0 kg/s and 0.005 kg/s. On the contrary, under-compression occurs in the two cases where the injection rate

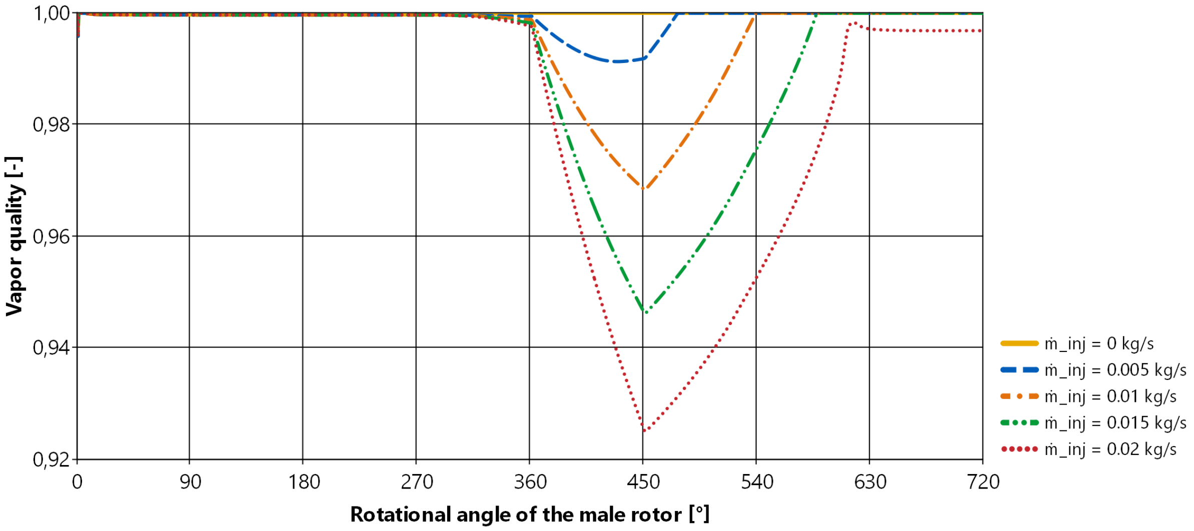


Figure 5-28: Vapor quality evolution with different injection flow rates, $\dot{m}_{inj}$.

is set to 0.015 kg/s and 0.02 kg/s. The effects of under- and over-compression can be studied closer by looking at the discharge flow rate. The flow rate out of each control volume's discharge port, as it rotates from 600° to 720°, is shown in Figure 5-30. The discharge port opens when a control volumes reaches an angle of 605°. If the pressure in the control volume is lower than the pressure at the discharge boundary, fluid flows from the discharge boundary back into the control volume. This type of backflow is represented by negative mass flow rates in Figure 5-30. A relatively large backflow occurs with an injection rate of 0.02 kg/s. In this case the discharge flow rate is -0.157 kg/s at its minimum. The backflow contributes to a rapid pressure increase in the control volume, and at a rotational angle of 612° the pressure in the control volume surpasses the discharge pressure, causing the discharge flow rate to become positive. In order to avoid any

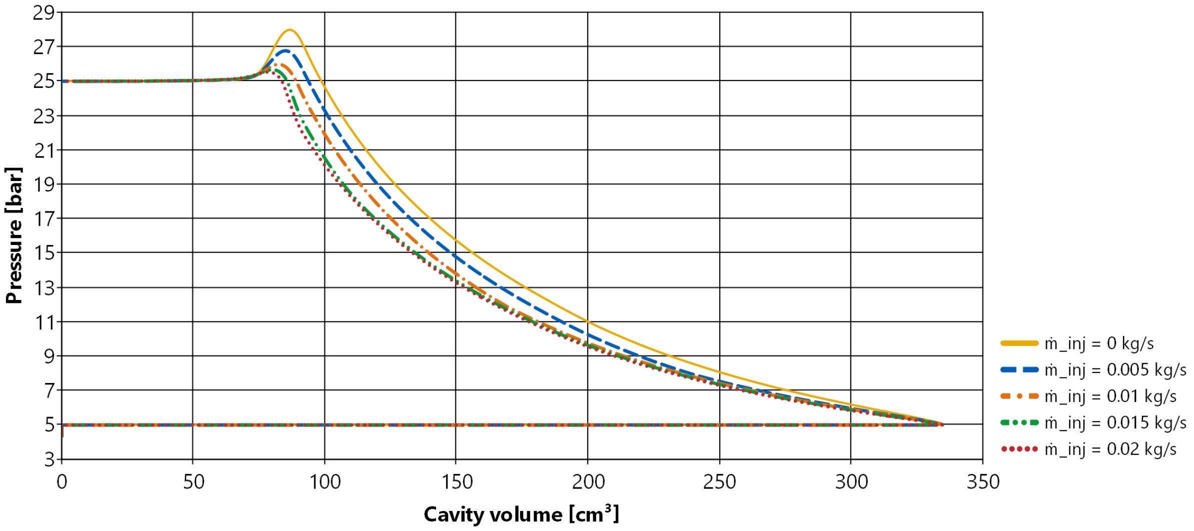


Figure 5-29: Pressure-volume diagram with different injection flow rates, $\dot{m}_{inj}$.

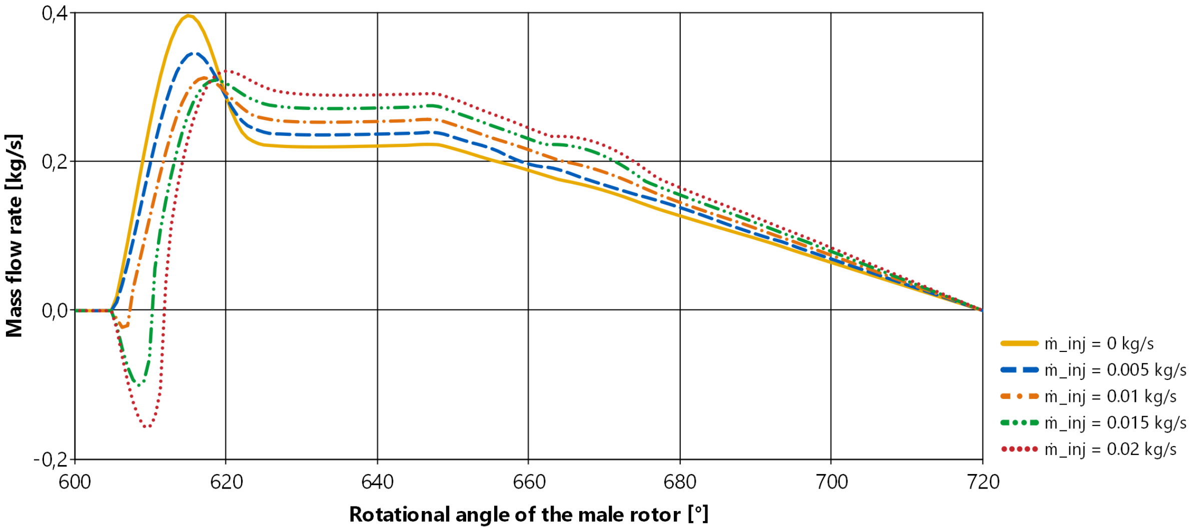


Figure 5-30: Discharge flow rate evolution with different injection flow rates, $\dot{m}_{inj}$.

under- or over-compression, the applied injection flow rate must be approximately 0.01 kg/s. With an injection rate of 0.01 kg/s, the compressor's pressure ratio matches the external pressure ratio and large flow pulsations in the discharge line are avoided. This injection rate results in a discharge temperature of 160 °C.

5.2.7 Injection Port Location

Lastly, simulations are carried out to investigate the effect of the injection port location. In these simulations the ammonia mass fraction of the injection liquid is set to 0.4, the injection temperature is set to 55 °C, and the injection flow rate is set to 0.01 kg/s. The injection angle is varied from 270° to 540°, and the results are presented in Figures 5-31 and 5-32.

The temperature evolution presented in Figure 5-31 shows that the discharge temperature increases slightly when the injection port is placed at a later stage in the compression cycle. The maximum temperature is substantially higher in the case where the injection port is located at an angle of 540° compared to the other three cases. The temperature evolution is close to identical for the two cases where the injection port is located at 270° and 360°. In other words, it does not affect the operating temperature of the compressor whether the liquid is injected during the suction phase or at the very beginning of the compression phase. With the injection port located at an angle of 450° the temperature rises rapidly at the beginning of the compression phase, but the temperature increase is slowed down once the control volume reaches the injection angle. The discharge temperature with $\theta_{inj} = 450°$ is only 2.5 °C higher than with $\theta_{inj} = 360°$.

The corresponding pressure-volume diagram is presented in Figure 5-32. The curves shown in the diagram demonstrate that the compression work load is at its lowest when liquid is injected during the beginning of the compression phase. The work load is highest in the case where liquid is injected at an angle of 540°. In the case whit liquid injection during the suction phase, $\theta_{inj} = 270°$, the required compression work is slightly higher than with $\theta_{inj} = 360°$. However, the simulation results also show that the throughput of the compressor is slightly higher with $\theta_{inj} = 270°$. The cooling effect of the liquid leads to an increase in the density of the suction fluid, and thus a higher suction flow rate. By calculating the compression work load per unit of mass sucked in by the compressor, equal values are obtained for $\theta_{inj} = 270°$ and $\theta_{inj} = 360°$.

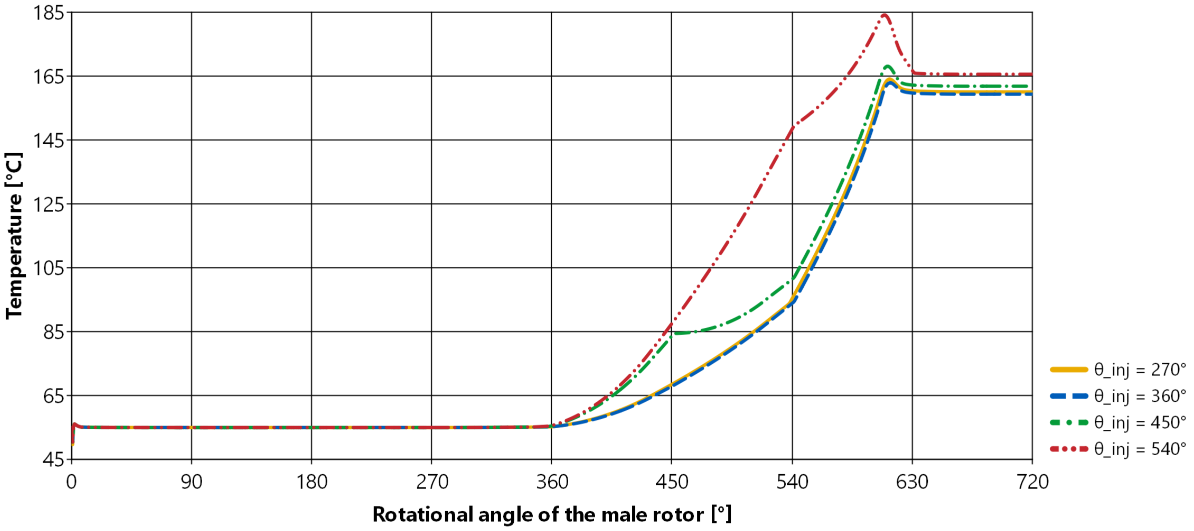


Figure 5-31: Temperature evolution with different injection port locations. θ_{inj} represents the rotational angle where injection begins.

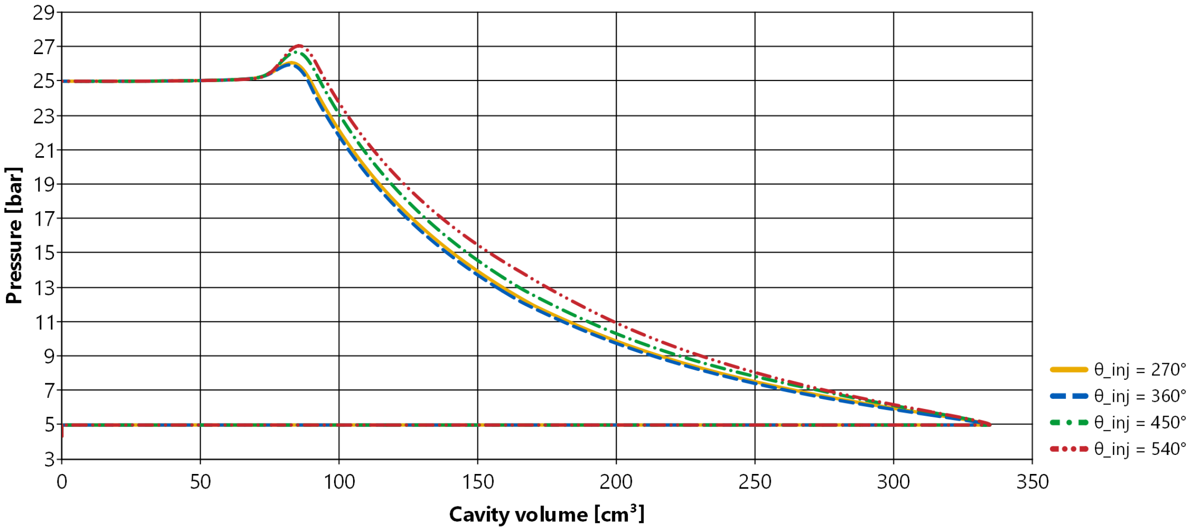


Figure 5-32: Pressure-volume diagram with different injection port locations. θ_{inj} represents the rotational angle where injection begins.

6 EES Two-Phase Compression Model

The Modelica model described in section 5 can give detailed insight into the process that takes place within a screw compressor during operation. However, the model has some analysis limitations. In each control volume the working fluid is treated as a single unit with homogeneous properties. This forces certain constraints on the state of the fluid, e.g., liquid and superheated vapor cannot be present simultaneously in one chamber. In reality, a thin film of liquid can be present along the inner surface of a compression chamber even though the vapor inside the chamber is in a superheated state. As a matter of fact, a portion of the working fluid always being in a liquid state is essential for safe and efficient operation of oil-free screw compressors. In addition, TILMedia used for ammonia-water property calculations in the Modelica model is based on a vapor-liquid equilibrium assumption. With this model it is not possible to investigate the potential consequences of non-equilibrium during high-speed operation. To compensate for any inaccuracies the Modelica model may have, a secondary calculation model is developed using the Engineering Equation Solver (EES). This model is to be used for detailed thermodynamic analysis of the two-phase process that takes place during the compression phase. Contrary to the Modelica model, the EES model disregards the suction and discharge phases. The aim is to get a better understanding of the intercorrelation between liquid and vapor during two-phase compression, and thereby being able to carry out comprehensive investigations on the effects of liquid injection.

6.1 Modeling Environment

EES is a commercial software, developed by F-Chart Software, that can numerically solve large sets of non-linear algebraic and differential equations (FCS, 2020a). The software is fitted with specialized functionality for solving problems related to heat transfer and thermodynamics, and it is a widely used tool in engineering academia related to these fields.

Performing calculations in EES is quite straightforward. First off, a set of equations has to be defined. The number of equations and the number of variables has to be equal in order to make the problem solvable. EES has inner functionality that decides in which order the equations are to be solved, thus the equations can be written down in any order. The correct solution to the set of equations is found numerically through a built-in iteration procedure. As long as the user provides reasonable guess values for each variable, i.e., the value that each parameter has at the beginning of the iteration procedure, EES is able to solve very large problem sets in a manner of seconds. A set of equations develped for analysis of two-phase compression will be presented in section 6.2.

As already discussed, reliable and efficient calculation of the working fluid's thermophysical properties is one of the main challenges when it comes to simulation of ammonia-water screw compressors, especially when dealing with the two-phase region. Similar to TILMedia used in the Modelica model, EES provides built-in thermophysical property data for a wide range of fluids, including the ammonia-water mixture (FCS, 2020b). With the built-in property functions, the thermodynamic state of the ammonia-water mixture can be found when three of the eight properties listed below are known:

- T – Temperature
- p – Pressure
- x – Ammonia mass fraction
- h – Specific enthalpy
- s – Specific entropy
- u – Specific internal energy
- v – Specific volume
- q – Vapor quality

The developed procedure for calculation of these properties is presented in the following section.

6.2 Model Description

The compression process is discretized by splitting the screw compressor into multiple segments, as illustrated in Figure 6-1, where the thermodynamic conditions within each segment are homogeneous. The total number of segments, N_{seg}, should be set as large as possible to minimize the errors that arise when a continuous process is split into discrete segments. However, a large number of segments results in a large number of equations and thereby heavy calculations. An EES model is limited to a maximum number of equations. With the version of EES used here, the limit is 12 000 equations. This must be taken into account when the number of segments is set. For the calculations and the results presented here, N_{seg} is set to 100 segments.

Each segment has a certain volume. The volume of the first segment is given by the volume of the cavity between the rotors at the point where the suction phase ends, while the volume of the last segment is given by the cavity volume at the point where the discharge phase begins. The difference in volume between adjecent segments is defined by a constant volume step, ΔV:

$$\Delta V = \frac{\left(\frac{1}{v_i} - 1\right) V_{suc}}{N_{seg} - 1} \tag{6.1}$$

where v_i is the built-in volume ratio of the compressor, V_{suc} is the cavity volume at the end of the suction phase, and N_{seg} is the number of segments. While the volume of the first segment is given by $V_1 = V_{suc}$, the remaining volumes are defined as:

$$V_j = V_{j-1} + \Delta V \tag{6.2}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$.

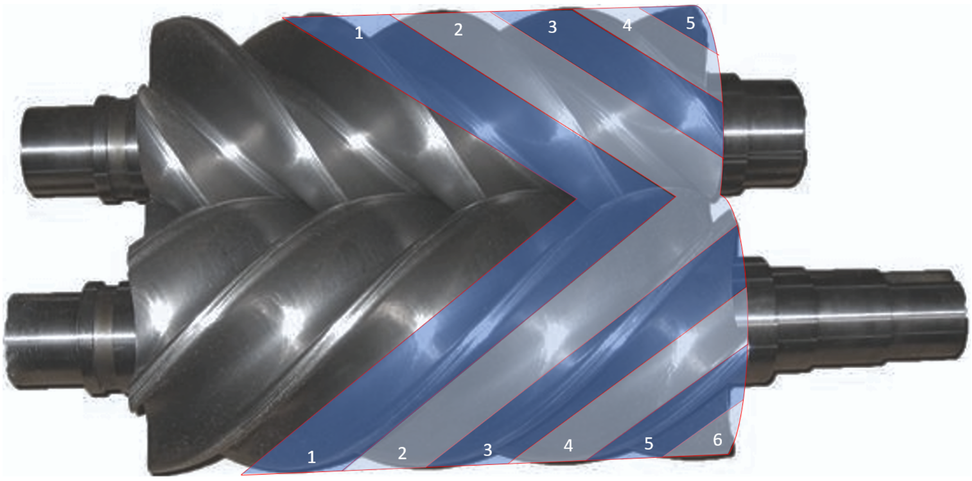


Figure 6-1: Simplified illustration of how the compressor is split into N_{seg} number of segments. Segment number 1 is located where the suction phase ends, while segment number N_{seg} is located where the discharge phase begins (which cannot be seen in the illustration because it is located on the bottom side of the rotors).

6.2.1 Mass Balance

Vapor and liquid flows through the segments towards the discharge end driven by the rotation of the rotors. In addition, a portion of the working fluid will flow back towards the suction end driven by pressure difference across the leakage gaps. Only the liquid part of the working fluid is assumed to flow backwards in this model, as liquid is used to seal the gaps in the compressor. This assumption is only valid in cases where enough liquid is injected to avoid dry compression, i.e., when compression of pure vapor is avoided in all stages of compression. Notice here that this approach is entirely different from the Modelica approach. In the Modelica model the working fluid is trapped within a control volume that expands and shrinks, while in this EES model the working fluid flows through segments of fixed size. The flow of vapor and liquid in segment number j is illustrated in Figure 6-2.

When the vapor is pushed through segments with smaller and smaller volumes, the pressure and the temperature of the vapor increases. If we assume that the liquid in each segment is in equilibrium with the vapor, i.e., that the pressure and temperature of the liquid is equal to that of the vapor, the composition of the liquid will change between each chamber. This type of composition change can be seen by looking back at Figure 4-4, which shows the bubble and dew point curves for ammonia-water at a constant pressure of 35 bar. At this pressure, if the temperature of the mixture rises from 140 °C to 160 °C, the ammonia mass fraction of saturated liquid decreases from around 0.4 to 0.3. In order to make such composition changes possible, a phase change must occur in a portion of the working fluid in each segment. E.g., in order to decrease the ammonia mass fraction of the liquid in a segment, either desorption of ammonia from the liquid, condensation of water from the vapor or a combination of both must occur.

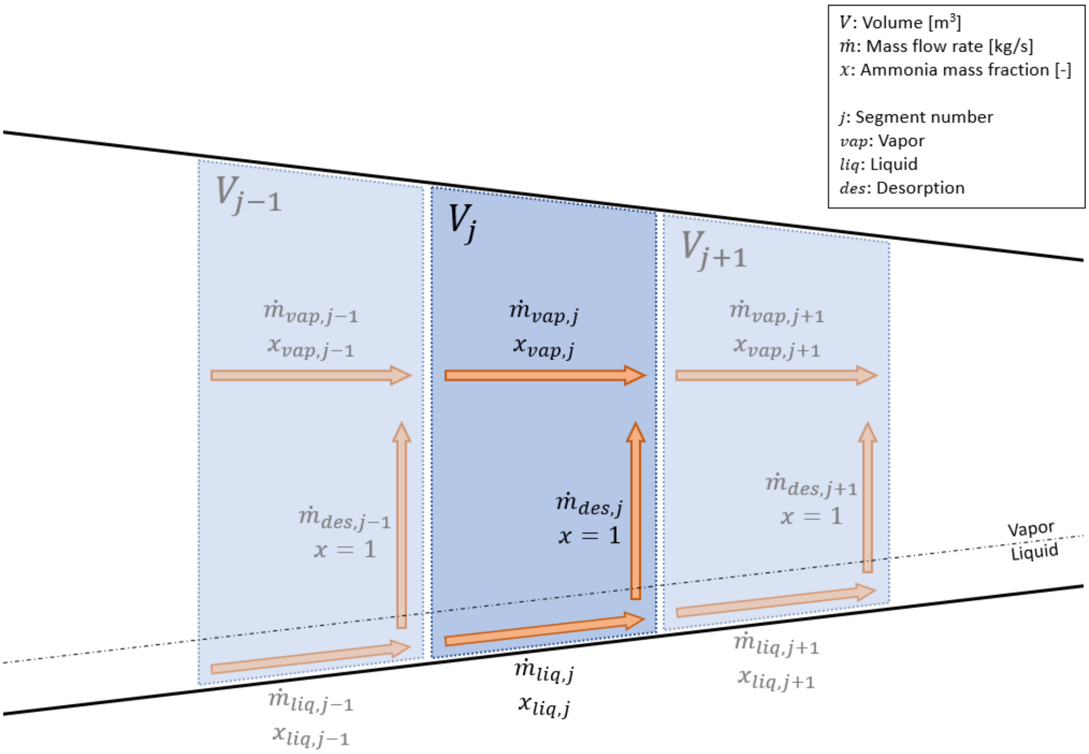


Figure 6-2: Schematic of the mass flows in each segment.

To simplify the calculations in the two-phase compression model, it is assumed that only the ammonia part of the working fluid goes through a phase change. This assumption is made due to the fact that pure water evaporates at a much higher temperature than pure ammonia at the same pressure. E.g., at a pressure of 20 bar the saturation temperature of pure water and pure ammonia is 212 °C and 49 °C respectively. With the cooling effect of liquid injection, the temperature of the superheated vapor in each segment is kept below the saturation temperature of pure water throughout the entire compression process, and therefore it is considered reasonable to assume that no water will evaporate. The rate of desorption of ammonia in each segment is represented by the parameter $\dot{m}_{des}$, as illustrated in Figure 6-2. Absorption of ammonia can also occur in a segment if the pressure and/or the temperature of the liquid decreases when it flows through the given segment. Calculation results will display negative values for $\dot{m}_{des}$ in segments where absorption occurs.

Conservation of mass is implemented in the model through four equations applied to each segment. Two equations regarding the vapor portion of the working fluid, and two equations regarding the liquid portion. Conservation of vapor mass is applied through:

$$\dot{m}_{vap,j} = \dot{m}_{vap,j-1} + \dot{m}_{des,j-1} \tag{6.3}$$

$$\dot{m}_{vap,j} x_{vap,j} = \dot{m}_{vap,j-1} x_{vap,j-1} + \dot{m}_{des,j-1} \tag{6.4}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Here $\dot{m}_{vap,j}$ represents the mass flow rate of vapor in segment j, and $x_{vap,j}$ represents the corresponding ammonia mass fraction of the vapor. The ammonia mass fraction corresponding to $\dot{m}_{des,j-1}$ does not have to be included in equation 6.4, since it is always equal to 1 based on the assumption that no water evaporates.

Conservation of liquid mass is applied in an equal manner, but with a special case for segments that are placed where liquid is injected into the compressor. If liquid is injected into segment number S, conservation of liquid mass in the subsequent segment is applied through:

$$\dot{m}_{liq,S+1} = \dot{m}_{liq,S} - \dot{m}_{des,S} + \dot{m}_{inj} \tag{6.5}$$

$$\dot{m}_{liq,S+1} x_{liq,S+1} = \dot{m}_{liq,S} x_{liq,S} - \dot{m}_{des,S} + \dot{m}_{inj} x_{inj} \tag{6.6}$$

where $\dot{m}_{inj}$ and x_{inj} represents the mass flow rate and the ammonia mass fraction of the injected liquid. Likewise, where injection does not occur:

$$\dot{m}_{liq,j} = \dot{m}_{liq,j-1} - \dot{m}_{des,j-1} \tag{6.7}$$

$$\dot{m}_{liq,j} x_{liq,j} = \dot{m}_{liq,j-1} x_{liq,j-1} - \dot{m}_{des,j-1} \tag{6.8}$$

for segment numbers $j \in [\![2,\ S]\!]$ and $j \in [\![S+2,\ N_{seg}]\!]$. Here $\dot{m}_{liq,j}$ represents the mass flow rate of liquid in segment j, while $x_{liq,j}$ represents the corresponding ammonia mass fraction. $\dot{m}_{liq,j}$ will obtain a negative value in segments where most of the liquid flows back towards the suction end due to leakage.

6.2.2 Energy Balance

In addition to conservation of mass, equations for conservation of energy are applied to each segment. Here, the work done by the compressor on the working fluid must be accounted for. The amount of work is equal to the area under the curve in a pressure-volume diagram. This area can be calculated through integration:

$$w = \int_{p_1}^{p_2} v \, \mathrm{d}p \tag{6.9}$$

where w is the specific work done by the compressor, v is the specific volume of the working fluid, and p is the pressure. p_1 and p_2 represent the pressure at the beginning and the end of the compression process respectively. What makes this work calculation challenging is the fact that the specific volume continually changes with the pressure in a way that is rather hard to predict. Usually some sort of polytropic relation is employed in order to simplify calculations related to compression. A polytropic process obeys the relation:

$$pv^n = C \tag{6.10}$$

where n is the polytropic index and C is a constant. In cases where the polytropic index, n, is set equal to the isentropic exponent, $\kappa = c_p/c_v$, the process is defined as isentropic. Using an isentropic process as a basis for calculation of compression work in the EES model can give inaccurate results, because isentropic processes are frictionless and without any transfer of heat or mass. The real process will involve continuous transfer of heat and mass between the vapor and the liquid portions of the working fluid, as well as between the working fluid and the compressor body. However, when the process is discretized and split into many small segments the error caused by assuming an isentropic process can be strongly reduced. The isentropic relation is only utilized to calculate the amount of work required to reduce the segment volume by ΔV between each segment, while the non-isentropic effects of heat and mass transfer are accounted for within each segment. In Figure 6-3 it is illustrated how the different energy flows, including compression work, are connected to each segment. By utilizing the relation presented in equation 6.10 and the isentropic exponent κ, equation 6.9 can be solved as follows:

$$p v^{\kappa} = p_1 v_1^{\kappa} \quad \Rightarrow \quad v = v_1 \left(\frac{p_1}{p} \right)^{1/\kappa} \tag{6.11}$$

$$w = \int_{p_1}^{p_2} v_1 \left(\frac{p_1}{p} \right)^{1/\kappa} \mathrm{d}p = \frac{v_1 p_1^{1/\kappa}}{1-1/\kappa} \left[p^{1-1/\kappa} \right]_{p_1}^{p_2} = \frac{v_1 p_1}{1-1/\kappa} \left[\left(\frac{p_2}{p_1} \right)^{1-1/\kappa} - 1 \right] \tag{6.12}$$

Thus the amount of work done by the compressor on the vapor between segment $j-1$ and j can be calculated as:

$$w_{vap,j} = \frac{v_{vap,j-1} p_{j-1}}{1-1/\kappa} \left[\left(\frac{p_j}{p_{j-1}} \right)^{1-1/\kappa} - 1 \right] \tag{6.13}$$

$$\dot{W}_{vap,j} = \dot{m}_{vap,j} w_{vap,j} \tag{6.14}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Here $w_{vap,j}$ is the specific work done on the vapor as it flows into segment j, $v_{vap,j-1}$ is the specific volume of the vapor in the preceding segment, $\dot{m}_{vap,j}$ is the mass flow rate of vapor in segment j, and $\dot{W}_{vap,j}$ is the compression work rate. The isentropic exponent, κ, varies slightly when the pressure and temperature of the working fluid varies, due to changes in the specific heat capacities of the working fluid, c_p and c_v. The variations are relatively small, typically κ has a value between 1.32 and 1.36 in the pressure and temperature ranges under investigation here. Therefore, it is considered reasonable to define κ as a constant in this calculation model. κ is set to 1.34, which corresponds to the isentropic exponent for saturated vapor with an ammonia mass fraction of 0.985 at a pressure of 5 bar according to data taken from REFPROP (Lemmon et al., 2018).

In addition to work required for compression of the vapor, a small amount of work must be added to compress the liquid portion of the working fluid. The changes in liquid density are very small in the pressure range under investigation here, and the work calculations are simplified by considering the liquid as incompressible. This yields:

$$w_{liq,j} = \left(p_j - p_{j-1} \right) v_{liq,j} \tag{6.15}$$

$$\dot{W}_{liq,j} = \dot{m}_{liq,j} w_{liq,j} \tag{6.16}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Here $w_{liq,j}$ is the specific work required for liquid compression in segment j, $v_{liq,j}$ is the specific volume of the liquid, $\dot{m}_{liq,j}$ is mass flow rate of the liquid, and $\dot{W}_{liq,j}$ is the compression work rate. Notice that $\dot{W}_{liq,j}$ will be negative in segments where $\dot{m}_{liq,j}$ is negative, presuming that $p_j > p_{j-1}$.

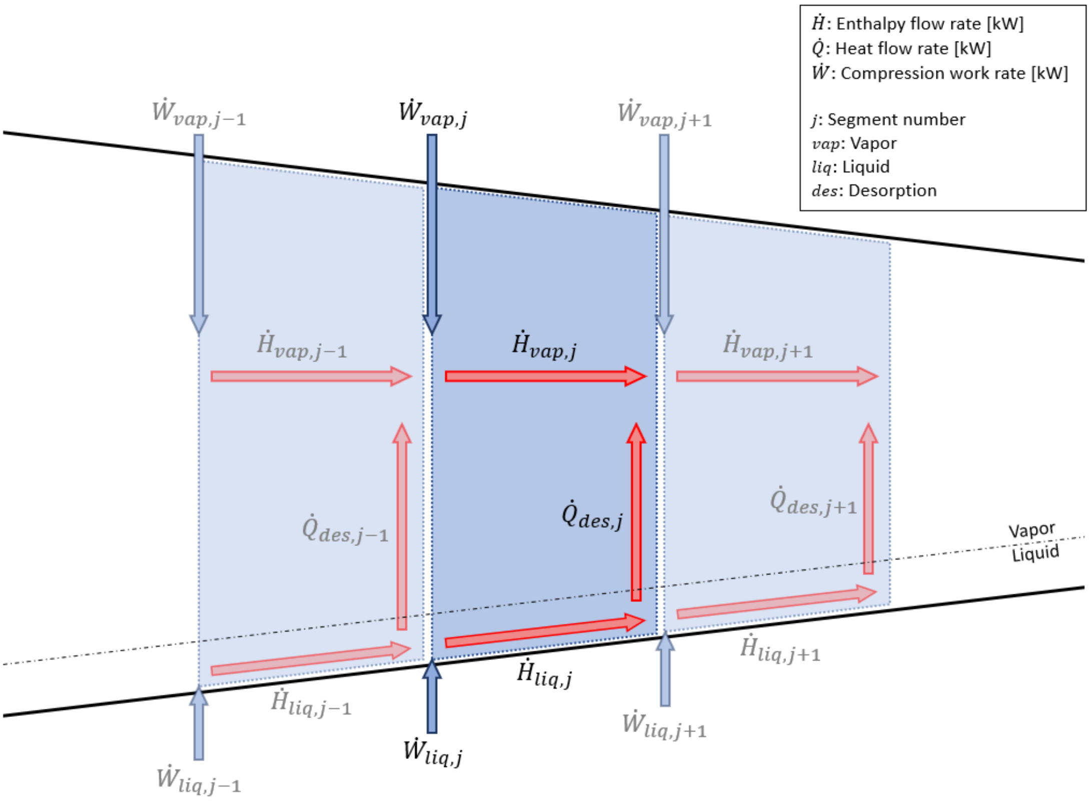


Figure 6-3: Schematic of the energy flows related to each segment.

When the compression work equations have been established, conservation of energy can be implemented in the model. First off, conservation of energy for the vapor:

$$\dot{H}_{vap,j} = \dot{H}_{vap,j-1} + \dot{W}_{vap,j} + \dot{Q}_{des,j-1} \tag{6.17}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Here $\dot{H}_{vap,j}$ is the vapor enthalpy flow rate through segment j, and $\dot{Q}_{des,j-1}$ is the heat flow from the the liquid to the vapor in segment $j-1$ due to desorption of ammonia. The enthalpy flow rate is given by:

$$\dot{H}_{vap,j} = \dot{m}_{vap,j} h_{vap,j} \tag{6.18}$$

where $h_{vap,j}$ is the specific enthalpy of vapor in segment j.

Likewise, conservation of energy for the liquid is implemented through:

$$\dot{H}_{liq,j} = \dot{H}_{liq,j-1} + \dot{W}_{liq,j} - \dot{Q}_{des,j-1} \tag{6.19}$$

for segment numbers $j \in [\![2,\ S]\!]$ and $j \in [\![S+2,\ N_{seg}]\!]$. The liquid enthalpy flow rate is given by:

$$\dot{H}_{liq,j} = \dot{m}_{liq,j} h_{liq,j} \tag{6.20}$$

where $h_{liq,j}$ is the specific enthalpy of liquid in segment j. There is a special case for segment number $S+1$, which is the subsequent segment to where liquid is injected into the compressor:

$$\dot{H}_{liq,S+1} = \dot{H}_{liq,S} + \dot{W}_{liq,S+1} - \dot{Q}_{des,S} + \dot{H}_{inj} + \dot{W}_{inj} \tag{6.21}$$

$\dot{H}_{inj}$ and $\dot{W}_{inj}$ represent the enthalpy flow rate from liquid injection and the required compression work rate for the injected liquid respectively. These values are calculated from equations 6.22 and 6.23.

$$\dot{H}_{inj} = \dot{m}_{inj} h_{inj} \tag{6.22}$$

$$\dot{W}_{inj} = \dot{m}_{inj} \left(p_s - p_{inj}\right) v_{inj} \tag{6.23}$$

Here h_{inj} is the specific enthalpy of the injected liquid, v_{inj} is the specific volume of the injected liquid, p_{inj} is the pressure in the injection line, and p_S is the pressure in the segment where injection occurs.

6.2.3 Thermodynamic Properties

In order to solve the conservation equations presented above, multiple thermodynamic properties must be calculated for each segment. Segment number 1 differs from the other segments, as it represents the transition from the suction phase to the compression phase. The pressure in segment 1 is given by the suction pressure:

$$p_1 = p_{suc} \tag{6.24}$$

Correspondingly, the ammonia mass fraction of the vapor in segment 1 is given by the suction ammonia mass fraction:

$$x_{vap,1} = x_{suc} \tag{6.25}$$

Under normal operating conditions it is reasonable to assume that the suction vapor is in a saturated state, since vapor and liquid is separated upstream of the compressor. With this, three of the eight thermodynamic properties listed on page 78 are known for segment 1: the pressure, the ammonia mass fraction and the vapor quality. Therefore, by using the built-in ammonia-water property functions in EES, all other thermodynamic properties can be calculated for segment 1. First off, the temperature:

$$T_1 = \text{temperature}\left(p_1;\ x_{vap,1};\ q = 1\right) \tag{6.26}$$

Regarding the notation used in equation 6.26, “temperature” represents the built-in function for ammonia-water temperature calculations in EES, and the variables listed in parenthesis represent the known variables used as input to that function. The same notation will be used in several equations presented below. In addition to temperature, the specific enthalpy for the vapor in segment 1 is calculated as:

$$h_{vap,1} = \text{enthalpy}\left(T_1;\ p_1;\ x_{vap,1}\right) \tag{6.27}$$

Similarly, the specific volume of the vapor in segment 1 is calculated as:

$$v_{vap,1} = \text{volume}\left(T_1;\ p_1;\ x_{vap,1}\right) \tag{6.28}$$

Equations 6.24 – 6.28 constitute one of the boundaries in the model, the suction boundary. The second model boundary is the injection boundary. The thermodynamic properties for this boundary is calculated through:

$$h_{inj} = \text{enthalpy}\left(T_{inj};\ p_{inj};\ x_{inj}\right) \tag{6.29}$$

$$v_{inj} = \text{volume}\left(T_{inj};\ p_{inj};\ x_{inj}\right) \tag{6.30}$$

where h_{inj} and v_{inj} represent the specific enthalpy and the specific volume of the injected liquid respectively. The injection temperature, T_{inj}, the injection pressure, p_{inj}, and the injection ammonia mass fraction, x_{inj}, must all be given as input parameters to the model.

When the boundary properties have been established, the thermodynamic properties for each segment can be calculated based on the equations for conservation of mass and energy. Equation 6.4, conservation of ammonia vapor mass, is used to calculate the ammonia mass fraction for the vapor in each segment, $x_{vap,j}$. Equations 6.17 and 6.18, conservation of vapor energy, is used to calculate the specific enthalpy for the vapor in each segment, $h_{vap,j}$. Following, the temperature of the vapor is obtained from:

$$T_j = \text{temperature}\left(p_j;\ x_{vap,j};\ h_{vap,j}\right) \tag{6.31}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Likewise, the specific volume of the vapor is defined by:

$$v_{vap,j} = \text{volume}\left(p_j;\ x_{vap,j};\ h_{vap,j}\right) \tag{6.32}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. In addition to equation 6.32, the specific volume of the vapor is calculated from:

$$v_{vap,j} = \frac{fV_j}{\dot{m}_{vap,j}} \tag{6.33}$$

for segment numbers $j \in [\![2,\ N_{seg}]\!]$. Here f is the operational frequency of the compressor. By multiplying it with the volume of a segment, V_j, the volume flow rate through the segment is obtained. Then the specific volume of the vapor can be obtained by dividing with the mass flow rate through the segment, $\dot{m}_{vap,j}$. In these calculations the volume of the liquid in each segment is neglected. The amount of liquid is much smaller than the amount of vapor (assuming that the amount of liquid injected is not superfluous) and the density of the liquid is much higher. Thus neglecting the volume of the liquid is considered viable. The reason for defining two equations for the specific volume is that equation 6.32 is not actually defined in order to calculate the specific volume. It is defined in order to calculate the pressure in each segment, p_j. By using the built-in iteration procedure, EES is able to find the value for p_j in equation 6.32 that results in the same specific volume as equation 6.33.

By assuming that the liquid is in equilibrium with the vapor in each segment, i.e., that the pressure and the temperature are homogeneous, the ammonia mass fraction for the liquid can be calculated from:

$$x_{liq,j} = \text{massfraction}\left(T_j;\ p_j;\ q = 0\right) \tag{6.34}$$

for segment numbers $j \in [\![1,\ N_{seg}]\!]$. Here the vapor quality, q, is set to zero to impose saturated liquid. From this, the specific enthalpy of the liquid is calculated:

$$h_{liq,j} = \text{enthalpy}\left(T_j;\ p_j;\ x_{liq,j}\right) \tag{6.35}$$

for segment numbers $j \in [\![1,\ N_{seg}]\!]$. Likewise, the specific volume of the liquid is given by:

$$v_{liq,j} = \text{volume}\left(T_j;\ p_j;\ x_{liq,j}\right) \tag{6.36}$$

for segment numbers $j \in [\![1,\ N_{seg}]\!]$. The specific volume is used to calculate liquid compression work, according to equation 6.15. The work calculation is simplified by considering the liquid as incompressible, but since the specific volume is re-evaluated for each segment with equation 6.36, the specific volume will in fact change slightly from segment to segment. This reduces the inaccuracy caused by assuming incompressible liquid.

6.2.4 Input Parameters

Equations 6.1 – 6.8 and 6.13 – 6.36 define the set of equations that must be solved in the two-phase compression model. In order to reduce the number of unknown variables to be equal to the number of equations, some variables must be given a predefined value. These values are referred to as the model's input parameters. Each one of them are described in what follows.

f: The operational frequency of the compressor is given as input to the model. This parameter is important if investigations on the capacity of a compressor are to be carried out. However, if the aim of an investigation is solely to predict the relative flows of mass and energy within the compressor to gain better understanding of the compression process itself, it is sufficient to set f equal to a trivial value such as 1 Hz.

V_{suc}: The volume of the first segment must be defined by the user of the model. This parameter will determine the size of the volume step, ΔV, according to equation 6.1. Furthermore, it will determine the suction mass flow rate, i.e., the mass flow rate of vapor flowing through the first segment, according to equation 6.37.

$$\dot{m}_{vap,1} = \frac{fV_{suc}}{v_{vap,1}} \tag{6.37}$$

where $v_{vap,1}$ is the specific volume of the vapor in the first segment, given by equation 6.28.

p_{suc}: The pressure in the suction line is given as input to the model. If the compressor is to be utilized in CACHP, this corresponds to the desorber pressure.

x_{suc}: The ammonia mass fraction for the suction vapor must be defined by the user. The value of this mass fraction is close to one in a CACHP under normal operating conditions, i.e., nearly pure ammonia vapor.

v_i: The compressor's built-in volume ratio is a crucial input parameter. It determines the degree of compression in the model, and thus the discharge pressure level. In cases where the value of v_i is lower than desirable, the calculated discharge pressure will be lower than the pressure in the discharge line, causing under-compression. Similarly, a value of v_i that is higher than desirable will cause over-compression.

($p_{N_{seg}}$): The calculated discharge pressure, i.e., the pressure in segment number N_{seg}, can be chosen as an input parameter. This parameter is written in parenthesis because it is not used when the volume ratio, v_i, is in use. $p_{N_{seg}}$ is simply a variable that can be used as an input parameter in cases where the user wants to find the optimal volume ratio for the compressor. In such cases $p_{N_{seg}}$ is set to a predefined value and the model itself finds the value for v_i that results in neither under- nor over-compression.

p_{inj}: The pressure in the injection line must be given by the user. This, together with x_{inj} and T_{inj}, defines the thermodynamic state of the injected liquid.

T_{inj}: The temperature of the injected liquid. To ensure that liquid injection is modeled, the injection temperature must be set lower than, or equal to, the saturation temperature at the given injection pressure.

x_{inj}: The ammonia mass fraction of the injected liquid is an input parameter. The mass fraction value can be anywhere in the range $[0,\ 1]$.

$\dot{m}_{inj}$: The injection mass flow rate must be predefined. In order to avoid an unreasonably high or low value, it is recommended to base the value of $\dot{m}_{inj}$ on the calculated suction flow rate, $\dot{m}_{vap,1}$. E.g., giving $\dot{m}_{inj}$ a value equal to 5 % of $\dot{m}_{vap,1}$.

N_{seg}: The number of segments is set by the user. It determines to size of the equation set. A higher value for N_{seg} gives a higher number of equations that must be solved, and thus heavier calculations. On the other hand, a higher value for N_{seg} reduces the magnitude of the errors caused by discretization.

S: The segment number for the segment where liquid injection occurs is a key input parameter. The value of S can be in the range $[\![1,\ N_{seg}-1]\!]$. It is also possible to implement liquid injection at multiple points by defining multiple values for S, e.g., S_1 and S_2.

$\dot{m}_{leak}$: The leakage mass flow rate is the final input parameter. This parameter represents the flow rate of liquid leaking out at the suction end, i.e., it gives the liquid mass flow rate in the first segment:

$$\dot{m}_{liq,1} = -\dot{m}_{leak} \tag{6.38}$$

Here $\dot{m}_{liq,1}$ is set to a negative value because the liquid flows opposite of the positive flow direction, which is defined as flow towards the discharge end. The value for $\dot{m}_{leak}$ is set based on the predicted maximum leakage flow in the compressor under investigation. Due to the assumption that no water evaporates during the compression process, all the water that leaks backwards at the point of maximum leakage must also leak out at the suction end.

6.2.5 Equilibrium Correction Factor

Additional functionality is added to the two-phase compression model by introducing an equilibrium correction factor, C_{eq}. This parameter is implemented in order to relax the vapor-liquid equilibrium restriction and thereby being able to investigate the effects of possible non-equilibrium states. In the equation set described above, the ammonia mass fraction for the liquid in each segment, $x_{liq,j}$, is calculated by assuming that the liquid temperature is equal to the vapor temperature, according to equation 6.34. Further, the desorption mass flow rate, $\dot{m}_{des,j}$, required to obtain this ammonia mass fraction in each segment is calculated, according to equations 6.5 – 6.8. However, at high speed operation there may not be enough time available for all that desorption to take place. This will result in non-equilibrium states and a different ammonia mass fraction for the liquid in each segment. To account for this, equation 6.34 is replaced by:

$$x_{liq,eq,j} = \text{massfraction}\left(T_j;\ p_j;\ q = 0\right) \tag{6.39}$$

Here *eq* is used to emphasize that this is the ammonia mass fraction at 100 % equilibrium. Next, the desorption mass flow rate required to obtain the equilibrium state, $\dot{m}_{des,eq,j}$, is calculated from equations 6.40 – 6.43.

$$\dot{m}_{liq,eq,j} = \dot{m}_{liq,j-1} - \dot{m}_{des,eq,j-1} \tag{6.40}$$

$$\dot{m}_{liq,eq,j} x_{liq,eq,j} = \dot{m}_{liq,j-1} x_{liq,j-1} - \dot{m}_{des,eq,j-1} \tag{6.41}$$

for segment numbers $j \in [\![2,\ S]\!]$ and $j \in [\![S+2,\ N_{seg}]\!]$. Here $\dot{m}_{liq,j-1}$ represents the actual liquid mass flow rate in the preceding segment, while $\dot{m}_{liq,eq,j}$ represents the liquid mass flow rate in segment j if complete equilibrium is obtained. Likewise, for the segment where liquid injection occurs:

$$\dot{m}_{liq,eq,S+1} = \dot{m}_{liq,S} - \dot{m}_{des,eq,S} + \dot{m}_{inj} \tag{6.42}$$

$$\dot{m}_{liq,eq,S+1} x_{liq,eq,S+1} = \dot{m}_{liq,S} x_{liq,S} - \dot{m}_{des,eq,S} + \dot{m}_{inj} x_{inj} \tag{6.43}$$

After calculating the desorption mass flow rate at equilibrium, the predicted actual desorption rate is found by utilizing the equilibrium correction factor:

$$\dot{m}_{des,j} = C_{eq}\dot{m}_{des,eq,j} \tag{6.44}$$

for segment numbers $j \in [\![1,\ N_{seg}-1]\!]$. The value of C_{eq} must be in the range $\langle 0,\ 1]$. Finally, by using the actual desorption mass flow rate, $\dot{m}_{des,j}$, the ammonia mass fraction for the liquid, $x_{liq,j}$, at non-equilibrium is calculated through equations 6.5 – 6.8.

At non-equilibrium the liquid temperature is not equal to the vapor temperature. The liquid temperature in each segment is calculated by assuming saturated liquid, $q = 0$, and by using the ammonia mass fraction, $x_{liq,j}$:

$$T_{liq,j} = \text{temperature}\left(p_j;\ x_{liq,j};\ q = 0\right) \tag{6.45}$$

for segment numbers $j \in [\![1,\ N_{seg}]\!]$. Furthermore, in order to define the boundary conditions for the liquid, it is assumed that there is an initial vapor-liquid equilibrium in the first segment. Thus:

$$T_{liq,1} = T_1 \tag{6.46}$$

where T_1 is the vapor temperature in the first segment, calculated from equation 6.26.

The complete and final equation set in the two-phase compression model is represented by equations 6.1 – 6.8, 6.13 – 6.33 and 6.35 – 6.46. The EES code for the model is given in appendix C.

6.3 Results

To demonstrate some of the capabilities of the developed EES model, two separate calculations are carried out. The first two-phase compression calculation deals with complete vapor-liquid equilibrium. In this calculation the equilibrium correction factor, C_{eq}, is set to 1. The second two-phase compression calculation aims to demonstrate the effects of non-equilibrium. This is done by giving the equilibrium correction factor a value much lower than 1. Results from calculations with $C_{eq} = 1$ and $C_{eq} = 0.2$ are presented in sections 6.3.1 and 6.3.2 respectively.

Values for the input parameters are chosen so that the operating conditions match the conditions in the Modelica simulations presented in section 5.2.7. The operational frequency of the compressor, f, is set to 50 Hz, and the volume of the first segment, V_{suc}, is set to 1286 cm^3. According to equation 6.37, this results in a vapor mass flow rate in section 1, $m_{vap,1}$, that matches the suction flow rate in the Modelica simulations. Notice here that the value of $V_{suc} = 1286$ cm^3 is substantially higher than $V_{max} = 335$ cm^3 used in the Modelica model. This is because V_{max} represents the maximum volume of one cavity, while V_{suc} represents the total volume that passes through the first segment during one rotation. Furthermore, the suction pressure, p_{suc}, is set to

5 bar and the suction ammonia mass fraction, x_{suc}, is set to 0.985. The compressor's built-in volume ratio, v_i, is set to 3.65. Liquid is injected at a rate of 0.01 kg/s into segment number 33. Based on the volume evolution, injection in segment 33 corresponds to injection at an angle of 450° in the Modelica model. The temperature of the injection liquid, T_{inj}, is set to 55 °C and the ammonia mass fraction, x_{inj}, is set to 0.4. Lastly, the leakage mass flow rate, $\dot{m}_{leak}$, is set to 0.002 kg/s. As there is no water evaporation, this means that about 20 % of the injected water will leak back towards the suction end, while the other 80 % will be pushed forwards and discharged with the vapor. This leakage distribution is an educated guess based on results from the Modelica simulations.

6.3.1 Equilibrium

Here the equilibrium correction factor is set to 1. Thus the temperature of the vapor and the temperature of the liquid are equal in each segment. The predicted temperature is presented in Figure 6-4. The results show that there is an instant temperature drop at the point of injection. The temperature is 4.7 °C lower in segment 34 than in segment 33, while there is steady temperature increase between all other segments. The temperature is found to be 179 °C at the end of the compression process.

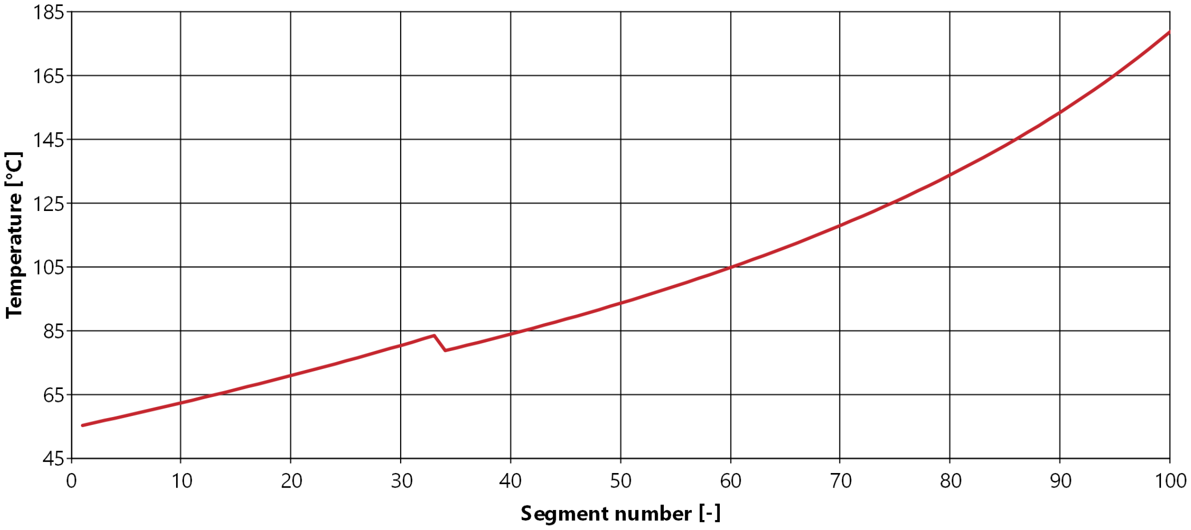


Figure 6-4: Temperature in each segment with complete vapor-liquid equilibrium.

In Figure 6-5 the pressure in each segment is plotted against the segment volume. The calculated pressure ratio suits the assumed pressure conditions in the CACHP system. The pressure in the last segment is calculated to be 24.9 bar, which corresponds to only 0.1 bar under-compression. There is a steady pressure increase between each segment, except for segments 33 and 34. The temperature drop leads to an increase in vapor density, and thus the pressure increase between segments 33 and 34 is close to zero.

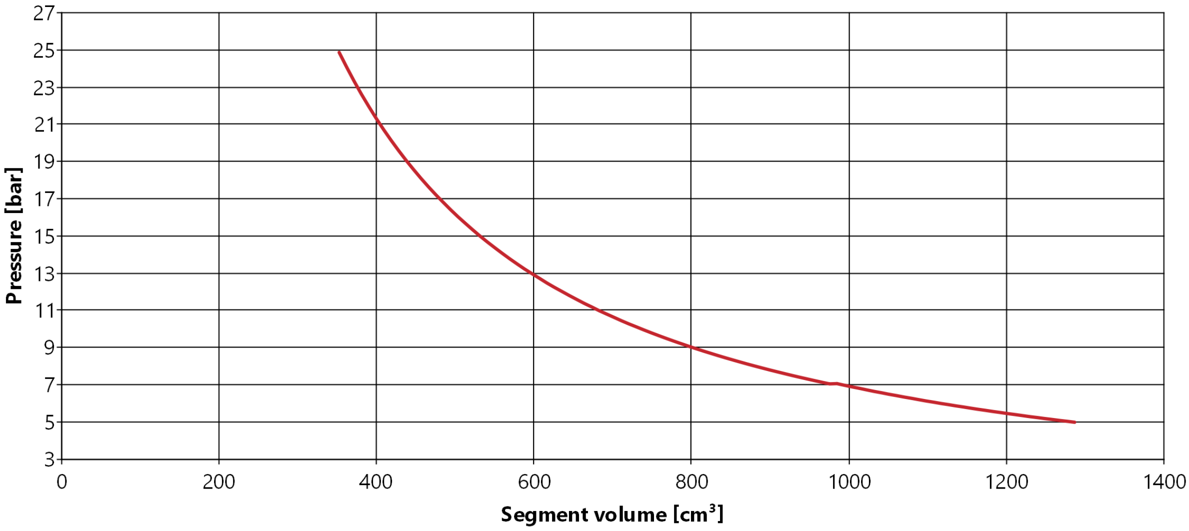


Figure 6-5: Pressure-volume diagram with complete vapor-liquid equilibrium.

Changes in liquid and vapor compositions throughout the compression process is studied to gain a better understanding of the two-phase process. The mass flow rates of liquid water and liquid ammonia in each segment are presented in Figure 6-6. For segments 1 – 33 the mass flow rate of the liquid is negative. This represents the portion of the injection liquid that leaks back to the suction end. The remaining portion of the injection liquid is driven towards the discharge end, resulting in positive liquid flow rates for segments 34 – 100. The flow rate of liquid water in either direction is constant, since evaporation of water does not occur. However, the flow rate of liquid ammonia varies. When there is an increases in the pressure and the temperature of

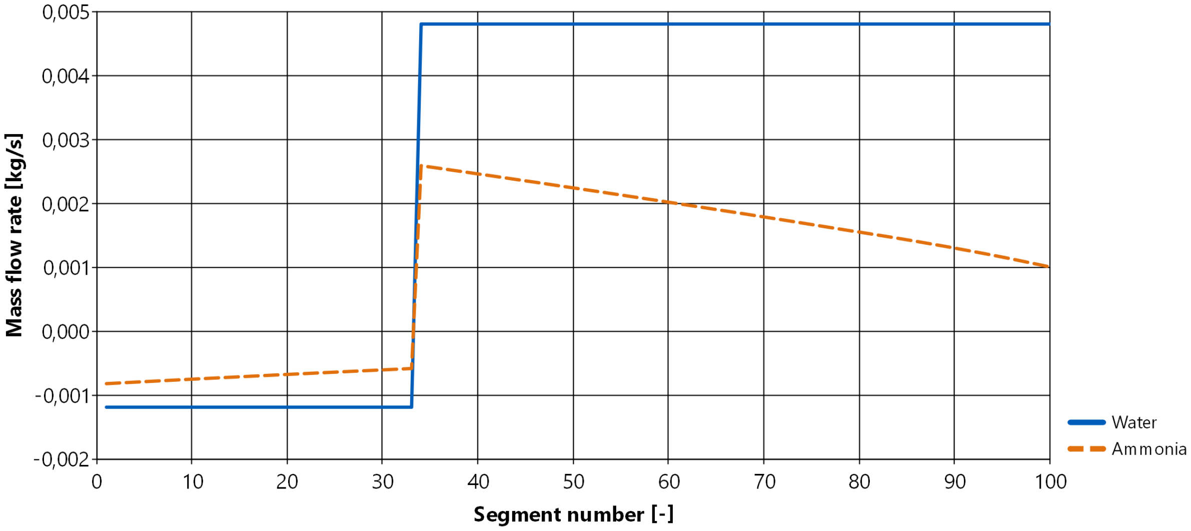


Figure 6-6: Mass flow rate of liquid in each segment with complete vapor-liquid equilibrium. Separated into liquid water and liquid ammonia. Negative flow rates represent flow towards the suction end.

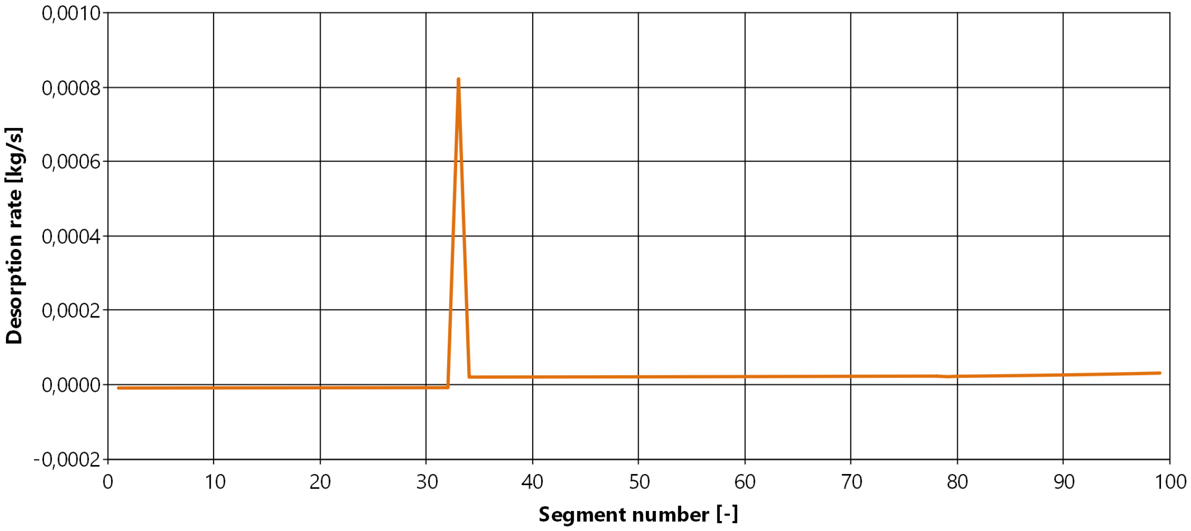


Figure 6-7: Ammonia desorption rate in each segment with complete vapor-liquid equilibrium. Negative desorption rates represent ammonia absorption.

the liquid, the ammonia mass fraction of the liquid decreases. For the liquid that flows towards the discharge end, this results in desorption of ammonia. For the liquid that flows towards the suction end, this results in absorption of ammonia. The rate of desorption and absorption in each segment is illustrated in Figure 6-7. Negative desorption rates for segments 1 - 32 demonstrate that absorption occurs in these segments, i.e., that ammonia is transferred from the vapor to the liquid. This can also be observed in Figure 6-6, as the absolute value of the ammonia flow rate increases towards the suction end. Likewise, since desorption occurs in segments 33 – 100, the mass flow rate of liquid ammonia decreases towards the discharge end. Furthermore, the results in Figure 6-7 show a very high rate of desorption in the injection segment. This is due to the difference between the ammonia mass fraction of the injection liquid and the ammonia mass fraction of saturated liquid in the segment. At the pressure and the temperature in segment 33, the ammonia mass fraction of saturated liquid is 0.33. The ammonia mass fraction of the injection liquid is 0.4. Thus, a relatively large amount of ammonia must be desorbed from the injection liquid in order to reach saturation.

6.3.2 Non-Equilibrium

To demonstrate the possible effects of non-equilibrium, calculations are carried out with the equilibrium correction factor set to 0.2. This correction factor relaxes the vapor-liquid equilibrium constraint, so that the temperature of the vapor and the temperature of the liquid do not have to be equal. The estimated temperature in each segment under non-equilibrium conditions is presented in Figure 6-8. The results show that the temperature of the liquid becomes slightly lower than the temperature of the vapor. The temperature difference is at its greatest in segment 34, subsequent to the injection segment. In this segment the liquid temperature is 12.5 °C lower than the vapor

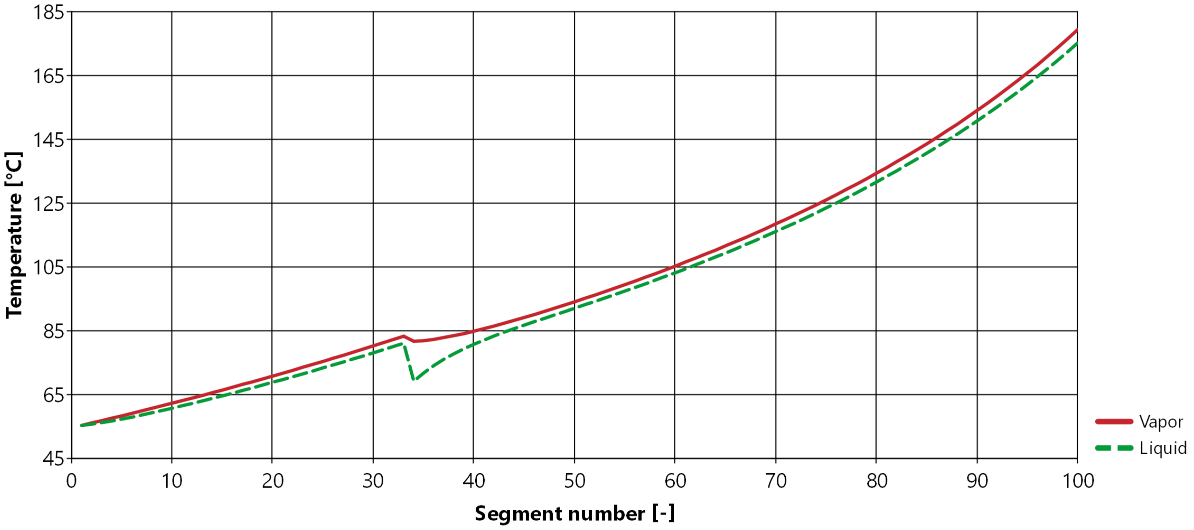


Figure 6-8: Temperature in each segment under non-equilibrium conditions.

temperature. There is a drop in the vapor temperature at the point of injection, but the cooling effect is not as intense as the cooling effect observed with complete equilibrium in Figure 6-4. In the last segment the vapor temperature is 179.7 °C and the liquid temperature is 175.5 °C. This vapor temperature is only 0.7 °C higher than the temperature obtained with complete equilibrium, which indicates that the overall cooling effect is not reduced substantially by non-equilibrium.

The mass flow rates of liquid water and liquid ammonia under non-equilibrium conditions are presented in Figure 6-9, and the corresponding desorption rates are presented in Figure 6-10. An equilibrium correction factor of 0.2 means that the desorption rate is only 20 % of the desorption rate that would be required to obtain complete vapor-liquid equilibrium in each segment. The

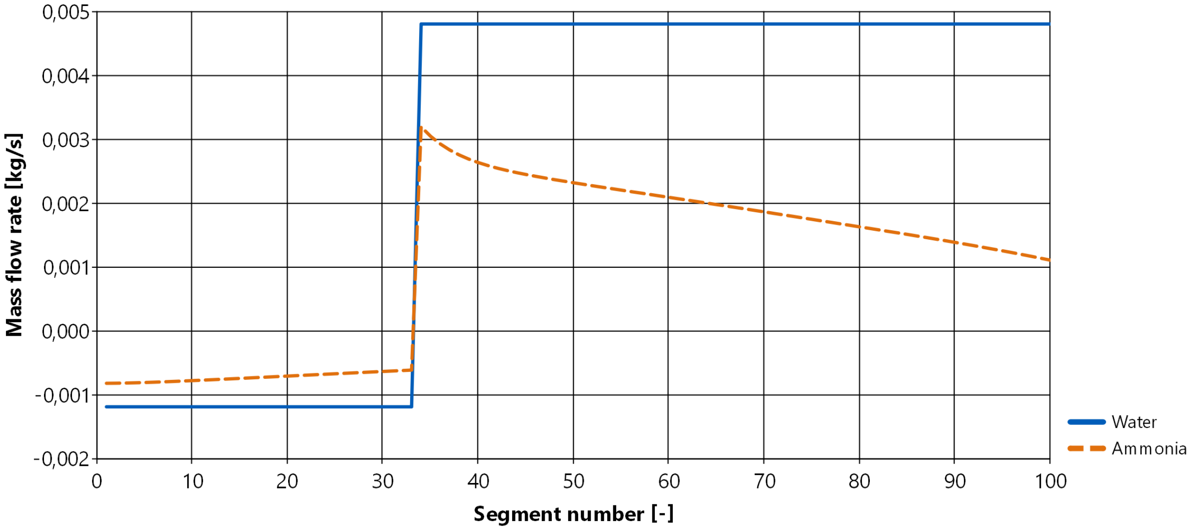


Figure 6-9: Mass flow rate of liquid in each segment under non-equilibrium conditions. Separated into liquid water and liquid ammonia. Negative flow rates represent flow towards the suction end.

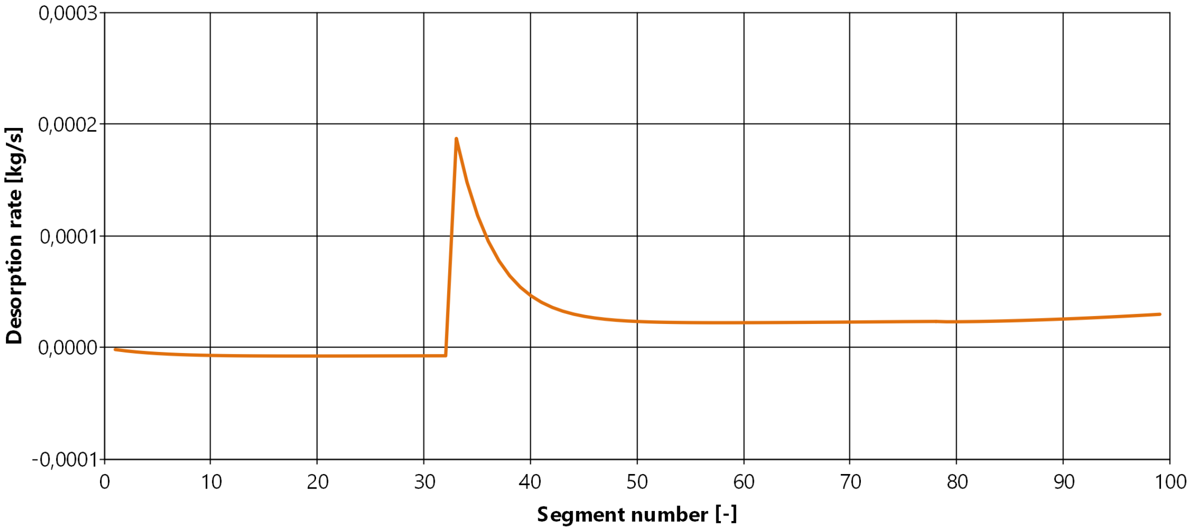


Figure 6-10: Ammonia desorption rate in each segment under non-equilibrium conditions. Negative desorption rates represent ammonia absorption.

reduction in desorption rate leads to a slightly higher ammonia mass fraction for the liquid. In the last segment the ammonia mass fraction in the liquid phase is 0.188, compared to an ammonia mass fraction of 0.173 in the case with complete equilibrium. Furthermore, the desorption rate curve in Figure 6-10 demonstrates how desorption of the injection ammonia is slowed down and spread out over multiple segments, contrary to the single desorption peak in Figure 6-7.

7 Discussion

Two different models have been developed, a screw compressor model in Modelica and a two-phase compression model in EES. The Modelica model is a quasi-one-dimensional numerical model, capable of both transient and steady-state analysis. It can be utilized for simulations of continuous operation of screw compressors, and it can be implemented into bigger system models. The EES model is a quasi-one-dimensional numerical model, capable of steady-state analysis of two-phase compression. This model does not involve the processes of suction and discharge, and it is not made for implementation into bigger system models. The Modelica model is made up of eight control volumes that run in a cycle between the compressor's suction and discharge port. The size of each control volume varies according to an established volume function, with a phase shift of 90° between adjacent control volumes. In contrast, the EES model is made of 100 segments with a fixed position and a fixed volume. The working fluid flows through the segments in the EES model, contrary to the working fluid which is trapped in the control volumes of the Modelica model. These models represent two vastly different modeling approaches, and it is therefore of interest and value to compare the results obtained with each model.

The investigations on the optimal injection port location, presented in section 5.2.7, resulted in a maximum operating temperature of 168 °C in the case with $\theta_{inj} = 450°$. With injection at the corresponding position, segment 33, in the two-phase compression model, the temperature at the end of compression was calculated to be 179 °C. This temperature difference is mainly caused by the different assumptions regarding evaporation in the two models. In the screw compressor model the working fluid is assumed to act as one single, homogeneous unit in each control volume. No distinction between the liquid phase and the vapor phase is made. As soon as the working fluid transitions into the superheated region, there cannot be any liquid present in the control volume. In two-phase compression model the liquid phase and the vapor phase are treated as two separate units. It is ensured that liquid is present in every segment, no matter how superheated the vapor is, by applying that evaporation of water does not occur. This restiction leads to a decrease in the cooling effect of the injection liquid, and thus the two-phase compression model provides a higher maximum temperature than the screw compressor model. The differences in operating temperature do not only concern the maximum temperature. There are also differences in the characteristics of the temperature evolution. By comparing the temperature evolution in Figure 5-31 with the segment temperatures in Figure 6-4, it can be seen that liquid injection in the two-phase compression model results in a much more sudden and intense temperature drop than liquid injection in the screw compressor model. The differences in temperature evolution are caused by the differences in model structure. In the Modelica model the control volume positions change continuously and each control volume passes by the injection port over a certain period of time. In the EES model the segments have a fixed position and all injection occurs at once in the position of the given injection segment. This gives a smooth cooling effect in the screw compressor model and an abrupt cooling effect in the two-phase compression model.

Comparisons of the Modelica results and the EES results also reveal a dissimilarity in terms of pressure evolution. With $\theta_{inj} = 450°$ the screw compressor simulation results in a maximum pressure of 26.7 bar, corresponding to 1.7 bar over-compression. In contrast, the two-phase compression model gives a maximum pressure of 24.9 bar, corresponding to 0.1 bar under-compression. One possible reason for the dissimilarity can be inaccuracies in the two-phase compression model caused by assuming an isentropic process with a constant isentropic exponent, $\kappa = 1.34$. This sort of assumption does not have to made in the Modelica model, because the TILMedia property procedure constantly re-evaluates the isentropic exponent based on the temperature and the pressure conditions. Another reason for the dissimilarity can be the assumption that no water evaporates in the two-phase compression model. The assumption leads to slightly less vapor in the EES calculations compared to the Modelica calculations, and thereby lower vapor density and lower vapor pressure. This may indicate that the developed screw compressor model tends to marginally over-estimate the pressure level. The liquid film, that must be present for safe and efficient operation of the screw compressor, is not present in the screw compressor model, and this may induce a pressure estimation that is slightly higher than the actual pressure level.

In order to establish the accuracy of the developed screw compressor model, and prove that the simulation results are reliable, some sort of model validation must be carried out. Comparison against experimental results can be an effective means for validation. However, experimental results are currently not available, since the compressor under investigation is a new and untested type of compressor. In the literature, a limited amount of research has been carried out on ammonia-water compressors with liquid injection. Furthermore, it is challenging to obtain data from an experimental setup that are detailed enough for complete validation. The amount of data generated in a simulation with the screw compressor model is immense, and a majority of this data is impossible to obtain experimentally. E.g., temperature sensors can easily be placed in the suction line and the discharge line connected to the compressor, but large restrictions apply for sensors that are to be placed within the working chamber of the compressor. Measuring the exact temperature evolution is therefore not possible. Besides, when dealing with two-phase compression, every temperature sensor is likely to be covered by liquid, and thereby they do not provide information about the temperature of the vapor in case of non-equilibrium. A more practical way of validating the screw compressor model can be to compare the simulation results with the results of screw compressor models developed by other researchers. E.g., the model developed by Tian et al. (2017a) was validated against experimental pressure recordings, as displayed in Figure 4-5. Therefore, a certain degree of model validation can be achieved by trying to recreate the same results. As different models require different inputs, this would require some effort to adapt the input parameters of Tian et al. (2017a) so that they suit the required parameters for the screw compressor model developed here. One beneficial feature of the Modelica model is that it is not restricted to a given type of working fluid. Even though only simulations with ammonia-water have been presented here, simulations with other working fluids can be carried out in the exact same manner. Thermophysical data for a wide range of fluids are available through TILMedia.

Experimental results with compression of ammonia-water will become available when the CACHP test rig is put into operation. Based on the simulation results presented in section 5.2, some basic recommendations can be made regarding implementation and operation of the screw compressor in the CACHP system. It is shown that it is desirable to inject liquid with a low ammonia mass fraction. The greatest cooling effect is obtained with injection of pure water. It is also shown that it is slightly beneficial with a low injection temperature. The main portion of the cooling effect is generated by latent heat, but it is proven that sensible heat has an impact as well. Furthermore, liquid should be injected during the beginning of the compression phase or during the suction phase. In the investigations on optimal injection port location by Tian et al. (2017b), it was found that the most energy efficient configuration was to inject 30 % of the liquid during suction and the remaining 70 % of the liquid during compression. The simulation results presented here indicate that such a double injection port configuration may be the most optimal solution, but further investigations are required to establish a complete injection strategy.

Once experimental results from the CACHP test rig have been obtained, parameter fitting can be carried out to find suitable values for the screw compressor model's input parameters. This involves values for the effective flow area of the suction path, A_{suc}, the effective flow area of the discharge path, A_{dis}, and especially the leakage flow coefficient, C_{leak}. Zaytsev (2003) states in his work that the most complicated problem in simulation of twin screw compressors is modeling of the internal flows. It is impracticable to establish a value for C_{leak} solely based on knowledge about the geometry of the compressor. The effective flow area of a leakage path can vary depending on the operating conditions of the compressor. E.g., high temperatures can cause a slight expansion of the rotors, and thus change the geometry of the leakage paths. Therefore, a certain amount of empirical data is required to be able to model leakage flows accurately. The Modelica modeling approach differs greatly from the EES modeling approach in terms of leakage flow modeling. Based on experiences acquired during development of these two models, the Modelica approach is considered to be the most practical approach. Having control volumes that represent moving cavities makes it relatively easy to comprehend what the different leakage paths represent and what their geometries could look like. This is not the case for the EES approach. Having fixed segments leads to a detachment from the actual geometry of the compressor, and thereby makes it challenging to model leakage flows in a comprehensible and correct manner.

In addition to the challenges of leakage flow modeling, it is found truly challenging to model liquid evaporation properly. The developed screw compressor model is based on a vapor-liquid equilibrium assumption. This assumption may not be valid, as liquid injection generates a thin film of liquid on the inner surfaces of the compressor body. At high-speed operation, the temperature of the liquid may not reach the same temperature as the compressed vapor. The two-phase compression model is developed to be able to investigate possible effects of non-equilibrium. However, multiple assumptions must be made in order to perform separate calculations on the liquid phase and the vapor phase. The EES model is therefore based on a higher number of assumptions and simplifications than the Modelica model. One of the primary assumptions is that no phase-change can occur in the water portion of the working fluid. Only the ammonia portion can transition between the two phases. It is considered reasonable to assume no water evaporation due to the fact that water has a much higher saturation temperature than ammonia at the same pressure. On the other hand, it is not reasonable to assume that water will not condense if the vapor is cooled down to saturation. The functionality of the model is therefore restricted, and the model should not be applied in investigations where the cooling effect is so large that the vapor reaches saturation. Furthermore, since water cannot evaporate, the temperature that is calculated with this model should be considered a worst-case scenario. The actual cooling effect will be larger than estimated is water evaporation occurs. Contrary to this, the results from the Modelica simulations should be considered the best-case scenario. The presented temperature evolution will only be achieved under complete equilibrium and with evaporation of all the injection liquid.

7.1 Proposals for Further Model Development

Two models with substantial analysis capabilities have been developed and implemented here. However, the full potential of the models has not been put into effect. Due to the high complexity of the thermodynamic process that takes place within a screw compressor, numerous features can be added to the models to expand their functionality and improve their value. The Modelica model is believed to have the greatest potential for further development. Simulations with this model have proven to be highly efficient and reliable. The TILMedia property procedure is very convenient, and the amount of data available through the procedure is extensive. The EES model is more restricted because of the many underlying assumptions, but it is believed that the model can provide valuable insight and act as supporting tool in further development of two-phase and non-equilibrium models.

To further develop the screw compressor model, mechanical elements should be added. In the current model the mechanical efficiency is 100 % and possible effects of friction are not taken into consideration. Friction does not only increase the work load of the compressor, it also generates heat. This heat it is either transferred to the working fluid or to the compressor body itself, and it should therefore be taken into consideration that friction can affect the performance of the compressor. Moreover, calculation of heat loss is greatly simplified in the current model.

The presented results indicate that heat loss to the compressor body can be neglected, but more comprehensive methods should be applied to establish whether this is valid or not. More detailed heat loss functionality can be added by modifying the heat loss equation for the control volume component, or by implementing additional heat loss components in the model. The TIL library involves multiple heat transfer components, such as heat boundaries, heat capacitors and heat ports. These components can be exploited to build a network of heat flow paths in the screw compressor model.

Improvements in the leakage path structure should be made to increase the accuracy of the screw compressor model. In the current model, only one type of leakage path is implemented. The implemented path represents one of the most influential leakage flows, the flow through the gap between the rotor tips and the compressor casing. Nevertheless, the works of Fleming and Tang (1995) and Zaytsev (2003) show that multiple leakage paths have a substantial influence on the performance of screw compressors. Therefore, leakage paths that represent the sealing line between the two rotors, the gaps at the rotor ends and the blow holes should be implemented in the model as well. The exact same model structure, with flow restrictor components and leakage area functions, can be utilized. Thus little effort is required in terms of modeling, but an intricate investigation to gather data on the geometry of the different paths is needed.

Lastly, it should be considered to implement separate property calculation procedures for the liquid phase and the vapor phase. The TILMedia procedure is based on a vapor-liquid equilibrium assumption. However, it is possible to represent non-equilibrium conditions within a control volume by appointing one TILMedia procedure to the liquid phase and another TILMedia procedure to the vapor phase. With separate property calculations it is possible to simulate cavities where liquid and superheated vapor is present simultaneously. The main challenge with such a solution is that additional equations for calculation of mass and heat transfer between the liquid phase and the vapor phase must be developed. This is where the developed EES model can act as a supporting tool. Calculation results from the two-phase compression model can be used as groundwork for development of suitable mass and heat transfer equations.

If the EES model is to be developed further, phase change of water should be incorporated. Water condensation must be taken into account when the vapor reaches saturation, and efforts should be made to investigate whether water evaporates or not in the high-temperature region. Furthermore, it should be considered to implement the equilibrium correction, C_{eq}, as a variable instead of a constant parameter. E.g., C_{eq} could be a function of temperature level, temperature difference between the liquid and the vapor and/or operational frequency of the compressor, as these are all factors that are likely to affect the degree of equilibrium in each segment. More comprehensive investigations on the effects of non-equilibrium can be carried out if suitable relations for C_{eq} are developed.

8 Conclusion

Computer modeling of screw compressors is a complex and challenging task. However, if managed properly, a computer model can be a powerful, time-saving and cost-effective tool for research and development of energy efficient screw compressors. There is a great variety in possible modeling approaches. Empirical and analytical screw compressor models were found to be unsuited for the objectives of this work. Thus the focus was set on numerical models. Quasi-one-dimensional models were addressed, and two different modeling approaches described by Chamoun et al. (2013) and Tian et al. (2017a) were closely investigated.

Based on knowledge obtained from the literature, a quasi-one-dimensional numerical model for simulation of screw compressors with liquid injection were developed. The model is written in the Modelica modeling language, and it is implemented and solved in the Dymola simulation environment. Taking into consideration the high complexity of the real process, the simulation model is relatively simple. Each model component comprises only a few basic thermodynamic equations that are easy to interpret. Contrary to many other models, no distinction between the suction, compression and discharge phases are made, i.e., the set of equations to be solved is equal throughout the entire compression cycle. Hence, continuous compressor operation can be simulated, and the model can easily be implemented into bigger system models for both steady-state and transient analysis. Additionally, a two-phase compression model were developed using the Engineering Equation Solver (EES). This model is more limited than the Modelica model, and it is only capable of steady-state analysis. However, opposed to the Modelica model, the EES model is capable of estimating effects of non-equilibrium. The most challenging modeling issues were found to be related to modeling of the internal leakage flows, and to develop a justifiable relation between the liquid and the vapor portions of the working fluid.

Simulations of an ammonia-water screw compressor with liquid injection were carried out successfully. Based on the simulation results it is recommended to inject liquid with a low ammonia mass fraction and a low temperature. Moreover, it is recommended to inject liquid at the very beginning of the compression phase or during the suction phase. Through this, some of the primary analysis capabilities of the models were demonstrated, and it reasoned that the Modelica model has great potential for further development. Suggestions on how to improve the model were given. These suggestions involve adding mechanical losses, investigating thermal losses further, expanding the leakage flow structure, and possibly separating the liquid and vapor property calculations.

Going forward, the developed model will be utilized to optimize the configuration and the operating conditions of the screw compressor in a CACHP test rig. To do this properly, a more accurate description of the screw compressor geometry must be developed and implemented in the model. This involves adjusting the model's volume function, suction area function, discharge area function and leakage area function. After doing so, the screw compressor model should be validated against experimental recordings from the test rig. Some degree of validation can also be achieved by recreating the simulation setup of other researchers and comparing the results. The model is flexible and simulations can be carried out with a wide range of different working fluids. Thus comparison with numerous models from the literature is possible. Furthermore, the model is built with focus on reusability and ease of implementation into bigger system models. To take advantage of the model's full potential, a complete HACHP system model should be developed. The TIL library in Modelica offers a range of components that can be applied in combination with the screw compressor model to construct the complete heat pump cycle.

References

ACA, 2018. *Rotary Screw Compressor Working Principle, The Basic Parts and Functions*. [online] Air Compressor Agency. Available at: <https://topaircompressorreviews.com/rotary-screw-compressor-parts-functions-working-principle/> [Accessed 1 Apr. 2020].

Ahrens, M. U., Hafner, A. and Eikevik, T. M., 2019. "Compressors for ammonia-water hybrid absorption-compression heat pumps". In: *8th IIR Conference: Ammonia and CO2 Refrigeration Technologies*. Ohrid: International Institute of Refrigeration.

Altenkirch, E., 1950. "Kompressionskältemaschine mit Lösungskreislauf". In: *Kältetechnik* 2, pp. 251–259, 279–284, 310–315.

Arpagaus, C., Bless, F., Uhlmann, M., Schiffmann, J. and Bertsch, S. S., 2018. "High temperature heat pumps: Market overview, state of the art, research status, refrigerants, and application potentials". In: *Energy* 152, pp. 985–1010.

Bahadori, A., 2017. *Fluid Phase Behavior for Conventional and Unconventional Oil and Gas Reservoirs*. [e-book] Available through the NTNU University Library. Gulf Professional Publishing, pp. 249–291. Available at: <https://www.ntnu.no/ub> [Accessed 20 May 2020].

Bejan, A., 2016. *Advanced Engineering Thermodynamics*. [e-book] Available through the NTNU University Library. John Wiley and Sons, Inc., pp. 1–38. Available at: <https://www.ntnu.no/ub> [Accessed 19 Nov. 2019].

Bommel, L. L. van, 2016. "Thermodynamic Model of a Screw Compressor". Master's thesis. TU Delft.

Brun, K. and Kurz, R., 2018. *Compression Machinery for Oil and Gas*. [e-book] Available through Scribd. Gulf Professional Publishing. Available at: <https://www.scribd.com/> [Accessed 6 Nov. 2019].

Chamoun, M., Rulliere, R., Haberschill, P. and Peureux, J., 2013. "Modelica-based modeling and simulation of a twin screw compressor for heat pump applications". In: *Applied Thermal Engineering* 58, pp. 479–489.

CIBSE, 2016. *Air Conditioning and Refrigeration - CIBSE Guide B3 - 2016*. [e-book] Available through the NTNU University Library. CIBSE, pp. 65–114. Available at: <https://www.ntnu.no/ub> [Accessed 23 Sept. 2019].

EIA, 2017. *International Energy Outlook 2017*. PowerPoint Presentation. [online] U.S. Energy Information Administration, pp. 17–18. Available at: <https://www.eia.gov/outlooks/ieo/pdf/0484(2017).pdf> [Accessed 10 Sept. 2019].

FCS, 2020a. *EES, Engineering Equation Solver*. [online] F-Chart Software. Available at: <http://www.fchartsoftware.com/ees/> [Accessed 3 Apr. 2020].

FCS, 2020b. *NH3H2O (Ammonia-Water)*. [online] F-Chart Software. Available at: <http://fchartsoftware.com/ees/eeshelp/hs700.htm> [Accessed 3 Apr. 2020].

Ferreira, C. A. Infante, Zamfirescu, C. and Zaytsev, D., 2006. “Twin screw oil-free wet compressor for compression–absorption cycle”. In: *International Journal of Refrigeration* 29, pp. 556–565.

Fleming, J. S. and Tang, Y., 1995. “The analysis of leakage in a twin screw compressor and its application to performance improvement”. In: *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 209 (2), pp. 125–136.

Ganesh, N. S. and Srinivas, T., 2011. “Evaluation of thermodynamic properties of ammonia-water mixture up to 100 bar for power application systems”. In: *Journal of Mechanical Engineering Research* 3, pp. 25–39.

Ganesh, N. S. and Srinivas, T., 2017. “Development of thermo-physical properties of aqua ammonia for Kalina cycle system”. In: *International Journal of Materials and Product Technology* 55, pp. 113–141.

Gaspar, P. D. and da Silva, P. D., 2015. *Handbook of Research on Advances and Applications in Refrigeration Systems and Technologies*. [e-book] Available through Knovel. IGI Global. Available at: <https://app.knovel.com/web/> [Accessed 24 Sept. 2019].

Gautschi, W., 2012. *Numerical Analysis*. [e-book] Available through the NTNU University Library. Birkhauser, pp. 325–398. Available at: <https://www.ntnu.no/ub> [Accessed 13 Dec. 2019].

Goossens, M., Teuillieres, C., Riviere, P., Cauret, O. and Marchio, D., 2017. “An Instrumented Method for the Evaluation of Compressor Heat Losses in Heat Pumps On-Field”. In: *12th IEA Heat Pump Conference*. Rotterdam: IEA.

Gudjonsdottir, V., Ferreira, C. A. Infante and Goethals, A., 2019. “Wet compression model for entropy production minimization”. In: *Applied Thermal Engineering* 149, pp. 439–447.

Heiyanthuduwage, M. A., Mounoury, S. and Kovacevic, A., 2011. “Performance prediction methods for screw compressors”. In: *7th International Conference on Compressors and their Systems*. London: Woodhead Publishing.

Huagen, W., Ziwen, X. and Pengcheng, S., 2004. “Theoretical and experimental study on indicator diagram of twin screw refrigeration compressor”. In: *International Journal of Refrigeration* 27, pp. 331–338.

IEA, 2019. *Global Energy & CO2 Status Report 2018*. Annual Report. Paris: International Energy Agency, p. 4.

Itard, L. C. M., 1998. “Wet Compression-Resorption Heat Pump Cycles: Thermodynamic Analysis and Design”. Ph.D. thesis. TU Delft.

Jensen, J. K., 2015. “Industrial heat pumps for high temperature process applications. A numerical study of the ammonia-water hybrid absorption-compression heat pump”. Ph.D. thesis. Kongens Lyngby: DTU Mechanical Engineering.

Jensen, J. K., Reinholdt, L., Markussen, W. B. and Elmegaard, B., 2014. “Investigation of Ammonia/Water Hybrid Absorption/Compression Heat Pumps for Heat Supply Temperatures

Above 100 °C". In: *Proceedings of the International Sorption Heat Pump Conference*. Vol. 1. Center for Environmental Energy Engineering, pp. 311–320.

KBC, 2019. *Software - Simulation and Optimization - Advanced Thermodynamics*. [online] KBC. Available at: <https://www.kbc.global/software/advanced-thermodynamics/> [Accessed 10 Dec. 2019].

Koai, K. and Soedel, W., 1990. "Gas Pulsations in Screw Compressors - Part I: Determination of Port Flow and Interpretation of Periodic Volume Source". In: *International Compressor Engineering Conference*. West Lafayette: Purdue University.

Kovacevic, A., Rane, S. and Stosic, N, 2016. "Computational fluid dynamics in rotary positive displacement screw machines". In: *16th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*. Honolulu: ISROMAC.

Lambers, K. J., 2008. "Isentropic and Volumetric Efficiencies for Compressors with Economizer Port". In: *19th International Compressor Engineering Conference*. West Lafayette: Purdue University.

Lemmon, E. W., Bell, I. H., Huber, M. L. and McLinden, M. O., 2018. *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties - REFPROP*. Version 10.0. Gaithersburg: National Institute of Standards and Technology.

Mansour, A. and Müller, N., 2019. "A Review of Flash Evaporation Phenomena and Resulting Shock Waves". In: *Experimental Thermal and Fluid Science* 107, pp. 146–168.

Mayekawa, 2015. *MYCOM i-Series Screw Compressor Instruction Manual*. [online] Available at: <https://www.mayekawamma.com/pdf/i-series2_IM20150601E%20tornillo.pdf> [Accessed 20 May 2020].

MCE, 2006. *Thermophysical Properties of {NH_3 + H_2O} Mixtures for the Industrial Design of Absorption Refrigeration Equipment*. [online] M. Conde Engineering. Available at: <http://www.mrc-eng.com/news.htm> [Accessed 22 Nov. 2019].

Modelica Association, 2017. *Modelica - A Unified Object-Oriented Language for Systems Modeling: Language Specification*. Version 3.4. [online] Modelica Association. Available at: <https://www.modelica.org/documents/ModelicaSpec34.pdf> [Accessed 12 Dec. 2019].

Mujic, E., Kovacevic, A., Stosic, N. and Smith, I. K., 2009. "Numerical modelling of gas pulsations in a screw compressor". In: *International Conference on Compressors and their Systems*. London: Woodhead Publishing.

NIST, 2019. *NIST Reference Fluid Thermodynamic and Transport Properties Database (REFPROP): Version 10*. [online] National Institute of Standards and Technology. Available at: <https://www.nist.gov/srd/refprop> [Accessed 13 Dec. 2019].

Nordtvedt, S. R., 2005. "Experimental and Theoretical Study of a Compression/Absorption Heat Pump with Ammonia/Water as Working Fluid". Ph.D. thesis. Trondheim: NTNU.

Osenbrück, A., 1895. "Verfahren zur Kälteerzeugung bei Absorptionsmaschinen". DRP 84084 (Germany). Kaiserliche Patentamt.

Rane, S. R., Kovacevic, A., Stosic, N. and Stupple, G., 2018. “Analysis of Water Evaporation in Twin Screw Compressors using Eulerian Multiphase Approach in CFD”. In: *24th International Compressor Engineering Conference*. West Lafayette: Purdue University.

Rattner, A. S. and Garimella, S., 2016. “Fast, stable computation of thermodynamic properties of ammonia-water mixtures”. In: *International Journal of Refrigeration* 62, pp. 39–59.

Singh, P. J. and Bowman, J. L., 1988. “Calculation of Blow-Hole Area for Screw Compressors”. In: *International Compressor Engineering Conference*. West Lafayette: Purdue University.

Sjöholm, L., 1986. “Variable Volume-Ratio and Capacity Control in Twin-Screw Compressors”. In: *International Compressor Engineering Conference*. West Lafayette: Purdue University.

Staedter, M. A. and Garimella, S., 2018. “Thermodynamic Considerations for Optimal Thermal Compressor Design”. In: *International Journal of Refrigeration* 91, pp. 28–38.

Stosic, N., Kovacevic, A., Smith, I. K. and Zhang, W. M., 2005a. *An Investigation of Liquid Injection in Refrigeration Screw Compressors*. [online] Available at: <https://www.researchgate.net/publication/237659424> [Accessed 9 Nov. 2019].

Stosic, N., Smith, I. K. and Kovacevic, A., 2005b. “Numerical Investigation of Heat Transfer on Screw Compressor Rotors”. In: *8th Argentine Congress on Computational Mechanics*. Buenos Aires: AMCA.

Stosic, N., Smith, I. K. and Kovacevic, A., 2005c. *Screw Compressors: Mathematical Modelling and Performance Calculation*. [e-book] Available through the NTNU University Library. Springer. Available at: <https://www.ntnu.no/ub> [Accessed 29 Oct. 2019].

Tabatabaian, M. and Rajput, R. K., 2017. *Advanced Thermodynamics: Fundamentals, Mathematics, Applications*. [e-book] Available through the NTNU University Library. Mercury Learning & Information, pp. 715–801. Available at: <https://www.ntnu.no/ub> [Accessed 16 Sept. 2019].

Tian, Y., Shen, J., Wang, C, Xing, Z. and Wang, X., 2017a. “Modeling and performance study of a water-injected twin-screw water vapor compressor”. In: *International Journal of Refrigeration* 83, pp. 75–87.

Tian, Y., Yuan, H., Wang, C., Wu, H. and Xing, Z., 2017b. “Numerical investigation on mass and heat transfer in an ammonia oil-free twin-screw compressor with liquid injection”. In: *International Journal of Thermal Sciences* 120, pp. 175–184.

TLK-Thermo GmbH, 2020. *TIL Suite - Simulates thermal systems*. [online] Available at: <https://www.tlk-thermo.com/index.php/en/software/til-suite> [Accessed 16 May 2020].

UN, United Nations, 2019. *About the Sustainable Development Goals*. [online] Available at: <https://www.un.org/sustainabledevelopment/sustainable-development-goals/> [Accessed 6 Sept. 2019].

Wang, Z., Wang, H., Wang, J., Li, Q. and Feng, Q., 2019. “Theoretical study on wear characteristics of single screw refrigeration compressor with multicolumn envelope meshing pair”. In: *International Journal of Refrigeration* 102, pp. 1–11.

Wennemar, J., 2009. “Dry Screw Compressor Performance and Application Range”. In: *Proceedings of the Thirty-Eighth Turbomachinery Symposium*. Houston: Turbomachinery Laboratory, Texas A&M University.

Zaytsev, D., 2003. “Development of Wet Compressor for Application in Compression-Resorption Heat Pumps”. Ph.D. thesis. TU Delft.

Zaytsev, D. and Ferreira, C. A. Infante, 2002. “Screw Compressor For Ammonia-Water Heat Pump Lubricated By The Process Mixture”. In: *16th International Compressor Engineering Conference*. West Lafayette: Purdue University.

Ziviani, D., Bell, I., Paepe, M. de and Broek, M van den, 2014. “Comprehensive Model of a Single Screw Expander for ORC-Systems”. In: *22nd International Compressor Engineering Conference*. West Lafayette: Purdue University.

Appendix A Modelica Code

Modelica code for each sub-model and the complete screw compressor model.

A.1 Volume Function

```
model VolumeFunction "Function for the volume of a cavity"

  Modelica.Blocks.Interfaces.RealOutput[3] y = {V, derV, theta} "Volume
     output"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
        rotation=90,
        origin={0,106})));

  parameter Modelica.SIunits.Angle thetaStart = 0 "Start value for
     rotational angle";

  outer Modelica.SIunits.Angle thetaCycle "Total rotation for one
     compression cycle";

  outer Modelica.SIunits.AngularVelocity w "Angular velocity";

  outer Modelica.SIunits.Time tCycle "Duration of one compression cycle"
     ;

  outer Modelica.SIunits.Volume Vmax "Maximum cavity volume";

  Modelica.SIunits.Volume Vmin = 1e-8 "Minimum cavity volume";

  Modelica.SIunits.Volume V "Cavity volume";

  Real derV "Time derivative of cavity volume";

  Modelica.SIunits.Angle theta "Rotational angle";

  Modelica.SIunits.Angle theta1 = thetaCycle/10;

  Modelica.SIunits.Angle theta2 = 4*thetaCycle/10;

  Modelica.SIunits.Angle theta3 = 5*thetaCycle/10;

  Modelica.SIunits.Angle theta4 = 6*thetaCycle/10;

  Modelica.SIunits.Angle theta5 = 9*thetaCycle/10;

  Modelica.SIunits.Time t = tCycle*theta/thetaCycle "Time since
     beginning of compression cycle";

  Modelica.SIunits.Time t1 = tCycle*theta1/thetaCycle;

  Modelica.SIunits.Time t2 = tCycle*theta2/thetaCycle;
  Modelica.SIunits.Time t3 = tCycle*theta3/thetaCycle;

  Modelica.SIunits.Time t4 = tCycle*theta4/thetaCycle;

  Modelica.SIunits.Time t5 = tCycle*theta5/thetaCycle;

  Real dVdt = 2*Vmax/(-t1+t2+t3) "Time derivative of volume in the
     linear region";

initial equation
  theta = thetaStart "Angle initialization";

equation
  der(theta) = w "Time derivative of theta";

  when (theta >= thetaCycle) then
    reinit(theta, 0);
  end when;

  V = smooth(1,
    if (theta < theta1) then dVdt*t^(2)/(2*t1) + Vmin
    else if (theta < theta2) then dVdt*w*(t1)^(2)/(2*theta1) + dVdt*(t-
       t1) + Vmin
    else if (theta < theta4) then dVdt*w*(t1)^(2)/(2*theta1) + dVdt*(t2-
       t1) + dVdt*(t3*t-t^(2)/2-t3*t2+t2^(2)/2)/(t3-t2) + Vmin
    else if (theta < theta5) then Vmax - dVdt*(t4^(2)/2-t3*t4+t3^(2)/2)
       /(t4-t3) - dVdt*(t-t4) + Vmin
    else Vmax - dVdt*(t4^(2)/2-t3*t4+t3^(2)/2)/(t4-t3) - dVdt*(t5-t4) -
       dVdt*(tCycle*t-t^(2)/2-tCycle*t5+t5^(2)/2)/(tCycle-t5) + Vmin);
  //Smooth used to specify that V is continous and continously
     differentiable up to order 1

  derV = smooth(0,
    if (theta < theta1) then dVdt*t/t1
    else if (theta < theta2) then dVdt
    else if (theta < theta4) then dVdt*(t3-t)/(t3-t2)
    else if (theta < theta5) then -dVdt
    else -dVdt*(tCycle-t)/(tCycle-t5));

  annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
     ={
          Rectangle(extent={{-100,100},{100,-100}},
                                                      lineColor={28,108,200},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid,
          lineThickness=0.5),
                                                        Text(
          extent={{-40,80},{40,0}},
          lineColor={28,108,200},
          textString="V"),
        Text(
          extent={{-100,-20},{100,-60}},
          lineColor={28,108,200},
          textString="%thetaStart")}),
                                                   Diagram(
        coordinateSystem(preserveAspectRatio=false)),
    experiment(StopTime=2));
end VolumeFunction;
```

A.2 Control Volume

```
model ControlVolume "Control volume, representing the volume of one
   cavity"

  /*Working fluid type*/
  outer TILMedia.VLEFluidTypes.BaseVLEFluid vleFluidType "VLE fluid type
     ";

  /*Working fluid model*/
  TILMedia.VLEFluid_ph workingFluid(
    final p=p,
    final h=h,
    final xi=xi,
    final vleFluidType=vleFluidType) "Working fluid model"
    annotation (Placement(transformation(extent={{-10,-12},{10,8}})));

  /*Connection ports*/
  TIL.Connectors.VLEFluidPort suctionPort(final vleFluidType =
     vleFluidType) "The cv's suction port"
    annotation (Placement(transformation(extent={{-110,30},{-90,50}})));
  TIL.Connectors.VLEFluidPort dischargePort(final vleFluidType =
     vleFluidType) "The cv's discharge port"
    annotation (Placement(transformation(extent={{90,-50},{110,-30}})));
  TIL.Connectors.VLEFluidPort injectionPort(final vleFluidType =
     vleFluidType) "The cv's injection port"
    annotation (Placement(transformation(extent={{-10,90},{10,110}})));
  TIL.Connectors.VLEFluidPort leakInPort(final vleFluidType =
     vleFluidType) "The cv's leak-in port"
    annotation (Placement(transformation(extent={{90,30},{110,50}})));
  TIL.Connectors.VLEFluidPort leakOutPort(final vleFluidType =
     vleFluidType) "The cv's leak-out port"
    annotation (Placement(transformation(extent={{-110,-50},{-90,-30}}))
       );

  /*Volume input*/
  Modelica.Blocks.Interfaces.RealInput volumeInput[3] "Volume input: {V,
      dVdt, theta}"
    annotation (Placement(transformation(extent
       ={{-140,-110},{-100,-70}})));

  /*Start values*/
  parameter Modelica.SIunits.AbsolutePressure pStart "Start value for
     pressure";

  parameter Modelica.SIunits.SpecificEnthalpy hStart "Start value for
     specific enthalpy";
```

```
parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiStart =
   zeros(vleFluidType.nc-1) "Start value for mass fraction";

/*Control volume properties*/
Modelica.SIunits.Volume V = volumeInput[1] "Volume of control volume";

Real dVdt = volumeInput[2] "Time derivative of volume";

Modelica.SIunits.Angle theta = volumeInput[3] "Rotational angle";

Modelica.SIunits.Mass m "Total fluid mass in control volume";

Modelica.SIunits.AbsolutePressure p "Pressure in control volume";

Modelica.SIunits.SpecificEnthalpy h "Specific enthalpy in control
   volume";

Modelica.SIunits.MassFraction[vleFluidType.nc-1] xi "Mass fraction of
   fluid component i";

Modelica.SIunits.SpecificInternalEnergy u "Specific internal energy in
    control volume";

/*Suction properties*/
outer Modelica.SIunits.AbsolutePressure pSuction "Pressure in suction
   line";

outer Modelica.SIunits.SpecificEnthalpy hSuction "Specific enthalpy in
    suction line";

outer Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiSuction "Mass
    fraction in suction line";

/*Compressor features*/
outer Modelica.SIunits.Angle thetaCycle "Total rotation for one
   compression cycle";

outer Modelica.SIunits.CoefficientOfHeatTransfer alpha "Heat transfer
   coefficient";

outer Modelica.SIunits.Temperature Tbody "Average compressor body
   temperature";

Modelica.SIunits.Power Wdot "Compression work rate";
```

```
  Modelica.SIunits.HeatFlowRate QdotBody "Heat flow rate from the
      working fluid to the compressor body";

  /*Initial equations*/
initial equation
  p = pStart "Pressure initialization";

  h = hStart "Specific enthalpy initialization";

  xi = xiStart "Mass fraction initialization";

  /*Equations*/
equation
  m = workingFluid.d*V "Mass given by density and volume";

  h = u + p/workingFluid.d "The definition of specific enthalpy";

  Wdot = -p*dVdt "Work rate";

  QdotBody = alpha*V^(2/3)*(workingFluid.T - Tbody) "Heat loss rate";

  der(m*u) = Wdot - QdotBody + noEvent(suctionPort.m_flow*actualStream(
      suctionPort.h_outflow) + dischargePort.m_flow*actualStream(
      dischargePort.h_outflow)
   + injectionPort.m_flow*actualStream(injectionPort.h_outflow) +
       leakInPort.m_flow*actualStream(leakInPort.h_outflow)
   + leakOutPort.m_flow*actualStream(leakOutPort.h_outflow)) "Energy
       balance";

  der(m) = suctionPort.m_flow + dischargePort.m_flow +
      injectionPort.m_flow + leakInPort.m_flow + leakOutPort.m_flow "
      Total mass balance";

  der(m*xi) = noEvent(suctionPort.m_flow*actualStream(
      suctionPort.xi_outflow) + dischargePort.m_flow*actualStream(
      dischargePort.xi_outflow)
   + injectionPort.m_flow*actualStream(injectionPort.xi_outflow) +
       leakInPort.m_flow*actualStream(leakInPort.xi_outflow)
   + leakOutPort.m_flow*actualStream(leakOutPort.xi_outflow)) "Mass
       balance for component i";

  p = suctionPort.p;
  p = dischargePort.p;
  p = injectionPort.p;
  p = leakInPort.p;
  p = leakOutPort.p; //Homogeneous pressure in the control volume
```

```
h = suctionPort.h_outflow;
h = dischargePort.h_outflow;
h = injectionPort.h_outflow;
h = leakInPort.h_outflow;
h = leakOutPort.h_outflow; //Homogeneous specific enthalpy in the
   control volume

xi = suctionPort.xi_outflow;
xi = dischargePort.xi_outflow;
xi = injectionPort.xi_outflow;
xi = leakInPort.xi_outflow;
xi = leakOutPort.xi_outflow; //Homogeneous mass fraction in the
   control volume

when (theta < thetaCycle/2) then
  reinit(p, pSuction);
  reinit(h, hSuction);
  if (vleFluidType.nc > 1) then
    reinit(xi, xiSuction);
  end if;
end when "Re-initialize when compression cycle starts over";

suctionPort.h_limit = -1e6;
dischargePort.h_limit = -1e6;
injectionPort.h_limit = -1e6;
leakInPort.h_limit = -1e6;
leakOutPort.h_limit = -1e6; //Unused limit parameters

/*Control volume model icon*/
annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
   ={
        Rectangle(
        extent={{-100,100},{100,-100}},
        lineColor={0,0,0},
        lineThickness=1,
        fillColor={188,189,255},
        fillPattern=FillPattern.Backward), Text(
        extent={{-60,60},{60,-60}},
        lineColor={0,0,0},
        lineThickness=1,
        fillColor={188,189,255},
        fillPattern=FillPattern.Solid,
        textString="CV"),
      Text(
        extent={{-100,-120},{100,-160}},
        lineColor={0,0,0},
        lineThickness=0.5,
          fillColor={0,0,0},
          fillPattern=FillPattern.Solid,
          textString="%name")}),                                 Diagram
             (
        coordinateSystem(preserveAspectRatio=false)));
end ControlVolume;
```

A.3 Flow Restrictor

```
model FlowRestrictor "Component for closing and opening of flow paths"

  /*Working fluid type*/
  outer TILMedia.VLEFluidTypes.BaseVLEFluid vleFluidType "VLE fluid type
     ";

  /*Area input*/
  Modelica.Blocks.Interfaces.RealInput effectiveFlowAreaInput "Effective
      flow area input"
    annotation (Placement(transformation(
        origin={0,50},
        extent={{-10,-10},{10,10}},
        rotation=270)));

  Modelica.SIunits.Area AreaEff = effectiveFlowAreaInput "Effective flow
      area";

  /*Connection ports*/
  TIL.Connectors.VLEFluidPort portA(final vleFluidType = vleFluidType) "
     portA"
    annotation (Placement(transformation(extent={{-90,-10},{-70,10}},
       rotation=
            0)));
  TIL.Connectors.VLEFluidPort portB(final vleFluidType = vleFluidType) "
     portB"
    annotation (Placement(transformation(extent={{70,-10},{90,10}},
       rotation=0)));

  /*Fluid models*/
  TILMedia.VLEFluid_ph vleFluidA(
    p=portA.p, h=inStream(portA.h_outflow),
    xi = inStream(portA.xi_outflow),
    final vleFluidType = vleFluidType)
    annotation (Placement(transformation(extent={{-100,20},{-80,40}},
       rotation=0)));
  TILMedia.VLEFluid_ph vleFluidB(
    p=portB.p,
    h=inStream(portB.h_outflow),
    xi = inStream(portB.xi_outflow),
    final vleFluidType = vleFluidType)
    annotation (Placement(transformation(extent={{80,20},{100,40}},
       rotation=0)));

  /*Equations*/
equation
```

```
portA.xi_outflow = inStream(portB.xi_outflow);
portB.xi_outflow = inStream(portA.xi_outflow); //Mass fraction

portA.h_outflow = inStream(portB.h_outflow);
portB.h_outflow = inStream(portA.h_outflow); //Specific enthalpy

portA.h_limit = -1e6;
portB.h_limit = -1e6; //Unused limit parameters

portB.m_flow + portA.m_flow = 0 "Mass balance";

if noEvent(portA.p > portB.p) then
  portA.m_flow = AreaEff*TIL.Utilities.Numerics.squareRootFunction(2*
     vleFluidA.d*(portA.p - portB.p), 1e-6);
  //Flow direction A to B, sqrtFunction approximation below sqrt(1e-6)
else
  portA.m_flow = AreaEff*TIL.Utilities.Numerics.squareRootFunction(2*
     vleFluidB.d*(portA.p - portB.p), 1e-6);
  //Flow direction B to A, sqrtFunction returns negative value
end if;

/*Flow restrictor model icon*/
annotation (defaultComponentName="flowRestrictor",Icon(
   coordinateSystem(
        preserveAspectRatio=true, extent={{-80,-40},{80,40}}),
                                              graphics={
      Polygon(
        points={{-10,60},{10,60},{0,40},{-10,60}},
        lineColor={0,0,255},
        pattern=LinePattern.None,
        fillColor={175,175,175},
        fillPattern=FillPattern.Solid),
      Rectangle(
        extent={{-80,20},{80,-20}},
        lineColor={0,0,0},
        lineThickness=0.5),
      Ellipse(
        extent={{-80,32},{80,8}},
        lineColor={0,0,0},
        lineThickness=0.5,
        startAngle=0,
        endAngle=180,
        fillColor={0,0,0},
        fillPattern=FillPattern.Solid),
      Ellipse(
        extent={{-80,-32},{80,-8}},
        lineColor={0,0,0},
          lineThickness=0.5,
          startAngle=0,
          endAngle=180,
          fillColor={0,0,0},
          fillPattern=FillPattern.Solid),
        Text(
          extent={{-80,-40},{80,-60}},
          lineColor={0,0,0},
          lineThickness=0.5,
          fillColor={0,0,0},
          fillPattern=FillPattern.Solid,
          textString="%name")}),
    Diagram(coordinateSystem(preserveAspectRatio=true, extent
       ={{-100,-100},{100,
            100}})));
end FlowRestrictor;
```

A.4 Suction Area Function

```
model SuctionAreaFunction "Function for the effective flow area of a
   suction path"

  Modelica.Blocks.Interfaces.RealOutput y = Aeff
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
        rotation=0,
        origin={106,0})));

  parameter Modelica.SIunits.Angle thetaStart = 0 "Start value for
     rotational angle";

  outer Modelica.SIunits.Angle thetaCycle "Total rotation for one
     compression cycle";

  outer Modelica.SIunits.AngularVelocity w "Angular velocity";

  outer Modelica.SIunits.Time tCycle "Duration of one compression cycle"
     ;

  outer Modelica.SIunits.Area Asuction "Maximum effective flow area of
     suction path";

  Modelica.SIunits.Area Aeff "Effective flow area of suction path";

  Modelica.SIunits.Angle theta "Rotational angle";

  Modelica.SIunits.Angle theta1 = thetaCycle*t1/tCycle;

  Modelica.SIunits.Angle theta2 = thetaCycle*t2/tCycle;

  Modelica.SIunits.Angle theta3 = thetaCycle*t3/tCycle;

  Modelica.SIunits.Angle theta4 = thetaCycle*t4/tCycle;

  Modelica.SIunits.Angle theta5 = thetaCycle*t5/tCycle;

  Modelica.SIunits.Time t = tCycle*theta/thetaCycle "Time since
     beginning of compression cycle";

  Modelica.SIunits.Time t1 = 0.5*tCycle/20;

  Modelica.SIunits.Time t2 = 2*tCycle/20;

  Modelica.SIunits.Time t3 = 2.5*tCycle/20;
  Modelica.SIunits.Time t4 = 9*tCycle/20;

  Modelica.SIunits.Time t5 = 10*tCycle/20;

  Real dAdt = 2*Asuction/(-t1+t2+t3) "Time derivative of area in the
     linear region";

initial equation
  theta = thetaStart "Angle initialization";

equation
  der(theta) = w "Time derivative of theta";

  when (theta >= thetaCycle) then
    reinit(theta, 0);
  end when;

  if noEvent(theta < theta1) then
    Aeff = dAdt*t^(2)/(2*t1);
  elseif noEvent(theta < theta2) then
    Aeff = dAdt*t1/2 + dAdt*(t-t1);
  elseif noEvent(theta < theta3) then
    Aeff = dAdt*t1/2 + dAdt*(t2-t1) + dAdt*(t3*t-t^(2)/2-t3*t2+t2^(2)/2)
       /(t3-t2);
  elseif noEvent(theta < theta4) then
    Aeff = Asuction;
  elseif noEvent(theta < theta5) then
    Aeff = Asuction*(1-(t-t4)/(t5-t4));
  else
    Aeff = 0;
  end if;

  annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
     ={
          Rectangle(extent={{-100,100},{100,-100}},
                                                      lineColor={238,46,47},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid),
                                                  Text(
          extent={{-40,80},{40,0}},
          lineColor={238,46,47},
          textString="A"),
        Text(
          extent={{-100,-20},{100,-60}},
          lineColor={238,46,47},
          textString="%thetaStart"),
                                                       Text(
     extent={{4,42},{100,10}},
     lineColor={238,46,47},
     textString="suc")}),                              Diagram(
    coordinateSystem(preserveAspectRatio=false)),
  experiment(StopTime=2));
end SuctionAreaFunction;
```

A.5 Discharge Area Function

```
model DischargeAreaFunction "Function for the effective flow area of a
   discharge path"

  Modelica.Blocks.Interfaces.RealOutput y = Aeff
    annotation (Placement(transformation(extent={{10,-10},{-10,10}},
        rotation=0,
        origin={-106,0})));

  parameter Modelica.SIunits.Angle thetaStart = 0 "Start value for
     rotational angle";

  outer Modelica.SIunits.Angle thetaCycle "Total rotation for one
     compression cycle";

  outer Modelica.SIunits.AngularVelocity w "Angular velocity";

  outer Modelica.SIunits.Time tCycle "Duration of one compression cycle"
     ;

  outer Modelica.SIunits.Area Adischarge "Maximum effective flow area of
      discharge path";

  outer Modelica.SIunits.Time tDischargeOpen "Point in time when the
     discharge port opens";

  Modelica.SIunits.Area Aeff "Effective flow area of suction path";

  Modelica.SIunits.Angle theta "Rotational angle";

  Modelica.SIunits.Angle theta1 = thetaCycle*tDischargeOpen/tCycle;

  Modelica.SIunits.Angle theta2 = thetaCycle*t2/tCycle;

  Modelica.SIunits.Angle theta3 = thetaCycle*t3/tCycle;

  Modelica.SIunits.Angle theta4 = thetaCycle*t4/tCycle;

  Modelica.SIunits.Time t = tCycle*theta/thetaCycle "Time since
     beginning of compression cycle";

  Modelica.SIunits.Time t2 = tDischargeOpen+0.4*(tCycle-tDischargeOpen);

  Modelica.SIunits.Time t3 = tDischargeOpen+0.5*(tCycle-tDischargeOpen);

  Modelica.SIunits.Time t4 = tDischargeOpen+0.6*(tCycle-tDischargeOpen);
  Real dAdt = 2*Adischarge/(-2*tDischargeOpen+t2+t3) "Time derivative of
      area in the linear region";

initial equation
  theta = thetaStart "Angle initialization";

equation
  der(theta) = w "Time derivative of theta";

  when (theta >= thetaCycle) then
    reinit(theta, 0);
  end when;

  if noEvent(theta < theta1) then
    Aeff = 0;
  elseif noEvent(theta < theta2) then
    Aeff = dAdt*(t-tDischargeOpen);
  elseif noEvent(theta < theta4) then
    Aeff = dAdt*(t2-tDischargeOpen) + dAdt*(t3*t-t^(2)/2-t3*t2+t2^(2)/2)
       /(t3-t2);
  else
    Aeff = dAdt*(t2-tDischargeOpen) - dAdt*(t-t4);
  end if;

  annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
     ={
          Rectangle(extent={{-100,100},{100,-100}},
                                                lineColor={238,46,47},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid),
                                            Text(
          extent={{-40,80},{40,0}},
          lineColor={238,46,47},
          textString="A"),
        Text(
          extent={{-100,-20},{100,-60}},
          lineColor={238,46,47},
          textString="%thetaStart"),
                                                 Text(
          extent={{4,42},{100,10}},
          lineColor={238,46,47},
          textString="dis")}),
                                                           Diagram(
        coordinateSystem(preserveAspectRatio=false)),
    experiment(StopTime=2));
end DischargeAreaFunction;
```

A.6 Leakage Area Function

```
model LeakageAreaFunction "Function for the effective flow area of a
    leakage path"

  Modelica.Blocks.Interfaces.RealInput x1[3] "Volume input 1: {V, dVdt,
      theta}"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
        rotation=90,
        origin={-40,-104})));
  Modelica.Blocks.Interfaces.RealInput x2[3] "Volume input 2: {V, dVdt,
      theta}"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
        rotation=90,
        origin={40,-104})));
  Modelica.Blocks.Interfaces.RealOutput y = Aeff "Effective flow area
      output"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
        rotation=90,
        origin={40,106})));

  Modelica.SIunits.Volume V1 = x1[1] "Volume of control volume 1";

  Real derV1 = x1[2] "Time derivative of volume 1";

  Modelica.SIunits.Angle theta1 = x1[3] "Rotational angle of control
      volume 1";

  Modelica.SIunits.Volume V2 = x2[1] "Volume of control volume 2";

  Real derV2 = x2[2] "Time derivative of volume 2";

  outer Real Cleak "Leakage flow coefficient";

  outer Modelica.SIunits.Angle thetaDischargeOpen "Rotational angle
      where discharge begins";

  Modelica.SIunits.Area Aeff "Effective flow area of leakage path";

equation
  if noEvent((derV2 < 0) and (theta1 < thetaDischargeOpen)) then
    Aeff = Cleak*min(V1, V2);
    //Leakage from cv2 to cv1 when cv2 is in the compression/discharge
        phase AND cv1 has not reached the discharge phase yet
  else
    Aeff = 0;
  end if;
  annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
     ={
          Rectangle(extent={{-100,100},{100,-100}},
                                                       lineColor={238,46,47},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid),
                                                   Text(
          extent={{-40,80},{40,0}},
          lineColor={238,46,47},
          textString="A"),
        Text(
          extent={{-100,-20},{100,-60}},
          lineColor={238,46,47},
          textString="Leak")}),
                                                              Diagram(
        coordinateSystem(preserveAspectRatio=false)),
    experiment(StopTime=2));
end LeakageAreaFunction;
```

A.7 Injector

```
model Injector "Component used to regulate the injection mass flow rate"

  /*Working fluid type*/
  outer TILMedia.VLEFluidTypes.BaseVLEFluid vleFluidType "VLE fluid type
     ";

  /*Connection ports*/
  TIL.Connectors.VLEFluidPort inPort(final vleFluidType=vleFluidType)
    "Port where fluid flows in from the injection line"
    annotation (Placement(transformation(extent={{-10,50},{10,70}}),
        iconTransformation(extent={{-10,50},{10,70}})));
  TIL.Connectors.VLEFluidPort outPort(final vleFluidType=vleFluidType)
    "Port where fluid flows out to the control volume"
    annotation (Placement(transformation(extent={{-10,-70},{10,-50}}),
        iconTransformation(extent={{-10,-70},{10,-50}})));

  /*Flow input*/
  Modelica.Blocks.Interfaces.RealInput m_flowInput "Mass flow rate input
     "
    annotation (Placement(transformation(extent={{70,-20},{30,20}})));

  Modelica.SIunits.MassFlowRate m_flow = m_flowInput "Mass flow rate";

  /*Equations*/
equation
  inPort.m_flow = m_flow;
  outPort.m_flow = -inPort.m_flow; //Mass flow rate

  inPort.xi_outflow = inStream(outPort.xi_outflow);
  outPort.xi_outflow = inStream(inPort.xi_outflow); //Mass fraction

  inPort.h_outflow = inStream(outPort.h_outflow);
  outPort.h_outflow = inStream(inPort.h_outflow); //Specific enthalpy

  inPort.h_limit = -1e6;
  outPort.h_limit = -1e6; //Unused limit parameters

  /*Injector model icon*/
  annotation (Icon(coordinateSystem(preserveAspectRatio=false, extent
     ={{-100,-100},{100,100}}), graphics={
                                        Line(
          points={{0,60},{0,-60}},
          color={0,0,0},
          thickness=1),
        Line(
          points={{0,-20},{20,20}},
          color={0,0,0},
          thickness=1),
        Line(
          points={{0,-20},{-20,20}},
          color={0,0,0},
          thickness=1),
        Text(
          extent={{-60,20},{60,-20}},
          lineColor={0,0,0},
          lineThickness=1,
          textString="%name",
          origin={-60,-7.10543e-15},
          rotation=-90)}),                                      Diagram
             (coordinateSystem(preserveAspectRatio=false, extent
             ={{-100,-100},{100,100}})));
end Injector;
```

A.8 Injection Flow Function

```
model InjectionFlowFunction "Function for the injection mass flow rate"

  Modelica.Blocks.Interfaces.RealOutput y = mDot "Mass flow rate output"
    annotation (Placement(transformation(extent={{10,-10},{-10,10}},
        rotation=0,
        origin={-106,0})));

  parameter Modelica.SIunits.Angle thetaStart = 0 "Start value for
     rotational angle";

  outer Modelica.SIunits.Angle thetaCycle "Total rotation for one
     compression cycle";

  outer Modelica.SIunits.AngularVelocity w "Angular velocity";

  outer Modelica.SIunits.Time tCycle "Duration of one compression cycle"
     ;

  outer Modelica.SIunits.MassFlowRate mDotInjection "Injection mass flow
      rate";

  outer Modelica.SIunits.Angle injectionAngle "Rotational angle where
     injection begins";

  Modelica.SIunits.MassFlowRate mDot "Mass flow rate";

  Modelica.SIunits.Angle theta "Rotational angle";

  Modelica.SIunits.Angle theta2 = injectionAngle + 0.02;

  Modelica.SIunits.Angle theta3 = injectionAngle + 0.04;

  Modelica.SIunits.Angle theta4 = theta6 - 0.04;

  Modelica.SIunits.Angle theta5 = theta6 - 0.02;

  Modelica.SIunits.Angle theta6 = injectionAngle+thetaCycle/8 + 0.04;

  Modelica.SIunits.Time t = tCycle*theta/thetaCycle "Time since
     beginning of compression cycle";

  Modelica.SIunits.Time t1 = tCycle*injectionAngle/thetaCycle;

  Modelica.SIunits.Time t2 = tCycle*theta2/thetaCycle;
  Modelica.SIunits.Time t3 = tCycle*theta3/thetaCycle;

  Modelica.SIunits.Time t4 = tCycle*theta4/thetaCycle;

  Modelica.SIunits.Time t5 = tCycle*theta5/thetaCycle;

  Modelica.SIunits.Time t6 = tCycle*theta6/thetaCycle;

  Real dmDotdt = 2*mDotInjection/(t3-t1) "Time derivative of mass flow
     rate at maximum slope";

initial equation
  theta = thetaStart "Angle initialization";

equation
  der(theta) = w "Time derivative of theta";

  when (theta >= thetaCycle) then
    reinit(theta, 0);
  end when;

  mDot = smooth(1,
    if (theta < injectionAngle) then 0
    else if (theta < theta2) then dmDotdt*(t^(2)/2 - t1*t + t1^(2)/2)/(
       t2-t1)
    else if (theta < theta3) then mDotInjection/2 + dmDotdt*(t3*t - t
       ^(2)/2 - t3*t2 + t2^(2)/2)/(t3-t2)
    else if (theta < theta4) then mDotInjection
    else if (theta < theta5) then mDotInjection - dmDotdt*(t^(2)/2 - t4*
       t + t4^(2)/2)/(t5-t4)
    else if (theta < theta6) then mDotInjection/2 - dmDotdt*(t6*t - t
       ^(2)/2 - t6*t5 + t5^(2)/2)/(t6-t5)
    else 0);

  annotation (Icon(coordinateSystem(preserveAspectRatio=false), graphics
     ={
          Rectangle(extent={{-100,100},{100,-100}},
                                                  lineColor={217,67,180},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid),
                                              Text(
          extent={{-80,80},{80,0}},
          lineColor={217,67,180},
          textString="flow"),
        Text(
          extent={{-100,-20},{100,-60}},
          lineColor={217,67,180},
          textString="%thetaStart")}),
                                                              Diagram(
        coordinateSystem(preserveAspectRatio=false)),
    experiment(StopTime=2));
end InjectionFlowFunction;
```

A.9 Junction

```
model Junction "Component used to merge or separate flows"

  /*Working fluid type*/
  outer TILMedia.VLEFluidTypes.BaseVLEFluid vleFluidType "VLE fluid type
     ";

  /*Connection ports*/
  TIL.Connectors.VLEFluidPort mergePort(final vleFluidType =
     vleFluidType) "Port for merged flow"
    annotation (Placement(transformation(extent={{-10,30},{10,50}})));
  TIL.Connectors.VLEFluidPort separatePort1(final vleFluidType=
     vleFluidType) "Port for separate flow 1"
    annotation (Placement(transformation(extent={{-50,-10},{-30,10}})));
  TIL.Connectors.VLEFluidPort separatePort2(final vleFluidType=
     vleFluidType) "Port for separate flow 2"
    annotation (Placement(transformation(extent={{-10,-50},{10,-30}})));

  /*Start values*/
  parameter Modelica.SIunits.SpecificEnthalpy hStart "Start value for
     specific enthalpy";

  parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiStart =
     zeros(vleFluidType.nc-1) "Start value for mass fraction";

  /*Junction properties*/
  Modelica.SIunits.Mass m = 1e-8 "Fluid mass in junction";

  Modelica.SIunits.SpecificEnthalpy h "Specific enthalpy in junction";

  Modelica.SIunits.MassFraction[vleFluidType.nc-1] xi "Mass fraction of
     fluid component i";

  /*Initial equations*/
initial equation
  h = hStart "Specific enthalpy initialization";

  xi = xiStart "Mass fraction initialization";

  /*Equations*/
equation
  separatePort1.m_flow + separatePort2.m_flow + mergePort.m_flow = 0 "
     Total mass balance";
  m*der(xi) = noEvent(separatePort1.m_flow*actualStream(
     separatePort1.xi_outflow) + separatePort2.m_flow*actualStream(
     separatePort2.xi_outflow)
    + mergePort.m_flow*actualStream(mergePort.xi_outflow)) "Mass balance
        for component i";

  m*der(h) = noEvent(separatePort1.m_flow*actualStream(
     separatePort1.h_outflow) + separatePort2.m_flow*actualStream(
     separatePort2.h_outflow)
    + mergePort.m_flow*actualStream(mergePort.h_outflow)) "Energy
       balance";

  mergePort.p = separatePort1.p;
  mergePort.p = separatePort2.p; //Pressure

  xi = separatePort1.xi_outflow;
  xi = separatePort2.xi_outflow;
  xi = mergePort.xi_outflow; //Mass fraction

  h = separatePort1.h_outflow;
  h = separatePort2.h_outflow;
  h = mergePort.h_outflow; //Specific enthalpy

  separatePort1.h_limit = -1e6;
  separatePort2.h_limit = -1e6;
  mergePort.h_limit = -1e6; //Unused limit parameters

  /*Junction model icon*/
  annotation (Icon(coordinateSystem(preserveAspectRatio=false, extent
     ={{-100,-100},{100,100}}), graphics={Ellipse(
          extent={{-80,80},{0,0}},
          lineColor={0,0,0},
          lineThickness=1,
          startAngle=0,
          endAngle=90,
          closure=EllipseClosure.None), Line(
          points={{0,40},{0,-40}},
          color={0,0,0},
          thickness=1)}),                                           Diagram
             (coordinateSystem(preserveAspectRatio=false, extent
             ={{-100,-100},{100,100}})));
end Junction;
```

A.10 The Complete Screw Compressor

```
model ScrewCompressor "Screw compressor model with 8 control volumes"

  /*SIM*/
  outer TIL.SystemInformationManager sim "System information manager";

  /*Working fluid type*/
  inner TILMedia.VLEFluidTypes.BaseVLEFluid vleFluidType =
      sim.vleFluidType1 "VLE fluid type";

  /*Compressor features*/
  parameter Modelica.SIunits.Frequency f = 2950/60 "Operational
      frequency of the compressor";

  inner Modelica.SIunits.AngularVelocity w = 2*Modelica.Constants.pi*f "
      Angular velocity";

  inner parameter Modelica.SIunits.Angle thetaCycle = 12.566370614359 "
      Total rotation for one compression cycle";

  inner Modelica.SIunits.Time tCycle = thetaCycle/w "Duration of one
      compression cycle";

  inner parameter Modelica.SIunits.Volume Vmax = 0.0003345 "Maximum
      cavity volume";

  parameter Real Vi = 3.65 "Built-in volume ratio";

  inner parameter Real Cleak = 0 "Leakage flow coefficient";

  inner parameter Modelica.SIunits.Area Asuction = 0.005 "Maximum
      effective flow area of suction path";

  inner parameter Modelica.SIunits.Area Adischarge = 0.001 "Maximum
      effective flow area of discharge path";

  inner Modelica.SIunits.Time tDischargeOpen = V1.t4 + ((Vmax - V1.dVdt
      *(V1.t4-V1.t3)/2)-Vmax/Vi)/V1.dVdt "Point in time when the
      discharge port opens";

  inner Modelica.SIunits.Angle thetaDischargeOpen = thetaCycle*
      tDischargeOpen/tCycle "Rotational angle where discharge begins";

  inner parameter Modelica.SIunits.Angle injectionAngle = 7
      .8539816339745 "Rotational angle where injection begins";
```

```
inner parameter Modelica.SIunits.MassFlowRate mDotInjection = 0 "
   Injection mass flow rate";

inner parameter Modelica.SIunits.CoefficientOfHeatTransfer alpha = 0 "
   Heat transfer coefficient";

inner parameter Modelica.SIunits.Temperature Tbody = 373.15 "Average
   compressor body temperature";

/*Suction properties*/
inner Modelica.SIunits.AbsolutePressure pSuction = suctionPort.p "
   Pressure in suction line";

inner Modelica.SIunits.SpecificEnthalpy hSuction = inStream(
   suctionPort.h_outflow) "Specific enthalpy in suction line";

inner Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiSuction =
   inStream(suctionPort.xi_outflow) "Mass fraction in suction line";

/*Start values*/
parameter Modelica.SIunits.AbsolutePressure pSuctionStart = 500000 "
   Start value for suction pressure"
  annotation(Dialog(group="Pressure", tab="Start values"));

parameter Modelica.SIunits.AbsolutePressure pStart_cv5 = 500000 "Start
    value for pressure in control volume 5"
  annotation(Dialog(group="Pressure", tab="Start values"));

parameter Modelica.SIunits.AbsolutePressure pStart_cv6 = 653000 "Start
    value for pressure in control volume 6"
  annotation(Dialog(group="Pressure", tab="Start values"));

parameter Modelica.SIunits.AbsolutePressure pStart_cv7 = 1213000 "
   Start value for pressure in control volume 7"
  annotation(Dialog(group="Pressure", tab="Start values"));

parameter Modelica.SIunits.AbsolutePressure pDischargeStart = 2500000
   "Start value for discharge pressure"
  annotation(Dialog(group="Pressure", tab="Start values"));

parameter Modelica.SIunits.SpecificEnthalpy hSuctionStart = 1737e3 "
   Start value for specific enthalpy at suction"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));

parameter Modelica.SIunits.SpecificEnthalpy hStart_cv5 = 1737e3 "Start
    value for specific enthalpy in control volume 5"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));
```

```
parameter Modelica.SIunits.SpecificEnthalpy hStart_cv6 = 1779e3 "Start
    value for specific enthalpy in control volume 6"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));

parameter Modelica.SIunits.SpecificEnthalpy hStart_cv7 = 1888e3 "Start
    value for specific enthalpy in control volume 7"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));

parameter Modelica.SIunits.SpecificEnthalpy hDischargeStart = 2034e3 "
   Start value for specific enthalpy at discharge"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));

parameter Modelica.SIunits.SpecificEnthalpy hInjectionStart = 1737e3 "
   Start value for specific enthalpy of injection fluid"
  annotation(Dialog(group="Specific enthalpy", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1]
   xiSuctionStart = zeros(vleFluidType.nc-1) "Start value for suction
   mass fraction"
  annotation(Dialog(group="Mass fraction", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiStart_cv5
    = zeros(vleFluidType.nc-1) "Start value for mass fraction in
   control volume 5"
  annotation(Dialog(group="Mass fraction", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiStart_cv6
    = zeros(vleFluidType.nc-1) "Start value for mass fraction in
   control volume 6"
  annotation(Dialog(group="Mass fraction", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1] xiStart_cv7
    = zeros(vleFluidType.nc-1) "Start value for mass fraction in
   control volume 7"
  annotation(Dialog(group="Mass fraction", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1]
   xiDischargeStart = zeros(vleFluidType.nc-1) "Start value for
   discharge mass fraction"
  annotation(Dialog(group="Mass fraction", tab="Start values"));

parameter Modelica.SIunits.MassFraction[vleFluidType.nc-1]
   xiInjectionStart = zeros(vleFluidType.nc-1) "Start value for
   injection mass fraction"
  annotation(Dialog(group="Mass fraction", tab="Start values"));
```

```
/*Additional variables for calculation of compressor characteristics*/
Modelica.SIunits.Mass mSucked_total "Total mass sucked in by the
   compressor";

Modelica.SIunits.Work W_total "Total compression work";

/*Control volumes*/
ScrewCompressorPackage.Components.ControlVolume cv1(pStart=
   pSuctionStart, hStart=hSuctionStart, xiStart=xiSuctionStart) "
   Control volume 1"
  annotation (Placement(transformation(extent={{-220,-10},{-200,10}}))
     );
ScrewCompressorPackage.Components.ControlVolume cv2(pStart=
   pSuctionStart, hStart=hSuctionStart, xiStart=xiSuctionStart) "
   Control volume 2"
  annotation (Placement(transformation(extent={{-160,-10},{-140,10}}))
     );
ScrewCompressorPackage.Components.ControlVolume cv3(pStart=
   pSuctionStart, hStart=hSuctionStart, xiStart=xiSuctionStart) "
   Control volume 3"
  annotation (Placement(transformation(extent={{-100,-10},{-80,10}})))
     ;
ScrewCompressorPackage.Components.ControlVolume cv4(pStart=
   pSuctionStart, hStart=hSuctionStart, xiStart=xiSuctionStart) "
   Control volume 4"
  annotation (Placement(transformation(extent={{-40,-10},{-20,10}})));
ScrewCompressorPackage.Components.ControlVolume cv5(pStart=pStart_cv5,
    hStart=hStart_cv5, xiStart=xiStart_cv5) "Control volume 5"
  annotation (Placement(transformation(extent={{20,-10},{40,10}})));
ScrewCompressorPackage.Components.ControlVolume cv6(pStart=pStart_cv6,
    hStart=hStart_cv6, xiStart=xiStart_cv6) "Control volume 6"
  annotation (Placement(transformation(extent={{80,-10},{100,10}})));
ScrewCompressorPackage.Components.ControlVolume cv7(pStart=pStart_cv7,
    hStart=hStart_cv7, xiStart=xiStart_cv7) "Control volume 7"
  annotation (Placement(transformation(extent={{140,-10},{160,10}})));
ScrewCompressorPackage.Components.ControlVolume cv8(pStart=
   pDischargeStart, hStart=hDischargeStart, xiStart=xiDischargeStart)
   "Control volume 8"
  annotation (Placement(transformation(extent={{200,-10},{220,10}})));

/*Volume functions*/
ScrewCompressorPackage.Functions.VolumeFunction V1(thetaStart=0) "
   Volume function 1"
  annotation (Placement(transformation(extent
     ={{-240,-250},{-220,-230}})));
ScrewCompressorPackage.Functions.VolumeFunction V2(thetaStart=1
   .5707963267949) "Volume function 2"
```

```
    annotation (Placement(transformation(extent
       ={{-180,-250},{-160,-230}})));
  ScrewCompressorPackage.Functions.VolumeFunction V3(thetaStart=3
     .1415926535898) "Volume function 3"
    annotation (Placement(transformation(extent
       ={{-120,-250},{-100,-230}})));
  ScrewCompressorPackage.Functions.VolumeFunction V4(thetaStart=4
     .7123889803847) "Volume function 4"
    annotation (Placement(transformation(extent={{-60,-250},{-40,-230}})
       ));
  ScrewCompressorPackage.Functions.VolumeFunction V5(thetaStart=6
     .2831853071796) "Volume function 5"
    annotation (Placement(transformation(extent={{0,-250},{20,-230}})));
  ScrewCompressorPackage.Functions.VolumeFunction V6(thetaStart=7
     .8539816339745) "Volume function 6"
    annotation (Placement(transformation(extent={{60,-250},{80,-230}})))
       ;
  ScrewCompressorPackage.Functions.VolumeFunction V7(thetaStart=9
     .4247779607694) "Volume function 7"
    annotation (Placement(transformation(extent={{120,-250},{140,-230}})
       ));
  ScrewCompressorPackage.Functions.VolumeFunction V8(thetaStart=10
     .995574287564) "Volume function 8"
    annotation (Placement(transformation(extent={{180,-250},{200,-230}})
       ));

  /*Connection ports*/
  TIL.Connectors.VLEFluidPort suctionPort(final vleFluidType=
     vleFluidType) "The compressor's suction port"
    annotation (Placement(transformation(extent={{-270,-10},{-250,10}}))
       );
  TIL.Connectors.VLEFluidPort dischargePort(final vleFluidType=
     vleFluidType) "The compressor's discharge port"
    annotation (Placement(transformation(extent={{250,-10},{270,10}})));
  TIL.Connectors.VLEFluidPort injectionPort(final vleFluidType=
     vleFluidType) "The compressor's injection port"
    annotation (Placement(transformation(extent={{-10,250},{10,270}})));

  /*Leakage paths*/
  Components.FlowRestrictor leak21 "Leakage from control volume 2 to 1"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={-180,0})));
  Components.FlowRestrictor leak32 "Leakage from control volume 3 to 2"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={-120,0})));
  Components.FlowRestrictor leak43 "Leakage from control volume 4 to 3"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={-60,0})));
  Components.FlowRestrictor leak54 "Leakage from control volume 5 to 4"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={0,0})));
  Components.FlowRestrictor leak65 "Leakage from control volume 6 to 5"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={60,0})));
  Components.FlowRestrictor leak76 "Leakage from control volume 7 to 6"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={120,0})));
  Components.FlowRestrictor leak87 "Leakage from control volume 8 to 7"
    annotation (Placement(transformation(extent={{-8,-4},{8,4}},rotation
       =180,origin={180,0})));
  Components.FlowRestrictor leak18 "Leakage from control volume 1 to 8"
    annotation (Placement(transformation(extent={{-8,4},{8,-4}},rotation
       =0,origin={208,-30})));

  /*Leakage area functions*/
  Functions.LeakageAreaFunction leakageAreaFunction21 "Effective flow
     area of leak21"
    annotation (Placement(transformation(extent
       ={{-194,-210},{-174,-190}})));
  Functions.LeakageAreaFunction leakageAreaFunction32 "Effective flow
     area of leak32"
    annotation (Placement(transformation(extent
       ={{-134,-210},{-114,-190}})));
  Functions.LeakageAreaFunction leakageAreaFunction43 "Effective flow
     area of leak43"
    annotation (Placement(transformation(extent={{-74,-210},{-54,-190}})
       ));
  Functions.LeakageAreaFunction leakageAreaFunction54 "Effective flow
     area of leak54"
    annotation (Placement(transformation(extent={{-14,-210},{6,-190}})))
       ;
  Functions.LeakageAreaFunction leakageAreaFunction65 "Effective flow
     area of leak65"
    annotation (Placement(transformation(extent={{46,-210},{66,-190}})))
       ;
  Functions.LeakageAreaFunction leakageAreaFunction76 "Effective flow
     area of leak76"
    annotation (Placement(transformation(extent={{106,-210},{126,-190}})
       ));
  Functions.LeakageAreaFunction leakageAreaFunction87 "Effective flow
     area of leak87"
    annotation (Placement(transformation(extent={{166,-210},{186,-190}})
       ));
```

```
Functions.LeakageAreaFunction leakageAreaFunction18 "Effective flow
   area of leak18"
  annotation (Placement(transformation(extent={{194,-210},{214,-190}})
     ));

/*Suction paths*/
Components.FlowRestrictor suctionPath1 "Suction path to control volume
    1"
  annotation (Placement(transformation(extent={{-248,36},{-232,44}}))
     ;
Components.FlowRestrictor suctionPath2 "Suction path to control volume
    2"
  annotation (Placement(transformation(extent={{-248,56},{-232,64}}))
     ;
Components.FlowRestrictor suctionPath3 "Suction path to control volume
    3"
  annotation (Placement(transformation(extent={{-248,76},{-232,84}}))
     ;
Components.FlowRestrictor suctionPath4 "Suction path to control volume
    4"
  annotation (Placement(transformation(extent={{-248,96},{-232,104}}))
     );
Components.FlowRestrictor suctionPath5 "Suction path to control volume
    5"
  annotation (Placement(transformation(extent={{-248,116},{-232,124}})
     ));
Components.FlowRestrictor suctionPath6 "Suction path to control volume
    6"
  annotation (Placement(transformation(extent={{-248,136},{-232,144}})
     ));
Components.FlowRestrictor suctionPath7 "Suction path to control volume
    7"
  annotation (Placement(transformation(extent={{-248,156},{-232,164}})
     ));
Components.FlowRestrictor suctionPath8 "Suction path to control volume
    8"
  annotation (Placement(transformation(extent={{-248,176},{-232,184}})
     ));

/*Suction area functions*/
Functions.SuctionAreaFunction suctionAreaFunction1(thetaStart=0) "
   Effective flow area of suction path 1"
  annotation (Placement(transformation(extent={{-290,40},{-270,60}}))
     ;
Functions.SuctionAreaFunction suctionAreaFunction2(thetaStart=1
   .5707963267949) "Effective flow area of suction path 2"
```

```
    annotation (Placement(transformation(extent={{-320,60},{-300,80}})))
      ;
  Functions.SuctionAreaFunction suctionAreaFunction3(thetaStart=3
     .1415926535898) "Effective flow area of suction path 3"
    annotation (Placement(transformation(extent={{-290,80},{-270,100}}))
      );
  Functions.SuctionAreaFunction suctionAreaFunction4(thetaStart=4
     .7123889803847) "Effective flow area of suction path 4"
    annotation (Placement(transformation(extent={{-320,100},{-300,120}})
      ));
  Functions.SuctionAreaFunction suctionAreaFunction5(thetaStart=6
     .2831853071796) "Effective flow area of suction path 5"
    annotation (Placement(transformation(extent={{-290,120},{-270,140}})
      ));
  Functions.SuctionAreaFunction suctionAreaFunction6(thetaStart=7
     .8539816339745) "Effective flow area of suction path 6"
    annotation (Placement(transformation(extent={{-320,140},{-300,160}})
      ));
  Functions.SuctionAreaFunction suctionAreaFunction7(thetaStart=9
     .4247779607694) "Effective flow area of suction path 7"
    annotation (Placement(transformation(extent={{-290,160},{-270,180}})
      ));
  Functions.SuctionAreaFunction suctionAreaFunction8(thetaStart=10
     .995574287564) "Effective flow area of suction path 8"
    annotation (Placement(transformation(extent={{-320,180},{-300,200}})
      ));

  /*Discharge paths*/
  Components.FlowRestrictor dischargePath1 "Discharge path for control
     volume 1"
    annotation (Placement(transformation(extent={{232,-184},{248,-176}})
      ));
  Components.FlowRestrictor dischargePath2 "Discharge path for control
     volume 2"
    annotation (Placement(transformation(extent={{232,-164},{248,-156}})
      ));
  Components.FlowRestrictor dischargePath3 "Discharge path for control
     volume 3"
    annotation (Placement(transformation(extent={{232,-144},{248,-136}})
      ));
  Components.FlowRestrictor dischargePath4 "Discharge path for control
     volume 4"
    annotation (Placement(transformation(extent={{232,-124},{248,-116}})
      ));
  Components.FlowRestrictor dischargePath5 "Discharge path for control
     volume 5"
    annotation (Placement(transformation(extent={{232,-104},{248,-96}}))
      );
  Components.FlowRestrictor dischargePath6 "Discharge path for control
     volume 6"
    annotation (Placement(transformation(extent={{232,-84},{248,-76}})))
      ;
  Components.FlowRestrictor dischargePath7 "Discharge path for control
     volume 7"
    annotation (Placement(transformation(extent={{232,-64},{248,-56}})))
      ;
  Components.FlowRestrictor dischargePath8 "Discharge path for control
     volume 8"
    annotation (Placement(transformation(extent={{232,-44},{248,-36}})))
      ;

  /*Discharge area functions*/
  Functions.DischargeAreaFunction dischargeAreaFunction1(thetaStart=0) "
     Effective flow area of discharge path 1"
    annotation (Placement(transformation(extent={{270,-180},{290,-160}})
      ));
  Functions.DischargeAreaFunction dischargeAreaFunction2(thetaStart=1
     .5707963267949) "Effective flow area of discharge path 2"
    annotation (Placement(transformation(extent={{300,-160},{320,-140}})
      ));
  Functions.DischargeAreaFunction dischargeAreaFunction3(thetaStart=3
     .1415926535898) "Effective flow area of discharge path 3"
    annotation (Placement(transformation(extent={{270,-140},{290,-120}})
      ));
  Functions.DischargeAreaFunction dischargeAreaFunction4(thetaStart=4
     .7123889803847) "Effective flow area of discharge path 4"
    annotation (Placement(transformation(extent={{300,-120},{320,-100}})
      ));
  Functions.DischargeAreaFunction dischargeAreaFunction5(thetaStart=6
     .2831853071796) "Effective flow area of discharge path 5"
    annotation (Placement(transformation(extent={{270,-100},{290,-80}}))
      );
  Functions.DischargeAreaFunction dischargeAreaFunction6(thetaStart=7
     .8539816339745) "Effective flow area of discharge path 6"
    annotation (Placement(transformation(extent={{300,-80},{320,-60}})))
      ;
  Functions.DischargeAreaFunction dischargeAreaFunction7(thetaStart=9
     .4247779607694) "Effective flow area of discharge path 7"
    annotation (Placement(transformation(extent={{270,-60},{290,-40}})))
      ;
  Functions.DischargeAreaFunction dischargeAreaFunction8(thetaStart=10
     .995574287564) "Effective flow area of discharge path 8"
    annotation (Placement(transformation(extent={{300,-40},{320,-20}})))
      ;

  /*Injection paths*/
  Components.Injector injector1 "Injection path to control volume 1"
    annotation (Placement(transformation(extent={{-220,210},{-200,230}})
      ));
  Components.Injector injector2 "Injection path to control volume 2"
    annotation (Placement(transformation(extent={{-160,210},{-140,230}})
      ));
  Components.Injector injector3 "Injection path to control volume 3"
    annotation (Placement(transformation(extent={{-100,210},{-80,230}}))
      );
  Components.Injector injector4 "Injection path to control volume 4"
    annotation (Placement(transformation(extent={{-40,210},{-20,230}})))
      ;
  Components.Injector injector5 "Injection path to control volume 5"
    annotation (Placement(transformation(extent={{20,210},{40,230}})));
  Components.Injector injector6 "Injection path to control volume 6"
    annotation (Placement(transformation(extent={{80,210},{100,230}})));
  Components.Injector injector7 "Injection path to control volume 7"
    annotation (Placement(transformation(extent={{140,210},{160,230}})))
      ;
  Components.Injector injector8 "Injection path to control volume 8"
    annotation (Placement(transformation(extent={{200,210},{220,230}})))
      ;

  /*Injection flow functions*/
  Functions.InjectionFlowFunction injectionFlowFunction1(thetaStart=0) "
     Mass flow through injector 1"
    annotation (Placement(transformation(extent={{-190,210},{-170,230}})
      ));
  Functions.InjectionFlowFunction injectionFlowFunction2(thetaStart=1
     .5707963267949) "Mass flow through injector 2"
    annotation (Placement(transformation(extent={{-130,210},{-110,230}})
      ));
  Functions.InjectionFlowFunction injectionFlowFunction3(thetaStart=3
     .1415926535898) "Mass flow through injector 3"
    annotation (Placement(transformation(extent={{-70,210},{-50,230}})))
      ;
  Functions.InjectionFlowFunction injectionFlowFunction4(thetaStart=4
     .7123889803847) "Mass flow through injector 4"
    annotation (Placement(transformation(extent={{-10,210},{10,230}})));
  Functions.InjectionFlowFunction injectionFlowFunction5(thetaStart=6
     .2831853071796) "Mass flow through injector 5"
    annotation (Placement(transformation(extent={{52,210},{72,230}})));
```

```
Functions.InjectionFlowFunction injectionFlowFunction6(thetaStart=7
    .8539816339745) "Mass flow through injector 6"
  annotation (Placement(transformation(extent={{110,210},{130,230}})))
      ;
Functions.InjectionFlowFunction injectionFlowFunction7(thetaStart=9
    .4247779607694) "Mass flow through injector 7"
  annotation (Placement(transformation(extent={{170,210},{190,230}})))
      ;
Functions.InjectionFlowFunction injectionFlowFunction8(thetaStart=10
    .995574287564) "Mass flow through injector 8"
  annotation (Placement(transformation(extent={{228,210},{248,230}})))
      ;

/*Suction junctions*/
Components.Junction suctionJunction1(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 1"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,40})));
Components.Junction suctionJunction2(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 2"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,60})));
Components.Junction suctionJunction3(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 3"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,80})));
Components.Junction suctionJunction4(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 4"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,100})));
Components.Junction suctionJunction5(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 5"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,120})));
Components.Junction suctionJunction6(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 6"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,140})));
Components.Junction suctionJunction7(hStart=hSuctionStart, xiStart=
    xiSuctionStart) "Separate suction flow 7"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
      rotation=180,origin={-256,160})));

/*Discharge junctions*/
Components.Junction dischargeJunction1(hStart=hDischargeStart, xiStart
    =xiDischargeStart) "Merge discharge flow 1"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-160})));
Components.Junction dischargeJunction2(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 2"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-140})));
Components.Junction dischargeJunction3(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 3"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-120})));
Components.Junction dischargeJunction4(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 4"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-100})));
Components.Junction dischargeJunction5(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 5"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-80})));
Components.Junction dischargeJunction6(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 6"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-60})));
Components.Junction dischargeJunction7(hStart=hDischargeStart, xiStart
   =xiDischargeStart) "Merge discharge flow 7"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=0,origin={256,-40})));

/*Injection junctions*/
Components.Junction injectionJunction1(hStart=hInjectionStart, xiStart
   =xiInjectionStart) "Separate injection flow 1"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=90,origin={-210,240})));
Components.Junction injectionJunction2(hStart=hInjectionStart, xiStart
   =xiInjectionStart) "Separate injection flow 2"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=90,origin={-150,240})));
Components.Junction injectionJunction3(hStart=hInjectionStart, xiStart
   =xiInjectionStart) "Separate injection flow 3"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=90,origin={-90,240})));
Components.Junction injectionJunction4(hStart=hInjectionStart, xiStart
   =xiInjectionStart) "Separate injection flow 4"
  annotation (Placement(transformation(extent={{-10,-10},{10,10}},
     rotation=90,origin={-30,240})));
Components.Junction injectionJunction5(hStart=hInjectionStart, xiStart
   =xiInjectionStart) "Separate injection flow 5"
```

```
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
       rotation=90,origin={30,240})));
  Components.Junction injectionJunction6(hStart=hInjectionStart, xiStart
     =xiInjectionStart) "Separate injection flow 6"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
       rotation=90,origin={90,240})));
  Components.Junction injectionJunction7(hStart=hInjectionStart, xiStart
     =xiInjectionStart) "Separate injection flow 7"
    annotation (Placement(transformation(extent={{-10,-10},{10,10}},
       rotation=90,origin={150,240})));

  /*Initial equations*/
initial equation
  mSucked_total = 0;

  W_total = 0;

    /*Equations*/
equation
  der(mSucked_total) = suctionPort.m_flow;

  der(W_total) = cv1.Wdot + cv2.Wdot + cv3.Wdot + cv4.Wdot + cv5.Wdot +
     cv6.Wdot + cv7.Wdot + cv8.Wdot;

  /*Connections*/
  connect(V1.y, cv1.volumeInput) annotation (Line(points={{-230,-229.4
     },{-230,-9},{-222,-9}},
                                Line color={28,108,200}));
  connect(V2.y, cv2.volumeInput) annotation (Line(points={{-170,-229.4
     },{-170,-9},{-162,-9}},
                                color={28,108,200}));
  connect(V3.y, cv3.volumeInput) annotation (Line(points={{-110,-229.4
     },{-110,-9},{-102,-9}},
                                color={28,108,200}));
  connect(V8.y, cv8.volumeInput) annotation (Line(points={{190,-229.4
     },{190,-9},{198,-9}},
                              color={28,108,200}));
  connect(V4.y, cv4.volumeInput) annotation (Line(points={{-50,-229.4
     },{-50,-9},{-42,-9}},
                              color={28,108,200}));
  connect(V5.y, cv5.volumeInput) annotation (Line(points={{10,-229.4
     },{10,-9},{18,-9}},
                        color={28,108,200}));
  connect(V7.y, cv7.volumeInput) annotation (Line(points={{130,-229.4
     },{130,-9},{138,-9}},
                              color={28,108,200}));
```

```
connect(V6.y, cv6.volumeInput) annotation (Line(points={{70,-229.4
   },{70,-9},{78,-9}},
                           color={28,108,200}));
connect(leakageAreaFunction21.x1, cv1.volumeInput) annotation (Line(
   points={{-188,-210.4},{-188,-220},{-230,-220},{-230,-9},{-222,-9}},
                                          color={28,108,200}));
connect(leakageAreaFunction21.x2, cv2.volumeInput) annotation (Line(
   points={{-180,-210.4},{-180,-220},{-170,-220},{-170,-9},{-162,-9}},
                                          color={28,108,200}));
connect(leakageAreaFunction32.x1, cv2.volumeInput) annotation (Line(
   points={{-128,-210.4},{-128,-220},{-170,-220},{-170,-9},{-162,-9}},
                                          color={28,108,200}));
connect(leakageAreaFunction32.x2, cv3.volumeInput) annotation (Line(
   points={{-120,-210.4},{-120,-220},{-110,-220},{-110,-9},{-102,-9}},
                                          color={28,108,200}));
connect(leakageAreaFunction43.x1, cv3.volumeInput) annotation (Line(
   points={{-68,-210.4},{-68,-220},{-110,-220},{-110,-9},{-102,-9}},
                                     color={28,108,200}));
connect(leakageAreaFunction43.x2, cv4.volumeInput) annotation (Line(
   points={{-60,-210.4},{-60,-220},{-50,-220},{-50,-9},{-42,-9}},
                                   color={28,108,200}));
connect(leakageAreaFunction76.x1, cv6.volumeInput) annotation (Line(
   points={{112,-210.4},{112,-220},{70,-220},{70,-9},{78,-9}},
                                 color={28,108,200}));
connect(leakageAreaFunction76.x2, cv7.volumeInput) annotation (Line(
   points={{120,-210.4},{120,-220},{130,-220},{130,-9},{138,-9}},
                                   color={28,108,200}));
connect(leakageAreaFunction87.x1, cv7.volumeInput) annotation (Line(
   points={{172,-210.4},{172,-220},{130,-220},{130,-9},{138,-9}},
                                   color={28,108,200}));
connect(leakageAreaFunction87.x2, cv8.volumeInput) annotation (Line(
   points={{180,-210.4},{180,-220},{190,-220},{190,-9},{198,-9}},
                                   color={28,108,200}));
connect(leakageAreaFunction18.x1, cv8.volumeInput) annotation (Line(
   points={{200,-210.4},{200,-220},{190,-220},{190,-9},{198,-9}},
                                   color={28,108,200}));
connect(leakageAreaFunction18.x2, cv1.volumeInput) annotation (Line(
   points={{208,-210.4
   },{208,-256},{-188,-256},{-188,-220},{-230,-220},{-230,-9},{-222,
        -9}},                                                    color
           ={28,108,200}));
connect(leak21.portB, cv1.leakInPort) annotation (Line(
    points={{-188,0},{-194,0},{-194,4},{-200,4}},
    color={153,204,0},
    thickness=0.5));
connect(cv2.leakOutPort, leak21.portA) annotation (Line(
    points={{-160,-4},{-166,-4},{-166,0},{-172,0}},
      color={153,204,0},
      thickness=0.5));
  connect(leak32.portB, cv2.leakInPort) annotation (Line(
      points={{-128,0},{-134,0},{-134,4},{-140,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv3.leakOutPort, leak32.portA) annotation (Line(
      points={{-100,-4},{-106,-4},{-106,0},{-112,0}},
      color={153,204,0},
      thickness=0.5));
  connect(leak43.portB, cv3.leakInPort) annotation (Line(
      points={{-68,0},{-74,0},{-74,4},{-80,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv4.leakOutPort, leak43.portA) annotation (Line(
      points={{-40,-4},{-46,-4},{-46,0},{-52,0}},
      color={153,204,0},
      thickness=0.5));
  connect(leak54.portB, cv4.leakInPort) annotation (Line(
      points={{-8,0},{-14,0},{-14,4},{-20,4}},
      color={153,204,0},
      thickness=0.5));
  connect(leak65.portB, cv5.leakInPort) annotation (Line(
      points={{52,0},{46,0},{46,4},{40,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv6.leakOutPort, leak65.portA) annotation (Line(
      points={{80,-4},{74,-4},{74,0},{68,0}},
      color={153,204,0},
      thickness=0.5));
  connect(leak76.portB, cv6.leakInPort) annotation (Line(
      points={{112,0},{106,0},{106,4},{100,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv7.leakOutPort, leak76.portA) annotation (Line(
      points={{140,-4},{134,-4},{134,0},{128,0}},
      color={153,204,0},
      thickness=0.5));
  connect(cv5.leakOutPort, leak54.portA) annotation (Line(
      points={{20,-4},{16,-4},{16,0},{8,0}},
      color={153,204,0},
      thickness=0.5));
  connect(leak87.portB, cv7.leakInPort) annotation (Line(
      points={{172,0},{166,0},{166,4},{160,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv8.leakOutPort, leak87.portA) annotation (Line(
      points={{200,-4},{194,-4},{194,0},{188,0}},
      color={153,204,0},
      thickness=0.5));
  connect(cv8.leakInPort, leak18.portB) annotation (Line(
      points={{220,4},{228,4},{228,-30},{216,-30}},
      color={153,204,0},
      thickness=0.5));
  connect(leak18.portA, cv1.leakOutPort) annotation (Line(
      points={{200,-30},{-228,-30},{-228,-4},{-220,-4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath1.portB, cv1.suctionPort) annotation (Line(
      points={{-232,40},{-228,40},{-228,4},{-220,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath2.portB, cv2.suctionPort) annotation (Line(
      points={{-232,60},{-166,60},{-166,4},{-160,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath3.portB, cv3.suctionPort) annotation (Line(
      points={{-232,80},{-106,80},{-106,4},{-100,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath4.portB, cv4.suctionPort) annotation (Line(
      points={{-232,100},{-46,100},{-46,4},{-40,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath5.portB, cv5.suctionPort) annotation (Line(
      points={{-232,120},{16,120},{16,4},{20,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath6.portB, cv6.suctionPort) annotation (Line(
      points={{-232,140},{74,140},{74,4},{80,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath7.portB, cv7.suctionPort) annotation (Line(
      points={{-232,160},{134,160},{134,4},{140,4}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionPath8.portB, cv8.suctionPort) annotation (Line(
      points={{-232,180},{194,180},{194,4},{200,4}},
      color={153,204,0},
      thickness=0.5));
  connect(cv8.dischargePort, dischargePath8.portA)
    annotation (Line(
      points={{220,-4},{224,-4},{224,-40},{232,-40}},
      color={153,204,0},
      thickness=0.5));
  connect(cv7.dischargePort, dischargePath7.portA)
    annotation (Line(
      points={{160,-4},{166,-4},{166,-60},{232,-60}},
      color={153,204,0},
      thickness=0.5));
  connect(cv6.dischargePort, dischargePath6.portA)
    annotation (Line(
      points={{100,-4},{106,-4},{106,-80},{232,-80}},
      color={153,204,0},
      thickness=0.5));
  connect(cv5.dischargePort, dischargePath5.portA)
    annotation (Line(
      points={{40,-4},{46,-4},{46,-100},{232,-100}},
      color={153,204,0},
      thickness=0.5));
  connect(cv4.dischargePort, dischargePath4.portA)
    annotation (Line(
      points={{-20,-4},{-14,-4},{-14,-120},{232,-120}},
      color={153,204,0},
      thickness=0.5));
  connect(cv3.dischargePort, dischargePath3.portA)
    annotation (Line(
      points={{-80,-4},{-74,-4},{-74,-140},{232,-140}},
      color={153,204,0},
      thickness=0.5));
  connect(cv2.dischargePort, dischargePath2.portA)
    annotation (Line(
      points={{-140,-4},{-134,-4},{-134,-160},{232,-160}},
      color={153,204,0},
      thickness=0.5));
  connect(cv1.dischargePort, dischargePath1.portA)
    annotation (Line(
      points={{-200,-4},{-194,-4},{-194,-180},{232,-180}},
      color={153,204,0},
      thickness=0.5));
  connect(dischargePath1.portB, dischargeJunction1.separatePort2)
    annotation (Line(
      points={{248,-180},{256,-180},{256,-164}},
      color={153,204,0},
      thickness=0.5));
  connect(dischargePath2.portB, dischargeJunction1.separatePort1)
    annotation (Line(
      points={{248,-160},{252,-160}},
      color={153,204,0},
      thickness=0.5));
```

```
connect(dischargeJunction1.mergePort, dischargeJunction2.separatePort2
   )
  annotation (Line(
    points={{256,-156},{256,-144}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath3.portB, dischargeJunction2.separatePort1)
  annotation (Line(
    points={{248,-140},{252,-140}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction2.mergePort, dischargeJunction3.separatePort2
   )
  annotation (Line(
    points={{256,-136},{256,-124}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath4.portB, dischargeJunction3.separatePort1)
  annotation (Line(
    points={{248,-120},{252,-120}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction3.mergePort, dischargeJunction4.separatePort2
   )
  annotation (Line(
    points={{256,-116},{256,-104}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath5.portB, dischargeJunction4.separatePort1)
  annotation (Line(
    points={{248,-100},{252,-100}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction4.mergePort, dischargeJunction5.separatePort2
   )
  annotation (Line(
    points={{256,-96},{256,-84}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath6.portB, dischargeJunction5.separatePort1)
  annotation (Line(
    points={{248,-80},{252,-80}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction5.mergePort, dischargeJunction6.separatePort2
   )
  annotation (Line(
    points={{256,-76},{256,-64}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath7.portB, dischargeJunction6.separatePort1)
  annotation (Line(
    points={{248,-60},{252,-60}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction6.mergePort, dischargeJunction7.separatePort2
   )
  annotation (Line(
    points={{256,-56},{256,-44}},
    color={153,204,0},
    thickness=0.5));
connect(dischargePath8.portB, dischargeJunction7.separatePort1)
  annotation (Line(
    points={{248,-40},{252,-40}},
    color={153,204,0},
    thickness=0.5));
connect(dischargeJunction7.mergePort, dischargePort)
  annotation (Line(
    points={{256,-36},{256,0},{260,0}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction1.mergePort, injectionPort)
  annotation (Line(
    points={{-214,240},{-220,240},{-220,256},{0,256},{0,260}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction1.separatePort2, injectionJunction2.mergePort
   )
  annotation (Line(
    points={{-206,240},{-154,240}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction2.separatePort2, injectionJunction3.mergePort
   )
  annotation (Line(
    points={{-146,240},{-94,240}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction3.separatePort2, injectionJunction4.mergePort
   )
  annotation (Line(
    points={{-86,240},{-34,240}},
    color={153,204,0},
    thickness=0.5));
```

```
connect(injectionJunction4.separatePort2, injectionJunction5.mergePort
   )
  annotation (Line(
    points={{-26,240},{26,240}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction5.separatePort2, injectionJunction6.mergePort
   )
  annotation (Line(
    points={{34,240},{86,240}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction6.separatePort2, injectionJunction7.mergePort
   )
  annotation (Line(
    points={{94,240},{146,240}},
    color={153,204,0},
    thickness=0.5));
connect(leakageAreaFunction21.y, leak21.effectiveFlowAreaInput)
  annotation (Line(points={{-180,-189.4},{-180,-5}}, color
     ={238,46,47}));
connect(leakageAreaFunction32.y, leak32.effectiveFlowAreaInput)
  annotation (Line(points={{-120,-189.4},{-120,-5}}, color
     ={238,46,47}));
connect(leakageAreaFunction43.y, leak43.effectiveFlowAreaInput)
  annotation (Line(points={{-60,-189.4},{-60,-5}}, color={238,46,47}))
     ;
connect(leakageAreaFunction54.y, leak54.effectiveFlowAreaInput)
  annotation (Line(points={{0,-189.4},{0,-5}}, color={238,46,47}));
connect(leakageAreaFunction65.y, leak65.effectiveFlowAreaInput)
  annotation (Line(points={{60,-189.4},{60,-5}}, color={238,46,47}));
connect(leakageAreaFunction76.y, leak76.effectiveFlowAreaInput)
  annotation (Line(points={{120,-189.4},{120,-5}}, color={238,46,47}))
     ;
connect(leakageAreaFunction87.y, leak87.effectiveFlowAreaInput)
  annotation (Line(points={{180,-189.4},{180,-5}}, color={238,46,47}))
     ;
connect(leakageAreaFunction18.y, leak18.effectiveFlowAreaInput)
  annotation (Line(points={{208,-189.4},{208,-35}}, color={238,46,47})
     );
connect(suctionAreaFunction1.y, suctionPath1.effectiveFlowAreaInput)
  annotation (Line(points={{-269.4,50},{-240,50},{-240,45}}, color
     ={238,46,47}));
connect(suctionAreaFunction2.y, suctionPath2.effectiveFlowAreaInput)
  annotation (Line(points={{-299.4,70},{-240,70},{-240,65}}, color
     ={238,46,47}));
connect(suctionAreaFunction3.y, suctionPath3.effectiveFlowAreaInput)
```

```
  annotation (Line(points={{-269.4,90},{-240,90},{-240,85}}, color
     ={238,46,47}));
connect(suctionAreaFunction4.y, suctionPath4.effectiveFlowAreaInput)
  annotation (Line(points={{-299.4,110},{-240,110},{-240,105}}, color
     ={238,46,
        47}));
connect(suctionAreaFunction5.y, suctionPath5.effectiveFlowAreaInput)
  annotation (Line(points={{-269.4,130},{-240,130},{-240,125}}, color
     ={238,46,
        47}));
connect(suctionAreaFunction6.y, suctionPath6.effectiveFlowAreaInput)
  annotation (Line(points={{-299.4,150},{-240,150},{-240,145}}, color
     ={238,46,
        47}));
connect(suctionAreaFunction7.y, suctionPath7.effectiveFlowAreaInput)
  annotation (Line(points={{-269.4,170},{-240,170},{-240,165}}, color
     ={238,46,
        47}));
connect(suctionAreaFunction8.y, suctionPath8.effectiveFlowAreaInput)
  annotation (Line(points={{-299.4,190},{-240,190},{-240,185}}, color
     ={238,46,
        47}));
connect(dischargeAreaFunction1.y,
   dischargePath1.effectiveFlowAreaInput)
  annotation (Line(points={{269.4,-170},{240,-170},{240,-175}}, color
     ={238,46,
        47}));
connect(dischargeAreaFunction2.y,
   dischargePath2.effectiveFlowAreaInput)
  annotation (Line(points={{299.4,-150},{240,-150},{240,-155}}, color
     ={238,46,
        47}));
connect(dischargeAreaFunction3.y,
   dischargePath3.effectiveFlowAreaInput)
  annotation (Line(points={{269.4,-130},{240,-130},{240,-135}}, color
     ={238,46,
        47}));
connect(dischargeAreaFunction4.y,
   dischargePath4.effectiveFlowAreaInput)
  annotation (Line(points={{299.4,-110},{240,-110},{240,-115}}, color
     ={238,46,
        47}));
connect(dischargeAreaFunction5.y,
   dischargePath5.effectiveFlowAreaInput)
  annotation (Line(points={{269.4,-90},{240,-90},{240,-95}}, color
     ={238,46,47}));
```

```
connect(dischargeAreaFunction6.y,
   dischargePath6.effectiveFlowAreaInput)
  annotation (Line(points={{299.4,-70},{240,-70},{240,-75}}, color
     ={238,46,47}));
connect(dischargeAreaFunction7.y,
   dischargePath7.effectiveFlowAreaInput)
  annotation (Line(points={{269.4,-50},{240,-50},{240,-55}}, color
     ={238,46,47}));
connect(dischargeAreaFunction8.y,
   dischargePath8.effectiveFlowAreaInput)
  annotation (Line(points={{299.4,-30},{240,-30},{240,-35}}, color
     ={238,46,47}));
connect(injector1.outPort, cv1.injectionPort) annotation (Line(
    points={{-210,214},{-210,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector2.outPort, cv2.injectionPort) annotation (Line(
    points={{-150,214},{-150,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector3.outPort, cv3.injectionPort) annotation (Line(
    points={{-90,214},{-90,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector4.outPort, cv4.injectionPort) annotation (Line(
    points={{-30,214},{-30,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector5.outPort, cv5.injectionPort) annotation (Line(
    points={{30,214},{30,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector6.outPort, cv6.injectionPort) annotation (Line(
    points={{90,214},{90,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector7.outPort, cv7.injectionPort) annotation (Line(
    points={{150,214},{150,10}},
    color={153,204,0},
    thickness=0.5));
connect(injector8.outPort, cv8.injectionPort) annotation (Line(
    points={{210,214},{210,10}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction1.separatePort1, injector1.inPort) annotation
    (Line(
    points={{-210,236},{-210,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction2.separatePort1, injector2.inPort) annotation
    (Line(
    points={{-150,236},{-150,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction3.separatePort1, injector3.inPort) annotation
    (Line(
    points={{-90,236},{-90,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction4.separatePort1, injector4.inPort) annotation
    (Line(
    points={{-30,236},{-30,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction5.separatePort1, injector5.inPort) annotation
    (Line(
    points={{30,236},{30,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction6.separatePort1, injector6.inPort) annotation
    (Line(
    points={{90,236},{90,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction7.separatePort1, injector7.inPort) annotation
    (Line(
    points={{150,236},{150,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionJunction7.separatePort2, injector8.inPort) annotation
    (Line(
    points={{154,240},{210,240},{210,226}},
    color={153,204,0},
    thickness=0.5));
connect(injectionFlowFunction1.y, injector1.m_flowInput)
  annotation (Line(points={{-190.6,220},{-205,220}}, color
     ={217,67,180}));
connect(injectionFlowFunction2.y, injector2.m_flowInput)
  annotation (Line(points={{-130.6,220},{-145,220}}, color
     ={217,67,180}));
connect(injectionFlowFunction3.y, injector3.m_flowInput)
  annotation (Line(points={{-70.6,220},{-85,220}}, color={217,67,180})
     );
connect(injectionFlowFunction4.y, injector4.m_flowInput)
```

```
    annotation (Line(points={{-10.6,220},{-25,220}}, color={217,67,180})
      );
  connect(injectionFlowFunction5.y, injector5.m_flowInput)
    annotation (Line(points={{51.4,220},{35,220}}, color={217,67,180}));
  connect(injectionFlowFunction6.y, injector6.m_flowInput)
    annotation (Line(points={{109.4,220},{95,220}}, color={217,67,180}))
      ;
  connect(injectionFlowFunction7.y, injector7.m_flowInput)
    annotation (Line(points={{169.4,220},{155,220}}, color={217,67,180})
      );
  connect(injectionFlowFunction8.y, injector8.m_flowInput)
    annotation (Line(points={{227.4,220},{215,220}}, color={217,67,180})
      );
  connect(leakageAreaFunction54.x1, V4.y) annotation (Line(points
     ={{-8,-210.4},{
          -8,-220},{-50,-220},{-50,-229.4}}, color={28,108,200}));
  connect(leakageAreaFunction54.x2, V5.y) annotation (Line(points
     ={{0,-210.4},{0,
          -220},{10,-220},{10,-229.4}}, color={28,108,200}));
  connect(leakageAreaFunction65.x1, V5.y) annotation (Line(points
     ={{52,-210.4},{
          52,-220},{10,-220},{10,-229.4}}, color={28,108,200}));
  connect(leakageAreaFunction65.x2, V6.y) annotation (Line(points
     ={{60,-210.4},{
          60,-220},{70,-220},{70,-229.4}}, color={28,108,200}));
  connect(suctionJunction1.mergePort, suctionPort) annotation (Line(
      points={{-256,36},{-256,0},{-260,0}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionJunction1.separatePort1, suctionPath1.portA) annotation
      (Line(
      points={{-252,40},{-248,40}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionJunction1.separatePort2, suctionJunction2.mergePort)
    annotation (Line(
      points={{-256,44},{-256,56}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionJunction2.separatePort1, suctionPath2.portA) annotation
      (Line(
      points={{-252,60},{-248,60}},
      color={153,204,0},
      thickness=0.5));
  connect(suctionJunction2.separatePort2, suctionJunction3.mergePort)
    annotation (Line(
      points={{-256,64},{-256,76}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction3.separatePort1, suctionPath3.portA) annotation
    (Line(
    points={{-252,80},{-248,80}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction3.separatePort2, suctionJunction4.mergePort)
  annotation (Line(
    points={{-256,84},{-256,96}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction4.separatePort1, suctionPath4.portA) annotation
    (Line(
    points={{-252,100},{-248,100}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction4.separatePort2, suctionJunction5.mergePort)
  annotation (Line(
    points={{-256,104},{-256,116}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction5.separatePort1, suctionPath5.portA) annotation
    (Line(
    points={{-252,120},{-248,120}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction5.separatePort2, suctionJunction6.mergePort)
  annotation (Line(
    points={{-256,124},{-256,136}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction6.separatePort1, suctionPath6.portA) annotation
    (Line(
    points={{-252,140},{-248,140}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction6.separatePort2, suctionJunction7.mergePort)
  annotation (Line(
    points={{-256,144},{-256,156}},
    color={153,204,0},
    thickness=0.5));
connect(suctionJunction7.separatePort1, suctionPath7.portA) annotation
    (Line(
    points={{-252,160},{-248,160}},
    color={153,204,0},
    thickness=0.5));
  connect(suctionJunction7.separatePort2, suctionPath8.portA) annotation
      (Line(
      points={{-256,164},{-256,180},{-248,180}},
      color={153,204,0},
      thickness=0.5));

  /*Screw compressor model icon*/
  annotation (Icon(coordinateSystem(preserveAspectRatio=false, extent
     ={{-260,-260},
            {260,260}}),
                    graphics={
        Ellipse(
          extent={{-260,260},{260,-260}},
          lineColor={0,0,0},
          fillColor={255,255,255},
          fillPattern=FillPattern.Solid,
          lineThickness=0.5),
        Line(
          points={{-166,200},{240,100}},
          color={0,0,0},
          thickness=0.5),
        Line(
          points={{-166,-200},{240,-100}},
          color={0,0,0},
          thickness=0.5)}),                                       Diagram
             (
        coordinateSystem(preserveAspectRatio=false, extent
           ={{-260,-260},{260,260}})),
    experiment(__Dymola_NumberOfIntervals=1000, __Dymola_Algorithm="
       Esdirk45a"));
end ScrewCompressor;
```

Appendix B Modelica Model Diagram

Close-ups of the model diagram for the screw compressor model.

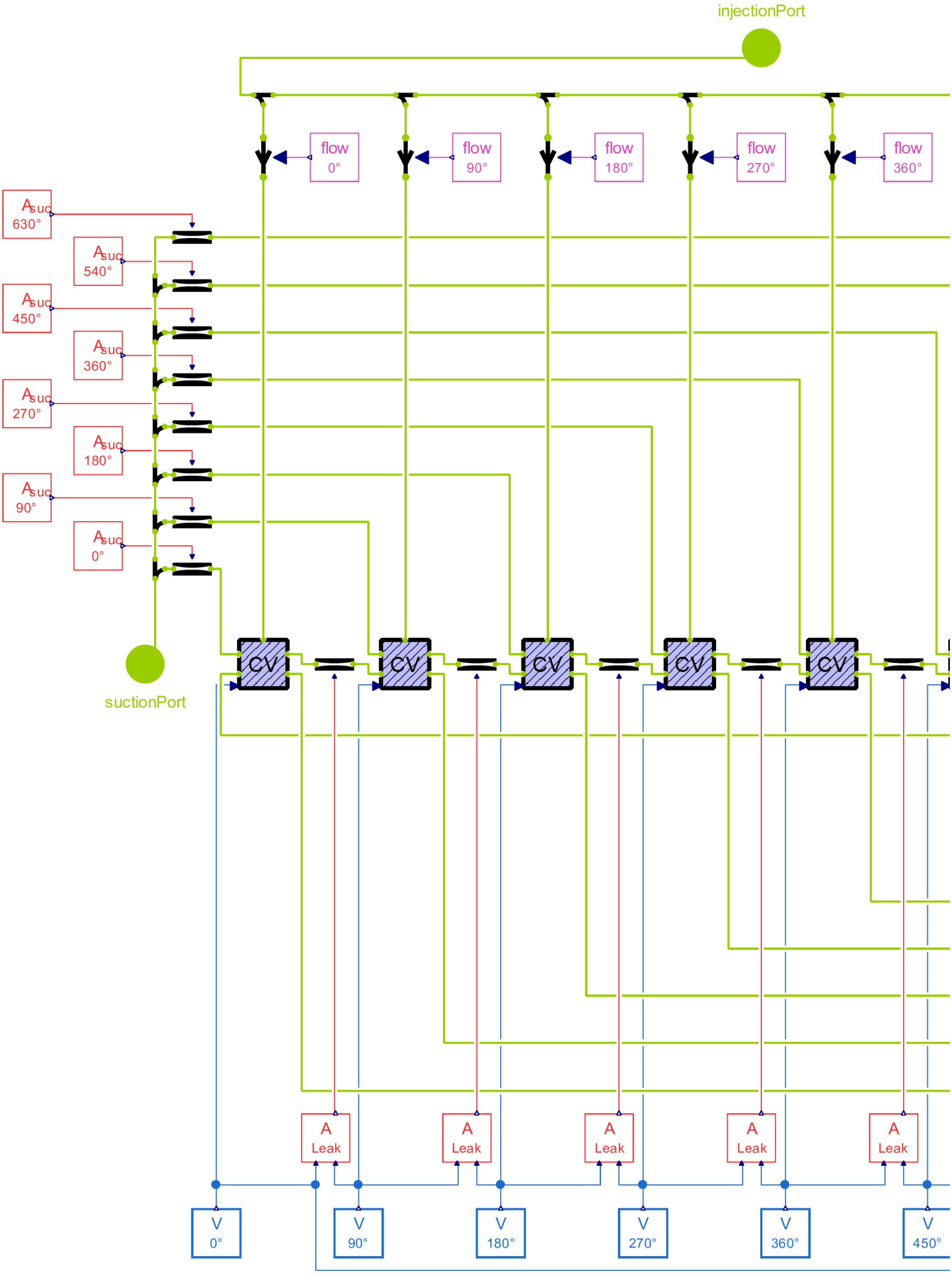


Figure B-1: Model diagram for the screw compressor model, close-up of left region.

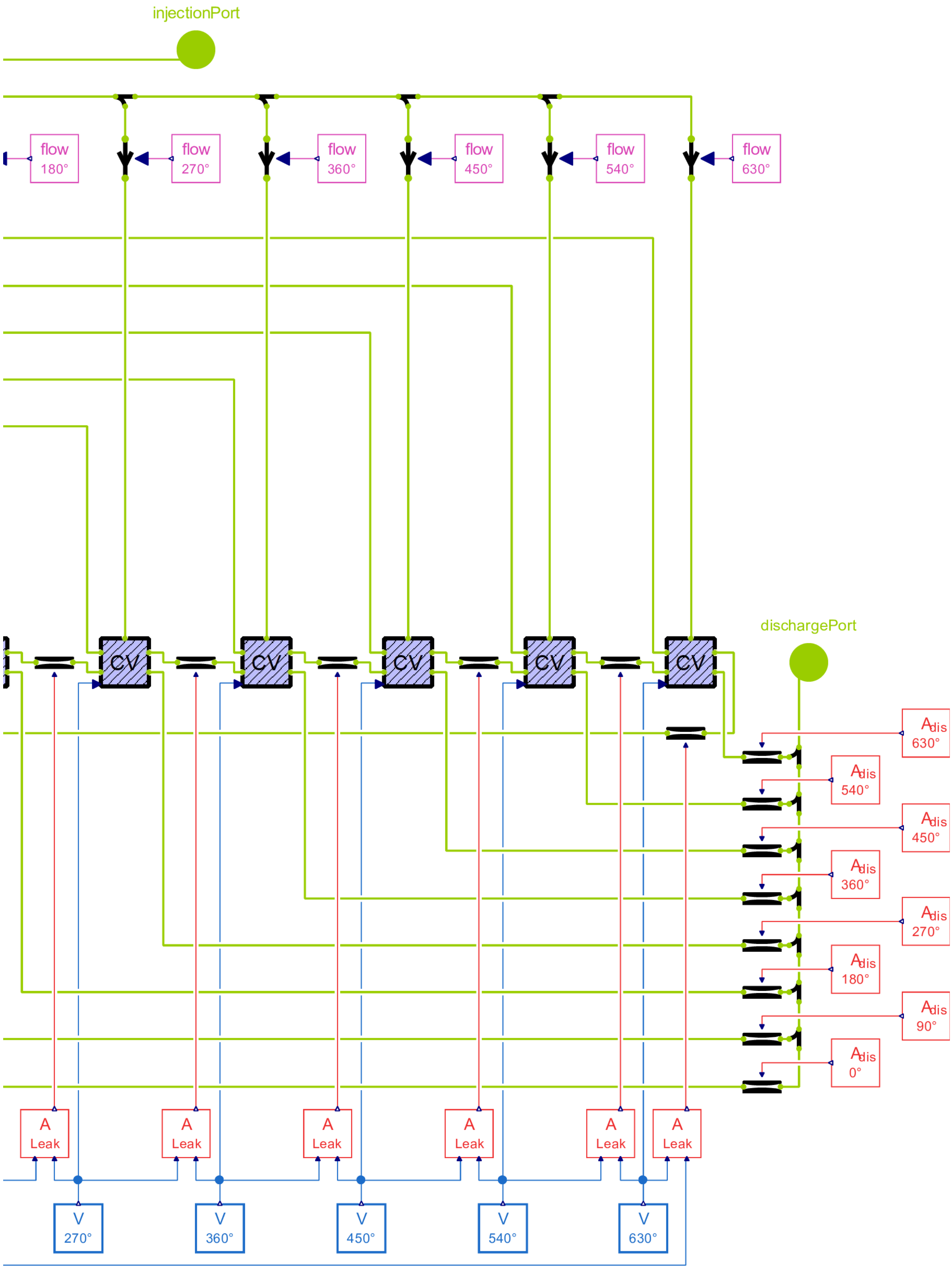


Figure B-2: Model diagram for the screw compressor model, close-up of right region.

Appendix C EES Code

EES code for the two-phase compression model.

```
$UnitSystem SI K bar kJ mass rad

"*****INPUT VALUES*****"

"!Suction"
p_suction = 5 [bar] "Suction pressure"
x_suction = 0,985   "Suction ammonia mass fraction"

"!Volume"
V_suction = 0,001286 [m^3]  "Suction volume"
VR = 3,65   "Built-in volume ratio"
{Note: Use this parameter if the volume ratio is fixed.}

"!Discharge"
{p[N] = 25 [bar]    "Discharge pressure"}
{Note: Use this parameter if the volume ratio is variable.}

"!Frequency"
f = 50 [Hz] "Operational frequency of the compressor"

"!Injection"
p_injection = 25 [bar]  "Injection pressure"
T_injection = converttemp(C; K; 55 [C]) "Injection temperature"
x_injection = 0,4   "Injection ammonia mass fraction"
m_dot_injection = 0,01 [kg/s]   "Injection mass flow rate"
S = 33  "Segment number where injection occurs"
{Note: This segment number must be lower than the total number of
   segments, N.}

"!Leakage"
m_dot_leakage = 0,002 [kg/s]    "Leakage mass flow rate"

"!Discretization"
N = 100 "Number of calculation segments"

"!Correction factor"
C_eq = 1    "Equilibrium correction factor"

"*****CALCULATIONS*****"

"!Suction vapor properties"
T[1] = temperature(NH3H2O; P=p_suction; X=x_suction; Q=1)   "Suction
   temperature, assuming saturated vapor"
v_vap[1] = volume(NH3H2O; T=T[1]; P=p_suction; X=x_suction) "Specific
   volume of suction vapor"
h_vap[1] = enthalpy(NH3H2O; T=T[1]; P=p_suction; X=x_suction)   "
   Specific enthalpy of suction vapor"
```

```
"!Volume"
DELTAV = (1 /VR - 1) * V_suction / (N - 1)  "Volume difference between
   each segment"
V[1] = V_suction    "Volume (1)"
Duplicate i = 2 ; N
    V[i] = V[i-1] + DELTAV  "Volume (i)"
End

"!Vapor mass balance"
m_dot_vap[1] = f * V_suction / v_vap[1]     "Vapor mass flow rate (1)"
x_vap[1] = x_suction        "Ammonia mass fraction vapor (1)"
Duplicate i = 2 ; N
    m_dot_vap[i] = m_dot_vap[i-1] + m_dot_des[i-1]      "Vapor mass flow
        rate (i)"
    m_dot_vap[i] * x_vap[i] = m_dot_vap[i-1] * x_vap[i-1] + m_dot_des[i
       -1]  "Ammonia vapor mass flow rate (i)"
End

"!Specific volume of vapor"
Duplicate i = 2 ; N
    v_vap[i] = f * V[i] / m_dot_vap[i]  "Specific volume vapor (i)"
End
{Note: Volume of liquid is neglected.}

"!Vapor energy balance"
kappa = 1,32        "Isentropic exponent"
p[1] = p_suction        "Pressure (1)"
Duplicate i = 2 ; N
    w_vap[i] = v_vap[i-1] * p[i-1] * convert(bar; kPa) / (1 - 1/kappa) *
        ((p[i] / p[i-1])^(1 - 1/kappa) - 1)    "Specific work required
       to compress vapor from pressure (i-1) to pressure (i)"
    m_dot_vap[i] * h_vap[i] = m_dot_vap[i-1] * h_vap[i-1] + m_dot_vap[i]
        * w_vap[i] + Q_dot_des[i-1]    "Energy balance vapor (i)"
    v_vap[i] = volume(NH3H2O; P=p[i]; X=x_vap[i]; h=h_vap[i])   "
       Specific volume vapor (i)"
End

"!Temperature"
Duplicate i = 2 ; N
    T[i] = temperature(NH3H2O; P=p[i]; X=x_vap[i]; h=h_vap[i])  "
       Temperature (i)"
End

"!Liquid injection properties"
v_injection = volume(NH3H2O; T=T_injection; P=p_injection; X=x_injection
   )  "Specific volume of injection liquid"
```

```
h_injection = enthalpy(NH3H2O; T=T_injection; P=p_injection; X=
   x_injection) "Specific enthalpy of injection liquid"

"!Ammonia mass fraction of liquid"
Duplicate i = 1 ; N
    x_liq_eq[i] = massfraction(NH3H2O; T=T[i]; P=p[i]; Q=0) "Ammonia
       mass fraction liquid (i), if pressure and temperature equilibrium
       "
End

"!Liquid mass balance, equilibrium"
m_dot_liq[1] = - m_dot_leakage      "Liquid mass flow (1)"
Duplicate i = 2 ; S
    m_dot_liq_eq[i] = m_dot_liq[i-1] - m_dot_des_eq[i-1]        "Liquid
       mass flow (i), if equilibrium"
    m_dot_liq_eq[i] * x_liq_eq[i] = m_dot_liq[i-1] * x_liq[i-1] -
       m_dot_des_eq[i-1] "Liquid ammonia mass flow (i), if equilibrium"
End
m_dot_liq_eq[S+1] = m_dot_liq[S] - m_dot_des_eq[S] + m_dot_injection
   "Liquid mass flow where injection occurs, if equilibrium"
m_dot_liq_eq[S+1] * x_liq_eq[S+1] = m_dot_liq[S] * x_liq[S] -
   m_dot_des_eq[S] + m_dot_injection * x_injection   "Liquid ammonia
   mass flow where injection occurs, if equilibrium"
Duplicate i = S+2 ; N
    m_dot_liq_eq[i] = m_dot_liq[i-1] - m_dot_des_eq[i-1]        "Liquid
       mass flow (i), if equilibrium"
    m_dot_liq_eq[i] * x_liq_eq[i] = m_dot_liq[i-1] * x_liq[i-1] -
       m_dot_des_eq[i-1] "Liquid ammonia mass flow (i), if equilibrium"
End

"!Equilibrium correction"
Duplicate i = 1 ; N-1
    m_dot_des[i] = C_eq * m_dot_des_eq[i]       "Desorption mass flow
       rate, if non-equilibrium"
End

"!Liquid mass balance, non-equilibrium"
Duplicate i = 2 ; S
    m_dot_liq[i] = m_dot_liq[i-1] - m_dot_des[i-1]      "Liquid mass
       flow (i)"
    m_dot_liq[i] * x_liq[i] = m_dot_liq[i-1] * x_liq[i-1] - m_dot_des[i
       -1]  "Liquid ammonia mass flow (i)"
End
m_dot_liq[S+1] = m_dot_liq[S] - m_dot_des[S] + m_dot_injection  "Liquid
   mass flow where injection occurs"
```

```
m_dot_liq[S+1] * x_liq[S+1] = m_dot_liq[S] * x_liq[S] - m_dot_des[S] +
   m_dot_injection * x_injection    "Liquid ammonia mass flow where
   injection occurs"
Duplicate i = S+2 ; N
    m_dot_liq[i] = m_dot_liq[i-1] - m_dot_des[i-1]      "Liquid mass
       flow (i)"
    m_dot_liq[i] * x_liq[i] = m_dot_liq[i-1] * x_liq[i-1] - m_dot_des[i
       -1]  "Liquid ammonia mass flow (i)"
End

"!Temperature of liquid"
T_liq[1] = T[1]      "Temperature of liquid (1) equal to temperature of
   vapor (1)"
Duplicate i = 1 ; N
    T_liq[i] = temperature(NH3H2O; P=p[i]; X=x_liq[i]; Q=0) "Temperature
        liquid (i), assuming equal pressure"
End

"!Liquid energy balance"
Duplicate i = 1 ; N
    v_liq[i] = volume(NH3H2O; T=T_liq[i]; P=p[i]; X=x_liq[i])   "
       Specific volume liquid (i)"
    h_liq[i] = enthalpy(NH3H2O; T=T_liq[i]; P=p[i]; X=x_liq[i]) "
       Specific enthalpy liquid (i)"
End
Duplicate i = 2 ; N
    w_liq[i] = (p[i] - p[i-1]) * convert(bar; kPa) * v_liq[i]       "
       Specific work due to pressure difference between liquid (i-1) and
        (i)"
End
w_injection = (p[S] - p_injection) * convert(bar; kPa) * v_injection
   "Specific work due to pressure difference between injection liquid
   and liquid (S)"
Duplicate i = 2 ; S
    m_dot_liq[i] * h_liq[i] = m_dot_liq[i-1] * h_liq[i-1] + m_dot_liq[i]
        * w_liq[i] - Q_dot_des[i-1]    "Energy balance liquid (i)"
End
m_dot_liq[S+1] * h_liq[S+1] = m_dot_liq[S] * h_liq[S] + m_dot_liq[S+1] *
    w_liq[S+1] - Q_dot_des[S] + m_dot_injection * (h_injection +
   w_injection) "Energy balance where injection occurs"
Duplicate i = S+2 ; N
    m_dot_liq[i] * h_liq[i] = m_dot_liq[i-1] * h_liq[i-1] + m_dot_liq[i]
        * w_liq[i] - Q_dot_des[i-1]    "Energy balance liquid (i)"
End

"*****SUPPLEMENTARY CALCULATIONS*****"
```

```
"!Vapor quality"
Duplicate i = 1 ; N
    vq[i] = quality(NH3H2O; T=T[i]; P=p[i]; X=x_suction)    "Vapor
       quality (i)"
End

"!Component specific mass flow"
Duplicate i = 1 ; N
    m_dot_liq_ammonia[i] = m_dot_liq[i] * x_liq[i]      "Liquid ammonia
       mass flow (i)"
    m_dot_liq_water[i] = m_dot_liq[i] - m_dot_liq_ammonia[i]    "Liquid
       water mass flow (i)"
    m_dot_vap_ammonia[i] = m_dot_vap[i] * x_vap[i]      "Ammonia vapor
       mass flow (i)"
    m_dot_vap_water[i] = m_dot_vap[i] - m_dot_vap_ammonia[i]    "Water
       vapor mass flow (i)"
End

"!Compression work"
Duplicate i = 2 ; N
    W_dot_liq[i] = m_dot_liq[i] * w_liq[i]  "Liquid compression work (i)
       "
    W_dot_vap[i] = m_dot_vap[i] * w_vap[i]  "Vapor compression work (i)"
End
W_dot_injection = m_dot_injection * w_injection "Liquid injection
   compression work"
```

Appendix D   Scientific Paper

Scientific paper for the 10th IIR International Conference on Compressors and Coolants.

PAPER ID: 0385
DOI: 10.18462/iir.compr.2020.0385

MODELLING APPROACH FOR A LIQUID-INJECTED AMMONIA-WATER SCREW COMPRESSOR

Marcel Ahrens, Even Tønsberg, Ignat Tolstorebrov, Armin Hafner, Trygve Eikevik

Norwegian University of Science and Technology, Department of Energy and Process Engineering,
Trondheim, 7491, Norway, marcel.u.ahrens@ntnu.no

ABSTRACT

The present work focuses on a modelling approach for a liquid-injected ammonia-water screw compressor for combined absorption-compression heat pumps (CACHP) at high-temperature operation. The CACHP system combines the technologies of an absorption and vapor compression heat pump with a mixture of ammonia and water as working fluid. The functionality of this process has already been proven in the industrial sector using standard refrigeration components. In recent years, several studies have investigated the CACHP system to identify challenges and opportunities to increase the achievable temperature range and optimize process performance. The compressor was identified as a main critical component for increasing the achievable sink outlet temperature level. For this purpose, this study identifies currently available modelling approaches of liquid-injected screw compressors and investigates their usability in a CACHP system. Based on this, an approach for the modelling of liquid-injected ammonia-water screw compressors is developed.

Keywords: Liquid-Injected Screw Compressor, Combined Absorption-Compression Heat Pump, Ammonia-Water Mixture

1. INTRODUCTION

Increasing energy consumption and environmental pollution are of the most important challenges in today’s society. According to IEA (2019), energy consumption worldwide grew by 2.3 % in 2018, nearly twice the average rate of growth since 2010. Hence improving the efficiency of energy systems in order to decrease the consumption of resources is of great interest. Using heat pumps with environmentally friendly working fluids such as the natural fluids ammonia and water instead of traditional boilers is an effective measure to reduce energy consumption related to industrial heating applications. Arpagaus et al. (2018) recognized large application potentials, especially in food, paper, metal and chemical industries. In their analysis of the European heat pump market, they found a technical potential of 113 PJ for process heat between 100 °C and 150 °C. A promising approach for industrial high-temperature heat pump applications is provided by combined absorption-compression heat pumps (CACHP) with a zeotropic ammonia-water mixture as working fluid. The working principle of the CACHP system extends a vapor compression heat pump (VCHP) by an additional solution cycle and is based on the Osenbrück cycle (Osenbrück, 1895). This extension offers the typical characteristics of CACHP systems, such as high achievable sink temperatures in combination with large temperature lifts and non-isothermal heat transfers. Nordtvedt et al. (2013) have already demonstrated the functionality of this system in the industrial sector by using standard refrigeration components and achieving sink outlet temperatures of up to 120 °C. In recent years, various authors, such as Jensen (2015) and Nordtvedt (2005), have examined the CACHP system to identify challenges and potentials for the optimization of process parameters. In this context, the compressor was highlighted as a critical component for achieving higher sink outlet temperatures with the associated pressure levels due to the limitation of the compressor discharge temperature. The use of a liquid-injected screw compressor was identified as a promising solution. The focus of this study is the development of a modelling approach of a liquid-injected screw compressor to be utilized in a CACHP system at high-temperature operation. A CACHP test facility is currently being built in a laboratory at the Department of Energy and Process Engineering, NTNU. Among other things, the test facility will be used to examine different compressor arrangements. Using computer modelling to optimize the compressor configuration, and later being able to compare experimental data from the test facility with theoretical data from simulations, is of great value.

2. THE COMBINED ABSORPTION-COMPRESSION HEAT PUMP

This section deals with the presentation of the combined heat pump cycle with ammonia-water mixture as working fluid as well as the characteristics and requirements for compressors used in the CACHP system.

2.1. Combined Heat Pump Cycle with Ammoina-Water Mixture as Working Fluid

The most basic type of CACHP is the Osenbrück cycle, named after its inventor (Osenbrück, 1895). The Osenbrück cycle is illustrated in Figure 1. In what follows, the working principle of this cycle is explained, based on theory presented by Nordtvedt (2005) and Jensen et al. (2014).

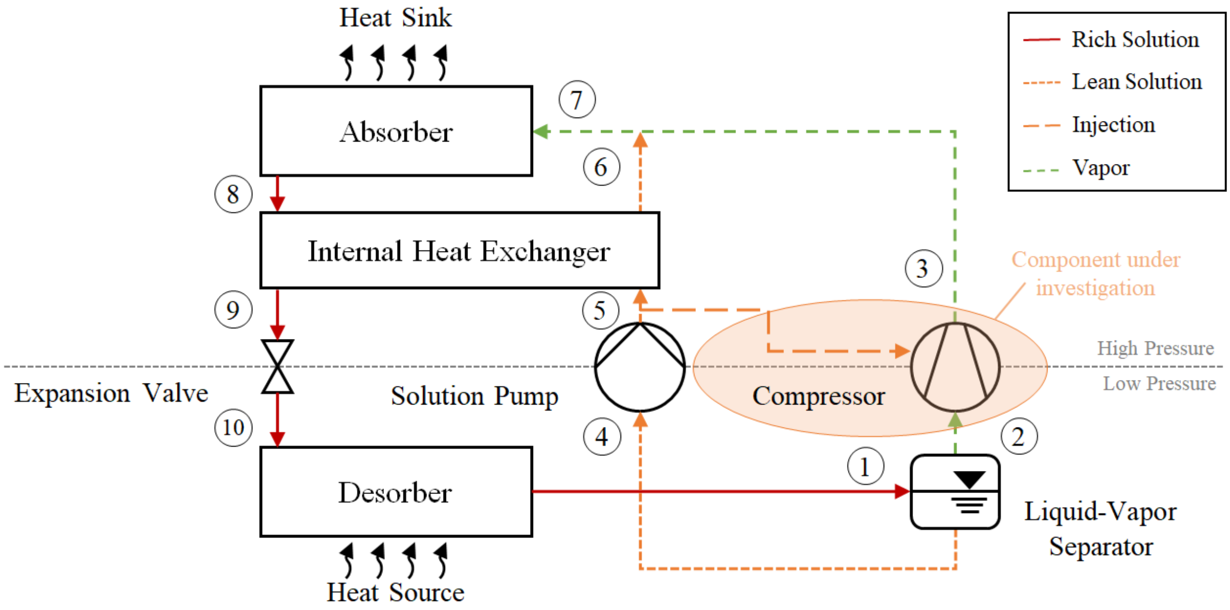


Figure 1: Simplified schematic of a combined absorption-compression heat pump cycle

In the CACHP, the evaporator and condenser are replaced by the desorber and absorber. The rich solution at low pressure (10) enters the desorber where heat from the source is transferred. This causes the temperature to rise and the solubility of the ammonia-water mixture to decrease. Hence vapor is generated, and a two-phase mixture leaves the desorber (1). The low-pressure vapor and lean solution are separated before they are drawn towards the compressor (2) and pump (4) respectively. The compressor increases the pressure and temperature of the vapor (2 to 3), while the pump elevates the the pressure of the lean solution correspondingly (4 to 5). The temperature rise over the pump is minor. Therefore, to improve the performance of the cycle, an internal heat exchanger (IHX) is added to the solution sub-cycle. Heat is exchanged between the lean and rich solutions, causing a temperature rise in the lean solution (5 to 6) and a temperature drop in the rich solution (8 to 9). The high-pressure lean solution (6) is mixed with the high-pressure vapor (3) at the entrance of the absorber (7). In the absorber, vapor is absorbed into the liquid phase rejecting heat to the heat sink fluid. The ammonia concentration in the liquid phase gradually increases, so that the absorption process leads to a saturated liquid at the absorber outlet. The rich solution eventually passes through the solution heat exchanger before being throttled to desorber pressure (8 to 9 to 10), thus completing the cycle.

Nordtvedt (2005) and Jensen et al. (2014) point out several advantages with utilization of CACHP. Compared to a VCHP using a single working fluid, the working fluid pair in a CACHP offers much more design flexibility. By altering the composition of the working pair, i.e., the refrigerant-to-absorbent ratio, the properties of the working fluid can be adapted to the heat source and heat sink properties, as well as properties of each heat pump component, to obtain optimal working conditions. Industrial heat supply and recovery of surplus heat requires large heat source and heat sink temperature glides. Contrary to the isothermal processes of evaporation and condensation in a VCHP, the absorption and desorption processes in a CACHP are non-isothermal processes. By matching the temperature glides of the working fluid with the temperature glides of the heat source and heat sink fluids, the thermal losses are reduced, resulting in a higher COP. This makes the CACHP suitable for many industrial applications. Another suitable feature of the CACHP is that using a zeotropic mixture reduces the required high-pressure level. The sink temperature from a conventional VCHP is limited because the condenser pressure becomes too high, whereas a CACHP can achieve much higher sink temperatures due to the reduced high-pressure level.

2.2. Characteristics and Requirements for Compressors Used in the CACHP System

Based on the results of a previous investigation of available compressors for use in an ammonia-water CACHP system, the following characteristics and requirements for the compressor are listed (Ahrens et al., 2019):

Material compatibility

Ammonia can cause corrosion in contact with materials such as copper, copper-based alloys and zinc. It is of high importance to choose ammonia-compatible materials in all parts of the compressor that are in contact with the working fluid. Stainless steel and aluminium are fully compatible with ammonia, and therefore among the most widely used materials in CACHP systems.

Discharge temperature

Due to the relatively low density and specific heat capacity of ammonia, high temperatures can arise during compression. Jensen (2015) identified the discharge temperature as the dominating constraint in an ammonia-water CACHP. Excessive temperatures can cause decomposition of the working fluid, breakdown of seals and compressor failure. It is therefore crucial to select a compressor capable of handling or curtailing high discharge temperatures.

Liquid resistance

Due to the design of the CACHP system and the properties of the ammonia-water mixture, it is possible that the suction vapor will contain droplets of water. Hence, it is important that the compressor is resistant to liquid carry-over. Choosing a liquid-resistant compressor also gives much more operational flexibility, e.g., the possibility of operating closer to the two-phase region and using liquid injection.

Oil-free operation

It is desirable to have a compressor capable of oil-free operation, since oil has been proven to diminish the performance of the rest of the CACHP system (Zaytsev and Infante Ferreira, 2002; Nordtvedt, 2005). Additionally, costs related to oil separation are avoided.

A screw compressor with injection of liquid ammonia-water should be capable of meeting all the above-mentioned characteristics and requirements. According to investigations of Zaytsev and Ferreira (2002) and other authors, liquid injection has a generally positive influence on the volumetric and isentropic efficiency of the compressor and can serve various functions, such as lubrication, sealing and decreasing of the discharge temperature through the compression process. However, the complexity of the processes involved in injection and compression also leads to various challenges. On the one hand, the optimal location for an effective injection must be determined. Then, the quantity and type of injection must be determined with respect to the effects on the compression process and performance. The use of the zeotropic ammonia-water mixture in a two-phase process represents a special challenge due to its thermodynamic properties.

As such a liquid-injected screw compressor is not yet commercially available, there is great uncertainty regarding the design and operation of the compressor. The analysis of the compressor with the help of thermodynamical and mathematical models can provide a better understanding and a reduction of the uncertainties. In the following, the developed modelling approach is presented accordingly.

3. IDENTIFICATION OF EXISTING MODELLING APPROACHES

The aim of the development of a modelling approach is the investigation of compressor behavior as well as a better understanding of the ongoing processes and associated phenomena. A computer model of the liquid-injected screw compressor can help to determine the required injection mass flows based on the fulfillment of the given requirements, such as the maximum discharge temperature. Additionally, the number and position of the inlet ports can be adjusted to optimize their placement for the intended operation. The CACHP test facility is capable of testing different compressor arrangements, as described previously. This involves varying the streams used for liquid injection. Specifically, both the lean solution downstream of the solution pump, e.g. upstream of the IHX (see Figure 1), and the strong solution upstream of the expansion valve can be used. As a result, the mass fraction of ammonia and water of the injected liquid varies depending on the stream used and the effects on compressor performance can be investigated. Subsequently an optimization can be carried out. Another important function of the computer model is the planning and preparation of experimental tests and the subsequent evaluation and validation of the results. Since the design of experiments and evaluation is also desirable for the entire CACHP system, the screw compressor model should be integrable into bigger system models. Further on, approaches available in the literature are presented and examined regarding their usability.

Liquid-Injected Screw Compressor Models in the Literature

A large number of models of varying complexity and scope are described in the literature. Three main types of modelling techniques are the most common when dealing with computer modelling of screw compressors, namely empirical, analytical and numerical methods. Empirical models are based on curve fitting or statistical analysis of test data. As very little test data is available for the type of compressor under investigation here, an empirical approach is considered unsuitable. Analytical models are based on pressure-volume diagrams developed from simple analytical curves, e.g., polytropic curves, or test data. Since ammonia-water is a zeotropic mixture with intricate thermophysical properties using an analytical approach may involve uncertainties and challenges. However, employing analytical curves to a certain extent can be an effective means to simplify complex thermodynamic problems. In the two-phase compression model discussed below, polytropic relations are employed to simplify calculations regarding compression work. In cases where test data and/or analytical relations are not sufficient, numerical models must be developed. Numerical screw compressor models can be either quasi-one-dimensional of three-dimensional, depending on its field of application. Quasi-one-dimensional models are based on mass and energy conservation equations applied to a relatively small number of chambers. In three-dimensional models on the other hand, the compressor is divided into countless finite-sized elements and conservation of momentum is applied in addition to conservation of mass and energy. Three-dimensional modelling, more commonly known as CFD analysis, involves heavy calculations and time-consuming developing procedures. For the model under development here it is desirable to be able to investigate multiple compressor configurations, e.g., injection port locations, in an efficient manner, and three-dimensional modelling is thus considered to be unsuitable. The focus is set on quasi-one-dimensional numerical models.

Zaytsev (2003) developed a numerical wet screw compressor model. He defined the control volume as the volume of one rotor cavity, and established equations for conservation of mass and energy within each control volume. Equations for both homogeneous and heterogeneous conditions were developed, but only the homogeneous model was implemented due to lack of sub-models for mass and heat transfer required in the heterogeneous model. Infante Ferreira et al. (2006) further developed the homogenous model and utilized it in theoretical investigations on the ideal injection port location. Zaytsev's equations are exclusively fitted for step-by-step calculation of changes in pressure and temperature of the ammonia-water mixture throughout the compression process, and thus the equations are rather complex and challenging to interpret. This makes the model hard to modify or further develop and unsuited for implementation into bigger system models. Chamoun et al. (2013) presented a different numerical approach to screw compressor modelling. Their model was developed and solved using the Modelica language [Modelica Association] and Dymola [Dassault Systemes, France]. Compared to Zaytsev, the approach developed by Chamoun et al. offers less complexity and higher flexibility with respect to implementation into bigger system models. However, the model is based on compression of pure water, and the liquid injection port is rigidly placed at the suction end. The Modelica model structure is considered to be a good starting point, but substantial adaptations must be made to the model in order to utilize it for investigations on ammonia-water screw compressors with optional injection port locations.

4. MODELLING APPROACH OF A LIQUID-INJECTED SCREW COMPRESSOR

A screw compressor simulation model is developed using the Modelica language, based on knowledge obtained from existing compressor models in the literature. The model takes into account the effect of leakage flows, heat losses and liquid injection, and it can be used for both steady-state and transient analysis of such phenomena. The model is aimed at efficient investigations on different compressor arrangements and operating conditions, as well as effortless implementation into bigger system models.

It is a quasi-one-dimensional numerical model with multiple control volumes, where each control volume is defined as the volume of one rotor cavity. Basic equations for conservation of mass and energy is applied to each control volume. The thermodynamic properties of ammonia-water are continuously calculated through utilization of the TILMedia library [TLK-Thermo GmbH, Germany] in Modelica, under the assumption that conditions within each control volume are homogeneous.

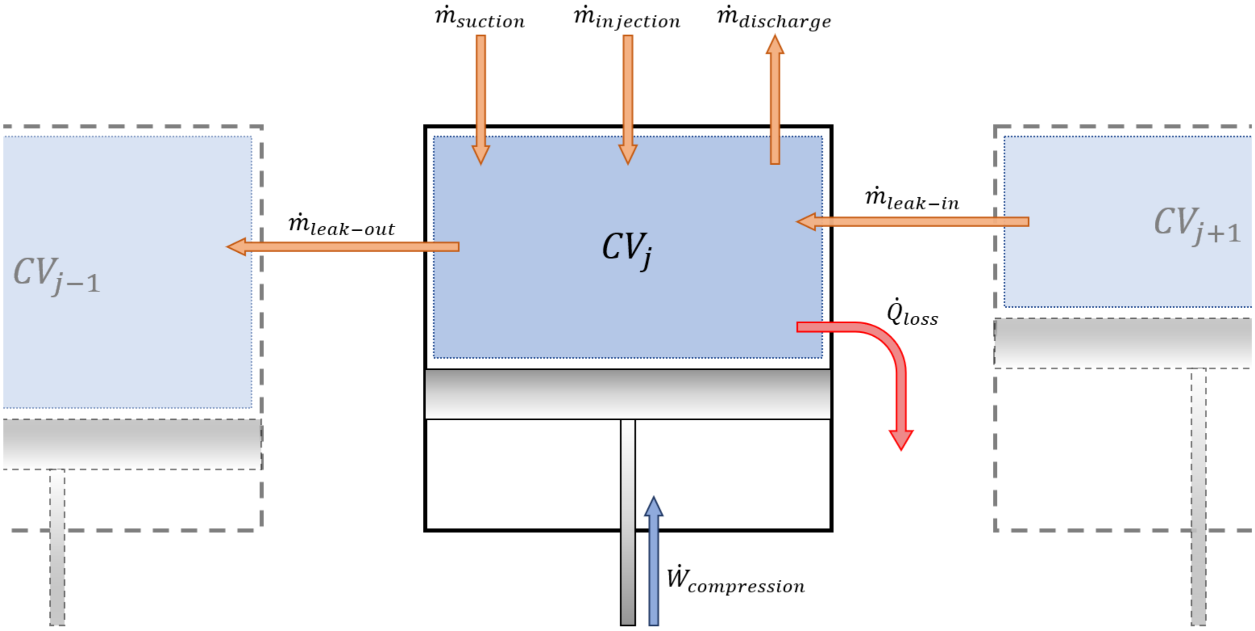


Figure 2: Schematic showing the mass and energy flows associated with each control volume

The size of the control volumes varies with time based on the rotational speed and the geometrical properties of the compressor. The volume increases when a control volume is in the suction phase, while it decreases during the compression and discharge phases. The volume variation is identical for all the control volumes, but there is a time delay between each one, e.g., control volume *j* can undergo a volume decrease simultaneously as control volume *j-1* undergoes a volume increase. This results in different pressure, temperature, mass and internal energy in each control volume at each point in time. The flow of mass and energy due to leakage gaps in the compressor is taken into account by interconnecting the control volumes as illustrated in Figure 2. Leakage mass flow rates are calculated from

$$\dot{m} = A_{eff}\sqrt{2\rho\Delta p} \qquad \text{Eq. (1)}$$

where A_{eff} is the effective flow area of the leakage gap, ρ is the density of the working fluid upstream of the flow, and Δp is the pressure difference across the gap. Mass flow rates related to suction, injection and discharge are also calculated using Equation 1. For instance, in calculations of suction mass flow rates A_{eff} respresents the effective flow area of the suction port, ρ represents the density of the fluid in the suction line, and Δp represent the pressure difference between the suction line and the control volume. Here A_{eff} varies with time as the control volume goes through the suction phase, while it is set to zero during the compression and discharge phases.

The model is implemented and solved using the Dymola modelling environment. Compared to screw compressor models found in the literature the developed model has low complexity, making it easier to interpret and to modify in accordance with special analysis requirements. Regardless of the relatively low complexity the model can be used to produce detailed data for numerous compressor configurations without the need of experimental data.

Two-Phase Compression Model

The previous described Modelica model can give detailed insight into the process that takes place within a screw compressor during operation. However, the model has some analysis limitations. In each compression chamber the working fluid is treated as a single unit with homogeneous properties. This forces certain constraints on the state of the fluid, e.g., liquid and superheated vapor cannot be present simultaneously in one chamber. In reality, a thin film of liquid can be present along the inner surface of a compression chamber even though the vapor inside the chamber is in a superheated state. As a matter of fact, a portion of the working fluid always being in a liquid state is essential for safe and efficient operation of oil-free screw compressors. In addition, TILMedia library used for ammonia-water property calculations in the Modelica model is based on a vapor-liquid equilibrium assumption. With this modelling approach it is not possible to investigate the potential consequences of non-equilibrium during high-speed operation.

To compensate for any inaccuracies the Modelica model may have, a secondary calculation model is developed using the Engineering Equation Solver (EES) [F-Chart Software, USA]. This model is to be used for detailed thermodynamic analysis of the two-phase process that takes place during the compression phase. Contrary to the Modelica model, the EES model disregards the suction and discharge phases. Polytropic relations are employed to simplify calculations regarding compression work. The aim is to get a better understanding of the intercorrelation between liquid and vapor during two-phase compression, and thereby being able to carry out comprehensive investigations on the impact of liquid injection. Figures 3 and 4 illustrate the schematic representation of the mass and energy flows associated with each segment.

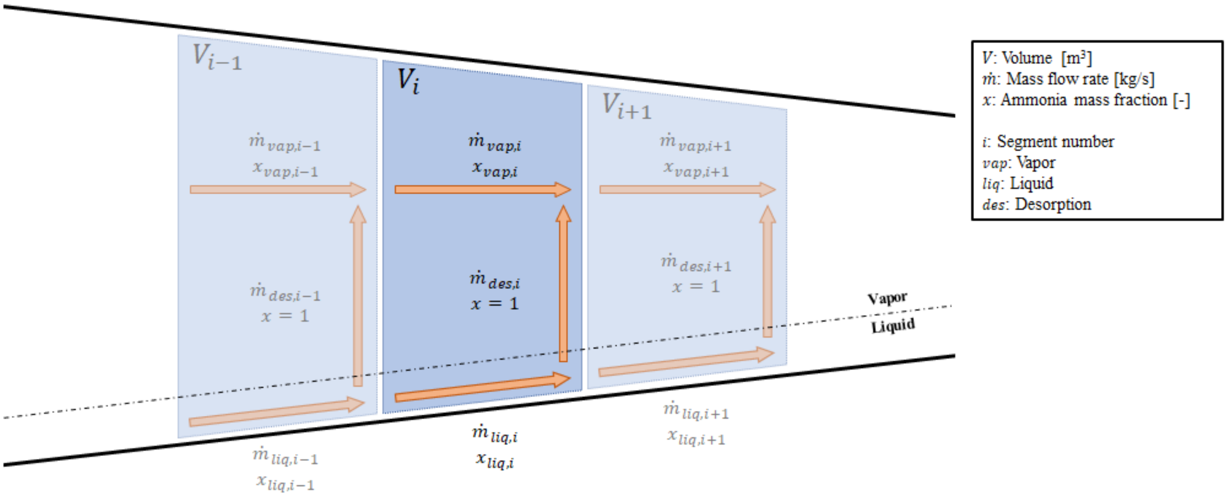


Figure 3: Schematic representation of the mass flows associated with each segment

The total volume of the compressor is divided into a variable definable number of segments. Mass and energy flows, as shown in the Figures 3 and 4, flow through the individual segments with their respective pressure, temperature and ammonia mass fraction for the gas and liquid phase. This is a different approach compared to the Modelica model, where the mass and energy flows are kept within the changing control volumes. The liquid injection is performed in single segments, which can be variably selected and varied. During evaporation of the liquid with heat extraction of the superheated vapor caused by the compression process, it is expected that ammonia desorption (*x=1*) from the liquid will occur due to the thermodynamic properties of the ammonia-water mixture. Generated backflows as well as temperature-related condensation and/or absorption processes are considered and represented with negative sign (contrary to the arrows shown). For the investigation of potential consequences, a non-equilibrium factor between vapor and liquid phase is implemented for each segment.

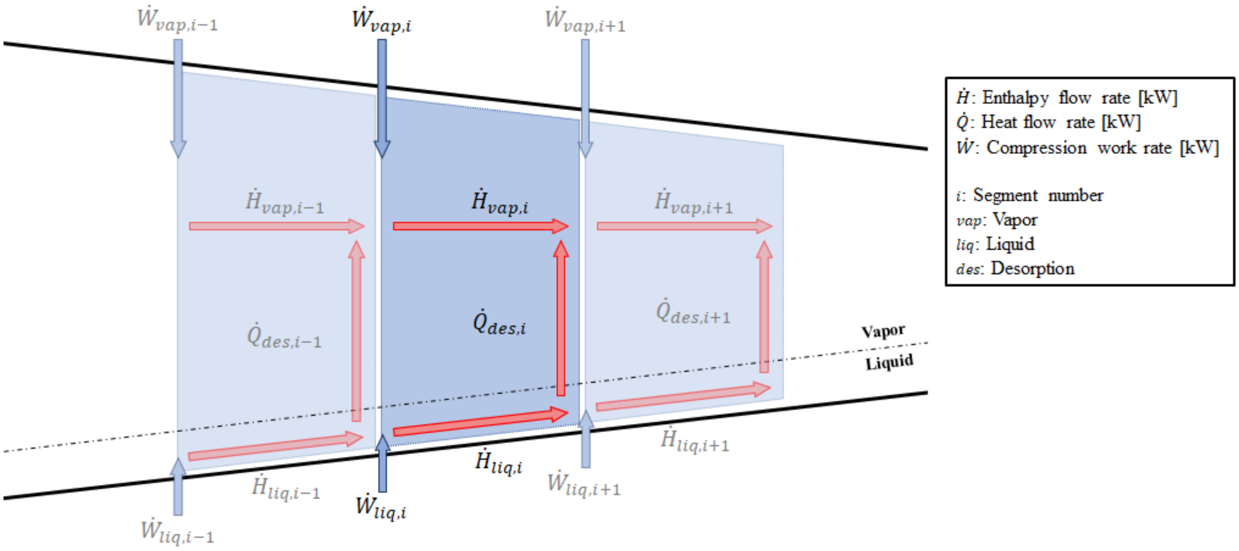


Figure 4: Schematic representation of the energy flows associated with each segment

The implemented functions in the two-phase compression model thus enable the detailed thermodynamic analysis originally described and provide a supporting approach to the Modelica model.

5. CONCLUSIONS

Within the scope of this study, a modelling approach for a liquid-injected ammonia-water screw compressor was developed, to be utilized in a CACHP system at high-temperature operation. At the beginning, the general functionality and advantages of the CACHP system were presented, followed by a discussion of the characteristics and requirements for the compressor to be used. Subsequently, the objectives for the development of the modelling approach were specified and, based on this, existing models in the literature were identified and examined with regard to their usability in a CACHP system. Based on this, a modelling approach for liquid-injected ammonia-water screw compressors was developed.

The developed screw compressor model was implemented and solved using the Modelica language with the modelling environment Dymola. The model takes into account the effect of leakage flows, heat losses and liquid injection, and it can be used for both steady-state and transient analysis of such phenomena. Compared to other screw compressor models in the literature, the developed model has a low complexity, which makes it easier to interpret and modify in accordance with the specific analysis requirements. Despite the relatively low complexity, the model can be used to generate detailed data for numerous compressor configurations without the need of experimental test data. An additional calculation model was developed to compensate for any inaccuracies the Modelica model may have. This model is intended to be used for a detailed thermodynamic analysis of the two-phase compression process and provides the possibility to vary the injection ports as well as the equilibrium factor to represent the vapor-liquid phase in non-equilibrium.

As defined in the requirements, the developed modelling approach aims at efficient investigations of different compressor arrangements and operating conditions as well as at an easy implementation into larger system models. In the further course of the project the model will be extensively tested, evaluated and verified. Subsequently, the simulation results will support the planning and preparation of different compressor arrangements as well as the experimental evaluation.

ACKNOWLEDGEMENTS

The work is part of HighEFF - Centre for an Energy Efficient and Competitive Industry for the Future, an 8-year Research Centre under the FME-scheme (Centre for Environment-friendly Energy Research, 257632). The authors gratefully acknowledge the financial support from the Research Council of Norway and user partners of HighEFF.

REFERENCES

Ahrens, M.U., Hafner, A., Eikevik, T.M., 2019. Compressors for ammonia-water hybrid absorption-compression heat pumps. In 8th IIR Conference of Ammonia and CO2 Refrigeration Technologies, Ohrid.

Arpagaus, C., Bless, F., Uhlmann, M., Schiffmann, J., Bertsch, S. S., 2018. High temperature heat pumps: Market overview, state of the art, research status, refrigerants, and application potentials. Energy 152, 985–1010.

Chamoun, M., Rulliere, R., Haberschill, P., Peureux, J., 2013. Modelica-based modeling and simulation of a twin screw compressor for heat pump applications. Applied Thermal Engineering 58, 479–489.

IEA, 2019. Global Energy & CO2 Status Report 2018. Annual Report. Paris: International Energy Agency, p. 4.

Infante Ferreira, C. A., Zamfirescu, C., Zaytsev, D., 2006. Twin screw oil-free wet compressor for compression–absorption cycle. International Journal of Refrigeration 29, 556–565.

Jensen, J.K., 2015. Industrial heat pumps for high temperature process applications – A numerical study of the ammonia-water hybrid absorption-compression heat pump. Ph.D. thesis, Technical University of Denmark.

Jensen, J. K., Reinholdt, L., Markussen, W. B., Elmegaard, B., 2014. Investigation of Ammonia/Water Hybrid Absorption/Compression Heat Pumps for Heat Supply Temperatures Above 100 °C. Proceedings of the International Sorption Heat Pump Conference. Vol. 1. Center for Environmental Energy Engineering, 311–320.

Nordtvedt, S.R., 2005. Experimental and theoretical study of a compression/absorption heat pump with ammonia/water as working fluid. Ph.D. thesis, Norwegian University of Science and Technology.

Nordtvedt, S.R., Horntvedt, B.R., Eikefjord, J., Johansen, J., 2013. Hybrid heat pump for waste heat recovery in Norwegian food industry. Thermally driven heat pumps for heating and cooling, 57-61.

Osenbrück, A., 1895. Verfahren zur Kälteerzeugung bei Absorptionsmaschinen. Deutsches Reichspatent DRP 84084.

Zaytsev, D., Infante Ferreira, C.A., 2002. Screw Compressor for Ammonia-Water Heat Pump Lubricated by The Process Mixture. International Compressor Engineering Conference. Paper 1581.

Zaytsev, D., 2003. Development of Wet Compressor for Application in Compression-Resorption Heat Pumps. Ph.D. thesis, Delft University of Technology.

Even Kristian Tønsberg

Modeling Approach for a Liquid-Injected NH_3-H_2O Screw Compressor

NTNU
Norwegian University of
Science and Technology