Modeling of a Stand-alone Liquid Air Energy Storage

Fabian Andersson

Thesis for the degree of Master of Science in Engineering
Division of Thermal Power Engineering
Department of Energy Sciences
Faculty of Engineering | Lund University

Modeling of a Stand-alone Liquid Air Energy Storage

Fabian Andersson

June 2020, Lund

This degree project for the degree of Master of Science in Engineering has been conducted at the Division of Thermal Power Engineering, Department of Energy Sciences, Faculty of Engineering, Lund University, and at Siemens Industrial Turbomachinery (SIT).

Supervisor at the Division of Thermal Power Engineering was Professor Magnus Genrup
Supervisor at SIT was Lennart Naes
Examiner at Lund University was Senior Lecturer Marcus Thern

Thesis for the Degree of Master of Science in Engineering

ISRN LUTMDN/TMHP-20/5457-SE

ISSN 0282-1990

Division of Thermal Power Engineering
Department of Energy Sciences
Faculty of Engineering, Lund University
Box 118, 221 00 Lund
Sweden

www.energy.lth.se

Abstract

The aim of this project was to create a model of the stand-alone LAES for use in future work at Siemens Industrial Turbomachinery (SIT). The model was created in IPSEpro with components developed from the software's standard library of power plant components. Required fluid properties were obtained from the Refprop DLL. This was accessed through the development of a DLL, functioning as a link between IPSEpro and the Refprop DLL. The model of the LAES was created from the plant layout proposed by Guizzi et al. (2015), along with presented stream data.

Simulations with the final model showed relatively small deviations in stream data and round-trip efficiency when compared to the reference. A test case was conducted to evaluate the model's capability of handling an off-design scenario with boil-off in the liquid air tank. The result showed that to compensate for the lost mass of liquid air the round-trip efficiency would be reduced from 53,23% to 50% over a stand-by duration of approximately ten days. The way the system compensated for the limited amount of cold energy by increasing the total compression ratio to maintain the rated capacity was considered reasonable. It could be concluded that the model could be useful for further evaluation of the energy storage system, but depending on the scenarios of interest, more development of the model may be necessary.

Acknowledgements

First of all, I would like to thank my supervisor at SIT, Lennart Naes, who has given me great support throughout this project. Next, I would like to thank Dr. Klas Jonshagen for the guidance and all the help, especially with the IPSEpro software. It would not have been possible to complete this project without it. Thanks to the people at the performance department for making me feel welcome during my time at the Siemens office, even though it was short given the unique circumstances. Many thanks to Prof. Magnus Genrup, not only for the support during this project, but also for all the guidance and help throughout my whole specialization. Finally, it has been a privilege to get the opportunity to work with this project at SIT, and I hope that the model will become useful in future projects regarding the LAES.

Nomenclature

Abbreviations

CAES	Compressed air energy storage	
CES	Cryogenic energy storage	
DCTES	Direct cold thermal energy storage	
DLL	Dynamic link library	
LAES	Liquid air energy storage	
LCTES	Liquid cold thermal energy storage	
LNG	Liquid natural gas	
PHES	Pumped hydroelectric energy storage	

Greek

Δ	Difference	
η	Efficiency	$-$
ρ	Density	m^3/kg

Latin

A	Area	m
D	Diameter	m
h	Height	m
h	Specific enthalpy	J/kg
k	Thermal conductivity coefficient	$W/m \cdot K$
P	Pressure	Pa
Q	Transferred heat	J
r	Radius	m
T	Temperature	K
t	Time	s

V	Volume	m^3
W	Work	J
w	Specific work	J/kg
x	Thickness	m
Y	Liquid air yield	

Subscripts

1	Initial or inlet state
2	Final or exit state
L	Liquefaction section
R	Recovery section
s	Isentropic

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last decade, there has been a significant increase in investments of renewable energy capacity, and the two technologies that have received the most attention are solar and wind. The reason for this increase in renewables is mainly a great improvement in cost-competitiveness. In many countries around the world, this has led to renewables becoming the cheapest option for new installment (Frankfurt School-UNEP Centre/BNEF 2019). However, what needs to be taken into consideration is that a full-scale transition towards 100% renewable energy requires a correspondingly large investment in energy storage capacity in order to make it viable. Renewable energy sources in the absence of energy storage or auxiliary power production systems will due to its intermittent behavior not be able to match the peak demands of energy consumption.

So, to increase the value of renewable energy sources, there is a need for the development of new grid-scale electric energy storage, and an interesting solution is the Liquid air energy storage (LAES) which belongs to the storage category of cryogenic energy storage. The storage method is based on the technology of using liquefied air as a storage medium. How the system operates can be separated into three main steps. As a first step, the charging process uses excess electrical energy to compress and liquefy air. The next step is the storage, where liquefied air is stored in a tank at a temperature of about -194°C and close to ambient pressure. The third and last step is the discharge section, where the energy is recovered through pumping, reheating, and expansion in a turbine.

Advantageous is that the LAES rely on already existing technology that is mature and well tested. A highly essential part of the LAES is an implementation of the technology around gas liquefaction, which is frequently used in the industry. And besides consisting of well-known technology, the LAES comes without the geographical limitation compared to other mature grid-scale storage systems such as compressed air (CAES) and pumped hydro (PHES), as it is not dependent on suitable underground geology or elevation difference between two reservoirs. Also, the fact that energy is stored in liquid air means that the storage volume needed has a significant reduction compared to the CAES (Guizzi et al. 2015). The technology has begun to be used commercially. With the technology already tested in a pilot plant and demonstration plant, the company Highview Power announced its first commercial power plant in late October 2019 (Highview Power 2020b). The plant has a rated capacity of 250 MWh and power of 50 MW, demonstrating that the LAES could have great potential as a solution for future grid-scale energy storage

(Highview Power 2020a).

1.1 Background

Plenty of research has been done in the area of the LAES. This can be separated into three main categories, where we have the optimization of air liquefaction, utilization of cold and heat storage, and integration with external technology.

Optimization of air liquefaction: Several studies have focused on the liquefaction process, which is the most power-consuming process of the system. Morgan et al. (2015) studied the twin-turbine Claude cycle, and suggested a rearrangement and increase of turbines due to restrictions on the reuse of cold energy. Abdo et al. (2015) made a comparison between Linde-Hampson, Claude, and the Collins cycle resulting in the conclusion that the Claude cycle had economic advantages over the Collins cycle, which still had a very similar power output. Borri et al. (2017) analyzed the three different liquefaction cycles for small-scale plants. These were the Lidne-Hampson, Claude, and Kapitza cycles, concluding that kapitza was determined to be the best option.

Cold and heat storage: During compression of the air in the liquefaction process, a large amount of heat is generated, and during the discharge process of the LAES, the liquid air holds much cold energy, which can be reused. The recovery and utilization of this heat and cold energy are crucial processes of the stand-alone LAES. Inefficient recycling of heat and cold energy has been shown to have a significant impact on the performance of the LAES, with a specific emphasis on the recovery of the cold energy (X. Peng, She, Cong, et al. 2018).

She et al. (2017) studied the stand-alone LAES in regard of energy storage, and how losses of heat and cold energy affected the round-trip efficiency. It was concluded that thermal insulation is essential, particularly for the liquid air storage tank.

Sciacovelli, Vecchi, and Ding (2017) looked into the possibilities of using packed beds for cold thermal energy storage and found that through recycling the cold from discharge, the energy needed to liquefy air was reduced by approximately 25%. H. Peng et al. (2018) studied the influence of using packed beds as hot thermal energy storage for the compression heat. The results showed that an efficiency in the range of 50-62% could be obtained. Hüttermann et al. (2019) went in the same direction as a comparison was made between different cold thermal energy storage devices. These were liquid cold thermal energy storage (LCTES), based on methanol and propane, and direct cold thermal energy storage (DCTES), based on packed beds of solid material.

However, more research is needed to make a proper evaluation of the use of packed beds. The downsides with it are the dynamic effects that occur because of thermal front propagation in the packed bed. Degradation of the thermal front in the packed bed leads to a decrease in the amount recycled cold during discharge. The efficiency of liquefaction is reduced with time during the charging process as the outlet temperature from the cold packed bed increases (Sciacovelli, Vecchi, and Ding 2017). In comparison, LCTES has a lower efficiency but is a steady straightforward process based on well-known technology, and the DCTES has a transient behavior that does not enable much control (Hüttermann et al. 2019).

Integration with external technology Y. Li et al. (2014) studied the integration of a LAES system with a nuclear power plant where the heat in the discharge section of the cycle was supplied by steam bled from the power plant. In terms of cold energy, the liquefaction and discharge section were coupled through a cold energy storage involving propane and methanol. Due to effective utilization of cold energy recovery and reheat in the turbine configuration, as well as optimistically chosen values of design parameters, the cycle was able to reach round trip efficiencies above 70%.

According to X. Peng, She, Cong, et al. (2018), the heat recovered from the compression was shown to be 20-45% in excess, depending on the charging and discharging pressures. The excessive amount of heat leaves space for the integration of systems that could utilize this heat. This was tested with the introduction of an organic rankine cycle (ORC), leading to a round-trip efficiency of over 60%. For storage, Thermal oil was used for the hot thermal storage and propane and methanol for cold thermal storage. Zhang et al. (2020) proposed a LAES in combination with a supercritical ORC, using a combination of thermal oil and water for thermal storage to reach higher temperatures. A round-trip efficiency of up to 56.86% was achieved.

In comparison to utilizing an external heat source Kim, Noh, and Chang (2018) proposed a LAES integrated with liquefied natural gas. They proposed a cycle where liquid natural gas (LNG) is integrated into the LAES to combine the effects of storage and power generation. This way a round-trip efficiency of 64.2% was obtained with the added energy from the fuel combustion. As stand-alone storage, the storage efficiency reached 73.4%. X. Peng, She, C. Li, et al. (2019) also investigated the possibilities of combining LAES with LNG and were able to conclude that one of the main reasons the stand-alone cycle has a limited cycle efficiency is because of the lack of cold energy input needed for the liquefaction of air. They found that a round-trip efficiency of the hybrid configuration could reach 88% by utilizing the cold energy during the regasification of LNG to liquidize the air before it is expanded in a turbine and delivered to customers.

The stand-alone system: Based on the work by Y. Li et al. (2014) with the integration of a nuclear power plant, Guizzi et al. (2015) performed a thermodynamic analysis of a LAES cycle to investigate what efficiencies could be obtained with a stand-alone cycle configuration without heat input from an external source. Instead, the excessive heat from the compression would be stored and used during discharge. The cycle resembles some parts of the cycle proposed by Y. Li et al. (2014), especially the liquefaction and cold storage. The use of thermal oil was proposed to make the system independent from external heat sources. For the cold thermal storage, the same fluids were used as proposed by Y. Li et al. (2014). Their analysis resulted in round-trip efficiencies around 54-55%. This value was considered a reasonable result as they estimated that the efficiency in practice could reach at least 50%.

The research has shown what possibilities there are with optimization of the cold and hot storage and utilization of external heat and cold energy sources for the improvement of the round-trip efficiency. It showed that the storage system could be combined with other technology to achieve higher performance. However, it is of interest to lay a foundation for further evaluation of the LAES. For this, a functioning model of the stand-alone system will be required. Moreover, the source considered suitable for this task is the work performed by Guizzi et al. (2015). The model will be based on the proposed plant layout, together with the associated data.

1.2 Objectives

The purpose of this thesis was to lay a foundation for future work regarding the LAES by creating a model of the stand-alone storage system. The expectations were to gain an increased understanding of the storage and what possibilities and potential the technology possesses. The main objectives of the work process are listed as follows.

- Conduct a literature study to gain knowledge from previous research and to find a suitable data source.
- Develop a DLL functioning as a link between the Refprop DLL and IPSEpro.
- Create a model of the stand-alone LAES in the software IPSEpro
- Validate the LAES model by comparison with the reference data.
- Run a test case with boil-off in the liquid air tank to see how the system handles a simple off-design scenario.

1.3 Limitations

There are a lot of details that can be looked into with the modeling of the LAES. Even though this thesis mainly focused on the thermodynamics of the storage system, the relatively short time frame meant that certain delimitations were be needed to limit the workload. For this reason, the following was considered.

- The model only handled steady-state calculations, whereas transient analysis was not included.
- Even though the streams could handle multiple components the separator, along with the recirculation, was only able to handle two fluids. Therefore, ambient air was considered to be a mixture of 77% nitrogen and 23% oxygen.
- Pressure drops were only expressed in terms of a percentage in heat exchangers, pressure losses in pipes was left out.
- Only the impact of losses in the liquid air storage tank was looked into with the intention of evaluating the models capability of handling a simple off-design scenario
- Work producing and consuming components was only expressed in terms of isentropic efficiencies.
- No economic aspects was included, yet the plant's round-trip efficiency was studied which is a performance parameter that can be directly linked to financial evaluations.

Chapter 2

Theory

This chapter will introduce the working principle of liquid air energy storage (LAES) together with the fundamental thermodynamic equations constituting the system process. The software used for the project will also be presented.

2.1 Liquid air energy storage

The LAES belongs to the storage category of cryogenic energy storage (CES), and the working principle of the CES is illustrated in Figure 2.1, where the left half can be described as the liquefaction section and the right as the recovery section. The general process steps with internal heat and cold recycling can be described as follows.

- **Compression:** Ambient air is first filtered and compressed to high pressure. Through intercooling, the excess heat from compression is stored in a hot thermal storage.

- **Purification:** Molecular sieves are used to remove impurities such as water vapor and carbon dioxide from the compressed air.

- **Cooling:** To liquefy the purified air, low temperatures much be reached. First, the air is cooled to approximately -175 °C using internally stored cold energy extracted during discharge and flows of cold air generated during the liquefaction process. By expansion in a cryogenic turbine, the compressed air is cooled further reaching partial liquefaction.

- **Separation and storage:** The partially liquefied air is separated, whereas the liquid air is stored, and the rest is used in the cooling process.

- **Evaporation** During the first part of the discharge section, the pressure of the liquid air is increased by a cryogenic pump. It is then evaporated, and the excess cold energy is stored in a high-grade cold storage.

- **Expansion:** The excess heat stored from the compression is used to superheat the air before it undergoes expansion where power is generated.

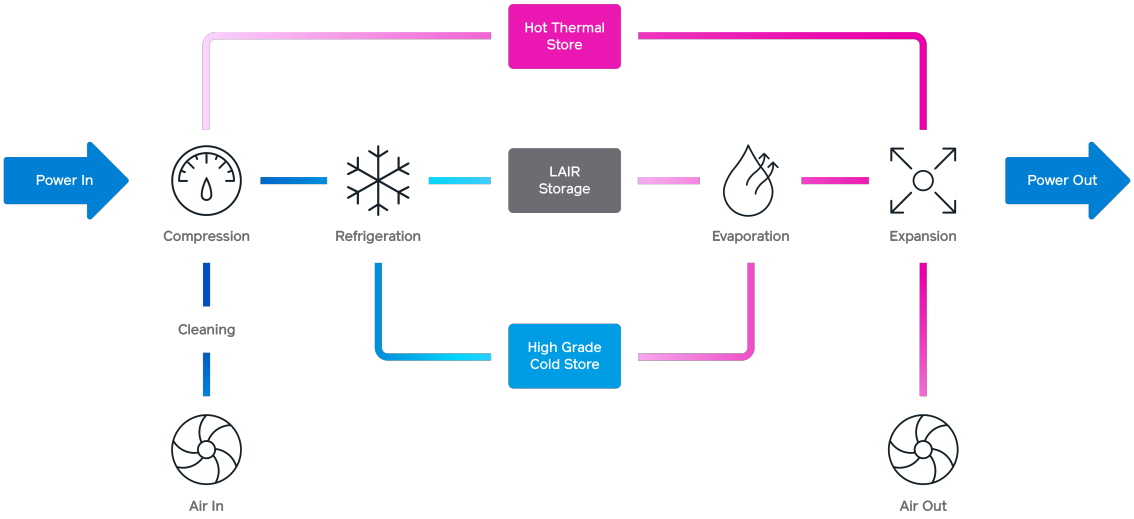

Figure 2.1: How cryogenic energy storage systems work (Highview Power 2020b)

2.1.1 System performance indices

The performance of the stand-alone LAES can be measured in terms of the round-trip efficiency, η_{RT}, which is a key parameter that is defined as the ratio between the net work output of the recovery section and the net work input of the liquefaction section,

$$n_{RT} = \frac{W_{out}}{W_{in}} = \frac{W_{air,tur} - W_{cryo,pump}}{W_{air,com} - W_{cryo,tur}} = \frac{m_R w_R}{m_L w_L} \tag{2.1}$$

where, $W_{air,tur}$ and $W_{cryo,tur}$ denotes the work produced by the air turbine and the cryogenic turbine, respectively. $W_{air,com}$ and $W_{cryo,pump}$ denotes the work input from the air compressor and the cryogenic pump. Also, w_R and w_L represent the net specific work of the recovery and liquefaction section. Assuming full discharge of the system during recovery, the total mass of liquid air pumped from the storage tank must be equal to the total mass of liquid air produced during liquefaction,

$$m_R = m_{liquid} = Y m_L \tag{2.2}$$

This introduces the liquid air yield, Y, which is the ratio between the liquefied air mass m_{liquid} and the mass of air compressed m_L. The liquid air yield is a key parameter representing the performance of any system involving air liquefaction. From the equation above, the liquid air yield also corresponds to the ratio between the total mass pumped into the recovery section and the amount compressed in the charging cycle

$$Y = \frac{m_R}{m_L} \tag{2.3}$$

This expression makes it possible to rewrite equation 2.1 for the round-trip efficiency in terms of the liquid air yield,

$$\eta_{RT} = Y \frac{w_R}{w_L} \tag{2.4}$$

2.2 General Thermodynamics

The thermodynamic definitions and equation introduced in this section comes from Çengel and Boles (2015).

2.2.1 Conservation of mass

For a steady-flow process, the total amount of mass within the control volume does not change with time. This means that the conservation of mass principle requires that the total mass entering the control volume must be equal to the total mass leaving. And for a steady-flow process, what is interesting is the amount of mass flowing per unit time. The conservation of mass principle for a general steady-flow system with multiple inlets and outlets is expressed in rate form as

$$\sum_{in} \dot{m} = \sum_{out} \dot{m} \tag{2.5}$$

2.2.2 The energy equation

The conservation of energy principle can be expressed as the net change of the total energy of a system during a process is equal to the total energy entering and the total energy leaving during that process. This can be expressed as the energy balance

$$E_{in} - E_{out} = \Delta E_{system} \tag{2.6}$$

And considering a steady-flow process involving a control volume the energy equation can be expressed in rate form as

$$(\dot{Q}_{in} - \dot{Q}_{out}) + (\dot{W}_{in} - \dot{W}_{out}) = \sum_{in} \dot{m}h - \sum_{out} \dot{m}h \tag{2.7}$$

2.2.3 Isentropic efficiencies

The performance of a turbine and a pump/compressor can be expressed in terms of the isentropic efficiency. The isentropic efficiency for a turbine is defined as the ratio between actual turbine work to the isentropic turbine work, or in expressed in enthalpies as

$$n_T = \frac{h_1 - h_2}{h_1 - h_{2s}} \tag{2.8}$$

where h_1 is the enthalpy at the entrance for the actual process while h_{2s} and h_2 are enthalpies at the exit for an isentropic and actual process, respectively.

The isentropic efficiency for a compressor or pump is defined as the ratio between the isentropic compressor/pump work to the actual compressor work. Expressed in terms of enthalpies the isentropic efficiency is

$$n_C = \frac{h_{2s} - h_1}{h_2 - h_1} \tag{2.9}$$

where h_1 is the enthalpy at the entrance for the actual process while h_{2s} and h_2 are enthalpies at the exit for an isentropic and actual process, respectively.

2.2.4 Gibbs Phase rule

In general, the number of independent variables of a multicomponent, multiphase system is given by the **Gibbs phase rule**, expressed as

$$IV = C - PH + 2 \tag{2.10}$$

Where IV equals the number of independent variables, C the number of components, and PH the number of phases present in equilibrium.

Single-component system

A pure substance ($C = 1$) in a single-phase condition ($PH = 1$) requires two independent intensive properties, such as pressure or temperature, to be specified to fix the equilibrium state. Based on the same rule, a pure component system ($C = 1$) in a double phase ($PH = 2$) requires the value of one independent intensive property to be specified. When the combination of temperature and pressure reaches a point where the system enters the two-phase region and separation into two phases ($P = 2$) starts, the number of independent variables decreases from 2 to 1. It will no longer be possible to control temperature and pressure independently.

Two-component system

A binary mixture consisting of two components ($C = 2$) in a double phase condition ($P = 2$) will require two independent intensive properties to be fixed to be in equilibrium. Besides temperature and pressure, the composition of each phase constitutes the other degree of freedom. The reason is that the two phases of a two-component system do not have the same composition in each phase. This is illustrated in Figure 2.2 for the two-phase mixture of oxygen and nitrogen at a pressure of 0.1 MPa. The vapor line on this diagram represents the equilibrium composition of the vapor phase at various temperatures, and the liquid line does the same for the liquid phase. The point on the right, were the liquid and vapor lines intercept, represents the boiling temperature for pure oxygen, and the left point shows the boiling point for pure nitrogen.

For the given values of temperature and pressure, two phases will be at equilibrium when the overall composition lies within the area of the two lines. A horizontal isotherm line at this temperature would intersect the vapor and liquid line corresponding to a certain composition of mole fractions on the x-axis for each phase. As an example, at a temperature of 82 K the mole fractions in the liquid phase are 55% oxygen and 45% nitrogen. Looking at the vapor phase at the same temperature the mole fractions of oxygen and nitrogen are about 20% and 80%, respectively.

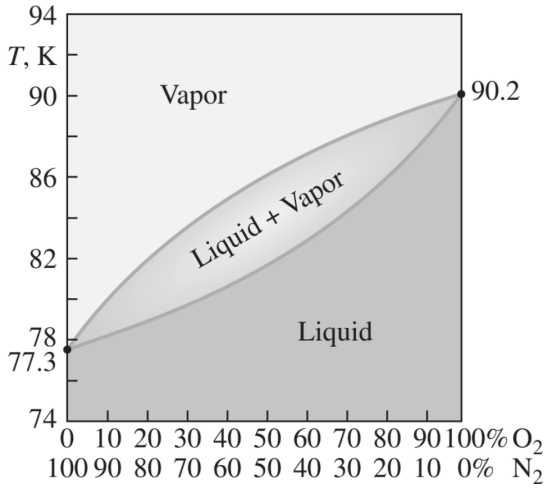

Figure 2.2: Equilibrium diagram for the two-phase mixture of oxygen and nitrogen at 0.1 MPa (Çengel and Boles 2015, p. 822).

2.2.5 Heat transfer

Heat can be transferred through conduction, convection, and radiation. However, for this work only conduction is considered relevant for the calculations of heat losses in the liquid air tanks. Therefore only conduction will be given a brief description.

Conduction

Conduction is the transfer of energy through collisions of particles where the more energetic particles of a substance transfer energy to less energetic ones. In gases and liquids, conduction occurs due to molecules colliding as a result of their random motion. In solids, the transfer occurs as a result of vibrations of molecules and transport of free electrons.

The rate of heat conduction $\dot{Q}_{cond}$ through a layer with the constant thickness Δx is proportional to the temperature difference ΔT across the layer and the area A normal to the direction of the heat transfer and is inversely proportional to the thickness of the layer. Therefore, the rate of heat conduction can be expressed as

$$\dot{Q}_{cond} = k_t A \frac{\Delta T}{\Delta x} (W) \tag{2.11}$$

where the k_t is the thermal conductivity of the material, which represents the materials' ability to conduct heat (Çengel and Boles 2015, p. 91).

2.3 IPSEpro and Newton-Raphson

IPSEpro is a software system developed by SimTech Simulation Technology for heat balance calculations and process simulation. The software allows for the graphical design of models consisting of networks of components. Together, these components are represented by systems of equations, and to solve these, they are put in a matrix. For time effectiveness, the software divides the matrix into sub-matrices to separate non-depending variables. The equations of every sub matrice are then solved simultaneously with the Newton-Raphson method.

2.3.1 IPSEpro

The IPSEpro software is consisting of several Program Modules where the main modules are the Model Development Kit (MDK) and the Process Simulation Environment (PSE). In MDK, the characteristics of the components are specified through adding and modifying equations and variables. This can be done by changing existing components or creating new ones and save them in library files. With the component library ready, a process model is set up in PSE, where the components can be arranged and connected in the desired order.

IPSEpro solver parameters

In order to be able to perform a full calculation of the model process, the total number of variables in relation to the number of equations have to match. If this is fulfilled, the software will then proceed to solve the system of equations. But besides fulfilling the requirement of having an equal number of variables and equations, other errors can occur due to, for example, insufficiently chosen solver parameters. This can result in the error where a group of variables does not converge within the maximum number of iteration steps. The software offers the user the possibility of adjusting certain solver parameters. If the calculations do not converge, this could be due to a lack of the number of iteration steps. It can also be solved by decreasing the level of accuracy, which can be set to a certain number of decimals.

2.3.2 Newton-Raphson

The IPSEpro software uses the Newton-Raphson method to solve multiple equations simultaneously. The method functions as a root-finding algorithm which successively gives better approximations of the root to a differentiable function $f(x)$. With an initial estimate $x = x_0$ which is reasonably close to to the actual root, a better approximation will be given by

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{2.12}$$

If expressed in the general form, the next approximation for any x_n is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.13}$$

This can be repeated until the desired accuracy for the solution is obtained.

The method can be described further with the help of the notations used in Figure 2.3. The tangent line $g(x)$ represents the slope $f'(x)$ which goes through the point $(x_n, f(x_n))$. The equation for this tangent line will therefore be given by

$$g(x) = f'(x_n)(x - x_n) + f(x_n) \tag{2.14}$$

By setting $g(x) = 0$ and $x = x_{n+1}$ an equation will obtained where the solution, also expressed by Equation 2.13, gives the new approximation.

Although it is a time-effective method, it lacks reliability as it is dependent on the starting estimate to be close to the actual root. With a poorly chosen initial estimate, there is a risk that calculations end up with the wrong or no roots.

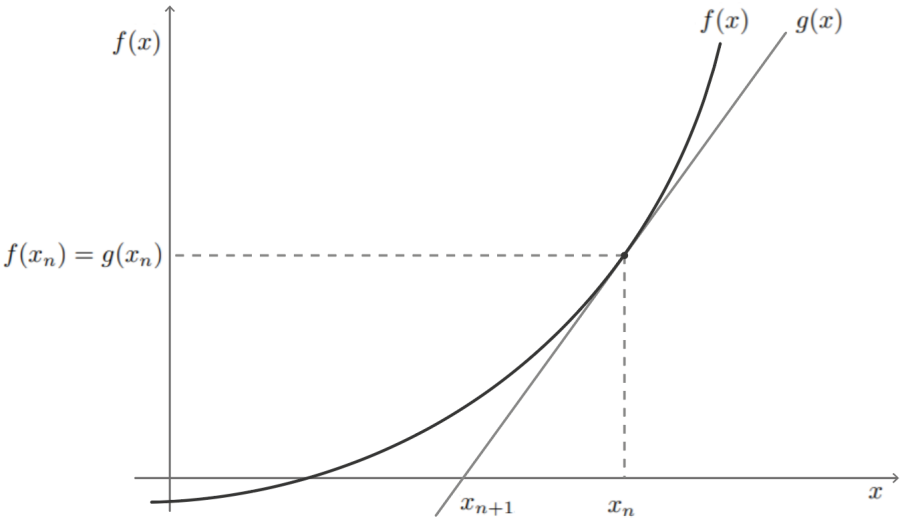

Figure 2.3: Illustration of the Newton-Raphson method

2.4 Refprop

Refprop is a program developed by the National Institute of Standards and Technology (NIST) which provides thermodynamic properties of a wide range of pure fluids and mixtures. The properties of these fluids and mixtures can be displayed through Tables and Plots in the graphical user interface. They can also be accessed, as done in this work, through user-written applications accessing the Refprop DLL.

Chapter 3

Methodology

The general approach to fulfill the purpose of this thesis was to start off by conducting a literature study to map previous work on the LAES and gain knowledge about the system. Then, with the use of suitable source data, create a model of the stand-alone liquid air energy storage (LAES). The model was constructed to give system performance indices and to enable the evaluation of how the system was able to handle an off-design scenario with boil-off in the liquid air tank.

3.1 Literature study

The literature study was conducted with the purpose to gain knowledge on previous research and to find suitable work to use as a base for the design of the model in IPSEpro. The work by Guizzi et al. (2015) on the stand-alone LAES was considered an appropriate source to use as a reference for the plant layout with associated data.

3.2 Selection of software

The software used during the work process was selected based on what programs were used at the SIT's department of research and development. Therefore, IPSEpro was selected for creating the model. Refprop was used to obtain necessary fluid data to perform thermodynamic calculations. The development and adjustment of a dynamic link library (DLL) implementing the subroutines in the Refprop DLL and allowing IPSEpro to communicate with the DLL was done in Microsoft Visual Studio 2017, with the programming language C++. The choice was made as a DLL with the purpose to make IPSEpro communicate with Refprop was already available at the Department of Energy Sciences at LTH. The fundamentals of the DLL follow the structure described by Mondejar (2015).

3.3 Development of the DLL

The DLL enables the software to use external functions or information that is not part of the program's executable code. Therefore, a DLL could be used to create a link between IPSEpro and the Refprop DLL to make external functions in IPSEpro access the fluid properties from the subroutines in the Refprop DLL. With the

already existing DLL, adjustments and addition of code were made to make it fit the problem.

3.3.1 Refprop Subroutines

The DLL was coded to be able to receive data from IPSEpro, which was needed to call the subroutines in the Refprop DLL. The fluid data that was obtained was then returned back to IPSEpro. The required input parameters varied depending on the routine. Still, a common shared input for the majority of the subroutines consisted of an array of the mole fractions of each fluid in the mixture. Listed below are the subroutines that were used in the DLL. The first three flash calculations were used to obtain the values of parameters as, for example, density or entropy. However, while inside the two-phase region, the vapor quality and compositions of the liquid and vapor phase returned from the flash calculations were on a mole basis. To solve this, the last subroutine listed below was used for conversion to a mass basis.

- **TPFLSHDdll**: Required temperature (K) and pressure (kPa) as input parameters.
- **PHFLSHdll:** Required enthalpy (J/mol) and pressure (kPa) as input parameters.
- **PQFLSHdll:** Required pressure (kPa) and vapor quality (mol/mol) as input parameters.
- **QMASSdll:** Required vapor quality and arrays of the liquid and vapor compositions as input parameters.

More details of all in and out parameters of these subroutines, which are also available in the Refprop DLL documentation, can be found in Appendix A.

3.3.2 Mass conversion

The parameters in IPSEpro were on a mass basis, while the functions in the Refprop DLL required the input to be on a mole basis. So, to keep the equations in IPSEpro relatively simple, a local function was written in the DLL that handled the conversion of the fractions from a mass basis to a mole basis before they were used to call the subroutines.

3.3.3 External functions in IPSEpro

The external function calls mentioned in previous sections were created in MDK to reach the developed DLL. These external functions created in IPSEpro consist of one main function and one or two derivatives for every value returned by the DLL. The number of derivatives depends on the number of required input parameters to the subroutines in the Refprop DLL. These derivatives were numerically calculated using the five-point stencil where, by finding four adjacent values to the input argument, the first derivative of a function $f(x)$ at a point x can be approximated according to

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{3.1}$$

3.3.4 Description of DLL functions

The main part of the DLL consisted of the implementation of the external functions made in IPSEpro. An example of one of the functions is presented in Appendix B. A more general description of the code for a function is as follows, where an external function can be separated into the main function and its derivatives.

- **Main function:** Input data is received from IPSEpro in the form of one or two main parameter values together with the mass fractions of the mixture.
  - The input data is used to call a local function, f_mole_frac which converts the mass fractions into mole fractions with the help of the molar mass for each fluid retrieved by a function f_wm. Returned is a matrix containing an array of mole fractions, an array of fluid IDs, and the number of fluids.
  - The data returned is used to call f_wm, which returns the molar mass of the mixture as well as creating an array of strings specifying which fluids the mixture contains. The molar mass for the mixture is needed to be able to convert parameters such as enthalpy from mass to mole basis.
  - A setup for the Refprop subroutine is done where, for instance, what and how many fluids the mixture contains is specified.
  - The subroutine is called with the main parameter values and an array of the mole fractions as input.
  - The sought output parameter value from the subroutine is retrieved and returned to IPSEpro.
- **Derivatives of the main function:** Depending on the number of main input parameter values, one ore two derivative functions are made with respect to each parameter.
  - These functions follow the same structure of the main function until the subroutine is called.
  - By finding four adjacent values to the argument value, the derivative can be calculated by Equation 3.1, where a call to the subroutine is made for each adjacent value.
  - The value of the derivative is calculated and returned to IPSEpro.

3.4 Creating the LAES model

The construction of the model in IPSEpro was done gradually as every component available in the standard library had to be adjusted to fit in the process.

3.4.1 Thermal oil

No suitable fluid for the hot thermal storage could be acquired from the available fluids in Refprop. Instead, the heat transfer fluid Dowtherm G was used with data available from Dow Chemical Company (1997). Separate calculations were made for this fluid from the available data. To perform calculations with Dowtherm G the

varying specific heat and densities had to be taken into consideration. Functions for the temperature, enthalpy, and specific volume had to be made. Both the specific heat and density of the fluid changes linearly with temperature, and an expression for both could be fitted from the table data in the form of a linear function

$$f(x) = kx + m \tag{3.2}$$

The enthalpy was determined through a function fitted from the table data. How the enthalpy varies with temperature could thereby be expressed by a polynomial function

$$f(x) = Ax + Bx^2 \tag{3.3}$$

3.4.2 Components

Most of the constituting components of the process were developed from the software's standard library of power plant components called advanced power plant library or applib. Components needed that was missing in the standard library had to be created. The most critical component of the model was the separator, which also turned out to be the most challenging component to design.

The separator

The main component of the process model is the separator. The function of the separator is to divide the flow into two separate flows containing 100% saturated liquid in one, and 100% saturated gas in the other. When implementing the separator, it had to be taken into consideration that the feed is consisting of more than one component in the two-phase region and, as mentioned in the theory section for **Gibbs phase rule**, the two phases do not consist of the same composition. This means that the mole fractions of each component are different depending on the phase. There must, therefore, also be mass balances for each fluid handling the different fractions in each phase. The main equations constituting the separator were

- An overall mass balance specifying that the inlet mass flow is equal to the total mass flows of the outlets.
- A fluid mass balance to ensure that the total mass of each fluid in the mixture remains constant.
- An energy balance

Other conditions were needed to make the separator work. By utilizing that the two drains will be either 100% liquid or 100% vapor, the mass of the drains will be equal to the mass of liquid and vapor in the feed and thereby also the condition that the liquid drain has the enthalpy of saturated liquid. Also, as the feed's liquid mass will be equal to the total mass of the liquid drain, the liquid mass fractions of the feed will constitute the overall composition of the liquid drain. Correspondingly, the same applies to the other drain but for the vapor fractions.

3.5 Validation of the model

To validate the model, a comparison was made between the stream data and round-trip efficiencies of the model and the reference. This in order to evaluate if the deviations of the model were reasonable. The deviations between model and reference data are expressed in terms of percentage error to the reference data.

$$\Delta x_{error}\% = \frac{x_{ref} - x_{model}}{x_{ref}} \tag{3.4}$$

3.6 Test case

A test was performed to see how the system would handle an off-design scenario with boil-off in the liquid air tank. The impact on the system was evaluated by looking at how the round-trip efficiency of the system would be affected of the system being held on stand-by. In addition, it was also looked into how the system compensated for the reduction of available liquid air. What is important to take into consideration is that only the losses during the stand-by duration was included.

3.6.1 Dimensioning

A requirement for being able to evaluate the impact of boil-off in the storage tank was to be able to perform dimensioning of the plant for a certain capacity and thereby get the required total mass required as well as the volume of the storage tanks. To do so, the two sections had to be coupled together. This was done with the help of so called globals, which enables a link to be created between certain components by being able to let them share specific parameters. The equation that was used to make the sections dependent on each other was given by Equation 2.3, where the liquefaction and recovery durations are not equal. This gave the expression

$$Y = \frac{m_R}{m_L} = \frac{t_R \dot{m}_R}{t_L \dot{m}_L} \tag{3.5}$$

which made sure that the total mass of liquid air from the liquefaction section would be equal to the total mass used during recovery. Similar relations in the form of mass balances without the liquid air yield Y were set to ensure that the total mass of cold fluids and thermal oil would be used during liquefaction and recovery.

3.6.2 Tank Geometry and insulation

To be able to calculate the heat transfer between the tank and the environment, the geometry and dimensions of the tank were needed. The assumed geometry of the tank was a cylinder with a hemispherical top and bottom. This gave the expression for the volume of the tank as

$$V = \pi r^2 h + \frac{4}{3}\pi r^3 \tag{3.6}$$

and the area expressed as

$$A = 2\pi r h + 4\pi r^2 \tag{3.7}$$

With an expression for the tank volume, the required volume to store the total amount of liquid air had to be determined to be able to calculate the heat transfer. This required volume was given by

$$V = \frac{\dot{m}t}{\rho} \tag{3.8}$$

where $\dot{m}$ is the mass flow of the feed to the tank, t is the charging/discharging duration and ρ the density of the fluid.

The insulation material used for the tanks in this scenario was perlite powder, which is also used in the LNG tank described by Yang et al. (2006). The Perlite powder has a thermal conductivity of 0.040 W/(mK).

3.6.3 Impact of boil-off

The heat is transferred by means of conduction, described in the theory section. With the introduction of heat losses in the liquid air tank, the boil-off will result in a loss of mass (Assuming that the vaporised air is released to the environment) according to

$$m_{vap} = \frac{Q_{loss}}{\Delta h_{vap}} \tag{3.9}$$

where Q_{loss} is the total heat transfer and h_{vap} the enthalpy of vaporisation. The effectiveness of the storage tank can be expressed as

$$\eta_{mass} = \frac{m_{tank} - m_{vap}}{m_{tank}} \tag{3.10}$$

And together with Equation 3.5 the new relation between the two sections for a system with losses in the air tank can be expressed as

$$Y\eta_{mass} = \frac{t_R\dot{m}_R}{t_L\dot{m}_L} = \frac{m_R}{m_L} \tag{3.11}$$

With this expression the round-trip efficiency for the plant including the losses in the liquid air tank is given by

$$\eta_{RT} = \eta_{mass} Y \frac{w_R}{w_L} \tag{3.12}$$

The expected compensation of having less mass available for recovery is that it would be possible to compensate by either increasing the mass flow or duration of the liquefaction process. This would result in an increased work input and reduction of the round-trip efficiency in itself. However, the loss of mass is expected to have other impacts on the system. The available cold energy that is stored during discharge will be limited by the total available mass for discharge, which means that to compensate for this, a higher pressure would be needed for the liquefaction at the expense of a larger amount of required work.

Chapter 4

Data and Results

4.1 Plant Layout

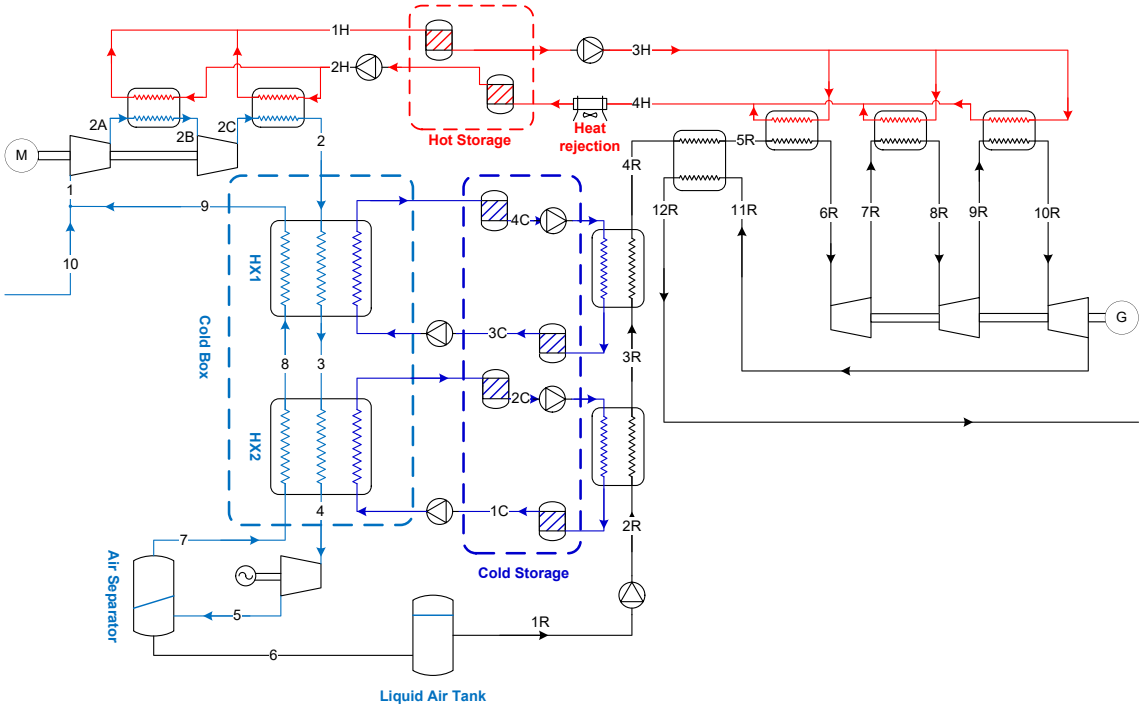

Figure 4.1: Reference plant layout (Guizzi et al. 2015)

The plant layout proposed by Guizzi et al. (2015) is illustrated in 4.1. The left section represents the liquefaction section. Air is first compressed to high pressure with a two-step intercooling where thermal oil is used to recover the heat and store it in a hot thermal storage. After compression, the air is cooled down in the Cold Box by the returning cold air from the separator together with the recovered cold energy stored in the cold storage. The stored cold energy is recovered during discharge and has two separate storages with working fluids being methanol and propane. The reason for this cascaded cold storage is that there is no fluid suitable for the required temperature range. Leaving the cold box, the cool air is then expanded in a cryo-turbine, lowering the temperature further, resulting in a mixture of vapor and liquid that is then separated into a gas stream and liquid stream. The produced liquid air is stored in a tank at close to atmospheric pressure and about 80 Kelvin.

During discharge, the liquid air is pumped from the storage tank by a cryogenic pump and heated up to almost ambient temperature by the cold fluids, which recover the cold energy and store it in the cold storage making available for the liquefaction. The temperature of the pumped air is increased in a regenerator, followed by a superheater, where the thermal oil provides heat from the hot storage. Expansion then takes place, divided into three steps with interheating provided by the thermal oil. The excess heat is rejected to the atmosphere, and thermal oil is led back to the hot storage where it is stored in an ambient-temperature tank.

4.1.1 Source data

Guizzi et al. (2015) performed calculations to find the optimum operating conditions for the set design parameters presented in Table 4.1. Their result showed that for a maximum pressure p_{2R} in the recovery section, an optimum compression ratio p_2/p_1 could be obtained. What was also shown was that an increasing recovery pressure resulted in a higher round-trip efficiency, but as it also increased the optimum pressure ratio p_2/p_1, the pressure p_{2R} was chosen to be the highest possible for a certain limit set on the overall compressor pressure ratio. For the chosen default design parameters, together with a limit on the compressor pressure ratio of approximately 180, the resulting maximum recovery pressure was $p_{2R} = 6.5$ MPa.

Table 4.1: Default design parameters

Parameter	**Value**	**Units**
Ambient temperature	25	°C
Ambient pressure	100	kPa
Liquid air storage pressure	100	kPa
Propane minimum temperature (T_{1C})	93	K
Propane maximum temperature (T_{2C})	214	K
Methanol minimum temperature (T_{3C})	214	K
Methanol maximum temperature (T_{4C})	288	K
Cold box HX pinch-point ΔT	5	K
Intercoolers pinch-point ΔT	10	K
Hot-end temperature approach at super heaters	10	K
Heat exchangers relative pressure loss	1%	
Isentropic efficiency of air turbines	85%	
Isentropic efficiency of air compressors	85%	
Isentropic efficiency of cryoturbine	70%	
Isentropic efficiency of cryogenic pump	70%	

The intermediate pressure ratios of the compressors and the turbines were selected with the minimization of compressor work and maximization of the individual turbine work output in mind. And thereby achieving the highest efficiency for the overall pressure ratios in the liquefaction and recovery section. These values together with the rest of the reference stream data of both sections is presented in Table 4.2 and 4.3, respectively. This data was used to create the model.

Table 4.2: Stream data in the liquefaction section for the reference system

	m/m_1	p[MPa]	T[K]	h[kJ/kg]	ρ[kg/m^3]	N_2[%]
1	1.000	0.100	296.24	299.50	1.17	79.5
2A	1.000	1.480	687.74	707.45	7.40	79.5
2B	1.000	1.465	308.15	308.73	16.48	79.5
2C	1.000	18.098	682.00	705.20	85.32	79.5
2	1.000	17.917	308.15	281.71	194.96	79.5
3	1.000	17.738	245.80	198.70	261.28	79.5
4	1.000	17.561	98.00	-77.38	825.80	79.5
5	1.000	0.102	78.91	-93.87	28.17	79.5
6	0.842	0.102	78.91	-126.21	871.26	77.0
7	0.158	0.102	78.91	78.16	4.58	93.0
8	0.158	0.101	237.80	244.26	1.45	93.0
9	0.158	0.100	286.28	294.33	1.19	93.0
10	0.842	0.100	298.15	300.47	1.16	77.0

Table 4.3: Stream data in the recovery section for the reference system

	m/m_1	p[MPa]	T[K]	h[kJ/kg]	ρ[kg/m^3]	N_2[%]
1R	0.842	0.100	78.74	-126.56	872.08	77.0
2R	0.842	6.500	81.89	-116.13	873.20	77.0
3R	0.842	6.435	209.00	180.44	120.39	77.0
4R	0.842	6.371	283.00	269.84	79.56	77.0
5R	0.842	6.307	436.27	436.35	49.13	77.0
6R	0.842	6.244	616.42	628.96	34.27	77.0
7R	0.842	1.590	450.55	454.68	12.18	77.0
8R	0.842	1.574	616.42	628.96	8.80	77.0
9R	0.842	0.401	451.23	456.27	3.08	77.0
10R	0.842	0.397	616.42	629.01	2.23	77.0
11R	0.842	0.101	451.42	456.69	0.78	77.0
12R	0.842	0.100	288.00	290.19	1.20	77.0

4.2 Validation of Model data

The model was validated through a comparison of stream data and round-trip efficiency η_{RT} with the reference system.

4.2.1 Stream data

The deviations from the reference data for the liquefaction an recovery section can be seen in Table 4.5 and 4.4. When calculating the deviations, the same level of

accuracy was used for the model data as for the values obtained from the reference. The comparison was also made with the liquefaction and recovery duration being set to equal to be able to compare fractional mass flows.

As can be seen in the tables, the deviations of the stream data stays below 0.7% with an exception for the pressure in stream 1R. The reason for this is that the temperature in stream 1R at a pressure of $p_{1R} = 0.1$ MPa does not correspond to the listed density. This could be due to a lack of accuracy with only two decimals, as the saturation temperature is a bit lower than 78.74 K for the given pressure of 0.1 MPa. These values would instead result in a density of $522 kg/m^3$. The temperature was set to the same temperature as in stream 6 with an assumed maintained pressure of 0.102 MPa.

Table 4.4: Deviations between model and reference stream data for the liquefaction section

	$\Delta \frac{\mathbf{m}}{\mathbf{m_1}}_{\mathbf{error}}$[%]	$\mathbf{\Delta P_{error}}$[%]	$\mathbf{\Delta T_{error}}$[%]	$\mathbf{\Delta \rho_{error}}$[%]	$\mathbf{\Delta N_{2error}}$[%]
1	0	0	0	0	0
2A	0	0	0	0,1351	0
2B	0	0	0	0	0
2C	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0,0569	0	-0,0036	0
5	0	0	0	0,0355	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0,6897	0
9	0	0	0	0	0
10	0	0	0	0	0

Presented in Table 4.6 and 4.7 are the stream data for the cold fluids and thermal oil for the reference and model. Note that a different reference for the enthalpy has been used for propane in the model. The different values of the thermal oil are due to the use Dowtherm G instead of Essotherm 650. Another reason why the model data differs from the reference is that the oil in the reference has been cooled by the ambient to a temperature of 15°C which is lower than the ambient temperature of 25°C. This has been taken into consideration in the model which could be the reason for the deviations.

Table 4.5: Deviations between model and reference stream data for the recovery section

	$\Delta \frac{m}{m_1}_{error}$[%]	ΔP_{error}[%]	ΔT_{error}[%]	$\Delta \rho_{error}$[%]	ΔN_{2error}[%]
1R	0	-2	-0,2159	0,0883	0
2R	0	0	-0,2198	0,0859	0
3R	0	0	-0,2010	0,3156	0
4R	0	0	-0,1095	0,1383	0
5R	0	0	0	0	0
6R	0	0	0	0	0
7R	0	0	-0,0022	0	0
8R	0	0	0	0	0
9R	0	0	-0,0111	0	0
10R	0	0	0	0	0
11R	0	0	0,0177	0	0
12R	0	0	-0,0903	0	0

Table 4.6: Stream data (cold fluids and thermal oil) for the reference

	$\frac{\dot{m}}{\dot{m}_1}$	T[K]	h[kJ/kg]	Fluid
1C	1.019	93.00	-182.18	propane
2C	1.019	214.00	62.72	propane
3C	0.437	214.00	-303.14	methanol
4C	0.437	288.00	-130.93	methanol
1H	0.999	626.42	849.94	Essotherm 650
2H	0.999	288.15	26.95	Essotherm 650
3H	0.999	626.42	849.94	Essotherm 650
4H	0.999	460.71	395.31	Essotherm 650

Table 4.7: Stream data (cold fluids and thermal oil) for the model

	$\frac{\dot{m}}{\dot{m}_1}$	T[K]	h[kJ/kg]	Fluid
1C	1,020	93	-282.54	propane
2C	1,020	214	-37.64	propane
3C	0,436	214	-303.14	methanol
4C	0,436	288	-130.93	methanol
1H	1,172	626,42	739.60	Dowtherm G
2H	1,172	298,15	37.99	Dowtherm G
3H	1,172	626,42	739.60	Dowtherm G
4H	1,172	467,13	352.08	Dowtherm G

4.2.2 Round-trip efficiency

The difference in the performance of the model and the reference is determined through a comparison of the round-trip efficiencies. Presented in Table 4.8 are the round trip efficiency for the reference $\eta_{RT_{ref}}$ and the model η_{RT} are together with the deviation.

Table 4.8: Round-trip efficiencies for the reference and the model

$\eta_{RT_{ref}}$	54.4%
η_{RT}	53.2%
Δ_{error}	2.21%

This difference in round-trip efficiency is due to the use of a mechanical efficiency of $n_m = 0.99$ used in turbines, compressors and pumps. Without it the same value for the round-trip efficiency was obtained.

4.3 Test case

4.3.1 Set-up

The purpose of the test was to test the model's capability to handle a simple off-design scenario caused by boil-off in the liquid storage tank. To see how the system compensated for the boil-off in the liquid air tank, a simulation was performed without losses for a plant with the capacity and rated power as presented in Table 4.9. This way, the values of parameters dependent on component geometry could be obtained and set. The mass flows in the liquefaction and recovery section were set for which the compressors and turbine were assumed to be designed. Next, the areas of the heat exchangers were fixed to let the pinch points depend on the operating conditions. The available cold energy was the limiting factor, which was determined from the given capacity and total mass of the recovery section. Instead, what was expected to make up for the lost mass was an increased pressure p_2, and therefore, the pressure ratio of the second compressor was allowed to increase to compensate for the reduced cooling available. As a consequence of the increase of pressure, and thereby required intercooling, the mass flow of thermal oil was allowed to increase. The reason why this could be done is that an increase in an already excess amount of hot oil would not affect the recovery section.

For this scenario, the plant's energy storage capacity was desired to be kept constant, meaning that it was assumed that the volume of the liquid air storage tank was not a limiting factor. This tank volume was determined by taking the required volume of air given by the simulation without losses and determine a size large enough for the expected increase. The specifications of the tank are presented in Table 4.10.

Table 4.9: Plant dimensions

Parameter	Value	Units
Capacity	330	MWh
Charging power	50	MW
Discharging power	50	MW

Table 4.10: Specifications of the liquid air tank

Parameter	Value	Units
Volume	2800	m^3
Height	15	m
Diameter	12.4	m
Insulation material	Perlite Powder	
Insulation layer thickness	0.635	m
Insulation thermal conductivity	0.040	W/mK

4.3.2 Outcome

Impact on performance

Fig 4.8 shows a diagram of how the round trip efficiency η_{RT} changes with the standby duration. It shows that the round-trip efficiency will be reduced down below 50% after a standby duration of 240 hours, or an equal of 10 days.

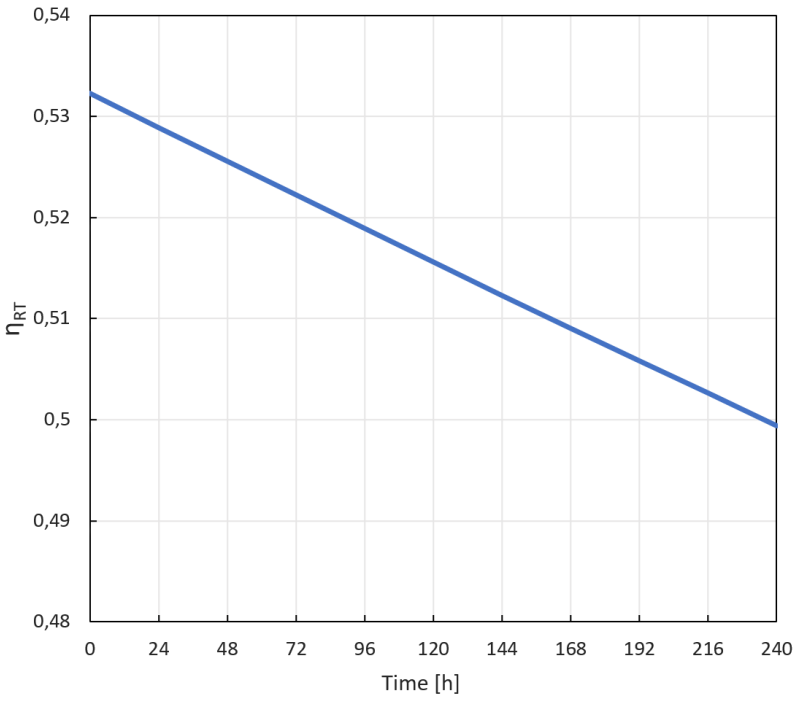

Figure 4.2: Diagram for how the round-trip efficiency η_{RT} changes with the stand-by duration

Impacts on the system

Presented in Figure 4.3 is a T-s diagram of the reference liquefaction section. The corresponding linear T-s diagram of the model is presented in Figure 4.4 which, represents the system's liquefaction section without losses and the re-calibrated system for a 10 day standby duration.

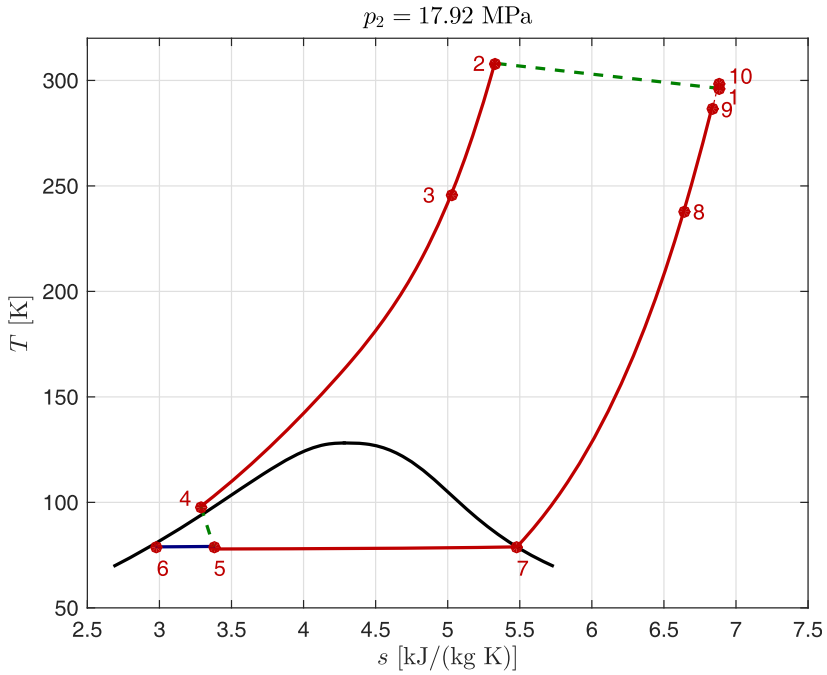

Figure 4.3: T-s diagram over the reference liquefaction section (Guizzi et al. 2015)

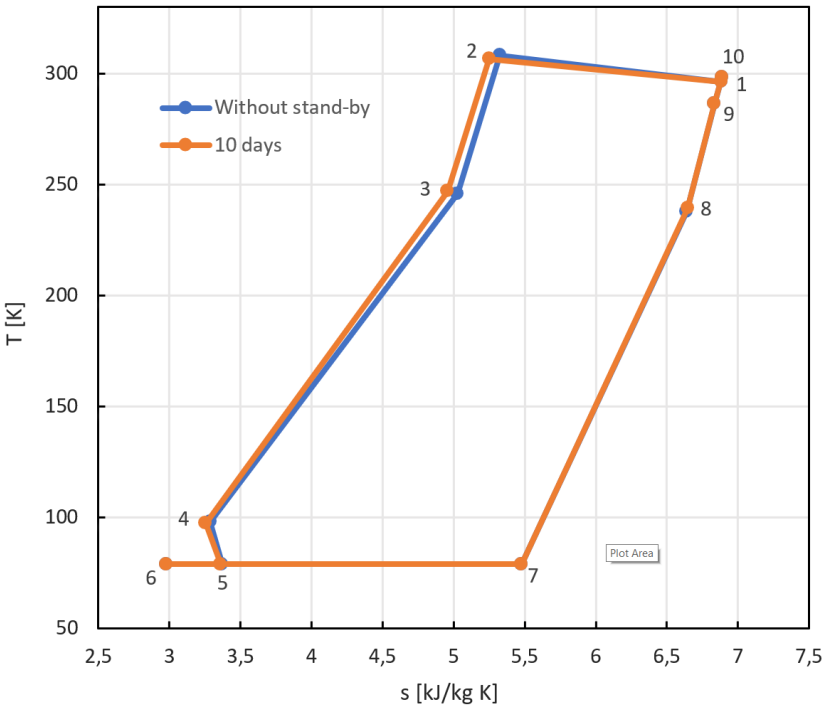

Figure 4.4: Linear T-s diagram for the model with instant discharge and a stand-by duration of 10 days

As can be seen in Figure 4.4, the placement of points 2, 3, and 4 are moved to the left, which corresponds to an increase in pressure as p_2 rises from 17.92 MPa to 21.62 MPa. The result of this pressure increase is that point 5 will also move to the left, resulting in a lower vapor quality and thereby a higher liquid air yield Y, meaning that a larger amount of liquid air will be obtained to compensate for the amount lost due to boil-off. The higher liquid air yield comes at the expense of an increased amount of work required, which results in the reduction of the round-trip efficiency seen in the diagram in Figure 4.8.

With the set parameters, the increased liquid air yield will also result in an increased charge duration. The explanation behind this is that through Equation 3.12, with the mass flows of each section specified along with the discharge duration and boil-off in the tank, the increase in liquid air yield shows that the duration will also have to increase in order to maintain the same capacity and discharge power.

Chapter 5

Discussion

5.1 Validation of data

The deviations from reference stream data seem to be below a reasonable error of 0.7%, and the round-trip efficiency without the use of mechanical losses matched the reference value. This confirms that the implementation of Refprop gas properties in IPSEpro has been successful, and the gas separation unit developed is working properly.

5.2 Test case

The off-design scenario demonstrated the model's capability to handle boil-off in the liquid air tank. The overall impact of the loss of mass resulted in a significant reduction of round-trip efficiency. The way the system compensated for the lack of available cold energy as a result of the loss of liquid air is considered reasonable for the parameters set. Although, setting only the pressure ratio of the second compressor as undefined should probably have been replaced by leaving both pressure ratios undetermined and instead set a relationship between the two intermediate pressure ratios. This way, by distributing the extra required work input on both compressors could probably lower the total extra work needed.

Another thing to have in mind is that the loss of mass was only calculated during the standby duration and not during charging. Setting this duration in relation to the total standby time required to lower the round-trip efficiency down to 50%, the decrease in time needed to reach this value would be significantly lower.

The heat transfer in the tanks will lead to certain mass being boiled off. To perform correct calculations of this, the composition of nitrogen and oxygen would change as a result. However, in this work the losses were considered small enough to assume a maintained composition. This should be taken into account when calculating losses in the future.

Chapter 6

Conclusions

6.1 Model and test case

Looking at the deviations of the model data, the differences of the stream data were relatively small. A round-trip efficiency without the use of mechanical losses would be equal to the one obtained in the reference. The overall impact of the boil-off in the liquid air tank resulted in an expected reduction of round-trip efficiency η_{RT}, and the system's compensation for the loss in the liquid air tank are considered reasonable. It can therefore be concluded that the model could be useful for further evaluation of the energy storage system, but depending on the scenarios of interest, more development of the model may be necessary.

6.2 Future work

- As far as the REFPROP DLL subroutines are concerned, it will be necessary to eventually switch to the High-level API with the functions included. The reason is that the subroutines used will be removed and replaced. However, the advantage of using these would reduce the number of subroutines needed as those included are considered more flexible.
- In this work, air has been assumed to be consisting of only nitrogen and oxygen. It could be of interest to enable the model to include more fluids in the composition.
- The work producing and consuming components are only expressed in terms of isentropic efficiencies, and more precise models of these components should be used to increase the accuracy of the model.
- It could be of interest to introduce losses in all tanks as well as look into more realistic pressure drops in heat exchangers and other components. Also, including losses in the pipes could be of interest.

Appendix A

REFPROP DLL subroutines

A.1 PHFLSHDLL

subroutine PHFLSHdll (*P*, *h*, *z*, *T*, *D*, *Dl*, *Dv*, *x*, *y*, *q*, *e*, *s*, *Cv*, *Cp*, *w*, *ierr*, *herr*, *herr_length*)

Flash calculation given pressure, bulk enthalpy, and bulk composition. (See subroutines ABFLSH or PBFLSH for the description of all variables.)

Parameters

- **P** *[double,in]* :: Pressure [kPa]
- **h** *[double,in]* :: Enthalpy [J/mol]
- **z** (20) *[double,in]* :: Bulk Composition (array of mole fractions)
- **T** *[double,out]* :: Temperature [K]
- **D** *[double,out]* :: Density [mol/K]
- **Dl** *[double,out]* :: Molar density of the liquid phase [mol/L]
- **Dv** *[double,out]* :: Molar density of the vapor phase [mol/L]
- **x** (20) *[double,out]* :: Composition of the liquid phase (array of mole fractions)
- **y** (20) *[double,out]* :: Composition of the vapor phase (array of mole fractions)
- **q** *[double,out]* :: Vapor quality [mol/mol]
- **e** *[double,out]* :: Internal energy [J/mol]
- **s** *[double,out]* :: Entropy [J/mol-K]
- **Cv** *[double,out]* :: Isochoric heat capacity [J/mol-K]
- **Cp** *[double,out]* :: Isobaric heat capacity [J/mol-K]
- **w** *[double,out]* :: Speed of sound [m/s]
- **ierr** *[int,out]* :: Error code (no error if ierr==0)
- **herr** *[char,out]* :: Error string (character*255)
- **herr_length** *[int]* :: length of variable `herr` (default: 255)

A.2 PQFLSHDLL

subroutine PHFLSHdll (*P*, *h*, *z*, *T*, *D*, *Dl*, *Dv*, *x*, *y*, *q*, *e*, *s*, *Cv*, *Cp*, *w*, *ierr*, *herr*, *herr_length*)

Flash calculation given pressure, bulk enthalpy, and bulk composition. (See subroutines ABFLSH or PBFLSH for the description of all variables.)

Parameters

- **P** *[double,in]* :: Pressure [kPa]
- **h** *[double,in]* :: Enthalpy [J/mol]
- **z** (20) *[double,in]* :: Bulk Composition (array of mole fractions)
- **T** *[double,out]* :: Temperature [K]
- **D** *[double,out]* :: Density [mol/K]
- **Dl** *[double,out]* :: Molar density of the liquid phase [mol/L]
- **Dv** *[double,out]* :: Molar density of the vapor phase [mol/L]
- **x** (20) *[double,out]* :: Composition of the liquid phase (array of mole fractions)
- **y** (20) *[double,out]* :: Composition of the vapor phase (array of mole fractions)
- **q** *[double,out]* :: Vapor quality [mol/mol]
- **e** *[double,out]* :: Internal energy [J/mol]
- **s** *[double,out]* :: Entropy [J/mol-K]
- **Cv** *[double,out]* :: Isochoric heat capacity [J/mol-K]
- **Cp** *[double,out]* :: Isobaric heat capacity [J/mol-K]
- **w** *[double,out]* :: Speed of sound [m/s]
- **ierr** *[int,out]* :: Error code (no error if ierr==0)
- **herr** *[char,out]* :: Error string (character*255)
- **herr_length** *[int]* :: length of variable `herr` (default: 255)

A.3 TPFLSHDLL

subroutine PHFLSHdll (*P*, *h*, *z*, *T*, *D*, *Dl*, *Dv*, *x*, *y*, *q*, *e*, *s*, *Cv*, *Cp*, *w*, *ierr*, *herr*, *herr_length*)
Flash calculation given pressure, bulk enthalpy, and bulk composition. (See subroutines ABFLSH or PBFLSH for the description of all variables.)

Parameters

- **P** *[double,in]* :: Pressure [kPa]
- **h** *[double,in]* :: Enthalpy [J/mol]
- **z** (20) *[double,in]* :: Bulk Composition (array of mole fractions)
- **T** *[double,out]* :: Temperature [K]
- **D** *[double,out]* :: Density [mol/K]
- **Dl** *[double,out]* :: Molar density of the liquid phase [mol/L]
- **Dv** *[double,out]* :: Molar density of the vapor phase [mol/L]
- **x** (20) *[double,out]* :: Composition of the liquid phase (array of mole fractions)
- **y** (20) *[double,out]* :: Composition of the vapor phase (array of mole fractions)
- **q** *[double,out]* :: Vapor quality [mol/mol]
- **e** *[double,out]* :: Internal energy [J/mol]
- **s** *[double,out]* :: Entropy [J/mol-K]
- **Cv** *[double,out]* :: Isochoric heat capacity [J/mol-K]
- **Cp** *[double,out]* :: Isobaric heat capacity [J/mol-K]
- **w** *[double,out]* :: Speed of sound [m/s]
- **ierr** *[int,out]* :: Error code (no error if ierr==0)
- **herr** *[char,out]* :: Error string (character*255)
- **herr_length** *[int]* :: length of variable `herr` (default: 255)

A.4 QMASSDLL

subroutine PHFLSHdll (*P*, *h*, *z*, *T*, *D*, *Dl*, *Dv*, *x*, *y*, *q*, *e*, *s*, *Cv*, *Cp*, *w*, *ierr*, *herr*, *herr_length*)

Flash calculation given pressure, bulk enthalpy, and bulk composition. (See subroutines ABFLSH or PBFLSH for the description of all variables.)

Parameters

- **P** *[double,in]* :: Pressure [kPa]
- **h** *[double,in]* :: Enthalpy [J/mol]
- **z** (20) *[double,in]* :: Bulk Composition (array of mole fractions)
- **T** *[double,out]* :: Temperature [K]
- **D** *[double,out]* :: Density [mol/K]
- **Dl** *[double,out]* :: Molar density of the liquid phase [mol/L]
- **Dv** *[double,out]* :: Molar density of the vapor phase [mol/L]
- **x** (20) *[double,out]* :: Composition of the liquid phase (array of mole fractions)
- **y** (20) *[double,out]* :: Composition of the vapor phase (array of mole fractions)
- **q** *[double,out]* :: Vapor quality [mol/mol]
- **e** *[double,out]* :: Internal energy [J/mol]
- **s** *[double,out]* :: Entropy [J/mol-K]
- **Cv** *[double,out]* :: Isochoric heat capacity [J/mol-K]
- **Cp** *[double,out]* :: Isobaric heat capacity [J/mol-K]
- **w** *[double,out]* :: Speed of sound [m/s]
- **ierr** *[int,out]* :: Error code (no error if ierr==0)
- **herr** *[char,out]* :: Error string (character*255)
- **herr_length** *[int]* :: length of variable `herr` (default: 255)

Appendix B

DLL function code

```
//#################################################################################################################
//#################################################################################################################
//  f_h_pt
//#################################################################################################################
//#################################################################################################################

double EXTERN_FUNCTION f_h_pt(double P, double T, double fluidID9, double x1, double x2, double x3, double x4,
double x5, double x6, double x7, double x8, double x9)
{
        vector< vector<double>> comp = f_mole_frac(x1, x2, x3, x4, x5, x6, x7, x8, x9, fluidID9);
        double fluids = comp[2][0];
        double* x = new double[fluids];
        double* fluid = new double[fluids];
        for (int n = 0; n < fluids; n++) {
                x[n] = comp[0][n];
                fluid[n] = comp[1][n];
        }
        long i = fluids;
        wm = f_wm(x, fluid, i);

        //// Then get pointers into the dll to the actual functions.
        SETUPdll = (fp_SETUPdllTYPE) GetProcAddress(RefpropdllInstance,"SETUPdll");

        SETREFdll = (fp_SETREFdllTYPE) GetProcAddress(RefpropdllInstance,"SETREFdll");
        //
        TPFLSHdll = (fp_TPFLSHdllTYPE) GetProcAddress(RefpropdllInstance,"TPFLSHdll");
        //
        ////...Call SETUP to initialize the program
        SETUPdll(&i, hf, hfmix, hrf, &ierr,
herr,refpropcharlength*ncmax,refpropcharlength,lengthofreference,errormessagelength);

        SETREFdll(hrf,&ixflag,x,&h0,&s0,&t0,&p0,&ierr,herr,lengthofreference,errormessagelength);

        P=P*100;
        T=T+273.15;

        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);

        delete[] x;
        delete[] fluid;
        return h/wm;
}

double EXTERN_FUNCTION df_h_pt_dp(double P, double T, double fluidID9, double x1, double x2, double x3, double
x4, double x5, double x6, double x7, double x8, double x9 )
{
        vector< vector<double>> comp = f_mole_frac(x1, x2, x3, x4, x5, x6, x7, x8, x9, fluidID9);
        double fluids = comp[2][0];
        double* x = new double[fluids];
        double* fluid = new double[fluids];
        for (int n = 0; n < fluids; n++) {
                x[n] = comp[0][n];
                fluid[n] = comp[1][n];
        }
        long i = fluids;
        wm = f_wm(x, fluid, i);
```

```
        // Then get pointers into the dll to the actual functions.
        SETUPdll = (fp_SETUPdllTYPE) GetProcAddress(RefpropdllInstance,"SETUPdll");SETREFdll =
(fp_SETREFdllTYPE) GetProcAddress(RefpropdllInstance,"SETREFdll");

        TPFLSHdll = (fp_TPFLSHdllTYPE) GetProcAddress(RefpropdllInstance,"TPFLSHdll");

        //...Call SETUP to initialize the program
        SETUPdll(&i, hf, hfmix, hrf, &ierr,
herr,refpropcharlength*ncmax,refpropcharlength,lengthofreference,errormessagelength);
        SETREFdll(hrf,&ixflag,x,&h0,&s0,&t0,&p0,&ierr,herr,lengthofreference,errormessagelength);


        P=P*100;
        T=T+273.15;
        double ep=P*sqrt(DBL_EPSILON);

        P=P+2*ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_1=h/wm;
        P=P-ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_2=h/wm;
        P=P-2*ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_3=h/wm;
        P=P-ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_4=h/wm;

        delete[] x;
        delete[] fluid;
        return (-h_1+8*h_2-8*h_3+h_4)/(12*ep/100);

}

double EXTERN_FUNCTION df_h_pt_dt(double P, double T, double fluidID9, double x1, double x2, double x3, double
x4, double x5, double x6, double x7, double x8, double x9 )
{
        vector< vector<double>> comp = f_mole_frac(x1, x2, x3, x4, x5, x6, x7, x8, x9, fluidID9);
        double fluids = comp[2][0];
        double* x = new double[fluids];
        double* fluid = new double[fluids];
        for (int n = 0; n < fluids; n++) {
                x[n] = comp[0][n];
                fluid[n] = comp[1][n];
        }
        long i = fluids;
        wm = f_wm(x, fluid, i);


        // Then get pointers into the dll to the actual functions.
        SETUPdll = (fp_SETUPdllTYPE) GetProcAddress(RefpropdllInstance,"SETUPdll");SETREFdll =
(fp_SETREFdllTYPE) GetProcAddress(RefpropdllInstance,"SETREFdll");

        TPFLSHdll = (fp_TPFLSHdllTYPE) GetProcAddress(RefpropdllInstance,"TPFLSHdll");

        //...Call SETUP to initialize the program
        SETUPdll(&i, hf, hfmix, hrf, &ierr,
herr,refpropcharlength*ncmax,refpropcharlength,lengthofreference,errormessagelength);
        SETREFdll(hrf,&ixflag,x,&h0,&s0,&t0,&p0,&ierr,herr,lengthofreference,errormessagelength);


        P=P*100;
        T=T+273.15;
        double ep = T * sqrt(DBL_EPSILON);

        T=T+2*ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_1=h/wm;
        T=T-ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_2=h/wm;
        T=T-2*ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_3=h/wm;
        T=T-ep;
        TPFLSHdll(&T,&P,x,&d,&dl,&dv,xliq,xvap,&q,&e,&h,&s,&cv,&cp,&w,&ierr,herr,errormessagelength);
        double h_4=h/wm;

        delete[] x;
        delete[] fluid;
        return (-h_1+8*h_2-8*h_3+h_4)/(12*ep);

}
```

Bibliography

Abdo, Rodrigo F. et al. (2015). *Performance evaluation of various cryogenic energy storage systems*. ID: 271090. DOI: https://doi.org/10.1016/j.energy.2015.08.008. URL: http://www.sciencedirect.com/science/article/pii/S0360544215010567.

Borri, E. et al. (2017). *A preliminary study on the optimal configuration and operating range of a "microgrid scale" air liquefaction plant for Liquid Air Energy Storage*. ID: 271098. DOI: https://doi.org/10.1016/j.enconman.2017.03.079. URL: http://www.sciencedirect.com/science/article/pii/S019689041730290X.

Çengel, Y. A and M. A Boles (2015). *Thermodynamics: An Engineering Approach*. 8th. New York: McGraw-Hill Education.

Dow Chemical Company (1997). *Product Technical Data DOWTHERM G Heat Transfer Fluid*. URL: http://msdssearch.dow.com/PublishedLiteratureDOWCOM/dh_0032/0901b803800325da.pdf?filepath=/heattrans/pdfs/noreg/176-01353.pdf&fromPage=GetDoc.

Frankfurt School-UNEP Centre/BNEF (Sept. 2019). *Global trends in renewable energy investment 2019*. URL: https://www.fs-unep-centre.org/wp-content/uploads/2019/11/GTR_2019.pdf.

Guizzi, Giuseppe Leo et al. (2015). *Thermodynamic analysis of a liquid air energy storage system*. ID: 271090. DOI: https://doi.org/10.1016/j.energy.2015.10.030. URL: http://www.sciencedirect.com/science/article/pii/S0360544215013985.

Highview Power (2020a). *Plants*. en-GB. URL: https://www.highviewpower.com/plants/ (visited on 01/28/2020).

— (2020b). *Technology*. en-GB. URL: https://www.highviewpower.com/technology/ (visited on 01/28/2020).

Hüttermann, Lars et al. (2019). *Investigation of a liquid air energy storage (LAES) system with different cryogenic heat storage devices*. ID: 277910. DOI: https://doi.org/10.1016/j.egypro.2019.01.776. URL: http://www.sciencedirect.com/science/article/pii/S1876610219308148.

Kim, Juwon, Yeelyong Noh, and Daejun Chang (2018). *Storage system for distributed-energy generation using liquid air combined with liquefied natural gas*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2017.12.092. URL: http://www.sciencedirect.com/science/article/pii/S0306261917318159.

Li, Yongliang et al. (2014). *Load shifting of nuclear power plants using cryogenic energy storage technology*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2013.08.077. URL: http://www.sciencedirect.com/science/article/pii/S0306261913007216.

Mondejar, Maria (2015). *User guide for developed DLLs for real fluid property models in IPSEpro process simulator*. URL: https://portal.research.lu.se/portal/files/6187440/5466302.pdf.

Morgan, Robert et al. (2015). *Liquid air energy storage – Analysis and first results from a pilot scale demonstration plant*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2014.07.109. URL: http://www.sciencedirect.com/science/article/pii/S0306261914008009.

Peng, Hao et al. (2018). *A study on performance of a liquid air energy storage system with packed bed units*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2017.11.045. URL: http://www.sciencedirect.com/science/article/pii/S0306261917316203.

Peng, Xiaodong, Xiaohui She, Lin Cong, et al. (2018). *Thermodynamic study on the effect of cold and heat recovery on performance of liquid air energy storage*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2018.03.151. URL: http://www.sciencedirect.com/science/article/pii/S0306261918304963.

Peng, Xiaodong, Xiaohui She, Chuan Li, et al. (2019). *Liquid air energy storage flexibly coupled with LNG regasification for improving air liquefaction*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2019.05.040. URL: http://www.sciencedirect.com/science/article/pii/S0306261919308979.

Sciacovelli, A., A. Vecchi, and Y. Ding (2017). *Liquid air energy storage (LAES) with packed bed cold thermal storage – From component to system level performance through dynamic modelling*. ID: 271429. DOI: https://doi.org/10.1016/j.apenergy.2016.12.118. URL: http://www.sciencedirect.com/science/article/pii/S0306261916319018.

She, Xiaohui et al. (2017). *Theoretical analysis on performance enhancement of stand-alone liquid air energy storage from perspective of energy storage and heat transfer*. ID: 277910. DOI: https://doi.org/10.1016/j.egypro.2017.12.236. URL: http://www.sciencedirect.com/science/article/pii/S1876610217359702.

Yang, YM et al. (2006). "Development of the world's largest above-ground full containment LNG storage tank". In: *23rd World Gas Conference, Amsterdam*, pp. 1–14.

Zhang, Tong et al. (2020). *Thermodynamic analysis of hybrid liquid air energy storage systems based on cascaded storage and effective utilization of compression heat*. ID: 271641. DOI: https://doi.org/10.1016/j.applthermaleng.2019.114526. URL: http://www.sciencedirect.com/science/article/pii/S1359431118379249.