

Masterthesis

Betriebsanalyse einer wassergekühlten Kälteanlage

Verfasser:

Stephan Vollmuth, BSc (Matrikelnummer: 01627652)

BetreuerIn:

Univ.Prof. Dipl.-Ing. Dr.techn. Tobias Pröll Dipl.-Ing. Dr. Magdalena Wolf

Masterstudium:

Umwelt- und Bioressourcenmanagement

Institut:

Institut für Verfahrens- und Energietechnik

Wien 2021

i. Eidesstattliche Erklärung

I declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Master Thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Location, Date, Signature

ii. Danksagung

Für die großartige Zusammenarbeit möchte ich vor allem meinem betreuenden Professor Tobias Pröll und meiner Betreuerin Magdalena Wolf danken. Beide standen mir bei Fragen immer zur Seite und konnten mir die komplexesten Fragen mit ihrer Fachkompetenz beantworten. Tobias Pröll Anmerkungen haben die Arbeit in einer wissenschaftlichen Weise verbessert. Magdalena Wolf hat mir die Techniken für IPSEpro und MatLab beigebracht und bei der Erstellung der Modelle sehr geholfen. Weiters möchte ich den beiden Mitarbeiterinnen der Firma Vasko + Partner, Barbara Beigelböck und Katharina Eder danken, die zusammen mit Magdalena Wolf mir die Messgeräte erklärt und beim Anbringen und Auslesen der Geräte sehr geholfen haben. Katharina Eder hat mich außerdem bei der Ausarbeitung des Pythonprogramms unterstützt. Es war sehr angenehm in solch einem Umfeld diese Masterthesis zu verfassen und erleichterte in vielerlei Hinsicht das Schreiben. Auch möchte ich meiner Freundin Itta und meiner Familie danken, die mich immer unterstützt haben.

iii. Abstract

In this master's thesis, the efficiency of the refrigeration system in the ship station on the Danube Canal in Vienna is investigated. With the help of various measuring devices, data for calculating the heat flows and the energy efficiency ratio (EER) are obtained. These data are processed in a specially written program and ultimately processed with IPSEpro. IPSEpro then provides results that allow a statement to be made about the energy and exergy loss of the system. In addition, the measurements themselves are validated in order to further improve the future use of the inexpensive Raspberry Pi measurement setup. The validity of the temperature data of the Raspberry Pi are heavily questioned, as they differ greatly from the temperatures recorded by the professional reference measuring device (Fluxus F601). This inaccuracy in the measurement data runs through all calculations in this work. Ultimately, however, there is a result that allows a statement to be made about the efficiency of the system. Although the power of the system changes in short time intervals of few minutes, the system runs at an EER above 4 and at an exergy efficiency of 22 %.

iv. Zusammenfassung

In der vorliegenden Masterarbeit wird die Effizienz der Kälteanlage in der Schiffstation am Donaukanal in Wien untersucht. Mit Hilfe verschiedener Messeinrichtungen werden Daten für die Berechnung der Wärmeströme und der Leistungszahl für Kältemaschinen (EER) gewonnen. Diese Daten werden in einem eigens erstellten Programm aufgearbeitet und letztendlich mit IPSEpro verarbeitet. IPSEpro liefert dann Ergebnisse, die eine Aussage über den Energie- und Exergieverlust der Anlage zulassen. Außerdem werden die Messungen selbst validiert, um die zukünftige Verwendung des günstigen Raspberry Pi Messaufbaus weiter zu verbessern. Die Validität der Temperaturdaten des Raspberry Pi werden stark in Frage gestellt, da sie von den Temperaturen, die vom professionellen Referenzmessgerät (Fluxus F601) aufgenommen werden, stark abweichen. Diese Ungenauigkeit in den Messdaten zieht sich durch alle Berechnungen in dieser Arbeit. Letztendlich entsteht aber doch ein Ergebnis, das eine Aussage über die Effizienz der Anlage zulässt. Obwohl die Leistung dieser Anlage kurzfristig stark schwankt, erreicht die Anlage Leistungszahlen (EER) von über 4 und einen exergetischen Gesamtwirkungsgrad von 22 %.

v. Begriffe, Formelzeichen, Abkürzungen

$\dot{B}_{Q_{Zu}}$	=	Anergiestrom	[kW]
COM	=	Component Object Model	
COP	=	Coefficient of Performance	[-]
c_p	=	spezifische Wärmekapazität	$\left[\frac{J}{kg\ K}\right]$
DN	=	Nennweite (aus dem französischen diamètre nominal)	[mm]
$\dot{E}x_{Qzu}$	=	Exergiestrom in den Kühlraum	[kW]
$\dot{E}x_v$	=	Exergieverluststrom	[kW]
$\dot{E}x_{v,KM}$	=	Exergieverlust Kältemaschine	[kW]
$\dot{E}x_{v,Mischer}$	=	Exergieverlust Mischer	[kW]
$\dot{E}x_{v,W\ddot{\mathrm{u}}}$	=	Exergieverlsut Wärmeübertrager	[kW]
EER	=	Energy Efficiency Ratio	[-]
EERc	=	Leistungszahl nach Carnot	[-]
EER _{Ges,Rsp}	=	EER der Gesamtanlage berechnet mit dem Raspberry Pi	[-]
EER _{KE,Flx2}	=	EER Kälteerzeugung berechnet mit dem Flexim 2 Gerät	[-]
$EER_{KE,Rsp}$	=	EER Kälteerzeugung berechnet mit dem Raspberry Pi	[-]
EER _{KM,Flx2}	=	EER der Kältemaschine berechnet mit dem Flexim 2 Gerät	[-]
$EER_{KM,Rsp}$	=	EER der Kältemaschine berechnet mit dem Raspberry Pi	[-]
Ges	=	Gesamtanlage	
Н	=	Enthalpie	[J]
I	=	Stromstärke	[A]
JAZ	=	Jahresarbeitszahl	[-]
KE	=	Kälteerzeugung	
KM	=	Kältemaschine	
MDK	=	Model Development Kit	
$\dot{m}_{Flx\#}$	=	Massenstrom des Flexim # Gerätes (1 oder 2)	$\left[\frac{kg}{s}\right]$
$\dot{m}_{PCEboku}$	=	Massenstrom des PCE Gerätes der Boku	$\left[\frac{kg}{s}\right]$
\dot{m}_{PCEvp}	=	Massenstrom des PCE Gerätes der Firma Vasko und Partner	$\left[\frac{kg}{s}\right]$
р	=	Druck	[Pa]
Р	=	Leistung	[kW]
P#	=	Bezeichnung für Pumpe # (1 bis 5)	
PCE_{boku}	=	PCE Messgerät der Universität für Bodenkultur	
PCE_{vp}	=	PCE Messgerät der Firma Vasko und Partner	
Pel	=	elektrische Leistung	[kW]
$P_{el,Fl\#}$	=	elektrische Leistung des Fluke X Gerätes (1 bis 3)	[kW]
$P_{el,Ges}$	=	elektrische Leistung der Gesamtanlage	[kW]
$P_{el,KE}$	=	elektrische Leistung innerhalb der Systemgrenze Kälteerzeugung	[kW]
$P_{\text{el},\text{KM}}$	=	elektrische Leistung der Kältemaschine	[kW]
PiT#	=	Temperaturfühler des Raspberry Pi Fühlers Nummer #	

рк	=	Druck nach Verdichter	[Pa]
Ркм	=	Leistung Kältemaschine	[kW]
PSE	=	Process Simulation Environment	
p_{v}	=	Druck vor Verdichter	[Pa]
Pw	=	Wirkleistung	[kW]
Q	=	Wärme	[J]
Q	=	Wärmestrom	[kW]
\dot{Q}_{ab}	=	Wärmestrom Abfluss	[kW]
$\dot{Q}_{ab,DoW,Rsp}$	=	Wärmestrom Donauwasserseite mit Raspberry gemessen	[kW]
$\dot{Q}_{ab,K\ddot{u}W,Rsp}$	=	Wärmestrom Kühlwasserseite mit Raspberry Pi gemessen	[kW]
$\dot{Q}_{ab,K\ddot{u}W,Flx1}$	=	Wärmestrom Kühlwasserseite mit Flexim 1 gemessen	[kW]
\dot{Q}_{env}	=	Wärmestrom der nicht erklärt werden kann	[kW]
\dot{Q}_{zu}	=	Wärmestrom Zufluss bzw. Kälteleistung	[kW]
$\dot{Q}_{zu,Abn,Rsp}$	=	Wärmestrom Abnehmerseite mit Raspberry gemessen	[kW]
$\dot{Q}_{zu,KaW,Flx2}$	=	Wärmestrom Kaltwasserseite mit Flexim 2 gemessen	[kW]
$\dot{Q}_{zu,KaW,Rsp}$	=	Wärmestrom Kaltwasserseite mit Raspberry Pi gemessen	[kW]
S	=	Entropie	$\left[\frac{J}{K}\right]$
SPF	=	Seasonal Performance Factor	[-]
Т	=	Temperatur	[°C]
T_{ab} / T_{U}	=	Temperatur an die Umgebung	[°C]
T _{Flx1.#}	=	Temperatur des Flexim 1 Gerätes Nummer # (1 oder 2)	[°C]
T _{Flx2.#}	=	Temperatur des Flexim 2 Gerätes Nummer # (1 oder 2)	[°C]
Τĸ	=	Kondensationstemperatur	[°C]
$T_{\text{Pi\#}}$	=	Temperatur des Raspberry Pi Fühlers #	[°C]
Tv	=	Siedetemperatur	[°C]
T_{zu}/T_0	=	Temperatur aus Kühlraum	[°C]
U	=	Innere Energie	[J]
U	=	Spannung	[V]
V	=	Volumen	[m ³]
\dot{V}	=	Volumenstrom [m³/s]	$\left[\frac{m^3}{s}\right]$
W	=	Arbeit	[J]
Griechisch			
$arepsilon_K$	=	Leistungszahl der Kältemaschine	[-]
$arepsilon_W$	=	Leistungszahl der Wärmepumpe	[-]
$\varepsilon_{C,K}$	=	Höchstmögliche Leistungszahl Kältemaschine mit Carnot-Faktor	[-]
ζ_{KM}	=	Exergetische Wirkungsgrad der Kältemaschine	[%]
ρ	=	Dichte [$\left[\frac{kg}{m^3}\right]$
arphi	=	Phasenverschiebungswinkel	[-]

 $\cos \varphi$ = Wirkleistungsfaktor [-]

vi. Glossar

Kaltwasser

In der Gebäudetechnik steht Kaltwasser für Wasser, das direkt für die Kühlung des Gebäudes genutzt wird, also die Wärme aus dem Gebäude aufnimmt. Das Kaltwasser stammt aus einem Kühlsystem (z.B. Kältemaschine oder Wärmeübertrager) und fließt in Richtung der Abnehmer.

Kühlwasser

In der Gebäudetechnik steht Kühlwasser für Wasser, das die Wärme aus dem Kälteprozess aufnimmt.

Kälteanlage

Die Kälteanlage beschreibt das Gesamtsystem mit Kühlwasser, Kältemaschine und Kaltwasser.

Kältemaschine

Die Kältemaschine ist ein Teil der Kälteanlage und dient der Kühlung auf eine Temperatur unterhalb der Umgebungstemperatur.

Gesamtsystem

Das Gesamtsystem umfasst alle Teilsysteme eines Systems. Im Falle der Kältemaschine wird das System vom Kühlwassereintritt bis zur Aufnahme der Wärme zum Kaltwasser betrachtet.

Vorlauf

Der Vorlauf bezeichnet ein wärme- oder kälteübertragendes Medium, das einem System zugeführt wird.

Rücklauf

Ein Rücklauf bezeichnet ein wärme- oder kälteübertragendes Medium, das aus einem System abgeführt wird.

vii. Inhaltsverzeichnis

	i.	Eidesstattliche Erklärung	2
	ii.	Danksagung	3
	iii.	Abstract	4
	iv.	Zusammenfassung	5
	٧.	Begriffe, Formelzeichen, Abkürzungen	7
	vi.	Glossar	10
	vii.	Inhaltsverzeichnis	12
1	Einleitu	ıng	16
2	Theore	tische Grundlagen	18
2.1	Der er	ste und zweite Hauptsatz der Thermodynamik	18
2.2		calisches Prinzip von Kältemaschinen	
2.3	-	der Technik von Kälteanlangen	
2.0	Otaria	doi 100mm von Randamangon	20
3	Ziele ur	nd Forschungsfragen	25
4	Method	en	26
4.1	Besch	reibung Kälteanlage	26
	4.1.1	Free Cooling	
	4.1.2	Kältemaschinenbetrieb	27
4.2	Besch	reibung Kältemaschine	28
4.3		reibung Messaufbau	
	4.3.1	Fluke	
	4.3.2	Qualistar	30
	4.3.3	Flexim	
	4.3.4	PCE-Durchflussmesser	31
	4.3.5	Raspberry Pi	31
4.4	Nutzbare Messungen		
	4.4.1	Nutzbare Temperaturmessungen	32
	4.4.2	Nutzbare Leistungsmessung	33
	4.4.3	Nutzbare Durchflussmessung	35
4.5	Daten	auswertung mit Python	37
	4.5.1	Daten einlesen	
	4.5.2	Anpassung des Zeitstempels der Raspberry Pi Fühler	38
	4.5.3	Berechnungen der Wärmeströme	39
	4.5.4	Berechnung der EER und EERc	39
4.6	IPSFn	ro	42

	4.6.1	Process Simulation Environment (PSE)	42
	4.6.2	Model Development Kit (MDK)	43
	4.6.3	PSValidate	47
	4.6.4	Randbedingungen:	47
	4.6.5	Daten in IPSEpro	49
	4.6.6	Messfehler in IPSEpro	50
	4.6.7	Statische Betrachtung	51
	4.6.8	Quasi-dynamische Modellierung	53
5	Auswer	rtung	54
5.1	Tempe	eraturmessung	54
5.2	Ergebi	nisse aus der statischen Simulation	59
	5.2.1	Systemgrenze Kältemaschine	
	5.2.2	Systemgrenze Kälteerzeugung	
	5.2.3	Systemgrenze 3 Gesamtanlage	
5.3	Ergebi	nisse aus der quasi-dynamischen-Simulation	
	5.3.1	Normaler Modus	
	5.3.2	Validierungsmodus	65
	5.3.3	Exergieverluste	
6	Diskuss	sionsion	72
6.1	Messe	erfahrung	72
	6.1.1	Ausfälle der Messgeräte	72
	6.1.2	Ungenauigkeiten der Messung	72
6.2	EER, I	EERc, Gütegrad, Temperaturhub	74
6.3		nprogramm	
6.4	-	pro	
7	Conclus	sio und Ausblick	77
8	Abbildu	ungsverzeichnis	80
9	Tabelle	nverzeichnis	82
10	Literatu	urverzeichnis	84
Δ	Δnhano	7	87

A.1.	MatLab Skript für die quasi-dynamische Simulation im normalen Modus	87
A.2.	MatLab Skript für die quasi-dynamische Simulation im	
	Validierungsmodus	87

1 Einleitung

Das Jahr 2018 war das Wärmste Jahr seit Beginn der Wetteraufzeichnungen. Die Temperaturen lagen etwa 1 °C über der Temperatur der vorindustriellen Zeit [1]. Mit dem Klimawandel einhergehend gibt es viele Veränderungen des Klimas wie z.B. die steigende Erhöhung der Mitteltemperatur, Meeresspiegel, Hitzeextreme, Starkniederschläge und die Zunahme von Dürren [2]. Vor allem die steigende Temperatur und die Zunahme von Hitzeextremen führt zu einem Zuwachs von Kälteanlagen Gebäudekühlung. In schnell wachsenden. zur warmen Schwellenländern wie z.B. Indien steigt die Anzahl an Klimaanlagen stark an. Allein im Jahr 2019 stieg der Absatz von Klimageräten um 15 % [3]. Aus diesen Gründen macht Sinn, sich mit der Effizienz von Kälteanlagen zu beschäftigen und Verbesserungsvorschläge zu erarbeiten.

Kälteenergie wird heutzutage in vielen Bereichen benötigt. So werden z.B. in Haushalten, der Gastronomie oder der Arzneiproduktion für die Kühlung von verderblichen Produkten oder für großflächige Raumkühlung in Gebäuden Kältemaschinen verwendet. Das führt zu einer erheblichen Energienachfrage, die vor allem in Zukunft durch höhere Komfortansprüche, steigende Temperaturen, aber auch durch verbesserte Wärmedämmung, da die inneren Wärmequellen nicht mehr abgeführt werden können und daher gekühlt werden muss, weiter steigen [4]. Durch die geringe Datengrundlage ist es schwer die Anzahl verbauter Kälteanlagen und die dadurch verbrauchte Energie einzuschätzen. Eine Studie des Umweltbundesamtes kam aber in Deutschland auf einen jährlichen elektrischen Energiebedarf der stationären Kältetechnik von 70.695 GWh/a, was 13,5 % des gesamten elektrischen Energiebedarfs Deutschlands entspricht. Außerdem geht diese Studie davon aus, dass die installierte Kälteleistung jährlich um etwa 3,5 bis 5 % steigt [5]. Roger Hitchin et al. schätzen, dass die installierte Kälteleistung in Europa bis 2025 um 55 bis 60 % im Vergleich zu 2010 steigt [6]. Um dieser enormen Steigerung der Kälteleistung entgegenzuwirken, soll laut der Europäischen Union nach dem Klima- und Energiepaket 2020, das eine Effizienzsteigerung von 20 % vorsah [7], im Zeitraum von 2021 bis 2030 eine Steigerung der Effizienz von 32,5 % vorgeschrieben werden [8].

Ein weiteres großes Problem bei dem Betrieb vieler Kälteanlagen ist die Verwendung von fluorierten Kohlenwasserstoffen (FKW). 2017 lag die Emission von FKW um 0,1

Mio. Tonnen CO₂-Äquivalent über dem Klimaschutzgesetz, was zum Großteil auf Kälteanlagen zurückzuführen ist. Mit den Maßnahmen der Verordnung Nr. 842/2006 des europäischen Parlaments und des Rates über bestimmte fluorierte Treibhausgase ist allerdings in früher Zukunft ein Rückgang zu erwarten [9].

Das in dieser Arbeit untersuchte Gebäude steht direkt am Donaukanal am Schwedenpatz in Wien. Darin befindet sich ein Restaurant bzw. Café und eine Anlegestelle für den touristischen Schiffsverkehr. Über die verbaute Kältemaschine werden nur das Restaurant und Büros gekühlt, also keine Kühlkammern oder Wohnungen. Für diese Raumkühlung nutzt die Kälteanlage den Donaukanal als Rückkühler, was eine Besonderheit in der Gebäudekühlung und für das Forschungsprojekt ViCC (Virtual Cooling Control) darstellt. Mit dem Forschungsprojekt ViCC soll die Effizienzsteigerung von Kompressionskältemaschinen durch verschiedene Regelstrategien untersucht werden. Zudem war die Firma Vasko und Partner, mit der die BOKU im Zuge des Forschungsprojekt ViCC zusammenarbeitet an der Planung und Umsetzung der Kälteanlage beteiligt, was den Zugang zur Kälteanlage ermöglicht. Da das Gebäude gewerblich genutzt wird, unterliegt es bestimmten Normen, die eine Planung von Kühlsystemen voraussetzt. Da der Typ und die Nutzung des untersuchten Gebäudes vielen Gebäuden mit moderner Architektur ähnelt, die Wasserrückkühlung aber selten in solchen Gebäuden zu finden ist, stellt die wissenschaftliche Analyse der Kälteanlage einen wichtigen Beitrag für die Optimierung von Kompressionskältemaschinen dar.

2 Theoretische Grundlagen

2.1 Der erste und zweite Hauptsatz der Thermodynamik

Laut dem ersten Hauptsatz der Thermodynamik kann Energie weder erzeugt noch vernichtet werden. Energie kann nur von einer Energieform in eine andere umgewandelt werden. Somit kann sich die Innere Energie U nur durch Energie in Form von Arbeit W oder Wärme Q durch Austausch über die Systemgrenzen ändern [10].

$$dU = \delta Q + \delta W \tag{2-1}$$

Auf Gleichung (2-1) aufbauend lässt sich die Enthalpie beschreiben. Addiert man zur Inneren Energie U das Produkt der Zustandsgrößen aus Druck und Volumen erhält man die Enthalpie H (2-2).

$$H = U + p \cdot V \tag{2-2}$$

Der Vorteil dieser extensiven Größe besteht darin, dass bei konstantem Druck (dp=0) die Veränderung der Enthalpie dH durch das Differential

$$dH = dU + p dV (2-3)$$

ausgedrückt werden kann. Das bedeutet, dass bei konstantem Druck die Enthalpieänderung gleich der differenzialen Wärme δQ ist [11].

Um aber eine Aussage darüber treffen zu können, ob und inwieweit eine Energie in eine andere Energieform umgewandelt werden kann, muss der zweite Hauptsatz der Thermodynamik betrachtet werden. So ist z.B. die vollständige Umwandlung von einem Wärmestrom in mechanische oder elektrische Energie nicht möglich, ansonsten würde es sich um ein Perpetuum Mobile 2. Art handeln. Das heißt, selbst bei einer reversiblen Wärmekraftmaschine kann nur ein Teil der Wärme in Energie umgewandelt werden und ein Teil muss an die Umgebung bei möglichst geringer Temperatur abgegeben werden. Andersherum lässt sich aber theoretisch mechanische oder elektrische Energie ohne Verluste in innere Energie oder Wärme umwandeln [12]. Bei einem idealen reversiblen Prozess wird keine Entropie erzeugt, was bedeutet, dass die thermodynamische Zustandsänderung jederzeit wieder umgekehrt werden kann (2-4).

$$dS = 0 (2-4)$$

Allerdings kann in einem abgeschlossenen System die Entropie niemals abnehmen, sondern nach dem zweiten Hauptsatz der Thermodynamik nur zunehmen. Daher werden Prozesse, bei denen Entropie erzeugt wird, als irreversibel bezeichnet. Sie können nur rückgängig gemacht werden, wenn sie die Entropie an ein anderes System abgeben (2-5).

$$dS \ge 0 \tag{2-5}$$

Bei reversiblen Prozessen ist die Änderung der Entropie der Quotient aus der Änderung der Wärme dQ und der Temperatur T (2-6).

$$dS = \frac{dQ}{T} \tag{2-6}$$

Beschränkt umwandelbare Energieformen sind außerdem vom Zustand der Umgebung abhängig und bringen Entropie in das betrachtete System. Im stationären Fall muss das System die Entropie in Form von Wärme- oder Stoffströmen wieder abgeben.

Energie hat einen unbeschränkt umwandelbaren Teil, der als Exergie bezeichnet wird und einen Teil, der nur beschränkt umwandelbar ist, der als Anergie bezeichnet wird. Daraus ergibt sich (2-7) [13]:

Betrachtet man einen Kühlraum, so fließt ein Exergiestrom $\dot{E}x_{Q_{zu}}$ in, und ein Anergiestrom $\dot{B}_{Q_{zu}}$ aus dem Kühlraum (Abbildung 2-1). Zusammen ergeben diese Ströme die Kälteleistung. Bei einer irreversibel arbeitenden Kältemaschine muss die Antriebsleistung P_{KM} um einen Exergieverluststrom $\dot{E}x_v$ ergänzt werden (2-8).

$$P_{KM} = \left| \dot{E} x_{Q_{zu}} \right| + \dot{E} x_{v}. \tag{2-8}$$

Somit verwandelt sich die zusätzlich benötigte Antriebsleistung in Anergie $\dot{E}x_v$ und erhöht den als Wärme an die Umgebung abzugebenden Anergiestrom (2-9).

$$\left|\dot{Q}\right| = \dot{B}_{Q_{zu}} + \dot{E}x_v \tag{2-9}$$

Der exergetische Wirkungsgrad von Kältemaschinen kann dann über das Verhältnis

$$\zeta_{KM} = \frac{\left| \dot{E} x_{Q_{Zu}} \right|}{P_{KM}} \tag{2-10}$$

berechnet werden.

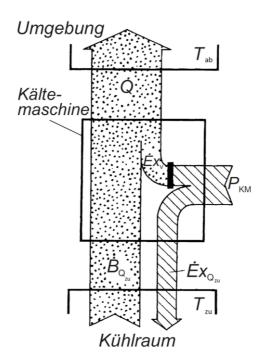


Abbildung 2-1: Anergie- und Exergiestrom bei einer Kältemaschine [12]

2.2 Physikalisches Prinzip von Kältemaschinen

Kälteanlagen sind ähnlich aufgebaut wie Wärmepumpen. Sie nehmen einen Wärmestrom bzw. Kühllast \dot{Q}_{zu} auf und heben ihn mithilfe einer Hilfsenergie, die meist eine mechanische Energie P darstellt, auf ein höheres Energieniveau \dot{Q}_{ab} . Man unterscheidet dabei zwischen Kälteanlage und Wärmepumpe nur aufgrund der aufgenommenen und abgegebenen Temperaturniveaus [14] (siehe Abbildung 2-2). Kälteanlagen führen also Wärme ab und geben diese über einen Rückkühler an die Umwelt ab. Die Effizienz von Anlagen wird mit der sogenannten Leistungszahl beschrieben. Der Energy Efficiency Ratio (EER) (2-11) gibt die Effizienz von Kälteanlagen als den Quotienten aus Kühllast und Hilfsenergie P an.

$$\varepsilon_{K} = EER := \frac{\dot{Q}_{zu}}{P}$$
(2-11)

Berücksichtigt man die Energieerhaltung erhält man Gleichung (2-12).

$$\dot{Q}_{ab} = \dot{Q}_{zu} + P$$
 (2-12)

Die höchste theoretische Effizienz von Kältemaschinen kann über die Leistungszahl nach Carnot (EERc) wie in Gleichung (2-13) beschrieben werden (mit Temperaturen in Kelvin). Die EERc kann nur für die Systemgrenze Kältemaschine verwendet werden,

denn nur beim Kältemaschinenprozess kann der Carnotprozess als Vergleichsprozess herangezogen werden.

$$\varepsilon_{\text{C,K}} = \text{EERc} = \frac{T_{\text{zu}}}{T_{\text{ab}} - T_{\text{zu}}}$$
 (2-13)

Diese Carnot-Leistungszahl kann dann zusammen mit der Leistungszahl der Kältemaschine in ein Verhältnis gesetzt werden was den Gütegrad der Kältemaschine ergibt. Dieser Gütegrad beschreibt wie nah die echte Leistungszahl (EER) an der theoretisch höchsten Leistungszahl (EERc) liegt (2-14). Umso höher dieser Gütegrad liegt umso effizienter läuft die Kältemaschine.

Gütegrad Kältemaschine =
$$\frac{EER}{EERc}$$
 (2-14)

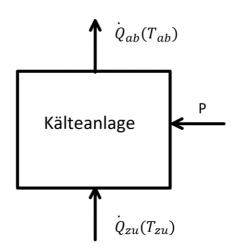


Abbildung 2-2: Schema einer Kälteanlage bei der die Wärme \dot{Q}_{ab} die Summe aus der Hilfsenergie P und der Kühllast \dot{Q}_{zu} darstellt

Hohe Leistungszahlen bedeuten, dass der Prozess eine geringe Menge an Hilfsenergie im Vergleich zur Kühllast benötigt. Die Leistungszahlen stellen somit einen Indikator für die Energieeffizienz dar. Allerdings können Leistungszahlen nur bedingt verglichen werden da sie nur bei gleichen Betriebspunkten vergleichbar sind [14]. Deshalb wird der Zeitfaktor in der Jahresarbeitszahl (JAZ), oder auch Seasonal Performance Faktor (SPF) genannt, wie in Gleichung (2-15) verwendet. Er besteht aus dem Quotienten des Intergrals der Kühllast mit der Zeit und dem Integral der elektrischen Leistung mit der Zeit. Diese Jahresarbeitszahl wird meist jedoch nur bei

Wärmepumpen verwendet und findet nur selten Anwendung bei der Bewertung von Kältemaschinen.

$$JAZ = \frac{\int \dot{Q}_{zu} dt}{\int P_{el} dt}$$
 (2-15)

Das mit Abstand am meisten verbreitete Prinzip von Kältemaschinen ist das Kaltdampfverfahren. Dabei wird, wie in Abbildung 2-3 zu sehen ist, zyklisch das Kältemittel verdampft, komprimiert, kondensiert und entspannt [14]. Idealisiert dargestellt, saugt zuerst ein Verdichter das gasförmige Arbeitsmedium mit niedrigem Druck p_√ an (Zustand 1) und hebt den Druck auf das Niveau p_K an. Dies geschieht oft in einem überhitzten Zustand, um den Verdichter zu schonen. Ist dieser Vorgang ideal, das heißt adiabat und reversibel, ist das Medium immer noch gasförmig und die Entropie bleibt gleich. Das Kältemittel gibt nun im Kondensator seine Wärme an die Umgebung oder ein externes Medium ab. Das Gas kondensiert bei gleichbleibendem Druck (Zustand 3) und wird im flüssigen Siedezustand isenthalp über ein Drosselventil entspannt. Dadurch sinkt sowohl der Druck als auch die Temperatur ab (Zustand 4). Das flüssige, kalte Kältemittel nimmt dann über den Verdampfer die Kühllast auf und der Kreislauf beginnt wieder von vorne. Die Zustandsänderung des Kältemittels ist in Abbildung 2-4 angegeben. Die Siedetemperatur Tv bei pv muss dabei aufgrund des 2. Hauptsatzes der Thermodynamik etwas niedriger als die Kühlraumtemperatur To sein. Die Kondensationstemperatur Tk muss etwas über der Umgebungstemperatur Tu liegen [12].

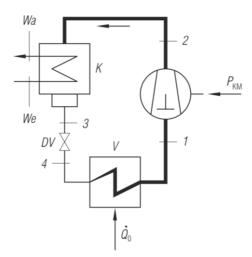


Abbildung 2-3: Schaltbild einer Kompressionskältemaschine Seite 596 [12]

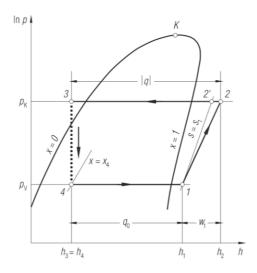


Abbildung 2-4: Zustandsänderung des Kältemittels einer Kompressionskältemaschine im In p, h-Diagramm (Seite 596) [12]

2.3 Stand der Technik von Kälteanlangen

Die Kälteerzeugung hat eine lange Geschichte. In den Anfängen wurde Natureis zur Kühlung verwendet, wobei im Sommer, wenn das Eis zur Neige ging, vereinzelt Nachtkühlungen, Erdreichwärmetauscher und Brunnenwasser eingesetzt wurden. Ende des 19. Jahrhundert wurden die ersten Absorptions-Kompressionsmaschinen in Betrieb genommen. Diese Maschinen produzierten meist mit Hilfe von Dampfmaschinen Kunsteis und lösten vor allem in der Industrie die Kühlung mit Natureis ab, wurden aber noch wenig für direkte Kältebereitstellung genutzt [15]. Aus heutiger Sicht stellen Kompressions-, Absorptions-, und Adsorptionskältemaschinen die wichtigsten Kälteerzeuger dar. Wie in Kapitel 2.2 beschrieben, wird bei der Kompressionskältemaschine die Verdampfungswärme beim Aggregatsübergang gasförmig/ flüssig genutzt. Die Absorptionskältemaschine nutzt einen thermischen Verdichter, bei dem das Kältemittel in einem Lösungsmittel bei niedriger Temperatur in einem zweiten Stoff absorbiert wird. Adsorptionskältemaschine kommt ein festes Sorptionsmittel mit großer innerer Oberfläche zum Einsatz an das sich gasförmige Fluide anlagern können und kondensieren. Diese Kondensationswärme wird dann frei [16]. Ein Vergleich der drei Anlagentypen ist in Tabelle 2-1 zu sehen. Bei Anlagen für mittlere und große Kälteleistungen werden die Maschinen meist als Kaltwassersätze ausgeführt, das heißt es wird eine Flüssigkeit gekühlt [17].

Literatur zur Kältetechnik gibt es viel. Über Boku:LITsearch, Google Scholar und anderen Internet- und Bibliothekssuchen wurde die Literatur für diese Arbeit herausgesucht. D. Baehr [12] liefert z.B. eine breite Grundlage zur Thermodynamik und deren Anwendungsgebiete, J. Dohmann [14] geht genauer auf die spezielle Anwendung von Kälteanlagen ein und K. Fitzner [15] erklärt die verschiedenen Kältemaschinentypen. Das Bundesministerium für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft [18] gibt einen Leitfaden für Energieaudits für Unternehmen heraus, um Kälteanlagen auf ihre Effizienz zu untersuchen.

Tabelle 2-1: Vergleich Kompressions-, Absorptions- und Adsorptions- kälteanlagen [16]

Übersicht über Kälteanlagen-Tech	nologien			
	Kompressions- kälteanlage	Absorptions- kälteanlage	Adsorptions- kälteanlage	DEC ¹ -Anlage
Physikalischer Kühlungs-Effekt	Verdampfen des Kä	iltemittels (Kaltdampf	prozess)	Verdunstung des Kältemittels
Verdichtungs- prinzip	mechanische Verdichtung	thermisch, Absorptions- lösungskreislauf	thermisch, Adsorption von Wasserdampf	sorptive Entfeuchtung
Antriebsenergie	Elektrische Energie	Wärmeenergie (85-180)°C	Wärmeenergie (55-95)°C	Wärmeenergie (50-100)°C
Kältemittel	chlorierte oder chlorfreie Koh- lenwasserstoffe	Wasser mit <i>LiBr</i> oder <i>NH</i> ₃ als Absorptionsmittel	Wasser mit Fest- stoff als Adsorpti- onsmittel (SILICA-Gel)	Wasser
Spezifischer Primärenergie- verbrauch ²	1,3–1,65	0,6–1,0	0,4–0,6	0,3

dessicative and evaporative cooling

Eine Zahl von 0,6–1,0 (Absorptionskälteanlage) bedeutet, dass je eingesetzter Kilowattstunde Wärme 0,6–1,0 Kilowattstunden Kälte bereitgestellt werden.

² Der spezifische Primärenergieverbrauch errechnet sich aus dem Verhältnis der erhaltenen Kältemenge zur eingesetzten elektrischen bzw. thermischen Energie.

3 Ziele und Forschungsfragen

Ziel der Arbeit ist es, die Energieeffizienz der Kälteanlage des touristisch genutzten Wirtschaftsgebäudes am Donaukanal zu bestimmen und Optimierungsmaßnahmen anzustellen. Dabei werden die Energie- und Exergieströme identifiziert und quantifiziert. So werden anhand einer konkreten Anlage die Effizienzsteigerungspotentiale erhoben und dabei die Effizienz in Bezug zur thermodynamisch möglichen Effizienz betrachtet, was eine verallgemeinerte Aussage erlaubt.

Im Detail werden folgende Forschungsfragen behandelt:

- 1. Wie lassen sich die benötigten Messdaten kostengünstig und ohne in das laufende System einzugreifen erfassen und aufbereiten?
- 2. Wie valide sind die Messdaten und lassen sich Schlüsse mit Hilfe eines Simulationsmodells in IPSEpro ziehen? Dafür wird sowohl eine statische Simulation als auch eine quasi-dynamische-Simulation durchgeführt.
- 3. Wie effizient arbeitet die Anlage? Mit Hilfe von Massen- und Exergieströmen sowie physikalische Parameter können Hinweise auf die Effizienz gezogen werden.

4 Methoden

4.1 Beschreibung Kälteanlage

Das gesamt betrachtete System, das hier als Kälteanlage bezeichnet wird, besteht aus einer Kältemaschine, einem Pufferspeicher, einem Wärmeübertrager, fünf Pumpen, Rohren, Filtern, Ventilen und weiteren Bauteilen, die für diese Arbeit nicht weiter relevant sind. Das System nutzt Donauwasser aus dem Donaukanal als Kühlmedium. Die gesamte Anlage läuft entweder im sogenannten "freecooling" Betrieb, bei dem ausschließlich das Donauwasser als Wärmesenke genutzt wird, oder bei wärmeren Temperaturen mit Hilfe der Kältemaschine, die das Kaltwasser bereitstellt.

Für diese Arbeit ist die Anlage in drei Systemgrenzen unterteilt (siehe Abbildung 4-1). Die Systemgrenze "Kältemaschine" (KM) umfasst die sowohl Kältemaschine als auch die zwei Kühlwasserleitungen und die zwei Kaltwasserleitungen (in Blau dargestellt). Das System Kälteerzeugung (KE) schließt die Systemgrenze Kältemaschine, den Wärmeübertrager und die Leitungen mit dem Donauwasser mit ein (in Grün dargestellt). Die Systemgrenze "Gesamtanlage" (Ges) umfasst alle Teile der Anlage mit Pufferspeicher und den Abnehmer Vor- und Rücklauf (in Gelb dargestellt).

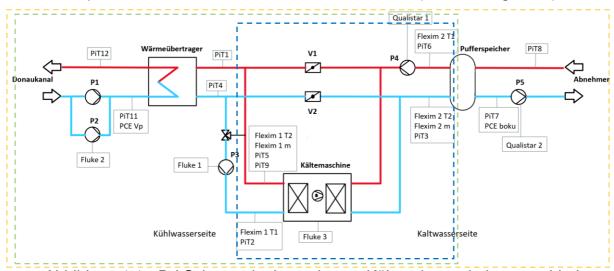


Abbildung 4-1: R-I Schema der betrachteten Kälteanlage mit den verschiedenen Messpunkten, Pumpen, Kalt-, und Warmwasserleitungen und der Kältemaschine, eingeteilt in drei Systemgrenzen (Kältemaschine blau, Kälteerzeugung grün, Gesamtanlage gelb)

4.1.1 Free Cooling

Ist die Kältemaschine nicht in Betrieb, läuft also die Kühlung ausschließlich über den Wärmeübertrager direkt zum Pufferspeicher, dann bezeichnet man dies als "freecooling" bzw. im deutschen als freie Kühlung (Abbildung 4-2). Die Anlage besitzt drei Kreisläufe im freecooling Betrieb. Der erste Kreislauf ist sowohl für den freecooling Betrieb als auch für den Kältemaschinenbetrieb identisch. Er wird ausschließlich mit Donauwasser betrieben und besteht aus Donauwasser Zu- und Ablauf, zwei tageweise abwechselnd laufenden Pumpen (P1 und P2) mit vorgeschaltetem Filter und einem Wärmeübertrager. Im zweiten Kreislauf des freecooling Betriebs befindet sich eine Pumpe (P4), die das Wasser zwischen Wärmeübertrager und Pufferspeicher zirkulieren lässt. Hier sind auch die Ventile verbaut, die im Kältemaschinenbetrieb den direkten Durchfluss zwischen Wärmeübertrager und Puffer blockieren. Im dritten Kreislauf. der ebenfalls für den freecooling **Betrieb** als auch Kältemaschinenbetrieb gleich ist, befördert eine Pumpe (P5) das Kaltwasser vom Pufferspeicher zu den Abnehmern.

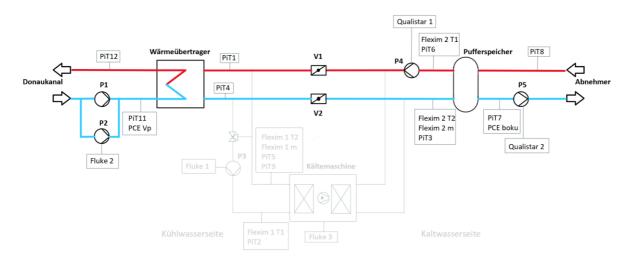


Abbildung 4-2: Schema des Freecooling-Betriebs der Kälteanlage

4.1.2 Kältemaschinenbetrieb

Ist die Kältemaschine in Betrieb, schließen sich die Sperrventile und der zweite Kreislauf teilt sich in einen Kühlwasserkreislauf und einen Kaltwasserkreislauf (siehe Abbildung 4-3). Das Kühlwasser wird durch eine Pumpe (P3), die nach einem Drei-Wege-Mischventil verbaut ist, im Kreis zwischen dem Wärmeübertrager und der Kältemaschine gepumpt. Damit die Eintrittstemperatur in die Kältemaschine nicht zu niedrig ist, und die Maschine somit in Störung gehen würde, mischt das Mischventil wärmeres Wasser aus der Kältemaschine kommend mit kälterem Wasser aus dem

Wärmeübertrager. Im Kaltwasserkreislauf wird das Wasser von der Pumpe P4 zwischen Kältemaschine und Pufferspeicher im Kreis gepumpt. Der vierte Kreislauf speist sich aus dem 700l Pufferspeicher und bedient damit die Abnehmer.

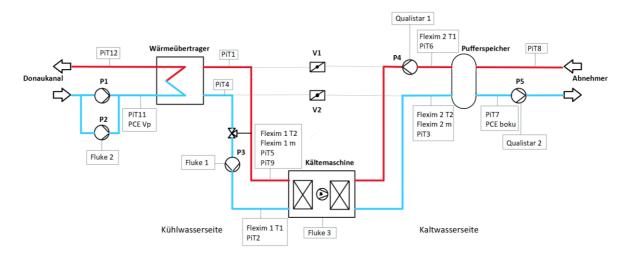


Abbildung 4-3: Schema des Kältemaschinenbetrieb der Kälteanlage

4.2 Beschreibung Kältemaschine

In der Kälteanlage ist ein modularer, wassergekühlt aufgebauter Kaltwassersatz der Firma Daikin verbaut. Dieser besteht aus insgesamt drei Kältemaschinen. Zwei des Typs EWWP065KAW1M und einer des Typs EWWP055KAW1M¹. Alle Module besitzen zwei Kältekreisläufe, zwei Gegenstrom-Plattenwärmeübertrager die mit zwei hermetischen Scrollverdichtern das Kältemittel R-407C transportieren. Laut Datenblatt haben die beiden 065 Modelle eine Nennleistung von 65 kW und das Modell 055 eine Nennleistung von 56 kW, was eine Gesamt-Nennkühlleistung von 186 kW ergibt. Insgesamt liegt die Leistungsaufnahme der gesamten Kältemaschine bei 50,7 kW mit Verdichter, Regelkreis und Pumpen. Dadurch liegt der EER bei Volllast bei 3,67. Die Anlage kann dabei in 6 Stufen betrieben werden [19].

_

¹ Das M am Ende der Bezeichnungen steht für "Modular". Das Datenblatt im Anhang ist für die 1N Modelle aber unterscheidet sich nicht von den Daten der M Modelle.

4.3 Beschreibung Messaufbau

In diesem Kapitel werden die verwendeten Messgeräte und die Messmethoden vorgestellt. Alle Messgeräte arbeiten mit sogenannten Clamp-on Fühlern und sind somit leicht direkt auf den Rohren zu installieren. Tabelle 4-1 zeigt alle Messfühler die an der Anlage angebracht werden, die jeweilige Messstelle und die Bezeichnung des Messwertes in dieser Arbeit.

Tabelle 4-1: Alle genutzten Messfühler mit der jeweiligen Messstelle, dem genutzten Messwert und der Bezeichnung

Messfühler	Messstelle	Bezeichnung	Messwert
PiT1	Kaltwasser/ Kühlwasser	T_{Pi1}	
FILL	Rücklauf	¹ Pi1	
PiT2	Kühlwasser Zulauf	T_{Pi2}	
PiT3	Kaltwasserzulauf	T_{Pi3}	
PiT4	Kaltwasser/ Kühlwasser Vorlauf	T_{Pi4}	
PiT5	Kühlwasser Rücklauf	T_{Pi5}	
PiT6	Kaltwasser Rücklauf	T_{Pi6}	
PiT7	Abnehmer Vorlauf	T_{Pi7}	[00]
PiT8	Abnehmer Rücklauf	T_{Pi8}	[°C]
PiT9	Kühlwasser Rücklauf	T_{Pi9}	
PiT11	Donauwasser Vorlauf	T_{Pi11}	
PiT12	Donauwasser Rücklauf	T_{Pi12}	
Flexim1.1	Kühlwasser Zulauf	$T_{Flx1.1}$	
Flexim1.2	Kühlwasser Rücklauf	$T_{Flx1.2}$	
Flexim2.1	Kaltwasser Rücklauf	$T_{Flx2.1}$	
Flexim2.2	Kaltwasser Zulauf	$T_{Flx2.2}$	
Flexim 1 m1	Kühlwasser Rücklauf	\dot{m}_{Flx1}	
Flexim 2 m2	Kaltwasserzulauf	\dot{m}_{Flx2}	[ma 3 /ha]
PCE Vp	Abnehmer Vorlauf	\dot{m}_{PCEvp}	[m ³ /h]
PCE Boku Donauwasser Vorlauf		$\dot{m}_{ extit{PCEboku}}$	
Fluke 1	Pumpe 3 (P3)	$P_{el,Fl1}$	
Fluke 2	Pumpe 2 (P2)	$P_{el,Fl2}$	
Fluke 3	Kältemaschine (KM)	$P_{el,Fl3}$	[kW]
Qualistar 1	Pumpe 4 (P4)	$P_{el,Q1}$	
Qualistar 2	Pumpe 5 (P5)	$P_{el,Q2}$	

4.3.1 Fluke

Um die elektrische Leistung der Pumpen und der Kältemaschine zu messen, werden insgesamt drei 1735 Power Logger Geräte der Marke Fluke in der Anlage angebracht. Fluke 1 an der Pumpe die Wasser zwischen Wärmeübertrager und Kälteanlage pumpt (P3), Fluke 2 an einer der Donauwasserpumpen (P2) und Fluke 3 an der Kältemaschine. Ziel der Geräte ist es, mit Hilfe der aufgenommen Spannung U, der Stromstärke I und der Phasenverschiebung φ aus der Formel (4-1) die Wirkleistung Pw zu berechnen. Bei der Leistungsmessung gibt der Hersteller einen Eigenfehler von ± 0.5 % des Messwerts und einen Betriebsfehler von ± 1.5 % des Messwertes an [20]. Da der Speicher der Geräte begrenzt ist, liegt das Zeitintervall zwischen den gespeicherten Messwerten anfangs bei einer Minute, im späteren Verlauf der Messungen bei zwei bzw. vier Minuten.

$$P_{W} = U * I * \cos \varphi \tag{4-1}$$

4.3.2 Qualistar

Für die Leistungsmessung der Pumpe P4, die das Kaltwasser zirkulieren lässt und der Pumpe P5, die die Abnehmer versorgt, werden zwei Analysatoren der Firma Chauvin Arnoux verwendet. Die Geräte des Typs Qualistar+ C.A 8331 berechnen die elektrische Leistung genau wie die Fluke Geräte über die Spannung, die Stromstärke und die Phasenverschiebung (Gleichung(4-1)). Das Gerät Qualistar1 misst die Leistung der Pumpe P4 und das Gerät Qualistar2 die Leistung von Pumpe P5 (siehe Abbildung 4-1 und Tabelle 4-1). Der Eigenfehler der Geräte liegt bei ± 1 % bei einer Phasenverschiebung von $cos \varphi \geq 0.8$ bzw. bei ± 1.5 % bei einer Phasenverschiebung von $0.2 \leq cos \varphi < 0.8$ [21]. Das Messintervall liegt über den gesamten Zeitraum der Messungen bei einer Minute.

4.3.3 Flexim

Das Gerät Fluxus F601 von der Firma Flexim ist ein Ultraschall Durchflussmessgerät. Zwei Geräte dieses Typs werden an der Kälteanlage angebracht, um die Wärmeleistungen auf der Kühlwasserseite (Flexim 1) und die Wärmeleistung auf der Kaltwasserseite (Flexim 2) zu messen. Dabei werden die Messdaten jede Minute gemittelt aufgezeichnet und gespeichert. Das Gerät berechnet selbstständig aus dem gemessenen Durchfluss, der spezifischen Wärmekapazität, der Dichte und der Temperaturdifferenz aus Wasser Vor- und Rücklauf der Wärmestrom \dot{Q} (siehe Gleichung (4-2) [22]. Bei der Volumenstrommessung gibt der Hersteller eine

Messunsicherheit des Messsystems von $\pm 0,3$ % des Messwertes $\pm 0,005$ m/s und bei der Messstelle von ± 1 % des Messwertes $\pm 0,005$ m/s an. Die Messungenauigkeit bei den zwei verwendeten Pt100 Clamp-on Temperaturfühlern für eine Messung von ΔT liegt bei $\leq 0,1$ K (3 K $<\Delta$ T<6 K) [23]. Das Zeitintervall ist über die gesamte Messperiode bei einer Minute.

$$\dot{Q} = \dot{V} * c_n * \rho * (T_2 - T_1)$$
 (4-2)

4.3.4 PCE-Durchflussmesser

Das Ultraschall Durchflussmessgerät PCE-TDS 100HSH arbeitet nach dem Laufzeit-Differenzverfahren. Dabei liegt die Genauigkeit für Rohre bei einer Nennweite DN ≥50mm bei ±1,5 % vom Messwert [24]. In Kombination mit den Temperaturfühlern des Raspberry Pi's wird daraus der Wärmstrom \dot{Q} berechnet. An der Kälteanlage werden zwei PCEs installiert, eines am Zulauf des Donauwassers (PCE_{vp}) und eines am Verbraucherkreislauf direkt nach dem Pufferspeicher (PCE_{boku}). Der Wärmestrom wurde ebenfalls mit der Gleichung (4-2) wie bei den Flexim Geräten berechnet. Beide PCE Geräte sind über Schnittstellen mit den Raspberry Pis verbunden, auf denen die Daten abgelegt und gespeichert werden. Die Messdaten werden dabei jede volle Minute aufgezeichnet.

4.3.5 Raspberry Pi

Um die Durchflüsse der PCEs zu speichern und 11 Temperaturfühler zu verwenden, werden zwei ein Platinen Computer der Firma Raspberry Pi verwendet. Die Temperaturfühler des Typs DS18S20 werden über eine 1-Wire-Schnittstelle von einem der Raspberrys über ein Programm ausgelesen und gespeichert. Zum einen werden die Temperaturdaten dafür verwendet, um die Flexim Geräte zu überprüfen und zum anderen, um daraus, zusammen mit den PCEs, den Wärmestrom zu berechnen. Alle Clamp-on-Temperaturfühler werden mit Wärmeleitpaste unter die aufgeschnittenen Rohrdämmungen geklemmt, um eine möglichst gut isolierte Temperaturmessung zu erhalten. Der Hersteller gibt eine Genauigkeit von ±0,5 °C für die Temperaturfühler an [25]. Ein vorrangegangener Test der Masterthesis von M. Rapport [26] zeigte bereits, dass die durchschnittliche Abweichung im Eiswasser bei ±0,19 °C und die größte Abweichung bei kochendem Wasser bei ±0,69 °C liegt. Diese Daten werden für die Kalibrierung der Messungen verwendet und mit Hilfe des Python Programms in die Datenauswertung miteingerechnet. Das Zeitintervall für die Temperaturen liegt bei etwa 10 Sekunden.

4.4 Nutzbare Messungen

Die gesamten Messungen fanden zwischen dem 29.05.19 und dem 30.07.19 statt. Aufgrund von Ausfällen können für diese Arbeit die Messungen zwischen dem 12.06.19 und dem 19.06.19 herangezogen werden. Die Gründe dafür werden im Folgenden erläutert.

4.4.1 Nutzbare Temperaturmessungen

Die Temperaturen werden an vier verschiedenen Stellen mit den zwei Flexim Geräten und mit dem Raspberry Pi an insgesamt zehn Stellen gemessen. Um einen Vergleich der Messgenauigkeit der beiden Geräte zu haben werden an den vier Messstellen der Flexim Geräte auch gleichzeitig die Temperaturen durch die Raspberry Pi Fühler aufgezeichnet. Vor allem Probleme bei der Aufzeichnung der Raspberry Pi Daten führen zu unbrauchbaren Zeiträumen. Wie Abbildung 4-4 zeigt gibt es Ausfälle zwischen dem

- 30.05 und 04.06
- 06.06 und 12.06
- 19.06 und 24.06
- 24.06 und 03.07
- 03.07 und 08.07
- 22.07 und 26.07

Durch die fehlenden Daten der Raspberry Pi Temperaturfühler sind nur 25 Tage der Messungen nutzbar. Die Flexim Geräte sind dagegen sehr ausfallsicher und die Daten nahezu vollständig. Nur zwischen dem 22.07 und dem 26.07 wurden keine Daten aufgezeichnet.

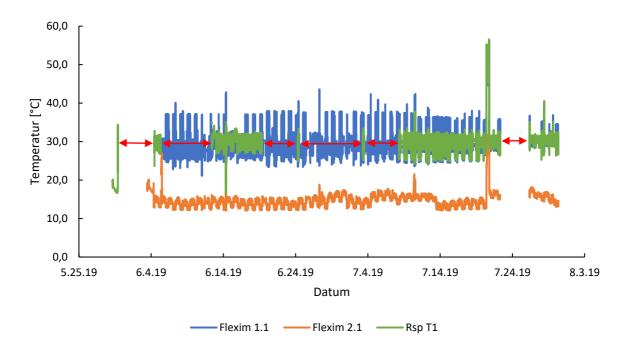


Abbildung 4-4: Temperaturdaten für die Bestimmung nutzbarer Messzeiträume.

Hier beispielhaft Flexim 1.1, Flexim 2.1 und PiT1. Rote Pfeile zeigen Problemstellen

4.4.2 Nutzbare Leistungsmessung

Der zweite Grund warum in den zwei letzten Drittel der Messperiode die Messwerte nicht genutzt werden können ist, dass das Messgerät Fluke 3 an der Kältemaschine ab dem 28.06.19 nur noch geringere Leistungen aufgezeichnet hat, obwohl die Außentemperatur in dieser Zeit höher war und somit die Leistung der Kälteanlage auch höher ausfallen müsste. Ab dem 12.07.19 zeigt das Fluke 3 Messgerät gar keine Leistung mehr an. Abbildung 4-5 zeigt die zwei Zeiträume der inkonsistenten Daten mit roten Pfeilen. Für das hier auftretende Problem kann kein Fehler ausfindig gemacht werden, wahrscheinlich ist jedoch, dass das Gerät verstellt worden ist, da sowohl Strom als auch Spannung weiter aufgezeichnet wurden aber die Phasenverschiebung $\cos \varphi$ fehlt.

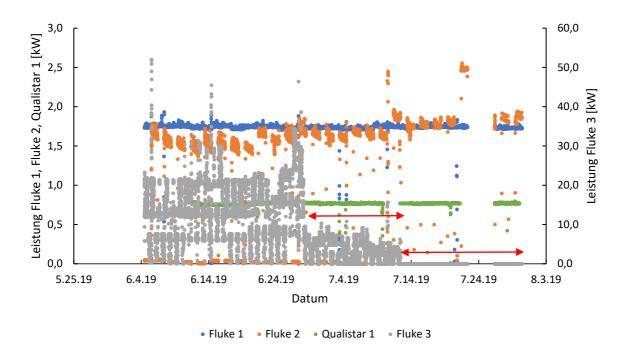


Abbildung 4-5: Leistungsdaten für die Bestimmung nutzbarer Messzeiträume.

Rote Pfeile zeigen Problemstellen

Die beiden anderen Fluke Geräte zeichnen über den gesamten Messzeitraum plausible Daten auf, nur zwischen dem 22.07 und dem 26.07 gab es bei beiden einen Ausfall. Das Fluke 2 Gerät hat über den gesamten Zeitraum nur einer der beiden Donaupumpen gemessen. Da die beiden verbauten Pumpen baugleich sind, kann davon ausgegangen werden, dass sie bei gleichem Durchfluss in etwa die gleiche Leistung haben. Fluke 2 und auch das PCE Messgerät zeigen über den gesamten Zeitraum in etwa gleichbleibende Werte an, daher wird in dem Zeitraum, indem die zweite (ungemessene) Donauwasserpumpe P1 läuft, der Mittelwert von 1,5 kW aus den Daten des Fluke 2 verwendet.

Das Strommessgerät Qualistar 1 hat ebenfalls immer wieder längere Ausfälle. So gibt es zwischen dem 23.06 und 24.06, dem 26.06 und 28.06, 02.07 und 03.07, 21.07 und 26.07 immer wieder Ausfälle. Ansonsten scheinen die Daten plausibel und relativ konstant zu sein. Dagegen hat Qualistar 2 über die ganze Messperiode keine brauchbaren Daten der Pumpe P5 aufgezeichnet und wird für diese Arbeit mit Hilfe der Durchflussmessung aus dem Datenblatt entnommen [27]. Die Pumpe P5 versorgt die Kälteabnehmer mit Kaltwasser, daher schwankt der Durchfluss im Tagesverlauf. Tagsüber liegt der Durchfluss bei etwa 22m³/h und nachts bei etwa 14m³/h. Somit ergeben sich zwei verschiedene Leistungsstufen, bei denen die Pumpe betrieben wird.

Laut Datenblatt (Abbildung 4-6) werden tagsüber 0,9 kW verbraucht und nachts etwa 0,8 kW.

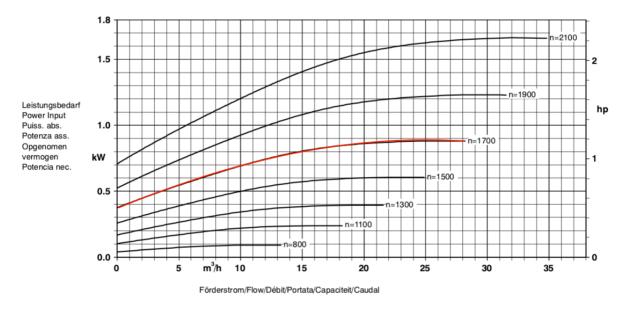


Abbildung 4-6: Leistungsbedarf der Pumpe der Baureihe und Typs Etaline PumpDrive 40-160/114. In rot ist die benötigte Kennlinie eingezeichnet [27].

4.4.3 Nutzbare Durchflussmessung

Weitere Gründe für den kurzen brauchbaren Zeitraum sind die abfallenden bzw. nicht aufgenommen Messqualitäten der PCE-Messgeräte. Die Messqualität wird von den Geräten selbst gemessen und zeigt an wie gut das Gerät misst. Sie wurde erst ab dem 19.06 aufgezeichnet und ist bis dahin unbekannt. Da aber die Durchflüsse beider PCE Geräte über die sieben Tage sehr konstant bleiben, wird die Qualität in diesem Zeitraum als brauchbar angenommen. Die Durchflussmessungen der beiden Flexim Geräte sind ebenfalls über den gesamten Zeitraum periodisch gleich und plausibel. Abbildung 4-7 zeigt die Problemzeiträume der Messdaten mit roten Pfeilen.

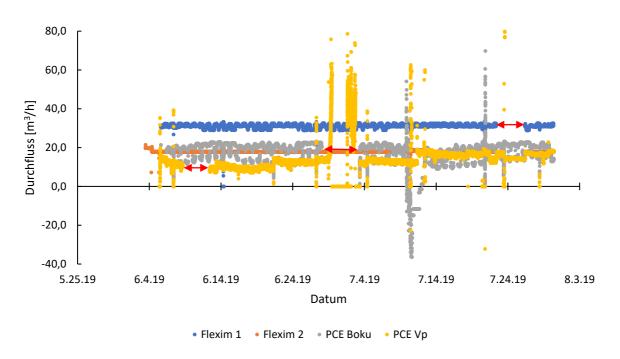


Abbildung 4-7: Durchflussdaten für die Bestimmung nutzbarer Messzeiträume.
Rote Pfeile zeigen Problemstellen

4.5 Datenauswertung mit Python

Zur Auswertung der Daten wird ein Python Skript (Python 3) verwendet. Python ist eine Programmiersprache, die entwickelt wurde, um möglichst knapp und übersichtlich Programme zu schreiben. Neben der bereits integrierten Standardbibliothek werden für die hier benötigten Anwendungen die "pandas" Bibliothek und die "bokeh" Bibliothek verwendet. Mit "pandas" ist es möglich Daten mit Hilfe von Tabellen und Zeitserien zu manipulieren und zu analysieren [28]. Die "bokeh" Bibliothek bietet viele Funktionen, um die Daten mit Hilfe eines Webbrowsers grafisch darzustellen [29]. Das für diese Arbeit entwickelte Programm erfüllt verschiedene Funktionen, die im weiteren Verlauf genauer erläutert werden.

4.5.1 Daten einlesen

Mit Hilfe der "pandas" Bibliothek werden die Daten aus den verschiedenen Messgeräten eingelesen. Die Raspberry Pi Fühler werden mit Hilfe der Arbeit von M. Rapport [26] auf die Temperaturabweichungen bei 0 °C kalibriert (siehe Tabelle 4-2), d.h. die gemessenen Temperaturen werden um den jeweiligen Kalibrierungswert korrigiert. Da nicht alle Messgeräte im gleichen Intervall messen, werden die Daten zusätzlich auf die jeweils volle Minute gemittelt, um einen übereinstimmenden Zeitstempel zu erhalten.

Tabelle 4-2: Gemessene Werte der Raspberry Pi Fühler im Eiswasser [26]

Sensornummer	Temperatur
	[°C]
PiT1	-0,125
PiT2	-0,437
PiT3	0,062
PiT4	-0,187
PiT5	-0,187
PiT6	-0,437
PiT7	0,062
PiT8	-0,375
PiT9	0,125
PiT11	-0,187
PiT12	-0,312

4.5.2 Anpassung des Zeitstempels der Raspberry Pi Fühler

Der Zeitstempel (in den Messgeräten), also die Uhrzeit und das Datum an denen die Daten der Messgeräte aufgezeichnet werden, wurden beim Start der Messungen nicht verglichen. Das führt zu einer nicht bekannten Zeitdifferenz zwischen den verschiedenen Messgeräten. Bekannt ist jedoch, dass die Uhrzeit zwischen den Geräten nicht mehr als 5 Minuten auseinander liegt. Deshalb wird mit Hilfe der kleinsten Fehlerquadrate die Zeit angepasst. Dafür wird die Temperaturdifferenz zwischen zwei gleichen Messzeitpunkten und gleicher Messstelle berechnet, quadriert und über den betrachteten Zeitraum summiert. Als Fixpunkt werden die Uhrzeiten der Flexim Geräte genutzt und die Daten aus den Raspberry Pi Temperaturen danach angepasst. Um den Zeitstempel der Raspberry Pi Temperaturen an die der Flexim Geräte anzupassen, wurde iterativ im Abstand von fünf Sekunden die Zeit soweit verschoben, bis die Fehlerquadratsumme aller vier verglichenen Temperaturen das Minimum erreicht hat. Da die beiden Messgeräte (Flexim und Raspberry Pi) an den vier Messstellen die gleiche Temperatur messen sollten, kann davon ausgegangen werden, dass bei einem Minimum der Fehlerquadratsumme die Übereinstimmung des Zeitstempels am höchsten ist. Ohne Zeitversschiebung der Raspberry Daten liegt die Fehlerquadratsumme bei 15.222. Nach einer Verschiebung um 85 Sekunden ist die Fehlerquadratsumme nur noch bei 4.314. Folglich wurde der Datensatz mit dem angepassten Zeitstempel für weitere Berechnungen benutzt. Tabelle 4-3 zeigt die Fehlerquadratsumme zwischen den Flexim und den Raspberry Pi Temperaturen.

Tabelle 4-3: Vergleich der Fehlerquadratsumme vor und nach dem Anpassen des Zeitstempels der Raspberry Pi Temperaturmessungen

Messwerte	Fehlerquadratsumme	Fehlerquadratsumme
	0 Sekunden	85 Sekunden
T _{Flx1.1} zu T _{Pi2}	6.062	1.558
T _{Flx1.2} zu T _{Pi5}	7.042	2.183
$T_{\text{Flx2.1}} \ zu \ T_{\text{Pi6}}$	829	389
T _{Flx2.2} zu T _{Pi3}	1.288	183
Summe	15.222	4.314

4.5.3 Berechnungen der Wärmeströme

Mit Hilfe des Python Programms werden insgesamt vier Wärmeströme berechnet. Die zwei verwendeten Flexim Geräte berechnen den jeweiligen Wärmestrom $\dot{Q}_{ab,K\ddot{u}W,Flx1}$ und $\dot{Q}_{zu,KaW,Flx2}$ direkt bei der Aufnahme der Messdaten nach den Gleichungen (4-3) und (4-4)

$$\dot{Q}_{ab,K\ddot{u}W,Flx1} = \dot{m}_{Flx1} \cdot c_{p} \cdot \rho \cdot (T_{Flx1.2} - T_{Flx1.1})$$
 (4-3)

$$\dot{Q}_{zu,KaW,Flx2} = \dot{m}_{Flx2} \cdot c_p \cdot \rho \cdot (T_{Flx2.1} - T_{Flx2.2}) \tag{4-4}$$

Mit der Gleichung (4-5) wird der Wärmestrom des Kühlwassers aus dem Massenstrom des Flexim 1 Gerätes und den Raspberry Temperaturen berechnet $(\dot{Q}_{ab,K\ddot{u}W,Rsp})$. Gleichung (4-6) zeigt die Berechnung des Wärmestroms auf der Kaltwasserseite mit Hilfe des Massenstroms aus dem Flexim 2 Messgerätes und den Temperaturen T_{Pi3} und T_{Pi6} $(\dot{Q}_{zu,KaW,Rsp})$. Gleichung (4-7) berechnet den Wärmestrom des Donauwassers aus den Temperaturfühlern T_{Pi11} und T_{Pi12} und dem Massenstrom des PCE_{vp} Gerätes $(\dot{Q}_{ab,DoW,Rsp})$ und Gleichung (4-8) zeigt die Berechnung des Wärmestroms der Abnehmerseite mit Hilfe der Messdaten T_{Pi7}, T_{Pi8} und dem Massenstrom des PCE_{boku} Gerätes $(\dot{Q}_{zu,Abn,Rsp})$.

$$\dot{Q}_{ab,K\ddot{u}W,Rsp} = \dot{m}_{Flexim 1} \cdot c_p \cdot \rho \cdot (T_{Pi5} - T_{Pi2}) \tag{4-5}$$

$$\dot{Q}_{zu,KaW,Rsp} = \dot{m}_{Flexim 2} \cdot c_p \cdot \rho \cdot (T_{Pi6} - T_{Pi3})$$
 (4-6)

$$\dot{Q}_{ab,DoW,Rsp} = \dot{m}_{PCEvp} \cdot c_p \cdot \rho \cdot (T_{Pi12} - T_{Pi11}) \tag{4-7}$$

$$\dot{Q}_{zu,Abn,Rsp} = \dot{m}_{PCEboku} \cdot c_p \cdot \rho \cdot (T_{Pi8} - T_{Pi7})$$
 (4-8)

4.5.4 Berechnung der EER und EERc

Für die Berechnung der EERs in der statischen Betrachtung werden die oben berechneten Wärmeleistungen verwendet. Außerdem wird die elektrische Leistung der Kältemaschine und der Pumpen herangezogen. Für die Systemgrenze Kältemaschine ist nur die elektrische Leistung der Kältemaschine zu beachten (Gleichung (4-9)). Für die Systemgrenze Kälteerzeugung sind die Pumpen P1 bzw. P2, P3 und P4 von Bedeutung (siehe Gleichung (4-10)). Die elektrische Leistung für die Gesamtanlage berechnet sich aus der Summe der elektrischen Leistungen aller Pumpen und der Kältemaschine (Gleichung (4-11)).

$$P_{el.KM} = P_{el.Fl3} \tag{4-9}$$

$$P_{el,KE} = P_{el,KM} + P_{el,Fl1} + P_{el,Fl2} + P_{el,CA1}$$
 (4-10)

$$P_{el,Ges} = P_{el,Fl1} + P_{el,Fl2} + P_{el,Fl3} + P_{el,CA1} + P_{el,CA2}$$
 (4-11)

Mit Hilfe der Formeln (4-12), (4-13), (4-14), (4-15), (4-16) wird der EER für die Systemgrenzen Kältemaschine, Kälteanlage und Gesamtsystem mit den verschiedenen Messdaten berechnet.

$$EER_{KM,Flexim2} = \frac{\dot{Q}_{zu,KaW,Flexim2}}{P_{el,KM}}$$
 (4-12)

$$EER_{KM,Rsp} = \frac{\dot{Q}_{zu,KaW,Rsp}}{P_{el,KM}}$$
 (4-13)

$$EER_{KE,Flexim2} = \frac{\dot{Q}_{zu,KaW,Flexim2}}{P_{el,KE}}$$

$$\dot{Q}_{zu,KaW,Rsp}$$
(4-14)

$$EER_{KE,Rsp} = \frac{\dot{Q}_{zu,KaW,Rsp}}{P_{el,KE}}$$
 (4-15)

$$EER_{Ges,Rsp} = \frac{\dot{Q}_{zu,Abn,Rsp}}{P_{el,Ges}}$$
 (4-16)

Die theoretisch mögliche Leistungszahl EERc der höchste Kältemaschine wird Anhand der Auslegungstemperaturen des Herstellers berechnet. Laut Datenblatt [19] liegen diese am Verdampfer bei 7 °C bzw. bei 12 °C. Am Kondensator gibt der Hersteller die Auslegungstemperatur mit 30 °C und 35 °C an. Berechnet man die EERc mit den "Spitzen", sprich der Austrittstemperatur des Kaltwassers (7 °C) und der Austrittstemperatur des Kühlwassers (35 °C), erhält man eine niedrigere EERc, als wenn man mit dem Mittelwert aus Ein- und Austrittstemperatur rechnet. Allerdings wird der Gütegrad (nach Gleichung (2-14) dadurch höher. Dieses Problem der nicht konstanten Temperaturniveaus tritt bei iedem thermodynamischen **Prozess** auf. In der Theorie laufen Zustandsänderungen bei konstanten Bedingungen ab, in der Praxis allerdings nicht. Beim Carnot-Faktor wird aber von konstanten Temperaturen ausgegangen. Die Carnot-Leistungszahl aus den mittleren Auslegungstemperaturen (9,5 °C am Verdampfer, 32,5 °C am Kondensator) des Herstellers liegt somit bei 12,29 (4-17). Der Gütegrad aus den Daten des Herstellers ergibt dann 0,3 (4-18).

$$EERc_{Hersteller} = \frac{273,15 + 9,5}{(273,15 + 32,5) - (273,15 + 9,5)} = 12,29$$
 (4-17)

Gütegrad Kältemaschine Hersteller =
$$\frac{EER}{EERc} = \frac{3,67}{12,29} = 0,3$$
 (4-18)

4.6 IPSEpro

IPSEpro ist ein Programm der Firma SimTech zur Berechnung von Wärmebilanzen und zur Simulierung von Prozessen. Das Programm besteht aus mehreren Programm Modulen. Für diese Arbeit sind vor allem drei Module von Bedeutung, das PSE (Process Simulation Environment), das PSValidate und das MDK (Model Development Kit) [30]. Dabei arbeitet IPSEpro mit dem Newtonverfahren bzw. dem Newton-Raphson-Verfahren.

4.6.1 Process Simulation Environment (PSE)

Das PSE bietet einen Flowsheet-Editor zum Einrichten von Prozessmodellen. Der Benutzer wählt die erforderlichen Komponenten aus dem Bibliotheksmenü aus und ordnet sie entsprechend an. Alle Prozessdaten werden direkt in das Flowsheet eingegeben. Die Ergebnisse werden dann im Flowsheet angezeigt. Das PSE generiert am Ende eines Simulationslaufs automatisch Ausgabeprotokolle [30]. In dieser Oberfläche wird die Kälteanlage modelliert. Das Schema enthält vier Pumpen mit jeweils einem Motor, den Wärmeübertrager, das Drei-Wege Ventil, die Kältemaschine, ein einfaches Pufferspeichermodell, eine Wärmequelle, eine Kühlwasserquelle, eine Kühlwassersenke und zwei Konnektoren, die die Kreisläufe schließen (siehe Abbildung 4-8).

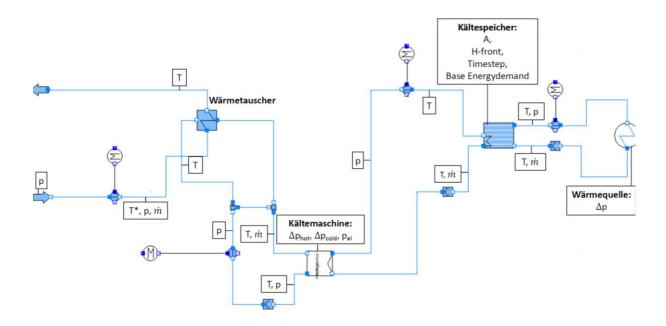


Abbildung 4-8: PSE Sheet der Kälteanlage. Mit * gekennzeichnete Temperatur wird im normalen Modus berechnet

4.6.2 Model Development Kit (MDK)

In der Programmierumgebung MDK, können neue Komponentenmodelle definiert werden, indem man ihnen einen Namen und eine entsprechende mathematische Gleichung zuweist. Dann werden diese in eine Form übersetzt, um im PSE verwendet werden zu können. Hierfür wurde bereits im Vorhinein von Magdalena Wolf ein entsprechendes Modell mit allen wichtigen Gleichungen für die Kältemaschine, den Mischer, den Pufferspeicher, den Wärmeübertrager und die Pumpen implementiert als auch die Grundlage für eine exergetische Betrachtung ergänzt. Ausführlich erklärt werden die Modelle in der Dissertation von Magdalena Wolf [31].

4.6.2.1 Kältemaschinenmodell

Das Modell vergleicht in Zeile 2 und 3 der Abbildung 4-9 eingehende und ausgehende Massenströme. In Zeile 6 und 7 werden die Druckverluste berechnet. Die Energiebilanz in den Zeilen 10 bis 13 werden aus den Enthalpien und den Massenströmen berechnet. Die Gleichung in Zeile 13 geht aus dem ersten Hauptsatz der Thermodynamik hervor und beschreibt die Energieerhaltung mit der Energie Q_{env} die als Verluste gezählt werden. Die Exergie wird dann in den Zeilen 16 und 17 berechnet. Die als COP bezeichneten Zeilen 20 und 21 werden in dieser Arbeit als EER bezeichnet. Die Zeilen 24 bis 27 beschreiben die Berechnung der Exergieeffizienz ζ_{KM} und den Exergieverlust.

```
1 #mass balance equations
2 f_mass_hot: feed_hot.mass = drain_hot.mass;
3 f mass cold: feed cold.mass = drain cold.mass;
5 # pressure drops
6 f_delta_p_hot: feed_hot.p - delta_p_hot = drain_hot.p;
7 f_delta_p_cold: feed_cold.p - delta_p_cold = drain_cold.p;
9 # energy balance
10 f_e_cold: feed_cold.mass * (drain_cold.h - feed_cold.h) = -1*Q_cool;
11 f_e_hot: feed_hot.mass * (drain_hot.h-feed_hot.h) = Q_heat;
                       Q_heat = Q_cool + P_el; #1st law
12 #f e total:
13 f e total: 0 = Q trans env + Q heat-Q cool-P el; #1st law
15 \# Exergy loss (Update by Magdalena Wolf, 21.02.2019)
16 fExergy_source: feed_cold.mass * (drain_cold.e - feed_cold.e) = E source;
17 fExergy_sink:
                          feed hot.mass * (drain hot.e - feed hot.e) = E sink;
19 # COP heating (vgl. Baehr Chapter 3.3.5)
20 fCOP1: COP_KM = abs(Q_cool)/P_el;
21 fCOP2: COP_KM = Zeta_KM/100*e_c_KM;
23 # Exergetic efficiency (vgl. Baehr Chapter 3.3.5)
24 fZeta: Zeta_KM = (1 - (E_loss/P_el))*100;
26 # Exergy loss (vgl. Baehr Chapter 3.3.5)
27 fExergy loss: E loss = P el - E source - E sink;
```

Abbildung 4-9: Kältemaschinenmodell im MDK von IPSEpro

4.6.2.2 Mischermodell

Das Mischermodell (Abbildung 4-10) berechnet zuerst die Massen- und Energiebilanzen (Zeile 1 bis 5). Dann werden in Zeile 8 bis 10 die Druckverluste überprüft. Die Berechnung der Exergieverluste setzt sich zusammen aus den beiden einströmenden Exergien und der ausströmenden Exergie. Am Schluss wird das Modell noch auf Randbedingungen getestet und, falls nötig ein Error ausgegeben.

```
1 # mass balances
2 f mass:
               feed1.mass + feed2.mass = drain.mass;
4 # energy balance
               feed1.h * feed1.mass + feed2.h * feed2.mass = drain.h * drain.mass;
5 f energy:
7 # pressure drop feed 1
8 f_p1: feed1.p - delta_p_1 = drain.p;
9 # pressure drop feed 2
               feed2.p - delta_p_2 = drain.p;
10 f_p2:
11
12 # Exergy loss (Update by Magdalena Wolf, 21.02.2019)
13 f_eLoss: E_loss = feed1.Exergy + feed2.Exergy - drain.Exergy;
15 # test for positive pressure drops
16 t1: test (delta_p_1>=0.0) warning "pressure drop delta_p_1 is negative";
17 t2: test (delta p 2>=0.0) warning "pressure drop delta p 2 is negative";
19 # test for positive mass flows
20 t3: test (feed1.mass >= 0.0) error "feed1 - mass flow is negative";
21 t4: test (feed2.mass >= 0.0) error "feed2 - mass flow is negative";
```

Abbildung 4-10: Mischermodell im MDK von IPSEpro

4.6.2.3 Pufferspeichermodell

In Zeile 1 und 2 der Abbildung 4-11, werden die Massenbilanzen erstellt, das heißt es wird überprüft ob die einströmende Masse der ausströmenden Masse entspricht. Dann werden in Zeile 5 und 6 die Drücke überprüft. In den Zeilen 9 bis 16 wird die Energiebilanz des Pufferspeichers berechnet. Die letzten beiden Zeilen testen das System und geben bei Bedarf ein Error oder eine Warnung aus.

```
# Mass balance equations
           Feedsolar.mass = Drainsolar.mass;
Feedcold.mass = Drainhot.mass;
1 fmass1:
2 fmass2:
4 # Pressure drops
5 fp1: Drainsolar.p = Feedsolar.p;
6 fp2: Drainhot.p = Feedcold.p;
8 # Energy balance
9 fH front new: H front new = H front + ((Q change/Feedsolar.h) *Feedsolar.v/A);
10
    fQ_change: Q_change = (Feedsolar.mass * Feedsolar.h + Feedcold.mass*Feedcold.h
11
Drainhot.mass*Drainhot.h - Drainsolar.mass * Drainsolar.h)*Timestep/60 - Base Energydemand *
Timestep/(24*60);
12
13 fQ store: Q store = A * H front new * 1/Feedsolar.v * Feedsolar.h;
15 fQ in: Q in = Feedcold.mass * Feedcold.h - Feedcold.mass * Drainhot.h;
16 fQ_out: Q_out = Feedsolar.mass * Feedsolar.h - Feedsolar.mass * Drainsolar.h;
17
18 # Test Storage Capacity
19 tH front: test(H front new <= (Volume/A))
                                                    warning "W Buffer is full";
20 t dt cold: test(Feedsolar.t < Drainhot.t)
                                                    error "Temperature crossover cold";
```

Abbildung 4-11: Pufferspeichermodell im MDK von IPSEpro

4.6.2.4 Wärmeübertrager

In den ersten sechs Zeilen der Abbildung 4-12 werden die Massenbilanzen aufgestellt und in Zeile 9 und 10 die Druckverluste berechnet. Die Energiebilanz aus Zeile 13 und 14 gibt den Wärmestrom aus, der durch den Wärmeübertrager läuft. Der Exergieverlust wird in Zeile 17 beschrieben. Die ein und ausströmenden Temperaturen werden in den Zeilen 22 bis 30 berechnet, anhängig davon ob es sich um einen Gegenstrom- oder Gleichstromwärmeübertrager handelt. In den Zeilen 35 bis 40 wird die mittlere Temperaturdifferenz berechnet, abhängig von den ein- und ausströmenden Temperaturen. Die letzten drei Zeilen geben Randbedingungen vor und bei Nichteinhaltung ein Error aus.

```
1 # mass balance equations
2 # hot side
3 f mass hot:
                      feed hot.mass = drain hot.mass;
6 f_mass_cold: feed cold.mass = drain cold.mass;
8 # pressure drops
                      feed_cold.p - delta_p_cold = drain_cold.p;
feed_hot.p - delta_p_hot = drain_hot.p;
9 f delta p cold:
10 f delta_p_hot:
11
12 # energy balance
13 f e hot: feed hot.mass * (feed hot.h - drain hot.h) * (1.0 - heat loss/100) = q trans;
14 f e cold:
               feed cold.mass * (drain cold.h - feed cold.h) = q trans;
1.5
16 # Exergy loss (Update by Magdalena Wolf, 08.05.2019)
17 f10: E_loss = feed_cold.Exergy + feed_hot.Exergy - (drain_hot.Exergy + drain_cold.Exergy);
19 # temperature differences
20 # They are differently defined for co and counter current heat exchangers.
2.1
22 ifl Type == cocurrent then
                           feed hot.t-dt in = feed cold.t;
       f dt in co:
2.4
       f_dt_out_co: drain_hot.t-dt_out = drain_cold.t;
25 endifl
27 ifl Type == counter current then
       f_dt_in_counter: drain_hot.t-dt_in = feed_cold.t;
29
       f_dt_out_counter:
                              feed_hot.t-dt_out = drain_cold.t;
30 endifl
32 # Mean temperature difference is the logarithmic temperature difference.
33 # For nearly equal inlet and outlet temperature differences, it is approximated
34 \# by the arithmetic mean (which is the asymptotic approximation).
35 f MTD: if abs(dt in/dt out) >=1.2 || abs(dt out/dt in) >=1.2 then
               MTD * ln(dt in/dt_out) = (dt_in-dt_out);
37
        else
38
               MTD = (dt in+dt out)/2.0;
40 f htc area: q trans = htc area * MTD;
41
42 # tests
43 t dt in:
               test(dt in>0.0)
                                       error "dt in <= 0.0";
43 t_dt_in: test(dt_in>0.0) error "dt_out <= 0.0";
44 t_dt_out: test(dt_out>0.0) error "dt_out <= 0.0";
45 t_q_trans: test(q_trans>0.0) error "q_trans <= 0.0";
```

Abbildung 4-12: Wärmeübertragermodell im MDK von IPSEpro

4.6.2.5 Pumpenmodell

Zeile 2 der Pumpenmodells (Abbildung 4-13) gibt den Massenstrom und Zeile 4 den Wirkungsgrad der Pumpen an. Der Exergieverlust der Pumpen wird in Zeile 7 definiert. Zwischen Zeile 9 und 23 werden, abhängig von der Nutzung der Pumpe, die Energiebilanzen aufgestellt. In den Zeilen 26 bis 29 werden Randbedingungen definiert und Errors bei Nichteinhaltung ausgegeben.

```
1 # mass balance
2 f_mass:
                feed.mass = drain.mass;
4 f eta p: feed.s = drain.fs(drain.p, feed.h+(drain.h - feed.h)*eta p,
0,0,0,0,0,0,0,0,0,0,0);
6 # Exergy Loss (Update by Magdalena Wolf, 21.02.2019)
7 fE loss: feed.Exergy + shaft in.power = E loss + drain.Exergy;
9 # both sides connected
10 ifl ref(shaft_in) && ref(shaft_out) then
                    (drain.h - feed.h)* feed.mass / eta_m = shaft_in.power
11
        f_powera:
shaft out.power;
12 endifl
13
14 # left side shaft only
15 ifl ref(shaft_in) && !ref(shaft_out) then
        f powerb:
                        (drain.h - feed.h) * feed.mass / eta m = shaft in.power;
17 endifl
18
19 # right side shaft only
20 ifl !ref(shaft in) && ref(shaft out) then
                      (drain.h - feed.h) * feed.mass / eta m = - shaft out.power;
        f_powerc:
22 endifl
23
24 # test conditions
25 t delta p: test((drain.p - feed.p) >= 0.0)
                                                          warning "outlet pressure lower than inlet
pressure";
pressure";
26 t_eta_p_low: test (eta_p >= 0.0) error "pump efficiency < 0.0";
27 t_eta_p_high: test (eta_p <= 1.0) error "pump efficiency > 1.0";
28 t_eta_m_low: test (eta_m >= 0.0) error "mechanical efficiency < 0.0";
29 t_eta_m_high: test (eta_m <= 1.0) error "mechanical efficiency > 1.0";
```

Abbildung 4-13: Pumpenmodell im MDK von IPSEpro

4.6.3 PSValidate

Das PSValidate Modul ist sowohl für die Erhöhung der Genauigkeit des Prozessmodells, die Anpassung der Komponentenmodelle an das erwartete Betriebsverhalten als auch für die Validierung der Ergebnisse geeignet. Bei der Verwendung von Messdaten ist zu beachten, dass die Messwerte niemals konsistent sind. Da Messungen immer fehlerbehaftet sind, genügen die Werte nicht den Modellgleichungen. Grundprinzipien wie die Erhaltung von Masse und Energie werden deshalb nicht erfüllt. Das PSValidate Module verwendet statistische Methoden, um Redundanzen zu beseitigen und genauere Daten zu erhalten [30]. Die quasidynamischen Simulation wird mit PSValidate mit einem Konfidenzintervall von 95 % durchgeführt.

4.6.4 Randbedingungen:

Randbedingungen müssen implementiert werden, um die möglichst fehlerfreie Berechnung in IPSEpro zu gewährleisten. Vor allem Bedingung (4-19) führt ohne Korrektur der Daten in der quasi-dynamischen Simulation zu erheblichen Fehlern, da die aufgezeichneten Temperaturen am Ausgang des Pufferspeichers zu den

Abnehmern oft niedriger liegen als am Eingang zum Pufferspeicher vom Kaltwasser und somit eine Error im Pufferspeichermodell hervorrufen. Die Randbedingungen wurden iterativ in IPSEpro getestet und mit Hilfe von Excel umgesetzt.

$$PiT7 = \begin{cases} PiT3 + 0.01 \text{ °C,} & PiT7 < PiT3 \\ PiT7, & PiT7 \ge PiT3 \end{cases}$$
 (4-19)

$$PiT2 = \begin{cases} Daten \ entfernt, & PiT2 < PiT4 \\ PiT2, & PiT2 \ge PiT4 \end{cases}$$
 (4-20)

$$PiT4 = \begin{cases} Daten \ entfernt, & PiT4 < PiT11 \\ PiT4, & PiT4 \ge PiT11 \end{cases}$$
 (4-21)

$$PiT5 = \begin{cases} Daten \ entfernt, & PiT5 < PiT2 \\ PiT5, & PiT5 \ge PiT2 \end{cases}$$
 (4-22)

Fluke3 =
$$\begin{cases} Daten entfernt, & Fluke3 < 0 \\ Fluke3, & Fluke3 \ge 0 \end{cases}$$
 (4-23)

So besagt Randbedingung (4-19), dass die Temperatur des Kaltwasservorlaufs (T_{pi3}) größer sein muss als die Temperatur des Abnehmervorlaufs (T_{Pi7}). Dies muss erfüllt sein, da sonst die thermodynamischen Bedingungen nicht erfüllt werden. Ist diese Bedingung nicht erfüllt, wird die Temperatur nach dem Kältespeicher angehoben indem man die Temperatur gleich der Temperatur des Kaltwassers vor dem Pufferspeicher setzt und 0,01 °C addiert.

Randbedingung (4-20) besagt, dass die Temperatur des Kühlwasservorlaufs direkt vor der Kältemaschine (T_{Pi2}) wärmer sein muss als die Temperatur des Kühlwasservorlaufs direkt nach dem Wärmeübertrager (T_{Pi4}). Ist diese Bedingung nicht erfüllt, werden die Daten entfernt. Diese Bedingung muss erfüllt sein, da zwischen diesen beiden Messpunkten ein Mischventil erwärmtes Kühlwasser aus der Kältemaschine mit dem Strang mischt und somit T_{Pi2} höher als T_{Pi4} liegen muss. Bei Randbedingung (4-21) wird überprüft, ob das austretende Kühlwasser aus der Kältemaschine größer gleich der eintretenden Temperatur ist. Ist diese Bedingung nicht erfüllt werden die entsprechenden Daten entfernt. Auch Randbedingung (4-22) besagt, dass bei einer niedrigeren Kühlwasser Vorlauf Temperatur (T_{Pi2}) im Vergleich zur Kühlwasser Rücklauftemperatur (T_{Pi5}) die aus der Kältemaschine kommt, die Daten entfernt werden. Ist die Bedingung nicht erfüllt, würde die Kältemaschine als Wärmepumpe laufen.

Randbedingung (4-23) muss eingeführt werden, da das Fluke 3 Gerät bei einer Leistung von 0 kW, also wenn die Kältemaschine nicht läuft, einen kleineren Wert als

0 kW misst. Dies führt zu unlogischen Folgeberechnungen und deshalb werden die Daten <0 kW aus den Gesamtdaten entfernt.

4.6.5 Daten in IPSEpro

Die gemessenen Daten aus den 7 Tagen werden entweder für die quasi-dynamische-Simulation über MatLab an IPSEpro übergeben oder entsprechend Tabelle 4-8 über gemittelte Daten für die statische Simulation per Hand in IPSEpro eingegeben. Bei der quasi-dynamischen-Berechnung erhält man für jeden minütlich aufgenommen Datenpunkt ein Ergebnis. Bei der Berechnung der statischen Daten werden über die Zeit die Daten gemittelt und dann in IPSEpro implementiert. In Tabelle 4-4 sind alle Messfühler aufgeführt, die zur Simulation der statischen Betrachtung und der Simulation im normalen IPSEpro Modus genutzt werden. Alle Werte werden dabei als "set" Wert implementiert.

Tabelle 4-4: Messwerte für die statische Simulation und den normalen Modus für die quasi dynamische Simulation, die an IPSEpro übergeben werden

Messwert	Daten
Temperatur	$T_{Pi2},T_{Pi3},T_{Pi4},T_{Pi5},T_{Pi6},T_{Pi7},T_{pi8},T_{Pi12}$
Durchfluss	$\dot{m}_{Flx1}, \dot{m}_{Flx2}, \dot{m}_{PCEvp}, \dot{m}_{PCEboku}$
Leistung	$P_{el,Fl3}$

Im Validierungsmodus der quasi-dynamischen Simulation können zusätzlich zu den Setzwerten ("set") noch gemessene (in IPSEpro als "measured" benannt) Werte implementiert werden. Für diese Simulation werden alle gemessenen Werte bis auf T_{Pi4} , T_{Pi7} , T_{Pi8} und der Massenstrom $\dot{m}_{PCEboku}$ als "measured" Wert implementiert, da diese sich aus vorrangegangenen Proberechnungen zur Einstellung des Modells nicht geändert haben. T_{Pi4} , T_{Pi7} , T_{Pi8} und $\dot{m}_{PCEboku}$ werden wie in Tabelle 4-5 zu sehen ist als "set" Wert gesetzt.

Tabelle 4-5: Implementierte Messwerte im Validierungsmodus der quasidynamsiche-Simulation

Messwert	"set" Daten	"measured" Daten
Temperatur	T_{Pi4},T_{Pi7},T_{Pi8}	T _{Pi2} , T _{Pi3} , T _{Pi5} , T _{Pi6} , T _{PiT11} T _{Pi12}
Durchfluss	$\dot{m}_{PCEboku}$	$\dot{m}_{Flx1}, \dot{m}_{Flx2}, \dot{m}_{PCEvp}$
Leistung	-	$P_{el,Fl3}$

Zusätzlich müssen bestimmte Werte, die nicht gemessen werden, in allen Simulationen als Setzwert implementiert werden. Diese sind über den gesamten Zeitraum konstant. Dazu gehören Volumen, Fläche, Höhe, Zeitschritte, Basisenergiebedarf des Kältespeichers, Druck in den Leitungen und die Druckverluste in der Kältemaschine und der Abnehmer. Tabelle 4-6 zeigt alle fest gesetzten Parameter.

Tabelle 4-6: Setzwerte, die in IPSEpro implementiert wurden. Mit * gekennzeichnete Größen werden nur im Validierungsmodus fest gesetzt.

Fest gesetzte Parameter	Setzwert
Druck Zulauf Donauwasser	2,0 bar
Druck Zulauf Donauwasser nach Pumpe	2,2 bar
Druck Kühlwasser nach Mischer	1,0 bar
Druck Zulauf Kühlwasser vor Kältemaschine	1,2 bar
Druckverlust Kaltwasserseite in Kältemaschine	0,1 bar
Druckverlust Warmwasserseite in Kältemaschine	0,2 bar
*Q _{env} Kältemaschine	0 kW
Fläche Kältespeicher	$0,5 \text{ m}^2$
Höhe Kältespeicher	1,6 m
Volumen Kältespeicher	700 I
Zeitschritt Kältespeicher	1 min
Basisenergiebedarf Kältespeicher	0 kWh/24 h
Druck Kaltwasserrücklauf Kältemaschine	2,2 bar
Druckverlust Abnehmer	0,2 bar
Druck Abnehmer	1,5 bar

4.6.6 Messfehler in IPSEpro

Im Validierungsmodus ist es zusätzlich möglich einen Messfehler für die als "measured" gesetzten Werte einzugeben. Dadurch kann man die Messungenauigkeiten der einzelnen Messgeräte in das Programm implementieren. Das hat den Vorteil, dass bei einer Eingabe ohne Messfehler im normalen Modus von IPSEpro, ein Wert der berechnet werden soll, bei ansonsten festgesetzten Werten die in der Realität einen Messfehler aufweisen, eine erhebliche Veränderung von beispielsweise mehreren Grad Celsius zur Folge hat. Im Validierungsmodus dagegen

werden alle als gemessen eingegebenen Werte innerhalb des Messfehlers korrigiert, das heißt es werden mehrere Werte leicht verändert (innerhalb des eingegebenen Messfehlers) anstatt ein Ergebnis stark zu verändern. Tabelle 4-7 zeigt alle eingegebenen Messfehler. Außerdem bringt der Validierungsmodus den Vorteil, alle gemessenen Werte eingeben zu können, um somit den vollen Umfang des Messaufbaus nutzen zu können. Das System wird also überbestimmt und lässt so eine statistische Auswertung zu, ob die Ergebnisse innerhalb des angegebenen Messfehlers bzw. innerhalb des Konfidenzintervalls liegen.

Tabelle 4-7: Messfehler die im Validierungsmodus in IPSEpro auf Grundlage der Datenblätter der Messgeräte implementiert wurden

Messwert	Messfehler
T _{Pi2} , T _{Pi3} , T _{Pi5} , T _{Pi6} , T _{PiT11} T _{Pi12}	0,5 °C
$P_{el,Fl3}$	0,5 kW
$\dot{m}_{Flx1}, \dot{m}_{Flx2}, \dot{m}_{PCEvp}$	0,3 kg/s

4.6.7 Statische Betrachtung

Aus den brauchbaren sieben Tagen werden mehrere Zeiträume herausgesucht, die einen möglichst stationären Verlauf aufweisen. Dies hat den Vorteil, dass die Temperatur sich einstellen kann und möglichst nahe am wahren Wert liegt. Diese Zeiträume werden dann jeweils gemittelt und in das IPSEpro Modell implementiert. Das statische Modell dient dazu die Berechnung für die dynamische Simulation einzustellen und einen möglichst geringen Fehleranteil zu erhalten. Außerdem liefern die über einen Zeitraum gemittelten konstanten Werte eine solide Information über statische Betriebszustände. In der statischen Berechnung werden die Temperaturen der Flexim Geräte verwendet, um die Validität der Raspberry Fühler zu überprüfen. Vor allem innerhalb der Systemgrenze Kältemaschine lässt es Schlüsse über die Genauigkeit zu, da hier entweder nur mit den Flexim Temperaturen oder nur mit den Raspberry Temperaturen gerechnet werden kann.

Insgesamt werden für die statische Berechnung die drei oben definierten Systemgrenzen untersucht. In der Systemgrenze Kältemaschine werden die Temperaturen $T_{Flx1.1}$, $T_{Flx1.2}$, $T_{Flx2.1}$, $T_{Flx2.2}$, T_{Pi2} , T_{Pi3} , T_{Pi6} , T_{Pi3} , die Durchflussmessung \dot{m}_{Flx1} und \dot{m}_{Flx2} sowie die Leistung der Kältemaschine $P_{el,Fl3}$ verwendet. In der Systemgrenze Kälteerzeugung die Temperaturen $T_{Flx1.1}$, $T_{Flx1.2}$, $T_{Flx2.1}$, $T_{Flx2.2}$, T_{Pi1} ,

 T_{Pi2} , T_{Pi5} , T_{Pi6} , T_{Pi3} , T_{Pi4} , T_{Pi11} , T_{Pi12} , die Durchflussmessung \dot{m}_{Flx1} , \dot{m}_{Flx2} und \dot{m}_{PCEvp} und die Leistungen $P_{el,Fl1}$, $P_{el,Fl2}$ und $P_{el,Fl3}$. In der Betrachtung der Gesamtanlage werden alle gemessenen Temperaturen, Massenströme und Leistungen (die Leistung der Pumpe 1 und der Pumpe 5 wird wie in Kapitel 4.4.2 verwendet) miteingerechnet. Abbildung 4-14 und Tabelle 4-8 zeigen die ausgesuchten statischen Zeiträume aus den sieben Tagen.

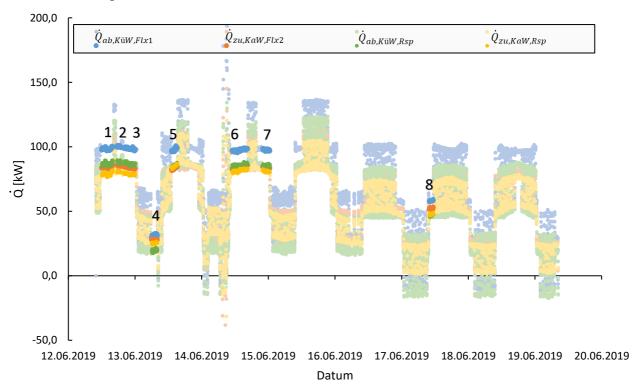


Abbildung 4-14: Die Messzeiträume für die statische Betrachtung sind hervorgehoben

Tabelle 4-8: Statische Messzeiträume

Bezeichnung	J Zeitraum
1	12.06.19 12:00 bis 12.06.19 16:00
2	12.06.19 17:30 bis 12.06.19 18:50
3	12.06.19 20:00 bis 13.06.19 00:00
4	13.06.19 06:00 bis 13.06.19 08:00
5	13.06.19 13:10 bis 13.06.19 15:10
6	14.06.19 11:00 bis 14.06.19 16:30
7	14.06.19 22:00 bis 15.06.19 00:30
8	17.06.19 10:00 bis 17.06.19 11:20

4.6.8 Quasi-dynamische Modellierung

IPSEpro bietet zwar die Möglichkeit einzelne Berechnungen einzugeben, aber in wenigen Schritten ganze Zeitreihen zu implementieren ist nicht möglich. Daher wird über eine COM Schnittstelle (Component Object Model), mit deren Hilfe IPSEpro mit MatLab kommuniziert, die Daten aus der untersuchten Zeitreihe an IPSEpro übergeben. Die COM Schnittstelle ermöglicht:

- das Öffnen, Schließen, Lösen von Berechnungen
- das Manipulieren von Daten
- Ändern von Werten (Parameter und Variablen)
- Ausgabe von Berechnungsergebnissen

Der dafür geschriebene MatLab Code (Skript ist im Anhang) muss mehrere Funktionen erfüllen. Mit dem Befehl "findObject" kann auf Objekte in IPSEpro zugegriffen werden, mit dem Befehl "findItem" auf den dort hinterlegten Parameter. Da die Berechnungen einmal ohne und einmal mit Validierungsmodus durchlaufen werden, kann entweder mit dem Befehl "setStatus" oder dem Befehl "setStatusMeasured" ein Wert für das jeweilige Item übergeben werden. Die Werte werden dabei in Form von Tabellen an MatLab übergeben. Alle nicht gesetzten Items nimmt IPSEpro als "estimate" an und werden berechnet.

Die Flexim-Temperaturfühler werden in der dynamischen Simulation nicht verwendet, um nur einheitlich mit Raspberry Pi Fühlern die Simulation durchzuführen und eine Verzerrung aufgrund größerer Differenzen zwischen den unterschiedlichen Messgeräten zu vermeiden.

5 Auswertung

Im folgenden Kapitel werden die Berechnungen aus dem Python Programm zusammen mit den Ergebnissen aus der statischen- sowie quasi-dynamischen-Simulation aufgearbeitet.

5.1 Temperaturmessung

Obwohl bereits die Raspberry Pi Fühler mit Hilfe der Diplomarbeit von M. Rapport [26] kalibriert wurden (siehe Tabelle 4-2), sind deutliche Differenzen zwischen $T_{Flx1.1}$ und T_{Pi2} , $T_{Flx1.2}$ und T_{Pi5} , $T_{Flx2.1}$ und T_{Pi6} , $T_{Flx2.2}$ und T_{Pi3} zu erkennen. Dabei liegt die Messgenauigkeit laut Datenblätter der Flexim Geräte bei $\pm 0,15$ °C und die der Raspberry Pi Temperaturfühler bei $\pm 0,5$ °C [23]. Daraus lässt sich schließen, dass höhere Differenzen als $\pm 0,65$ °C nicht aus den Messungenauigkeiten der Messgeräte abgeleitet werden können, sondern andere Ursachen haben muss.

Nachdem die Daten nach den Randbedingungen (Siehe Kapitel 4.6.4) bereinigt wurden, schwankt der Temperaturunterschied zwischen den Flexim und den Raspberry Pi Fühlern im Bereich zwischen -3,4 °C und 1,8 °C. In den betrachteten Zeiträumen, in denen die Temperaturen relativ stationär sind, sind die Differenzen allerdings nicht allzu groß und betragen nur noch maximal ±0,3 °C. Genauer betrachtet liegt der durchschnittliche Temperaturunterschied in den betrachtenden sieben Tagen zwischen $T_{Flx1.1}$ und T_{Pi2} bei 0,17 °C (maximal 1,5 °C, minimal -2,7 °C) und zwischen $T_{Flx1.2}$ und T_{Pi5} bei -0,16 °C (maximal 1,8 °C, minimal -3,4 °C). In Abbildung 5-1 und Abbildung 5-2 ist zu sehen, dass die höheren Differenzen in den Bereichen in denen die Temperatur um bis zu 6°C schwankt deutlich größer sind als bei gering dynamischen Temperaturverläufen. Etwas anders verhält sich der Vergleich zwischen dem Flexim 2 Gerät und den zugehörigen Temperaturen der Raspberry Pi Fühler. Abbildung 5-3 und Abbildung 5-4 zeigen, dass zwischen $T_{Flx2.1}$ und T_{Pi6} die Temperaturdifferenz nur noch bei maximal 0,4 °C und minimal -0,7 °C (im Durchschnitt bei -0,17 °C) und bei den Temperaturen $T_{Flx2.2}$ und T_{Pi3} bei maximal 0,4 °C und minimal -0,6 °C (im Mittel bei -0,03 °C) liegen. Die Temperaturen des Flexim 2 Gerätes liegen also im Vergleich mit den Temperaturen des Flexim 1 Gerätes näher an denen der Raspberry Temperaturen. Tabelle 5-1 fasst die Ergebnisse des Vergleichs zwischen den Flexim und Raspberry Temperaturen zusammen.

Tabelle 5-1: Temperaturunterschied zwischen den Flexim und den Raspberry Daten

Messwert	Min	Max	Mittel
$\overline{T_{Pi2} - T_{Flx1.1}}$	-2,66 °C	1,50 °C	0,17 °C
$T_{\text{Pi5}}\!-T_{\text{Flx1.2}}$	-3,41 °C	1,82 °C	-0,16 °C
$T_{\text{Pi6}} - T_{\text{Flx2.1}}$	-0,71 °C	0,39 °C	-0,17 °C
$T_{\text{Pi3}} - T_{\text{Flx2.2}}$	-0,58 °C	0,44 °C	-0,03 °C

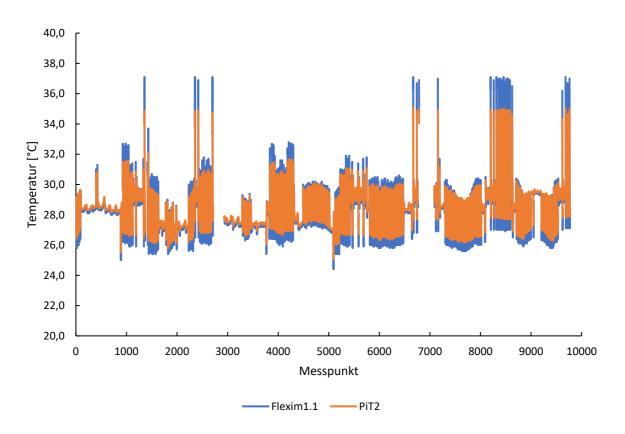


Abbildung 5-1: Temperaturverläufe von Flexim1.1 und PIT2

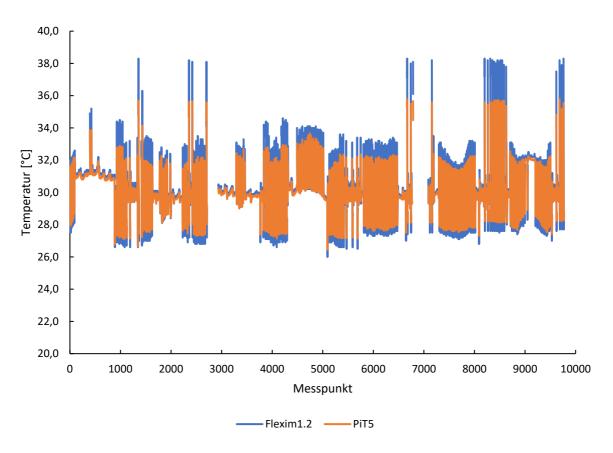


Abbildung 5-2: Temperaturverläufe von Flexim1.2 und PiT5

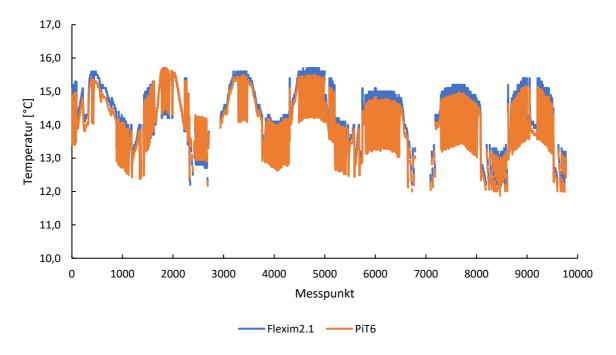


Abbildung 5-3: Temperaturverläufe von Flexim2.1 und PiT6

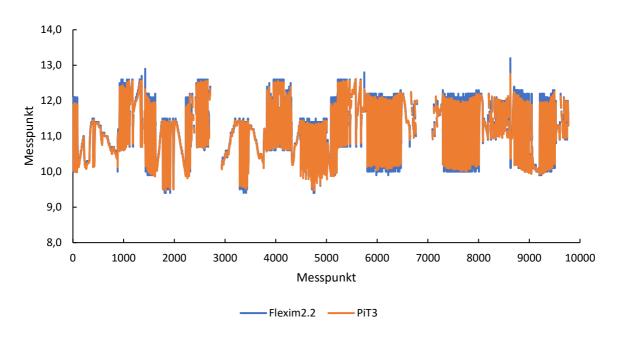


Abbildung 5-4: Temperaturverläufe von Flexim2.2 und PiT3

Diese Differenzen zwischen den Messgeräten können völlig unterschiedliche Ursachen haben. In Frage kommen Messungenauigkeiten, unterschiedliche Mengen Wärmeleitpaste, unterschiedliche Isolierung, Messfehler, unterschiedliche Genauigkeiten und die Ansprechzeit der Fühler. Besonders der extreme Unterschied in den dynamischen Temperaturverläufen lässt darauf schließen, dass die Messungen an den Raspberry Pi Fühlern bei einem schnellen Temperaturanstieg oder -abfall nicht schnell genug den Temperatursprung aufnehmen können und so die Spitzen "abgeschnitten" werden. Beispielhaft zeigt Abbildung 5-5, dass $T_{Flx1.1}$ ab 13:46 eine sinkende Temperatur gemessen hat, T_{Pi2} aber zeigt die Temperaturänderung verspätet, bzw. träger an und reagiert erst um 13:48 Uhr darauf. Die Flexim Geräte sind bei Temperaturschwankungen außerdem deutlich steiler. Dadurch entsteht ein "Kreuzen" der Temperaturverläufe und die Differenzen schwanken im Plus- und Minusbereich. Die Temperaturen aus den Flexim Geräten erscheinen plausibler, da sie stärkere Amplituden aufzeichnen, die nur aus wahren Temperaturen resultieren können. Somit entsprechen diese Messungen mit größerer Wahrscheinlich den echten Temperaturen. Allerdings werden auch Temperaturen an Messstellen nur mit den Raspberry Fühlern gemessen. Deshalb kann in dieser Arbeit nicht vollständig auf diese Daten verzichtet werden. Außerdem stellen die Messungen mit dem Raspberry Pi eine günstige Alternative zu den teuren Flexim Geräten da.

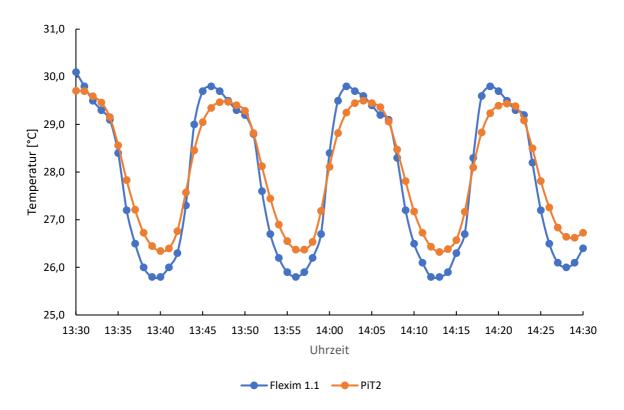


Abbildung 5-5: Ausschnitt aus dem Python Programm vom 16.06.19 13:30 Uhr bis 14:30 Uhr zum Vergleich der Ansprechzeit des PiT2 und Flexim 1.1 Fühler

Die Temperaturmessungen sind deshalb so wichtig, weil sie für die Berechnung der Wärmeströme benötigt werden. Sie fließen als Differenz zwischen Vor- und Rücklauf in die Berechnung mit ein. Generell lässt sich für die statische Berechnung festhalten, dass die Temperaturdifferenzen zwischen den Flexim 1 Vor- und Rücklauffühler im Mittel höher sind als die Differenzen zwischen den Raspberry Pi Fühlern. Dies liegt daran, dass die Temperatur $T_{Flx1.1}$ kälter als T_{PiT2} und $T_{Flx1.2}$ wärmer als T_{PiT5} ist. Somit ist die Spreizung bzw. die Temperaturdifferenz größer und letztlich auch der Wärmestrom $\dot{Q}_{zu,KM,Flx1}$ höher. Ähnlich verhält sich das Verhältnis der Flexim 2 zu den Raspberry Pi Temperaturen. Allerdings sind hier sowohl $T_{Flx2.1}$ als auch $T_{Flx2.2}$ höher als T_{PiT6} und T_{PiT3} , aber in einem Verhältnis, so dass die Temperatur $T_{Flx2.1}$ deutlich höher als T_{PiT6} ist und somit die Temperaturdifferenz des Flexim 2 Gerätes höher ausfällt als die Differenz der Raspberry Fühler.

5.2 Ergebnisse aus der statischen Simulation

5.2.1 Systemgrenze Kältemaschine

Die Ergebnisse zeigen, dass die Wärmeströme aus den Raspberry Berechnungen geringer ausfallen als die mit den Flexim Geräten. Dadurch resultiert ebenfalls ein kleinerer EER_{KM} bzw. COP_{KM}. Dies liegt an den im Vergleich höheren Temperaturdifferenzen.

$$dT_{ab,Rsp} < dT_{ab,Flx1}$$

führt zu

$$\dot{Q}_{ab,KM,Rsp} < \dot{Q}_{ab,KM,Flx1}$$

und schließlich zu

$$COP_{KM,Rsp} < COP_{KM,Flx1}$$

Und

$$dT_{zu,Rsp} < dT_{zu,Flx2}$$

führt zu

$$\dot{Q}_{\mathrm{zu,KM,Rsp}} < \dot{Q}_{zu,KM,Flx2}$$

Und schließlich zu

Die Daten aus Tabelle 5-2 zeigen genau diese Ergebnisse. In allen statisch betrachteten Zeiträumen, außer Zeitraum 5, liegen die Wärmeströme aus der Berechnung mit den Flexim Geräten höher als bei der Berechnung aus den Raspberry Temperaturen. Dabei ist der Unterschied zwischen $\dot{Q}_{ab,KM,Rsp}$ und $\dot{Q}_{ab,KM,Flx1}$ höher als bei dem Wärmestrom $\dot{Q}_{zu,KM,Rsp}$ und $\dot{Q}_{zu,KM,Flx2}$, was an der geringeren Temperaturdifferenzen zwischen Vor- und Rücklauf der Raspberry Fühler liegt. Zeitraum 5 zeigt ein etwas anderes Bild. Hier liegt der Wärmestrom aus den Raspberry Temperaturen $\dot{Q}_{zu,KM,Rsp}$ knapp höher als der Wärmestrom aus dem Flexim 2 Gerät $(\dot{Q}_{zu,KM,Flx2})$ und somit erhöht sich auch der EER_{KM,Rsp}.

Tabelle 5-2: Ergebnisse der Systemgrenze Kältemaschine

Zeitraum	P _{el,KM}	$\dot{\mathbf{Q}}_{ab,\mathrm{KM,Rsp}}$	$\dot{Q}_{zu, ext{KM,Rsp}}$	EER _{KM,Rsp}	$\dot{Q}_{ab,KM,Flx1}$	$\dot{Q}_{zu,KM,Flx2}$	EER _{KM,Flx2}
	[kW]	[kW]	[kW]	[-]	[kW]	[kW]	[-]
1	20,53	87,06	78,93	3,85	98,52	84,06	4,10
2	20,58	88,54	80,70	3,92	100,31	85,64	4,16
3	20,57	87,04	79,02	3,84	98,63	84,08	4,09
4	6,42	19,79	25,48	3,97	30,83	27,81	4,33
5	20,04	85,13	84,96	4,24	97,84	83,33	4,16
6	20,14	85,74	81,73	4,06	97,51	84,69	4,21
7	19,98	85,16	81,07	4,06	97,58	84,17	4,21
8	12,85	47,51	47,96	3,73	58,14	52,08	4,05

Vergleicht man die berechneten EER mit den Carnot-Leistungszahlen EERc (Temperaturhub aus den Mittelwerten der jeweiligen Temperaturen an Kaltwasserund Kühlwasserseite) erhält man einen Gütegrad zwischen 0,21 und 0,25 (Tabelle 5-3). Das bedeutet das die Kältemaschine in etwa 1/5 bis 1/4 der theoretisch möglichen Leistungszahlen schafft. Insgesamt liegt dieses Verhältnis bei den Berechnungen aus den Flexim Daten etwas höher als bei den Raspberry Pi Daten.

Tabelle 5-3: Temperaturhub, EERc und Gütegrad der Systemgrenze Kältemaschine berechnet aus den Raspberry Pi und den Flexim Temperaturdaten

	Temperaturhub		EER _{KM,Rsp}	Temperaturhul)	EER _{KM,Flx}
Zeitraum	Raspberry	$EERc_Rsp$	EERc _{Rsp}	Flexim	EERc _{Flx}	EERc _{Flx}
	[°C]	[-]	[-]	[°C]	[-]	[-]
1	17,30	16,51	0,23	17,15	16,66	0,25
2	16,57	17,28	0,23	16,44	17,44	0,24
3	16,97	16,84	0,23	16,83	16,99	0,24
4	16,85	16,97	0,23	16,74	17,09	0,25
5	15,85	18,03	0,24	15,82	18,06	0,23
6	15,92	17,97	0,23	15,82	18,08	0,23
7	16,16	17,67	0,23	16,06	17,79	0,24
8	16,37	17,48	0,21	16,26	17,60	0,23

5.2.2 Systemgrenze Kälteerzeugung

Bei der Systemgrenze Kälteerzeugung ist die benötigte Hilfsenergie höher als bei der Systemgrenze Kältemaschine, da nun auch die Pumpen P1 bzw. P2, P3 und P4 mit dazu gerechnet werden müssen. Da diese Pumpen relativ konstant laufen liegt der zusätzliche Leistungsaufwand konstant bei etwa 4 kW. Diese zusätzlich benötigte Hilfsenergie wirkt sich negativ auf den EER aus. Bei einer Wärmeleistung $\dot{Q}_{zu,Rsp}$ von etwa 80 kW liegt der EER_{KE,Rsp} ca. 0,66 niedriger als der EER_{KM,Rsp}. Bei der berechneten Wärmeleistung $\dot{Q}_{zu,Flx2}$ ist der EER_{KE,Flx2} bei einer Wärmeleistung von ca. 80 kW um etwa 0,69 niedriger. Dieser Unterschied wird umso extremer, umso niedriger der Wärmestrom \dot{Q}_{zu} wird da sich das Verhältnis von Wärmestrom zu Hilfsenergie ändert. So liegt der EER_{KE,Rsp} bei einem Wärmestrom von 25,48 kW (Zeitraum 4) nur noch bei 2,45 im Vergleich zu dem EER_{KM,Rsp} von 3,97. Es lässt sich also generell schließen, dass bei einer geringeren Wärmeleistung, aber einer zusätzlich benötigten Hilfsenergie aus den Pumpen, die Leistungszahl sinkt und somit ein Indikator für eine geringere Effizienz ist. Tabelle 5-4 zeigt die beschriebenen Ergebnisse.

Tabelle 5-4: Elektrische Leistung, Wärmestrom und EER der Systemgrenze Kälteerzeugung mit den Daten aus den Raspberry Pi, den Flexim und den Fluke Geräten

Zeitraum	P _{el,KE}	$\dot{Q}_{ab,DW,Rsp}$	$\dot{Q}_{zu,KM,Rsp}$	EER _{KE, Rsp}	$\dot{Q}_{zu,KM,Flx2}$	EER _{KE, Flx2}
Zeitraum	[kW]	[kW]	[kW]	[-]	[kW]	[-]
1	24,53	128,14	78,93	3,22	84,06	3,43
2	24,58	133,73	80,70	3,28	85,64	3,48
3	24,56	131,53	79,02	3,22	84,08	3,42
4	10,41	40,23	25,48	2,45	27,81	2,67
5	24,02	119,64	84,96	3,54	83,33	3,47
6	24,14	126,91	81,73	3,39	84,69	3,51
7	23,99	127,18	81,07	3,38	84,17	3,51
8	16,83	80,93	47,96	2,85	52,08	3,10

5.2.3 Systemgrenze 3 Gesamtanlage

Bei der Betrachtung der Gesamtanlage werden alle Pumpen und deren aufgenommene elektrische Leistung in die Berechnung der EER miteinbezogen. Da nur Pumpe 5 im Vergleich zu der Systemgrenze Kälteerzeugung noch zusätzlich

miteingerechnet werden liegt $P_{\text{el,Ges}}$ tagsüber um 0,9 kW und nachts um 0,8 kW höher als $P_{\text{el,KE}}$. Anders als bei der Betrachtung der Systemgrenze Kälteerzeugung wird jetzt der EER nicht mehr aus $\dot{Q}_{zu,Flx2}$ bzw. $\dot{Q}_{zu,Rsp}$ berechnet, sondern nur aus dem Wärmestrom $\dot{Q}_{zu,Abn,Rsp}$ der Abnehmerseite. Wie in Tabelle 5-5 zu sehen ist, liegen in allen acht betrachteten statischen Zeiträumen die EER_{Ges,Rsp} bei über 4, bzw. in den Zeiträumen in denen die Wärmeströme $\dot{Q}_{zu,Abn,Rsp}$ höher als 84 kW liegen sogar bei über 4,8. Auffallend ist bei der Betrachtung des Gesamtsystems, dass die Wärmeströme $\dot{Q}_{zu,Abn,Rsp}$ öfters über $\dot{Q}_{ab,DW,Rsp}$ liegen. Außerdem ist Gleichung (2-12) eindeutig nicht erfüllt was zu einer Verlustleistung $\dot{Q}_{env,Ges}$ bei der Betrachtung der Systemgrenze Gesamtanlage führt (Gleichung (5-1)). Die hohen EER für diese Systemgrenze im Vergleich zur Systemgrenze Kälteerzeugung, resultieren aus dem höheren Temperaturhub auf der Abnehmerseite bei einer vergleichsweise gering höheren Leistungsaufnahme $P_{\text{el,Ges}}$. Dies ist durch den Pufferspeicher zu erklären, der hier nicht weiter im Detail untersucht wird.

$$\dot{Q}_{env,Ges} = \dot{Q}_{zu,Abn,Rsp} + P_{el,Ges} - \dot{Q}_{ab,DW,Rsp}$$
 (5-1)

Tabelle 5-5: Ergebnisse für die Systemgrenze Gesamtanlage

Zeitraum	$P_{el,Ges}$	$\dot{Q}_{ab,DW,Rsp}$	$\dot{Q}_{zu,Abn,Rsp}$	EER _{Ges,Rsp}	$\dot{Q}_{env,Ges}$
Lomann	[kW]	[kW]	[kW]	[-]	[kW]
1	25,43	128,14	122,64	4,82	19,93
2	25,48	133,73	122,65	4,81	14,4
3	25,46	131,53	121,78	4,78	15,71
4	11,21	40,23	47,58	4,24	18,56
5	24,92	119,64	122,79	4,93	28,07
6	25,04	126,91	120,76	4,82	18,89
7	24,89	127,18	120,46	4,84	18,17
8	17,63	80,93	84,51	4,79	21,21

5.3 Ergebnisse aus der quasi-dynamischen-Simulation

Für die quasi-dynamischen Simulationen im normalen sowie im Validierungsmodus wurden die gemessenen Daten verwendet. Die Ergebnisse sind in Tabelle 5-7 zusammengefasst.

5.3.1 Normaler Modus

In der IPSE Simulation ohne Validierungsmodus werden, wie aus der Tabelle 4-4 und Tabelle 4-6 zu entnehmen ist, die Daten aus den Messgeräten implementiert. Die Ausnahme ist die Temperatur des PiT11 Fühler, die nicht eingegeben wurde, da sonst die Berechnung überbestimmt wäre und somit keine Simulation möglich wäre. Dies führt dazu, dass TPi11 ebenfalls berechnet werden muss. Die Ergebnisse aus der normalen IPSEpro quasi-dynamischen-Simulation zeigen eine deutliche Differenz zu den erhobenen Messdaten. Zu betrachten sind hier die Differenzen bei den Temperaturen T_{Pi1} , T_{Pi7} und T_{Pi11} sowie der Wärmestrom \dot{Q}_{env} , der durch die Berechnung des IPSEpro Programms aus der elektrischen Leistung Pel und den Wärmeströmen \dot{Q}_{zu} und \dot{Q}_{ab} (siehe Gleichung (5-2)) eine Differenz zu den erhobenen Messdaten zeigt. TPi11 zeigt in der normalen Simulation eine hohe prozentuale Differenz von +21,7 % bzw. im Mittel eine Abweichung von +3,74 °C im Vergleich zu den erhobenen Messdaten. Dies ist vor allem auf die Messungenauigkeiten aller anderen Raspberry Pi Messfühler zurückzuführen da das Programm, anstatt mehrere Temperaturen zu korrigieren, nur T_{Pi1} und T_{Pi11} korrigiert bzw. berechnet, wobei T_{Pi1} aus thermodynamischer Sicht in etwa die Gleiche Temperatur aufweisen muss wie TPi5 und somit der Fehler nur auf TPi11 Einfluss nimmt. TPi7 wird aufgrund der Randbedingung (4-19) korrigiert, und zwar im Schnitt um +0,14 °C.

$$\dot{Q}_{env,KM} = P_{el} + \dot{Q}_{zu} - \dot{Q}_{ab} \tag{5-2}$$

Tabelle 5-6: Differenzen im Mittel zwischen aufgenommenen Messdaten und den Ergebnissen aus dem normalen IPSEpro Modus

Messstelle	Mittelwert	Prozentuale Messgröß	
		Differenz	
PiT1	+0,05	+0,18 %	[°C]
PiT7	+0,14	+1,28 %	[°C]
PiT11	+3,74	+21,7 %	[°C]

Tabelle 5-7: Ergebnisse aus den quasi-dynamischen Berechnungen im Vergleich zu den gemessenen Daten

Management	Gemessen	Normaler	Validierungs-	Messgröße	
Messwert		Modus	Modus		
EER _{KM,Rsp}	3,96	3,97	4,42	[-]	
EERc _{KM,Rsp}	17,36	17,37	17,37	[-]	
Güte _{Rsp}	0,23	0,23	0,25	[-]	
EER _{KM,Flx}	4,14	-	-	[-]	
EERc _{KM,Flx}	17,51	-	-	[-]	
Güte _{Flx}	0,24	-	-	[-]	
Temperaturhub Rsp	16,45	16,45	16,45	[°C]	
$\dot{E}x_{v,KM}$	-	14,43	13,75	[kW]	
$\dot{E}x_{v,Mischer}$	-	0,36	0,41	[kW]	
$\dot{E}x_{v,W\ddot{\mathrm{u}}}$	-	0,81	1,75	[kW]	
$P_{\text{el},\text{KM}}$	18,54	18,54	18,55	[kW]	
T _{Pi1}	30,14	30,20	30,49	[°C]	
T _{Pi2}	28,01	28,01	27,69	[°C]	
T _{Pi3}	10,89	10,89	10,71	[°C]	
T _{Pi4}	24,99	24,99	24,99	[°C]	
T _{Pi5}	30,18	30,18	30,49	[°C]	
T _{Pi6}	14,39	14,39	14,57	[°C]	
T _{Pi7}	10,81	10,95	11,00	[°C]	
T _{Pi8}	15,30	15,30	15,30	[°C]	
T _{Pi11}	17,23	20,97	17,39	[°C]	
T _{Pi12}	28,09	28,09	27,93	[°C]	
$\dot{Q}_{zu, ext{KM}, ext{Rsp}}$	72,62	72,87	81,08	[kW]	
$\dot{Q}_{zu,KM,Flx2}$	75,87	-	-	[kW]	
$\dot{Q}_{ab,KM,Rsp}$	76,75	76,60	99,63	[kW]	
$\dot{Q}_{ab,KM,Flx1}$	88,24	-	-	[kW]	
$\dot{Q}_{env,Rsp}$	-	14,81	-	[kW]	
\dot{m}_{Flx1}	8,49	8,49	8,53	[kg/s]	
\dot{m}_{Flx2}	4,97	4,97	5,02	[kg/s]	
$\dot{m}_{PCEboku}$	6,00	6,00	6,00	[kg/s]	
\dot{m}_{PCEvp}	2,57	2,57	2,29	[kg/s]	

Im normalen Modus ergibt sich eine mittlere Leistungszahl von 3,97. Bei einem Temperaturhub von 16,45 °C erhält man eine Leistungszahl nach Carnot von 17,37. Die Güte der Kältemaschine liegt also bei der normalen Simulation in IPSEpro bei 0,23 bzw. bei 23%. Da nach dem ersten Hauptsatz der Thermodynamik die Energie in einem geschlossenen System konstant ist, wurde die Größe $\dot{Q}_{env,Rsp}$ nach Gleichung (5-2) eingeführt. Sie beschreibt die Wärmeleistung die theoretisch bei Einhaltung des ersten Hauptsatzes verloren geht, bzw. aufgrund der Messungenauigkeiten berechnet wird. Sie gibt somit einen Indikator für die Validität der Messdaten. Mit einem Mittelwert von 14,81 kW zeigt $\dot{Q}_{env,Rsp}$ eine eindeutige Messungenauigkeit an.

Tabelle 5-8: Ergebnis des Qenv aus der normalen IPSEpro Simulation

Messwert	Min	Max	Mittelwert	Messgröße
$\dot{Q}_{env,Rsp}$	-5,12	28,88	14,81	[kW]

5.3.2 Validierungsmodus

Für die Berechnung in IPSEpro im Validierungsmodus wurden die Messwerte wie in Tabelle 4-5 und die Setzwerte wie in Tabelle 4-6 implementiert. Die Korrekturen der Daten die IPSEpro aufgrund der eingegebenen Messtoleranz berechnet hat und die deshalb veränderten Ergebnisse werden genauer betrachtet.

5.3.2.1 Korrektur durch die Messtoleranz

Wie oben beschrieben werden die Messdaten innerhalb der eingegebenen Messtoleranz korrigiert (Tabelle 4-7). Die Auswertung der Ergebnisse aus dem Validierungsmodus im Vergleich zu den Originaldaten aus dem Python Programm werden in Tabelle 5-9 zusammengefasst. Sie zeigen unterschiedlich starke Veränderung der Messwerte innerhalb der eingegebenen Messtoleranzen. Jeweils die Paare von Zu- und Rücklauf werden gegengesetzt verschoben. Das bedeutet, dass TPi2 und TPi5, TPi3 und TPi6, TPi11 und TPi12 sich komplementär verhalten. Wird also z.B. der Wert von TPi2 nach oben korrigiert, dann wird der Wert von TPi5 um den gleichen Betrag nach unten korrigiert. Die Größte Verschiebung findet bei den Messfühlern TPi2 und TPi5 statt. Im Detail heißt das, um einen Mittelwert von ±0,32 °C bzw. einer Veränderung von -1,14 % bei TPi2 und +1,05 % bei TPi5 im Vergleich zu den erhobenen Messdaten. Da sich die Temperatur im Kühlwasserzulauf verringert und die im Kühlwasserrücklauf erhöht, wird die Spreizung ΔT größer und wirkt sich direkt mit einer Erhöhung des Wärmestroms aus. Genauso, aber weniger ausgeprägt verhält sich TPi3 und TPi6. Dabei liegen die Differenzen der Ergebnisse aus dem Validierungsmodus im

Vergleich zu den originalen Messdaten im Mittel bei $\pm 0,18$ °C. Das wirkt sich somit auch weniger stark auf den Wärmestrom \dot{Q}_{zu} aus. Die Korrektur der gemessenen Massenströme der Flexim Geräte ist dagegen im Vergleich deutlich geringer. \dot{m}_{Flx1} wird im Mittel nur um 0,04 kg/s ($\pm 0,45$ %), und \dot{m}_{Flx2} um 0,05 kg/s ($\pm 0,02$ %) korrigiert. Anders verhält sich die Korrektur der \dot{m}_{PCEvp} Daten. Die Veränderung beträgt hier im Mittel $\pm 0,28$ kg/s, was einer Veränderung von $\pm 1,07$ % im Vergleich zu den gemessenen Daten entspricht und die größte prozentuale Korrektur des IPSEpro Programms in Bezug auf die Messtoleranz darstellt.

Die Werte von den fest gesetzten Größen T_{Pi4} , T_{Pi8} und $\dot{m}_{PCEboku}$ werden erwartungsgemäß nicht korrigiert. Nur T_{Pi7} wird nach Randbedingung (4-19) entsprechend verändert. Dabei liegt die Veränderung des Wertes aus der Validierungssimulation im Vergleich zu den gemessenen Werten im Mittel bei 0,19 °C.

Tabelle 5-9: Mittlere Korrekturen der Messdaten im Validierungsmodus von IPSEpro zu den originalen Daten

Messwert	Mittelwert	Prozentuale	Messgröße	
	Korrektur	Korrektur		
T _{Pi1}	0,35	1,16 %	[°C]	
T _{Pi2}	-0,32	-1,14 %	[°C]	
T _{Pi3}	-0,18	-1,64 %	[°C]	
T_{Pi4}	0,00	0,00 %	[°C]	
T_{Pi5}	0,32	1,05 %	[°C]	
T _{Pi6}	0,18	1,24 %	[°C]	
T _{Pi7}	0,19	1,73 %	[°C]	
T _{Pi8}	0,00	0,00 %	[°C]	
T _{Pi11}	0,16	0,95 %	[°C]	
T _{Pi12}	-0,16	-0,58 %	[°C]	
\dot{m}_{Flx1}	0,04	0,45 %	[kg/s]	
\dot{m}_{Flx2}	0,05	1,02 %	[kg/s]	
$\dot{m}_{PCEboku}$	0,00	0,00 %	[kg/s]	
\dot{m}_{PCEvp}	-0,28	-11,07 %	[kg/s]	
$P_{el,Fl3}$	0,01	0,06 %	[kW]	

5.3.2.2 Ergebnisse aus dem Validierungsmodus

Aus den zum Teil hohen Korrekturen folgen starke Veränderungen bei der Berechnung der Wärmeleistungen. Die Kaltwasserwärmeleistung liegt laut Tabelle 5-7 nach der Korrektur der Temperaturen und des Durchflusses im Validierungsmodus durchschnittlich bei 81,08 kW im Vergleich zu 72,62 kW aus der Berechnung mit den originalen Messdaten, was einer Veränderung von +11,65 % entspricht (siehe Tabelle 5-10). Die Veränderung bei der Kühlwasserwärmeleistung ist noch höher und liegt bei +29,81 %. Dies lässt sich hauptsächlich durch die erhöhte Temperaturspreizung von T_{Pi2} zu T_{Pi5} erklären, da der Massenstrom aus dem Flexim1 Gerät sich kaum verändert (+0,45 % bzw. 0,04 kg/s, Tabelle 5-9). Der EER Unterschied aus der Validierungsberechnung zu den Berechnungen aus dem Python Programm liegt bei +11,6 %.

Tabelle 5-10: Differenzen der berechneten Wärmeströme und EERs im Vergleich zwischen gemessenen Daten aus den Raspberry Temperaturen und Ergebnissen aus dem Validierungsmodus

	Mittelwert	Prozentuale	Messgröße
	Differenz	Veränderung	
$\dot{Q}_{ab,Rsp}$	22,88	29,81 %	[kW]
$\dot{Q}_{zu,Rsp}$	8,46	11,65 %	[kW]
EER _{KM,Rsp}	0,46	11,6 %	[-]

Wie erwartet schlagen die ungenauen Messungen der Raspberry Pi Fühler stark in die Berechnung bzw. Simulation der Wärmeströme und somit auch der EER's ein. Im Vergleich dazu kann man die direkten Ergebnisse aus der Leistungsmessungen der Flexim Geräte im Vergleich mit den Ergebnissen aus dem Validierungsmodus betrachten (Tabelle 5-11). Die Spreizung der jeweiligen Zu- und Rücklauftemperaturen ist im Vergleich zu den Raspberry Pi Fühlern geringer und ähneln mehr den Berechnungen aus dem Validierungsmodus. So liegt die Kühlwasserwärmeleistung direkt aus dem Flexim1 bei durchschnittlich 88,24 kW und die aus dem Validierungsmodus mit den Raspberry Pi Temperaturen bei 99,63 kW was einen Unterschied von 11,39 kW ergibt, bzw. +12,9 %. Dies ist im Vergleich zu den 22,88 kW bzw. +29,81 % Differenz aus der Berechnung mit den Raspberry Pi Temperaturen um mehr als die Hälfte geringer. Auch die Differenz des Kaltwasserwärmestroms

zwischen dem Flexim 2 und der Simulation im Validierungsmodus ist mit +5,21 kW (6,87 %) im Vergleich zu +8,46 kW (11,65 %) deutlich geringer. Der berechnete EER mit den Daten aus dem Flexim 2 beträgt im Mittel 4,14 was 0,28 bzw. +6,73 % über dem aus der Validierungssimulation liegt. Somit liegen die Ergebnisse aus den Flexim Geräten deutlich näher bei denen aus dem Validierungsmodus als die Ergebnisse, die mit den Raspberry Pi Fühlern erzielt wurden.

Im Validierungsmodus ergibt sich eine mittlere Leistungszahl von 4,42. Mit einem Temperaturhub von 16,45 °C erhält man somit eine Leistungszahl nach Carnot von 17,37. Die Güte der Kältemaschine liegt mit 0,25 etwas höher als bei der normalen Simulation in IPSEpro, da die Leistungszahl um 0,45 höher ist.

Tabelle 5-11: Differenzen der berechneten Wärmeströme und EERs im Vergleich zwischen gemessenen Daten aus den Flexim Temperaturen und Ergebnissen aus dem Validierungsmodus

	Messgröße	Mittewert	Prozentuale	
		Differenz	Veränderung	
$\dot{Q}_{ab,{ m Flx}}$	[kW]	11,39	12,9 %	
$\dot{Q}_{zu,Flx}$	[kW]	5,21	6,87 %	
$EER_{KM,Flx}$	[-]	0,28	6,73 %	

Die Ergebnisse aus der Berechnung im Validierungsmodus zeigen eine schnell regelnde Kältemaschine. Das gewählte Intervall von einer Minute zeigt deutlich eine schnell wechselnde Temperatur und Leistung. In Abbildung 5-6 und Tabelle 5-12 sieht man die verschiedenen aufgenommenen Leistungsstufen der Daikin Kältemaschine. Von den insgesamt sechs Stufen, die die Maschine schalten kann, wurden in den betrachteten sieben Tagen nur vier verwendet. Stufe 1 hat eine elektrische Leistung von ca. 6,4 kW, Stufe 2 ca. 12,6 kW, Stufe 3 ca. 20,5 kW und Stufe 4 ca. 28,9 kW. Demnach sollte die Stufe 5 bei einem Stufenanstieg von 8 kW bei ca. 36 kW und Stufe 6 bei ca. 45 kW liegen. Das Leistungsblatt gibt dagegen eine Leistungsaufnahme von 50,7 kW an, das heißt bei sechs gleichen Stufen sollte jeweils eine Leistungsstufe bei 8,45 kW liegen. Stufe 1 hat einen Anteil an den gemessenen Daten von 19 %, Stufe zwei von 27 %, Stufe 3 von 46 % und Stufe 4 von 1 %. Somit lief die Kältemaschine in knapp der Hälfte der Zeit bei 20,5 kW. Betrachtet man den EER in den verschiedenen

Leistungsstufen fällt auf, dass der EER bei Stufe 1 mit 4,5 unter dem der Stufe 2 mit 4,66 liegt. Damit arbeitet die Anlage bei dem Vergleich des EER bei Stufe 2 (12,6 kW) am effizientesten.

Tabelle 5-12: Leistungsstufen der Kältemaschine nach Validierungsberechnung

Leistung	Kälteleistung	EER	Anteil
[kW]	[kW]	[-]	
6,4	29,9	4,5	19 %
12,6	58,57	4,66	27 %
20,5	87,94	4,3	46 %
28,9	108,67	3,76	1 %
	[kW] 6,4 12,6 20,5	[kW] [kW] 6,4 29,9 12,6 58,57 20,5 87,94	[kW] [kW] [-] 6,4 29,9 4,5 12,6 58,57 4,66 20,5 87,94 4,3

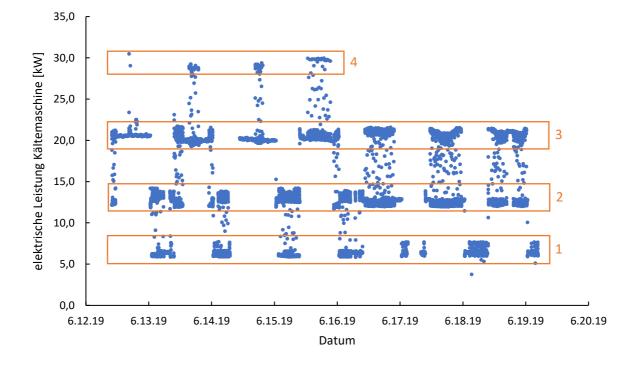


Abbildung 5-6: Gemessene elektrische Leistung der Kältemaschine mit den vier Leistungsstufen in orangen Kästen

5.3.3 Exergieverluste

Die Exergiebilanz der Gesamtanlage setzt sich aus den zugeführten Exergieströmen, den abgeführten Exergieströmen und den Exergieverlusten innerhalb des Systems zusammen. Die Gesamtbilanz wurde mit Hilfe von IPSEpro und den statischen Daten vom 12.06.19 12:00 Uhr bis 12.06.19 16:00 Uhr berechnet. Der Gesamtanlage wird über den Donaukanal 1,11 kW und über die Pumpen und der Kältemaschine ein

Gesamtexergiestrom von 21.37 kW zugeführt. Außerdem sind im Puffer 0,85 kW gespeichert. Daraus ergibt sich ein Gesamtexergiestrom im System von 23,33 kW. Aus dem Gesamtsystem werden 0,7 kW wieder zurück in den Donaukanal geleitet, 5,13 kW gelangen zu den Abnehmern und insgesamt 17,5 kW gehen als Exergieverlust an den Pumpen, dem Wärmetauscher, dem Mischer und der Kältemaschine verloren. Der austretende Gesamtexergiestrom beläuft sich somit auf 23,33 kW. In Abbildung 5-7 ist die Gesamtbilanz detailliert zu sehen. Der Exergetische Wirkungsgrad der Anlage liegt somit bei 22% (= 5,13 kW / 23,33 KW). Der größte Verlust entsteht erwartungsgemäß mit 15,07 kW bzw. einem Anteil von 86% an der Kältemaschine. Etwa 13% der Verluste entstehen am Wärmetauscher.

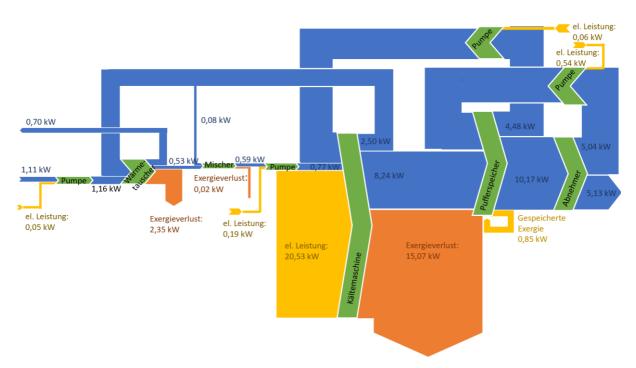


Abbildung 5-7 Exergetische Gesamtbilanz. Die Verluste der Pumpen sind so gering, dass sie in der Darstellung nicht abgebildet wurden

6 Diskussion

6.1 Messerfahrung

Laut der ersten Forschungsfrage wird ein kostengünstiges Messkonzept, das nicht in das laufende System eingreift, untersucht. Da aus zwei Monaten Messzeitraum nur knapp sieben Tage brauchbar waren muss das Messkonzept grundsätzlich überarbeitet werden.

6.1.1 Ausfälle der Messgeräte

Der Hauptgrund für den kurzen brauchbaren Messzeitraum ist vor allem der Ausfall der verschiedenen Messgeräte. Bei den Temperaturmessungen mit dem Raspberry Pi war vor allem das Programm an den fehlenden Daten schuld, da es immer wieder zu Ausfällen kam. Dieses sollte noch einmal überprüft werden, um Komplettausfälle zu vermeiden. Der Ausfall des Fluke 3 Geräts, das die Leistung der Kältemaschine messen sollte, kann nicht rekonstruiert werden. Es kann aber davon ausgegangen werden, dass die Einstellungen direkt am Messgerät verändert wurden und somit die Leistung bzw. die Phasenverschiebung nicht mehr mit aufgezeichnet wurde. Ein Generelles Problem bei der Aufzeichnung von vorübergehend angebrachten Messergeräten ist die Speichermöglichkeit. In unserem Fall konnten nur die Temperaturen der zehn Messfühler und die Daten aus den PCEs auf den Raspberry Pis gespeichert werden. Das heißt, hier konnten (fast) beliebig lange und niedrige Zeitintervalle gewählt werden, da hier genug Speicher vorhanden war. Vor allem aber die Fluke Geräte mit ihren kleinen Speichern und den vielen Messwerten mussten entweder oft ausgelesen oder auf relativ große Intervalle gestellt werden. Deshalb kam es zu vollen Speichern, was im Falle der Fluke Geräte dazu führte, dass alte Messwerte gelöscht wurden. Ausfälle von anderen Messgeräten können nur schwer nachvollzogen werden. Ein Versuch dieses Problem zu lösen wäre, sie in einem Labor davor zu testen und den Problemen auf den Grund zu gehen.

6.1.2 Ungenauigkeiten der Messung

Die zweite Forschungsfrage befasst sich mit der Validität der Messdaten. Vor allem die Temperaturmessungen haben bei dieser Arbeit Probleme gemacht. Es wurde auf zwei verschiedene Arten die Temperatur gemessen:

- Temperaturmessung mit dem Raspberry Pi
- Temperaturmessung mit den Flexim Geräten

Bei den Messungen mit dem Raspberry Pi wurden die zylindrischen Temperaturfühler mit Wärmeleitpaste versehen und dann zwischen die Isolierung der Rohre geklemmt. In diesem Fall spielt die Menge an Wärmeleitpaste, die Luft um den Messfühler, der Druck auf das Rohr und die Position des Fühlers eine große Rolle. Jede kleinste Veränderung der Position kann das Ergebnis deutlich verfälschen. Es ist also von großer Wichtigkeit alle baugleichen Messfühler mit möglichst gleichen Bedingungen anzubringen. Bei der Temperaturmessung mit den Flexim Geräten wurden die Oberflächenfühler mit Wärmeleitpaste versehen und fest an das Rohr gebunden. Da sie deutlich fester sitzen als die Fühler des Raspberry Pis sind sie weniger von Erschütterungen beeinflusst. Generell lässt sich sagen, dass es immer zu gewissen Ungenauigkeiten bei Temperaturmessungen kommen kann. Vergleicht man noch dazu Temperaturen von unterschiedlichen Messgeräten miteinander, kann es leicht passieren, dass diese sich deutlich unterscheiden.

Ein weiters Problem bei Clamp-on Fühlern ist, dass sie nur an das Rohr angebracht werden und nicht direkt im Medium im Rohr messen. Im Falle der Temperaturmessung wird also die Oberflächentemperatur der Rohre und nicht die Temperatur des Fluids im Rohr gemessen. Das führt dazu, dass die Wärmeleitfähigkeit der Rohre theoretisch in die Messung mit eingerechnet werden muss. Das zweite Problem bei der Verwendung von Clamp-on Fühlern ist, dass es bei niedrigen Temperaturen zur Kondensation oder sogar Eisbildung zwischen Rohrwand und Fühler kommen kann, was die Messung zusätzlich verfälscht.

Bei der Temperaturmessung zeigen mehrere Messdaten deutliche Unsicherheiten. Vergleicht man die Ergebnisse aus den Flexim Geräten, mit denen aus den Raspberry Daten ergeben sich Temperaturdifferenzen von bis zu 3,4 °C. Rechnet man mit diesen Differenzen die Wärmeleistungen und den EER aus, führt dies zu vollkommen verschiedenen Ergebnissen. So liegt die Wärmeleistung des Kühlwassers bei der Berechnung mit den Flexim Temperaturen bei 88,24 kW und bei der Verwendung der Raspberry Temperaturen bei 76,75 kW. Die Wärmeleistung des Kaltwassers liegt bei den Flexim Daten bei 75,87 kW und bei den Raspberry Daten bei 72,62 kW. Weitere Indikatoren für die nicht immer gegebene Validität ist das berechnete Q_{env} von 14,81 kW im normalen Simulationsmodus von IPSEpro.

Bei der Verwendung von Durchflussmessgeräten spielt die Wasserqualität und die Verschmutzung mit Partikeln im Rohr eine große Rolle. Da die in dieser Arbeit verwendeten Geräte mit Ultraschall den Durchfluss berechnen, können Verschmutzungen die Ergebnisse beeinflussen. Vor allem der Durchfluss des Donaukanalwassers führt in dieser Hinsicht zu Problemen, auch wenn das Wasser durch einen Filter läuft. Aus diesem Grund sollte in regelmäßigen Abständen die Wasserqualität und die Partikel im System überprüft werden und bei der Verwendung der Durchflussdaten mit einbezogen werden.

6.2 EER, EERc, Gütegrad, Temperaturhub

Mit Hilfe der dritten Forschungsfrage wird die Effizienz der Anlage untersucht. Vergleicht man nun die Ergebnisse aus dieser Arbeit mit den Herstellerangaben ergeben sich die folgenden Erkenntnisse. Zum einen gibt der Hersteller eine Leistungszahl von 3,67 bei einem mittleren Temperaturhub von 23 °C, einer Nennkühlleistung von 186 kW und einer Leistungsaufnahme von 50,7 kW unter Volllast an. Der EERc liegt bei diesen Angaben bei 12,29, was eine Kältemaschinengüte von 0,3 ergibt. Zum anderen ergeben sich bei der statischen Betrachtung in dieser Arbeit Leistungszahlen der Systemgrenze Kältemaschine von 3,96 (über alle acht Zeiträume gemittelt). Der Temperaturhub dieser acht Zeiträume liegt bei etwa 16,5 °C, also 6,5 °C unter dem Nenntemperaturhub des Herstellers. Außerdem liegt die Kühlleistung bei nur etwa 42% und die elektrische Leistungsaufnahme bei nur 40% der Herstellerangaben. Das bedeutet, dass die Kühlleistung im Verhältnis etwas höher liegt und somit den höheren EER erklärt. Bei Betrachtung der höheren Leistungszahl nach Carnot in den acht Zeiträumen (17,34) ergibt sich ein Gütegrad von 0,23. Im Vergleich zu den Herstellerangaben liegt dieser um 0,07 unter dem Gütegrad des Herstellers. Dieses Ergebnis lässt sich durch den hohen Unterschied im Temperaturhub und dem geringeren Unterschied der EER erklären. Genauso verhalten sich die Ergebnisse aus der normalen quasidynamischen-Simulation und der Simulation im Validierungsmodus. Der Temperaturhub, die aufgenommene elektrische Leistung und der EERc sind in beiden Simulationen gleich. Allerdings unterscheidet sich die Kühlleistung deutlich (normaler Modus: 72,87 kW, Validierungsmodus: 81,08 kW). Somit unterscheidet sich auch die Kältemaschinengüte mit 0,23 im normalen Modus und 0,25 im Validierungsmodus. In allen Berechnungen in dieser Arbeit liegt die Kältemaschinengüte somit unter der des Herstellers. Des Weiteren sind die hohen EER bei der statischen Betrachtung der

Systemgrenze "Gesamtanlage" (4,24 bis 4,93) mit Vorsicht zu betrachten. In den Berechnungen wurde der Pufferspeicher nicht genügend mit einbezogen und es kann somit zu Verzögerungen der Kälteleistung kommen, was zu einer Verzerrung der Ergebnisse führt.

6.3 Pythonprogramm

Das geschriebene Pythonprogramm zur Zusammenführung der Messdaten und der Erleichterung der Aufarbeitung der Messdaten kann in zukünftigen Projekten viele Arbeitsschritte abnehmen. Der Vorteil bei der Verwendung von Python ist, dass es so gut wie keine Grenzen für die Anwendung gibt und bei richtiger Programmierung in Zukunft nahtlos für ähnliche Projekte verwendet werden kann. Allerdings muss beachtet werden, dass bei einer Erhöhung der Funktionen, das Programm deutlich komplexer und unübersichtlicher wird.

6.4 IPSEpro

Mit Hilfe von IPSEpro konnten aus den gemessenen Daten zum einen die Exergieflüsse berechnet werden (Forschungsfrage 3) und zum anderen die Daten validiert werden (Forschungsfrage 2). Die Exergieverluste waren erwartungsgemäß bei der Kältemaschine am höchsten. Der Validierungsmodus zeigt im direkten Vergleich vor allem bei den berechneten Wärmeströmen große Differenzen zu den erhobenen Daten und auch zum normalen Modus. Viele Temperaturen wurden von dem Programm innerhalb der eingegebenen Messtoleranz verändert, was einen Hinweis auf die schlechte Genauigkeit der Raspberry Pi Temperaturen liefert. Im Vergleich zwischen den Temperaturen aus dem Validierungsmodus und den erhobenen Temperaturdaten aus den beiden Flexim Geräten sind die Unterschiede deutlich geringer. Daher liefert das Programm IPSEpro ebenfalls deutliche Hinweise darauf, dass die teuren Flexim Geräte genauer messen als der Aufbau mit dem Raspberry Pi. Allein die vom Hersteller der one-wire Temperatursensoren angegebene Messtoleranz von 0,5 °C im Vergleich zu der Messtoleranz der Flexim Temperaturmessung von <0,1 K gibt Aufschluss über die Genauigkeit der beiden Messarten.

7 Conclusio und Ausblick

Die Ergebnisse dieser Arbeit zeigen, dass der EER der Kältemaschine bei allen Berechnungen und Simulationen zwischen 3,73 und 4,42 liegt. Nur bei der Betrachtung der Systemgrenze Kälteerzeugung sinkt der EER deutlich unter 4. Die Leistungszahlen der Gesamtanlage in der statischen Betrachtung von weit über 4 haben allerdings aufgrund des wenig untersuchten Pufferspeichers nur wenig Aussagekraft. Ein gutes Bild der Effizienz liefert der berechnete Gütegrad der Kältemaschine. Dieser liegt in allen Berechnungen unter dem des Herstellers. Also auch wenn alle berechneten EER in dieser Arbeit höher liegen als vom Hersteller angegeben, erreicht die Kältemaschine nie den Gütegrad des Herstellers. Betrachtet man diese Ergebnisse allerdings genauer, und unter welchen Bedingungen sie zustande gekommen sind, muss die Validität der Daten stark in Frage gestellt werden. Die Daten zeigen einen hohen \dot{Q}_{env} oder unlogischen Kühl- und Kaltwasserströme, die zum Teil aufgrund von nicht messbaren Wärmeströmen an die Umgebung entstehen. Um die Ungenauigkeiten, die solche Fehler hervorrufen in Zukunft zu minimieren, muss vor allem der Messaufbau der Raspberry Pi Fühler geändert werden. Weitere Arbeiten könnten sich bei der Verwendung kostengünstiger Messaufbauten mit einem Raspberry Pi mit den Effekten von Eis- oder Kondensatbildung, Druck des Sensors auf das Rohr, unterschiedlicher Menge Wärmeleitpaste, verbesserter Isolierung des Messsensors, verunreinigtem Fluid im Rohr oder Ähnlichem beschäftigen. Dennoch muss erwähnt werden, dass bei verbesserter Anwendung das Potential von Einplatinencomputern und passenden Messgeräten bei der Überprüfung von ähnlichen Anlagen riesig ist. Sowohl der Preis als auch die unzähligen Möglichkeiten, die ein solches Setup in Kombination mit den passenden Programmen ermöglichen, können in Zukunft die schnelle Überprüfung von Kälteanlagen erheblich erleichtern.

Die Exergieverluste am Mischer und dem Wärmeübertrager halten sich generell in Grenzen. Nur der Exergieverlust an der Kältemaschine ist mit 15,07 kW relativ hoch. Bei der ersten Betrachtung der Daten fällt dazu sofort auf, dass die Anlage viel und stark regelt. Der Sprung von einer Leistungsstufe auf die nächst Höhere und wieder zurück passiert teilweise in wenigen Minuten mehrmals. Dieses Verhalten gilt als ineffizient und sollte nach Möglichkeit vermieden werden. Hervorzuheben ist dagegen, dass die Anlage in sechs verschiedenen Leistungsstufen betrieben werden kann und somit immer die möglichst beste Kälteleistung bereitstellen kann. Da die dritte

Leistungsstufe (20,5 kW elektrische Leistung, 87,94 kW Kälteleistung) in dem betrachteten Zeitraum zu 46 % der Zeit lief, sollte in zukünftigen Optimierungsmaßnahmen besonders auf diese Leistungsstufe Wert gelegt werden.

Die Verwendung von IPSEpro hat sich als sehr vorteilhaft bei ungenauen Messdaten herausgestellt. Die Daten der Raspberry Pi Sensoren konnten damit den Daten aus den Flexim Geräten angenähert werden und liefern so bessere Ergebnisse. In Zukunft ist bei Verwendung von günstigen, ungenaueren Messsensoren die Simulation mit IPSEpro im Validierungsmodus sehr zu empfehlen. In weiteren Arbeiten könnte man die statistischen Protokolle, die IPSEpro ausgibt genauer betrachten und statistische Signifikanzen mit auswerten.

8 Abbildungsverzeichnis

Abbildung 2-2: Schema einer Kälteanlage bei der die Wärme <i>Qab</i> die Summe aus der Hilfsenergie P und der Kühllast <i>Qzu</i> darstellt	Abbildung 2-1:	Anergie- und Exergiestrom bei einer Kältemaschine [12]	. 20
Abbildung 2-4: Zustandsänderung des Kältemittels einer Kompressionskältemaschine im In p, h-Diagramm (Seite 596) [12]			. 21
maschine im In p, h-Diagramm (Seite 596) [12]	Abbildung 2-3:	Schaltbild einer Kompressionskältemaschine Seite 596 [12]	. 22
verschiedenen Messpunkten, Pumpen, Kalt-, und Warmwasserleitungen und der Kältemaschine, eingeteilt in drei Systemgrenzen (Kältemaschine blau, Kälteerzeugung grün, Gesamtanlage gelb)			. 23
Abbildung 4-3: Schema des Kältemaschinenbetrieb der Kälteanlage	ve W Sy	rschiedenen Messpunkten, Pumpen, Kalt-, und armwasserleitungen und der Kältemaschine, eingeteilt in drei vstemgrenzen (Kältemaschine blau, Kälteerzeugung grün,	. 26
Abbildung 4-4: Temperaturdaten für die Bestimmung nutzbarer Messzeiträume. Hier beispielhaft Flexim 1.1, Flexim 2.1 und PiT1. Rote Pfeile zeigen Problemstellen	Abbildung 4-2:	Schema des Freecooling-Betriebs der Kälteanlage	. 27
Hier beispielhaft Flexim 1.1, Flexim 2.1 und PiT1. Rote Pfeile zeigen Problemstellen	Abbildung 4-3:	Schema des Kältemaschinenbetrieb der Kälteanlage	. 28
Rote Pfeile zeigen Problemstellen	Hi	er beispielhaft Flexim 1.1, Flexim 2.1 und PiT1. Rote Pfeile	. 33
PumpDrive 40-160/114. In rot ist die benötigte Kennlinie eingezeichnet [27]			. 34
Rote Pfeile zeigen Problemstellen	Pu	umpDrive 40-160/114. In rot ist die benötigte Kennlinie	. 35
wird im normalen Modus berechnet			. 36
Abbildung 4-10: Mischermodell im MDK von IPSEpro			. 42
Abbildung 4-11: Pufferspeichermodell im MDK von IPSEpro	Abbildung 4-9:	Kältemaschinenmodell im MDK von IPSEpro	. 43
Abbildung 4-12: Wärmeübertragermodell im MDK von IPSEpro	Abbildung 4-10:	Mischermodell im MDK von IPSEpro	. 44
Abbildung 4-13: Pumpenmodell im MDK von IPSEpro	Abbildung 4-11:	Pufferspeichermodell im MDK von IPSEpro	. 45
Abbildung 4-14: Die Messzeiträume für die statische Betrachtung sind hervorgehoben	Abbildung 4-12:	Wärmeübertragermodell im MDK von IPSEpro	. 46
hervorgehoben	Abbildung 4-13:	Pumpenmodell im MDK von IPSEpro	. 47
Abbildung 5-2: Temperaturverläufe von Flexim1.2 und PiT5	•		. 52
Abbildung 5-3: Temperaturverläufe von Flexim2.1 und PiT6	Abbildung 5-1:	Temperaturverläufe von Flexim1.1 und PIT2	. 55
Abbildung 5-4: Temperaturverläufe von Flexim2.2 und PiT3	Abbildung 5-2:	Temperaturverläufe von Flexim1.2 und PiT5	. 56
Abbildung 5-5: Ausschnitt aus dem Python Programm vom 16.06.19 13:30 Uhr bis 14:30 Uhr zum Vergleich der Ansprechzeit des PiT2 und Flexim	Abbildung 5-3:	Temperaturverläufe von Flexim2.1 und PiT6	. 56
bis 14:30 Uhr zum Vergleich der Ansprechzeit des PiT2 und Flexim	Abbildung 5-4:	Temperaturverläufe von Flexim2.2 und PiT3	. 57
1. I T UIIICI	bis	·	. 58

Abbildung 5-6: Gemessene elektrische Leistung der Kältemaschine mit den vier Leistungsstufen in orangen Kästen	
Abbildung 5-7 Exergetische Gesamtbilanz. Die Verluste der Pumpen sind so gering, dass sie in der Darstellung nicht abgebildet wurden	70

9 Tabellenverzeichnis

Tabelle 2-1:	Vergleich Kompressions-, Absorptions- und Adsorptions- kälteanlagen [16]	24
Tabelle 4-1:	Alle genutzten Messfühler mit der jeweiligen Messstelle, dem genutzten Messwert und der Bezeichnung	29
Tabelle 4-2:	Gemessene Werte der Raspberry Pi Fühler im Eiswasser [26]	37
Tabelle 4-3:	Vergleich der Fehlerquadratsumme vor und nach dem Anpassen des Zeitstempels der Raspberry Pi Temperaturmessungen	38
Tabelle 4-4:	Messwerte für die statische Simulation und den normalen Modus für die quasi dynamische Simulation, die an IPSEpro übergeben werden	49
Tabelle 4-5:	Implementierte Messwerte im Validierungsmodus der quasi- dynamsiche-Simulation	49
Tabelle 4-6:	Setzwerte, die in IPSEpro implementiert wurden. Mit * gekennzeichnete Größen werden nur im Validierungsmodus fest gesetzt.	50
Tabelle 4-7:	Messfehler die im Validierungsmodus in IPSEpro auf Grundlage der Datenblätter der Messgeräte implementiert wurden	51
Tabelle 4-8:	Statische Messzeiträume	52
Tabelle 5-1:	Temperaturunterschied zwischen den Flexim und den Raspberry Daten	55
Tabelle 5-2:	Ergebnisse der Systemgrenze Kältemaschine	60
Tabelle 5-3:	Temperaturhub, EERc und Gütegrad der Systemgrenze Kältemaschine berechnet aus den Raspberry Pi und den Flexim Temperaturdaten	60
Tabelle 5-4:	Elektrische Leistung, Wärmestrom und EER der Systemgrenze Kälteerzeugung mit den Daten aus den Raspberry Pi, den Flexim und den Fluke Geräten	61
Tabelle 5-5:	Ergebnisse für die Systemgrenze Gesamtanlage	62
Tabelle 5-6:	Differenzen im Mittel zwischen aufgenommenen Messdaten und den Ergebnissen aus dem normalen IPSEpro Modus	63
Tabelle 5-7:	Ergebnisse aus den quasi-dynamischen Berechnungen im Vergleich zu den gemessenen Daten	64
Tabelle 5-8:	Ergebnis des Q _{env} aus der normalen IPSEpro Simulation	65
Tabelle 5-9:	Mittlere Korrekturen der Messdaten im Validierungsmodus von IPSEpro zu den originalen Daten	66
Tabelle 5-10	:Differenzen der berechneten Wärmeströme und EERs im Vergleich zwischen gemessenen Daten aus den Raspberry Temperaturen und Ergebnissen aus dem Validierungsmodus	67
Tabelle 5-11	:Differenzen der berechneten Wärmeströme und EERs im Vergleich zwischen gemessenen Daten aus den Flexim Temperaturen und Ergebnissen aus dem Validierungsmodus	68

Tabelle 5-12:Leistungsstufen der Kältemaschine nach Validierungsberechnung 69

10 Literaturverzeichnis

- [1] World Meteorological Organization, "2018 Annual Report WMO for the Twenty-first Century," 2019.
- [2] V. Masson-Delmotte, H.-O. Pförtner, J. Skea, A. Pirani, R. Pidcock, Y. Chen, E. Lonnoy, T. Maycock, W. Moufouma-Okia, S. Connors, X. Zhou, M. Tignor, P. Zhai, D. Roberts, P. R. Shukla und P, IPCC, 2018: Summary for Policymakers. In: Global Warming of 1.5 °C. An IPCC Special Report on the impacts of global warming of 1.5 °C above preindustrial levels and related global greenhouse gas emission pathways, in the context of strengthening the globa, I. P. o. C. Change, Hrsg., Genf: World Meteorological Organization, 2018.
- [3] INDUSTRIEMAGAZIN Verlag GmbH, "HLK," 2019. [Online]. Available: https://hlk.co.at/a/indien-klimaanlagen-gegen-die-klimakrise. [Zugriff am 15 11 2019].
- [4] C. Wulf, M. Kaltschmitt und F. Dzukowski, "Betriebsoptimierung großtechnischer Kälteversorgungs- verbunde bei limitierter Datengrundlage Methodenentwicklung und Verifizierung," *Z Energiewirtsch 42*, p. 337–351, 2018.
- [5] C. Heinrich, S. Wittig, P. Albring, L. Richter, M. Safarik, U. Böhm und A. Hantsch, Climate Change 25/2014 Nachhaltige Kälteversorgung in Deutschland an den Beispielen Gebäudeklimatisierung und Industrie, Umweltbundesamt, Hrsg., Umweltbundesamt, 2014.
- [6] R. Hitchin, C. Pout und P. Riviere, "Assessing the market for air conditioning systems in European buildings," in *Energy and Buildings*, Bd. 58, 2013, pp. 355-362.
- [7] Europäische Kommission, "Klima- und Energiepaket 2020," [Online]. Available: https://ec.europa.eu/clima/policies/strategies/2020_de. [Zugriff am 2 3 2020].
- [8] Europäische Kommission, "Rahmen für die Klima- und Energiepolitik bis 2030," [Online]. Available: https://ec.europa.eu/clima/policies/strategies/2030_de. [Zugriff am 2 3 2020].
- [9] K. Geiger, B. Gugele, M. Gössl, M. Anderl, S. Haider, C. Heller, N. Ibesich, T. Köther, T. Krutzler, V. Kuschel, C. Lampert, H. Neier, K. Pazdemik, D. Perl und S. Poupa, Klimaschutzbericht 2019, U. GmbH, Hrsg., Wien, 2019.

- [10] G. J. Lauth und J. Kowalczyk, Thermodynamik Eine Einführung, Springer Spektrum, 2015.
- [11] A. Heintz, Thermodynamik Grundlagen und Anwendungen, 2. Auflage Hrsg., Springer, 2017.
- [12] H. D. Baehr und S. Kabelac, Thermodynamik Grundlagen und technische Anwendungen, 16 Hrsg., Springer Vieweg, 2016.
- [13] H. Herwig, Energie Richtig bewerten und sinnvoll nutzen, Springer Vieweg, 2016.
- [14] J. Dohmann, Thermodynamik der Kälteanlagen und Wärmepumpen, Grundlagen und Anwendungen der Kältetechnik, Springer Viehweg, 2016.
- [15] K. Fitzner und J. Masuch, Raumklimatechnik 16. Auflage, Bd. Band 4: Physik des Gebäudes, K. Fitzner, Hrsg., Springer Vieweg, 2013.
- [16] F. Wosnitza und H. G. Hilgers, Energieeffizienz und Energiemanagement Ein Überblick heutiger Möglichkeiten und Notwendigkeiten, Springer Spektrum, 2012.
- [17] C. Hainbach und S. Schädlich, "Kältetechnik," in *Dubbel*, Berlin, Springer, 2011, pp. M1-M58.
- [18] K. Kulterer, O. Mair, T. Sulzer, A. Betrand, M. Blaser und J. Saar, Leitfaden für Energieaudits in Kältesystemen, U. u. W. Ö. E. Bundesministerium Ffür Landund Forstwirschaft, Hrsg., Wien, 2015.
- [19] Daikin, "Technische Daten Wassergekühlt EWWP014-065KAW1N 16Hp 72Hp," 2009.
- [20] Fluke Corporation, Fluke 1735 Power Logger Bedienungshandbuch, 2006.
- [21] Chauvin Arnoux Group, "Bedienungsanleitung C.A 8331 C.A 8333 Analysator für Drehstromnetze," 2019.
- [22] Flexible Industriemesstechnik GmbH, "Flexim when measuring matters," 2019. [Online]. Available: https://www.flexim.com/de/produkte/portable-durchflussmesser-fuer-fluessigkeiten/fluxus-f601. [Zugriff am 24 09 2019].
- [23] FLEXIM GmbH, "Technische Spezifikation FLUXUS F601," 01 05 2019.
- [24] PCE Deutschland GmbH, "Durchflussmessgerät PCE-TDS 100HSH," 2019.

 [Online]. Available: https://www.pce-instruments.com/deutsch/messtechnik/messgeraete-fuer-alle-parameter/durchflussmessgeraet-durchflussmesser-pce-instruments-

- durchflussmessgeraet-pce-tds-100hsh-det_5885583.htm?_list=qr.art&_listpos=1. [Zugriff am 24 09 2019].
- [25] Maxim Integrated, DS18S20 High-Precision 1-Wire Digital Thermometer, 2015.
- [26] M. Rapport, Analysis of the Chillers in the BOKU Location Muthgasse II, D. o. M. S. a. P. E. BOKU, Hrsg., Wien: Master's Thesis, 2019.
- [27] KSB, Kennlinienheft Etaline PumpDrive, Pegnitz, 2007.
- [28] W. McKinney, "Data structures for statistical computing in python," in *Proceedings* of the 9th Python in Science Conference, Austin, Texas, 2010, pp. 51-56.
- [29] Bokeh Development Team , "Bokeh: Python library for interactive visualization," 2019. [Online]. Available: https://bokeh.org. [Zugriff am 2 3 2020].
- [30] SimTech GmbH, "What is IPSEpro?," 2019. [Online]. Available: http://simtechnology.com/CMS/index.php/ipsepro. [Zugriff am 3 11 2019].
- [31] M. Wolf, Application potential and optimization of high heat pump Systems for process heat supply, Wien: Institut für Verfahrens- und Energietechnik an der Universität für Bodenkultur Wien, 2019.
- [32] Bokeh Development Team, "Bokeh: Python library for interactive visualization," 2019. [Online]. Available: https://bokeh.org.
- [33] F. Bernhard, Handbuch der Technischen Temperaturmessung, 2. Auflage Hrsg., F. Bernhard, Hrsg., Springer Vieweg, 2014.

A. Anhang

- A.1. MatLab Skript für die quasi-dynamische Simulation im normalen Modus
- A.2. MatLab Skript für die quasi-dynamische Simulation im Validierungsmodus

A.1. MatLab Skript für die quasi-dynamische Simulation im normalen Modus

```
%% initialize com, define variables :
which invoke
app=actxserver('PSE.application');
proj=invoke(app,'openProject',
'C:\Users\Ivet43\Desktop\Vollmuth\DynamischeSimulation\Schiffstation_GesamtSystem_V4_1206_1200.pro');
j=0;
%% Ein- Ausgabeobjekte aus Ipse
NachPumpeDWzu=invoke(proj, 'findObject', 'W Stream001');
DWab=invoke(proj, 'findObject', 'W Stream002');
WTRKab=invoke(proj, 'findObject', 'W_Stream005');
WTRKzu=invoke(proj, 'findObject', 'W Stream004');
NachConPrimRKzu=invoke(proj, 'findObject', 'W Stream010b');
PrimRKab=invoke(proj, 'findObject', 'W_Stream008');
NachConPrimKWab=invoke(proj, 'findObject', 'W Stream002a');
VorPumpePrimKWzu=invoke(proj, 'findObject', 'W Stream007');
VorConSekKWab=invoke(proj, 'findObject', 'W_Stream012');
NachPumpeSekKWzu=invoke(proj, 'findObject', 'W_Stream011');
KM=invoke(proj, 'findObject', 'refrigeration machine001');
Mischer=invoke(proj, 'findObject', 'W_Mixer001');
WT=invoke(proj, 'findObject', 'W_Htex002');
%% Variablen benennen
t WTDzu=invoke(NachPumpeDWzu, 'finditem', 0, 't');
m WTDzu=invoke(NachPumpeDWzu, 'finditem', 0, 'mass');
E Loss WT=invoke(WT, 'finditem', 0, 'E loss');
t WTDab=invoke(DWab, 'finditem', 0, 't');
t WTRKab=invoke(WTRKab, 'finditem', 0, 't');
E_Loss_Mischer=invoke(Mischer, 'findItem', 0, 'E_loss');
t KMRKzu=invoke(NachConPrimRKzu, 'finditem', 0, 't');
P el KM=invoke(KM, 'finditem', 0, 'P el');
E Loss KM=invoke(KM, 'finditem', 0, 'E loss');
COP KM=invoke(KM, 'finditem', 0, 'COP KM');
Q cool KM=invoke(KM, 'finditem', 0, 'Q cool');
Q_heat_KM=invoke(KM, 'finditem', 0, 'Q_heat');
Q_trans_env_KM=invoke(KM, 'finditem', 0, 'Q_trans_env');
t_KMRKab=invoke(PrimRKab, 'finditem', 0 , 't');
m_KMRKab=invoke(PrimRKab, 'finditem', 0 , 'mass');
t WTRKzu=invoke(WTRKzu, 'finditem', 0, 't');
t KMKWab=invoke(NachConPrimKWab, 'finditem', 0, 't');
m KMKWab=invoke(NachConPrimKWab, 'finditem', 0, 'mass');
t KMKWzu=invoke(VorPumpePrimKWzu, 'finditem', 0, 't');
t_S_ab=invoke(VorConSekKWab, 'finditem', 0, 't');
m S ab=invoke(VorConSekKWab, 'finditem', 0 , 'mass');
t_S_zu=invoke(NachPumpeSekKWzu, 'finditem', 0, 't');
for i=1:10
%% Werte setzen: für Messwerte die in der IPSE Simulatin fehler verursachen
  if IWET messpi01T3C(i)> IWET messpi01T7C(i)
     IWET messpi01T7C(i) = IWET messpi01T3C(i)+0.01;
     invoke(t_S_ab, 'setStatus', 1, IWET_messpi01T7C(i));
     fprintf('IWET_messpi01T3C(%d) > IWET_messpi01T7C(%d) \n',i, i);
  end
```

```
%% Temperauren nur mit Rsp
  %invoke(t WTRKzu, 'setStatus', 1, IWET messpi01T1C(i));
  invoke(t_KMRKzu, 'setStatus', 1, IWET_messpi01T2C(i));
  invoke(t_KMKWab, 'setStatus', 1, IWET_messpi01T3C(i));
  invoke(t WTRKab, 'setStatus', 1, IWET messpi01T4C(i));
  invoke(t KMRKab, 'setStatus', 1, IWET messpi01T5C(i));
  invoke(t KMKWzu, 'setStatus', 1, IWET messpi01T6C(i));
  invoke(t S ab, 'setStatus', 1, IWET messpi01T7C(i));
  invoke(t_S_zu, 'setStatus', 1, IWET_messpi01T8C(i));
  %invoke(t_WTDzu, 'setStatus', 1, IWET_messpi01T11C(i));
  invoke(t_WTDab, 'setStatus', 1, IWET_messpi01T12C(i));
%% Durchflussmessungen PCEs
  invoke(m WTDzu, 'setStatus', 1, Flowmeter vp massenstrom (i));
  invoke(m_S_ab, 'setStatus', 1, Flowmeter_boku_massenstrom (i));
%% Durchflussmessungen Flexims
  invoke(m_KMRKab, 'setStatus', 1, Flexim1Massenstrom(i));
  invoke(m KMKWab, 'setStatus', 1, Flexim2Massenstrom(i));
%% Fluke3
  invoke(P_el_KM, 'setStatus', 1, Fluke3WirkleistungTotalAvgkW(i));
%% Simulation
  run=invoke(proj, 'runSimulation',0);
  if run >1
    run;
    j=j+1;
    save=invoke(proj, 'save');
    fprintf('Fehler in Punkt(%d)\n',i);
    fprintf('Fehler insgesamt(%d)\n',i);
  else
    invoke(proj, 'importEstimates', 0);
    PiT1(i)=invoke(t WTRKzu, 'resultValue');
    PiT2(i)=invoke(t_KMRKzu, 'resultValue');
    PiT3(i)=invoke(t_KMKWab, 'resultValue');
    PiT4(i)=invoke(t_WTRKab, 'resultValue');
    PiT5(i)=invoke(t_KMRKab, 'resultValue');
    PiT6(i)=invoke(t_KMKWzu, 'resultValue');
    PiT7(i)=invoke(t S ab, 'resultValue');
    PiT8(i)=invoke(t S zu, 'resultValue');
    PiT11(i)=invoke(t WTDzu, 'resultValue');
    PiT12(i)=invoke(t_WTDab, 'resultValue');
    PCE Boku(i)=invoke(m S ab, 'resultValue');
    PCE_Vp(i)=invoke(m_WTDzu, 'resultValue');
    P el sim(i)=invoke(P el KM, 'resultValue');
    Q_trans_env_sim(i)=invoke(Q_trans_env_KM, 'resultValue');
    E Loss KM sim(i)=invoke(E Loss KM, 'resultValue');
    E_Loss_WT_sim(i)=invoke(E_Loss_WT, 'resultValue');
    E_Loss_Mischer_sim(i)=invoke(E_Loss_Mischer, 'resultValue');
    E_Loss_Mischer_sim(i)=invoke(E_Loss_Mischer, 'resultValue');
    COP_KM_sim(i)=invoke(COP_KM, 'resultValue');
    Q cool KM sim(i)=invoke(Q cool KM, 'resultValue');
    Q_heat_KM_sim(i)=invoke(Q_heat_KM, 'resultValue');
```

```
end
end
app.release
fig = figure();
subplot(3,2,1); hold on;
plot(PiT1,'DisplayName', 'PiT1');
plot(PiT2,'DisplayName', 'PiT2');
plot(PiT3,'DisplayName', 'PiT3');
plot(PiT4, 'DisplayName', 'PiT4');
plot(PiT5,'DisplayName', 'PiT5');
plot(PiT6,'DisplayName', 'PiT6');
plot(PiT7,'DisplayName', 'PiT7');
plot(PiT8,'DisplayName', 'PiT8');
plot(PiT11, 'DisplayName', 'PiT11');
plot(PiT12, 'DisplayName', 'PiT12');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(3,2,2); hold on;
plot(PCE_Boku,'DisplayName', 'PCE Boku');
plot(PCE_Vp,'DisplayName', 'PCE Vp');
xlabel('time [Minutes]'); ylabel('Massenstrom');
legend
subplot(3,2,3); hold on;
plot(P_el_sim,'DisplayName', 'P el sim');
plot(Q_cool_KM_sim,'DisplayName', 'Q cool KM sim');
plot(Q_heat_KM_sim,'DisplayName', 'Q heat KM sim');
plot(Q_trans_env_sim,'DisplayName', 'Q trans env sim');
xlabel('time [Minutes]'); ylabel('Leistung in kW');
legend
subplot(3,2,4); hold on;
plot(E_Loss_KM_sim,'DisplayName', 'E Loss KM sim');
plot(E_Loss_WT_sim,'DisplayName', 'E Loss WT sim');
plot(E_Loss_Mischer_sim, 'DisplayName', 'E Loss Mischer sim');
xlabel('time [Minutes]'); ylabel('Exergieverlust in kW');
legend
subplot(3,2,5); hold on;
plot(COP_KM_sim,'DisplayName', 'COP KM sim');
```

xlabel('time [Minutes]'); ylabel('COP');

legend

A.2. MatLab Skript für die quasi-dynamische Simulation im Validierungsmodus

```
%% initialize com, define variables:
app=actxserver('PSE.application');
proj=invoke(app,'openProject',
'C:\Users\Ivet43\Desktop\Vollmuth\DynamischeSimulationValidierung\ValidierungV3.pro');
invoke(proj, 'getSolverConfiguration', 3);
%% Ein- Ausgabeobjekte aus Ipse
NachPumpeDWzu=invoke(proj, 'findObject', 'W Stream001');
DWab=invoke(proj, 'findObject', 'W Stream002');
WTRKab=invoke(proj, 'findObject', 'W_Stream005');
WTRKzu=invoke(proj, 'findObject', 'W Stream004');
NachConPrimRKzu=invoke(proj, 'findObject', 'W Stream010b');
PrimRKab=invoke(proj, 'findObject', 'W_Stream008');
NachConPrimKWab=invoke(proj, 'findObject', 'W Stream002a');
VorPumpePrimKWzu=invoke(proj, 'findObject', 'W Stream007');
VorConSekKWab=invoke(proj, 'findObject', 'W Stream012');
NachPumpeSekKWzu=invoke(proj, 'findObject', 'W Stream011');
KM=invoke(proj, 'findObject', 'refrigeration machine001');
Mischer=invoke(proj, 'findObject', 'W_Mixer001');
WT=invoke(proj, 'findObject', 'W_Htex002');
t WTDzu=invoke(NachPumpeDWzu, 'finditem', 0, 't');
m_WTDzu=invoke(NachPumpeDWzu, 'finditem', 0 , 'mass');
E_Loss_WT=invoke(WT, 'finditem', 0 , 'E_loss');
t_WTDab=invoke(DWab, 'finditem', 0, 't');
t WTRKab=invoke(WTRKab, 'finditem', 0, 't');
E Loss Mischer=invoke(Mischer, 'findItem', 0, 'E loss');
t KMRKzu=invoke(NachConPrimRKzu, 'findItem', 0, 't');
P_el_KM=invoke(KM, 'finditem', 0, 'P_el');
E Loss KM=invoke(KM, 'finditem', 0, 'E loss');
COP KM=invoke(KM, 'finditem', 0, 'COP KM');
Q_cool_KM=invoke(KM, 'finditem', 0, 'Q_cool');
Q heat KM=invoke(KM, 'finditem', 0, 'Q heat');
Q_trans_env_KM=invoke(KM, 'finditem', 0, 'Q_trans_env');
t KMRKab=invoke(PrimRKab, 'finditem', 0, 't');
m KMRKab=invoke(PrimRKab, 'finditem', 0, 'mass');
t WTRKzu=invoke(WTRKzu, 'findItem', 0, 't');
t KMKWab=invoke(NachConPrimKWab, 'finditem', 0, 't');
m KMKWab=invoke(NachConPrimKWab, 'finditem', 0, 'mass');
t KMKWzu=invoke(VorPumpePrimKWzu, 'finditem', 0, 't');
t S ab=invoke(VorConSekKWab, 'finditem', 0, 't');
m S ab=invoke(VorConSekKWab, 'finditem', 0, 'mass');
t S zu=invoke(NachPumpeSekKWzu, 'finditem', 0, 't');
for i=1:8382
%% Werte setzen: für Messwerte die in der IPSE Simulatin fehler verursachen
  if IWET messpi01T3C(i)> IWET messpi01T7C(i)
     IWET messpi01T7C(i) = IWET messpi01T3C(i)+0.1;
     invoke(t_S_ab, 'setStatus', 1, IWET_messpi01T7C(i));
     fprintf('IWET_messpi01T3C(%d) > IWET_messpi01T7C(%d) \n',i, i);
  end
%% Temperauren nur mit Rsp
  invoke(t KMRKzu, 'setStatusMeasured', IWET messpi01T2C(i), StdDevTemp(i));
  invoke(t KMKWab, 'setStatusMeasured', IWET messpi01T3C(i), StdDevTemp(i));
```

```
invoke(t WTRKab, 'setStatus', 1, IWET messpi01T4C(i));
  invoke(t KMRKab, 'setStatusMeasured', IWET messpi01T5C(i), StdDevTemp(i));
  invoke(t_KMKWzu, 'setStatusMeasured', IWET_messpi01T6C(i), StdDevTemp(i));
  invoke(t_S_ab, 'setStatus', 1, IWET_messpi01T7C(i));
  invoke(t S zu, 'setStatus', 1, IWET messpi01T8C(i));
  invoke(t WTDzu, 'setStatusMeasured', IWET messpi01T11C(i), StdDevTemp(i));
  invoke(t_WTDab, 'setStatusMeasured', IWET_messpi01T12C(i), StdDevTemp(i));
%% Durchflussmessungen PCEs
  invoke(m_WTDzu, 'setStatusMeasured', Flowmeter_vp_massenstrom (i), StdDevMasse(i));
  invoke(m_S_ab, 'setStatus', 1, Flowmeter_boku_massenstrom (i));
%% Durchflussmessungen Flexims
  invoke(m KMRKab, 'setStatusMeasured', Flexim1Massenstrom(i), StdDevMasse(i));
  invoke(m KMKWab, 'setStatusMeasured', Flexim2Massenstrom(i), StdDevMasse(i));
%% Fluke3
  invoke(P_el_KM, 'setStatusMeasured', Fluke3WirkleistungTotalAvgkW(i), StdDevTemp(i));
  %invoke(P el th KM, 'setStatusMeasured', Fluke3WirkleistungTotalAvgkW(i), StdDevMasse(i));
%% Simulation
  run=invoke(proj, 'runSimulation',0);
  if run >1
    run;
    j=j+1;
    save=invoke(proj, 'save');
    fprintf('Fehler in Punkt(%d)\n',i);
    fprintf('Fehler insgesamt(%d)\n',j);
    invoke(proj,'importEstimates',0);
    PiT1(i)=invoke(t WTRKzu, 'resultValue');
    PiT2(i)=invoke(t KMRKzu, 'resultValue');
    PiT3(i)=invoke(t_KMKWab, 'resultValue');
    PiT4(i)=invoke(t_WTRKab, 'resultValue');
    PiT5(i)=invoke(t_KMRKab, 'resultValue');
    PiT6(i)=invoke(t_KMKWzu, 'resultValue');
    PiT7(i)=invoke(t_S_ab, 'resultValue');
    PiT8(i)=invoke(t S zu, 'resultValue');
    PiT11(i)=invoke(t WTDzu, 'resultValue');
    PiT12(i)=invoke(t WTDab, 'resultValue');
    PCE Boku sim(i)=invoke(m S ab, 'resultValue');
    PCE Vp sim(i)=invoke(m WTDzu, 'resultValue');
    Flexim1Massenstrom_sim(i)=invoke(m_KMRKab, 'resultValue');
    Flexim2Massenstrom_sim(i)=invoke(m_KMKWab, 'resultValue');
    E_Loss_KM_sim(i)=invoke(E_Loss_KM, 'resultValue');
    E Loss WT sim(i)=invoke(E Loss WT, 'resultValue');
    E_Loss_Mischer_sim(i)=invoke(E_Loss_Mischer, 'resultValue');
    P_el_sim(i)=invoke(P_el_KM, 'resultValue');
    COP_KM_sim(i)=invoke(COP_KM, 'resultValue');
    Q_cool_KM_sim(i)=invoke(Q_cool_KM, 'resultValue');
    Q heat KM sim(i)=invoke(Q heat KM, 'resultValue');
    Q trans env sim(i)=invoke(Q trans env KM, 'resultValue');
```

```
end
end
app.release
fig = figure();
subplot(8,2,1); hold on;
plot(PiT1,'DisplayName', 'PiT1');
plot(PiT2,'DisplayName', 'PiT2');
plot(PiT3,'DisplayName', 'PiT3');
plot(PiT4, 'DisplayName', 'PiT4');
plot(PiT5,'DisplayName', 'PiT5');
plot(PiT6,'DisplayName', 'PiT6');
plot(PiT7,'DisplayName', 'PiT7');
plot(PiT8,'DisplayName', 'PiT8');
plot(PiT11, 'DisplayName', 'PiT11');
plot(PiT12, 'DisplayName', 'PiT12');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,2); hold on;
plot(PCE_Boku_sim,'DisplayName', 'PCE_Boku sim');
plot(PCE_Vp_sim,'DisplayName', 'PCE Vp sim');
plot(Flexim1Massenstrom_sim,'DisplayName', 'Flexim1Massenstrom sim');
plot(Flexim2Massenstrom_sim, 'DisplayName', 'Flexim2Massenstrom sim');
xlabel('time [Minutes]'); ylabel('Massenstrom');
legend
subplot(8,2,3); hold on;
plot(P_el_sim,'DisplayName', 'P el sim');
plot(Q cool KM sim, 'DisplayName', 'Q cool KM sim');
plot(Q_heat_KM_sim,'DisplayName', 'Q heat KM sim');
plot(Q_trans_env_sim,'DisplayName', 'Q trans env sim');
xlabel('time [Minutes]'); ylabel('Leistung in kW');
legend
subplot(8,2,4); hold on;
plot(E_Loss_KM_sim,'DisplayName', 'E Loss KM sim');
plot(E_Loss_WT_sim,'DisplayName', 'E Loss WT sim');
plot(E Loss Mischer sim, 'DisplayName', 'E Loss Mischer sim');
xlabel('time [Minutes]'); ylabel('Exergieverlust in kW');
legend
subplot(8,2,5); hold on;
plot(COP_KM_sim,'DisplayName', 'COP KM sim');
xlabel('time [Minutes]'); ylabel('COP');
legend
subplot(8,2,6); hold on;
plot(PiT1,'DisplayName', 'PiT1');
plot(IWET_messpi01T1C,'DisplayName', 'IWET_messpi01T1C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,7); hold on;
```

plot(PiT2, 'DisplayName', 'PiT2');

```
plot(IWET messpi01T2C, 'DisplayName', 'IWET messpi01T2C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,8); hold on;
plot(PiT3, 'DisplayName', 'PiT3');
plot(IWET_messpi01T3C,'DisplayName', 'IWET_messpi01T3C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,9); hold on;
plot(PiT4, 'DisplayName', 'PiT4');
plot(IWET_messpi01T4C,'DisplayName', 'IWET_messpi01T4C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,10); hold on;
plot(PiT5,'DisplayName', 'PiT5');
plot(IWET messpi01T5C,'DisplayName', 'IWET messpi01T5C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,11); hold on;
plot(PiT6,'DisplayName', 'PiT6');
plot(IWET messpi01T6C, 'DisplayName', 'IWET messpi01T6C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,12); hold on;
plot(PiT7,'DisplayName', 'PiT7');
plot(IWET messpi01T7C, 'DisplayName', 'IWET messpi01T7C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,13); hold on;
plot(PiT8,'DisplayName', 'PiT8');
plot(IWET messpi01T8C,'DisplayName', 'IWET messpi01T8C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,14); hold on;
plot(PiT11, 'DisplayName', 'PiT11');
plot(IWET_messpi01T11C,'DisplayName', 'IWET_messpi01T11C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
subplot(8,2,15); hold on;
plot(PiT12, 'DisplayName', 'PiT12');
plot(IWET_messpi01T12C,'DisplayName', 'IWET_messpi01T12C');
xlabel('time [Minutes]'); ylabel('Temperatur');
legend
```