

Optimierte Einbindung von Energiespeichern in industrielle Prozesse

von

Christoph MOSER

als Diplomarbeit der Kommission zur Abhaltung der dritten Diplomprüfung
der Studienrichtung Wirtschaftsingenieurwesen-Maschinenbau an der
Technischen Universität Graz über das Institut für Wärmetechnik vorgelegt.

Betreuer

Dipl.-Ing. Dr. techn. Richard Heimrath

Dipl.-Ing. Dr. mont. Hermann Schranzhofer

Beurteiler:

Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. René Rieberer

Graz, im November 2012

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

(Unterschrift)

KURZFASSUNG

Titel: Optimierte Einbindung von Energiespeichern in industrielle Prozesse

Autor: Christoph Moser

1. Stichwort: Thermische Energiespeicher
2. Stichwort: Thermische Anlagensimulation
3. Stichwort: Effizienz

Industrielle Produktionsprozesse weisen hohe Energiedichten, unterschiedliche Temperaturniveaus und einen zeitlich differierenden Energiebedarf auf. Zur Optimierung von derartigen Prozessen sind komplexe Modelle für Wärmetauschernetzwerke mit integrierten Speichern zum Ausgleich der zeitlichen Differenzen notwendig. In dieser Arbeit wird auf die speziellen Anforderungen derartiger Speichermodelle eingegangen.

Dazu wurden Messdaten von zehn Großspeichern (18 m³ bis 2050 m³ Volumen) erhoben und mit Modellberechnungen in der Simulationsumgebung TRNSYS (Transient System Simulation Tool) verglichen. Die Wiedergabegüte hing in großem Maße von den vorhandenen Daten ab, wobei die Rohrleitungen als ein kritischer Punkt ausgemacht werden konnten. Der Kontakt mit den Anlagenbetreibern stellte sich als essentiell für den Aufbau der Simulation heraus. Die Simulation zeigte, dass die Abweichungen der Simulationsergebnisse von den Messdaten bei größeren Speichern tendenziell geringer waren als bei kleineren Speichern. Auch die berechneten Effizienzen zeigten eine starke Abhängigkeit vom Speichervolumen.

Als wesentliche Parameter für einen thermischen Energiespeicher wurden die Dämmung, die Wärmeverluste der Anschlüsse und die Belade- und Entladehöhe identifiziert.

ABSTRACT

Title: Optimized Integration of Energy Stores in industrial Processes

Author: Christoph Moser

1st keyword: Thermal Energy Store

2nd keyword: Thermal Facility Simulation

3rd keyword: Efficiency

Industrial Processes are characterized by high energy rates, different temperature levels and discontinuous energy demand. To optimize these processes, complex models for heat exchanger networks with integrated storages to compensate the time difference have to be created. This thesis has the main focus on the special requirements of a storage model.

For this purpose measured data from ten large storages (18 m³ to 2050 m³ filling volume) were collected and compared with model calculations using the simulation environment TRNSYS (Transient System Simulation Tool). The agreement of the simulation results mainly depends on the available data. The pipes could be identified as a critical issue. Essential for simulation development was the communication with the facility operators. The simulation shows that the deviations of the simulation results from measured data for larger storages tend to be lower than for smaller storages. Also the calculated efficiencies showed a strong dependence on the storage volume.

As important parameters for thermal energy storage the insulation, the heat loss of the ports and the height of the inlets and outlets were identified.

VORWORT

Im Laufe meines Studiums Wirtschaftsingenieurwesen – Maschinenbau mit der Vertiefung „Energie- und Umwelttechnik“ wurde der Grundstein für die spätere Wahl dieser Arbeit gelegt. Diese Arbeit bot die Möglichkeit einen genaueren Einblick in industrielle Prozesse zu bekommen und diese näher zu analysieren.

Mein besonderer Dank gilt meinen Betreuern Herrn Dipl.-Ing. Dr. techn. Richard Heimrath und Herrn Dipl.-Ing. Dr. mont. Hermann Schranzhofer, welche mir diese Arbeit im Rahmen des Forschungsprojektes SOCO ermöglichten und mir stets mit Rat und Tat zur Seite standen.

Weiters möchte ich mich bei Herrn Dipl.-Ing. (FH) Dr. techn. Andreas Heinz und Herrn Dipl.-Ing. Werner Lerch für ihre Tipps und Anregungen bedanken.

Vielen lieben Dank an meine Eltern Johann und Dorothea, ohne deren Unterstützung die Absolvierung meines Studiums und in weiterer Folge die Verfassung dieser Arbeit nicht möglich gewesen wäre.

Graz, im November 2012

Christoph Moser

INHALTSVERZEICHNIS

1	<u>EINLEITUNG</u>	1
1.1	Motivation	1
1.2	Stand der Technik	1
1.3	Vorgehensweise	2
2	<u>GRUNDLAGEN</u>	4
2.1	Speichermodelle	4
2.1.1	Eindimensionales Modell	4
2.1.2	Mehrdimensionale Modelle	10
2.2	Simulationsumgebung TRNSYS	11
2.2.1	Mathematisch- physikalisches Modell	12
2.2.2	Informationsflüsse im Simulationsaufbau	14
2.3	Speichermodelle in TRNSYS	14
2.3.1	Schichtspeicher (Type4)	15
2.3.2	Detaillierter Flüssigkeitsspeicher (Type60)	16
2.3.3	Ideal geschichteter Speicher (Type38)	17
2.3.4	MULTIPORT – Speichermodell (Type340)	18
2.3.5	Speicher mit Phasenwechselmaterial (Type840)	19
2.3.6	Speicher mit variabler Füllhöhe (Type39)	20
2.3.7	Speicher mit variabler Füllhöhe (Type844)	20
2.4	Verwendete Modelle in TRNSYS	21
2.4.1	Regler (Type2b)	21
2.4.2	Einfacher Massenstromteiler (Type11f)	21
2.4.3	Rohrleitungsmodell (Type709)	21
2.4.4	Rohrleitungsmodell (Type604a)	22
2.4.5	Erweiterter Massenstromteiler (Type647)	22
2.4.6	Mischer (Type649)	22
2.5	Vergleich der Speichermodelle	23
2.5.1	Beschreibung des Lastprofils	23
2.5.2	Testaufbau des Prüfzyklus	24
2.5.3	Ergebnisse des Speichertest	27
2.6	Datenerhebung - Messdaten	29
2.7	Effizienz	29
2.7.1	Gesamteffizienz	30
2.7.2	Periodeneffizienz	31
3	<u>SIMULATIONSAUFBAU UND ERGEBNISSE</u>	33
3.1	Energiespeicher (93 m ³) – STIEGL	33
3.2	Warmwasserspeicher (220 m ³) – STIEGL	40
3.3	Graz (2150 m ³) – Energie Steiermark	47
3.4	Rottenmann (54 m ³) – Energie Steiermark	52
3.5	Rottenmann (250 m ³) – Energie Steiermark	57
3.6	Knittelfeld (105 m ³) – Energie Steiermark	60
3.7	Knittelfeld (320 m ³) – Energie Steiermark	66
3.8	Gleinstätten (44 m ³) – Nahwärme	70

3.9	Mattsee (40 m³) – Nahwärme	77
3.10	Golling (18 m³) – Nahwärme	83
4	<u>EFFIZIENZ</u>	<u>88</u>
4.1	Effizienz des Modellprozesses	88
4.1.1	Beschreibung der prEN 12977-3:2008 (E)	88
4.1.2	Beschreibung des Prüfzyklus	89
4.1.3	Modellierung in TRNSYS	90
4.1.4	Ergebnisse	91
4.2	Effizienz des realen Prozesses	93
4.2.1	Ermittlung der Effizienz im realen Prozess	93
4.2.2	Ergebnisse	95
5	<u>SCHLUSSFOLGERUNGEN</u>	<u>97</u>
	<u>LITERATUR</u>	<u>99</u>
	<u>ANHANG</u>	<u>101</u>

Nomenklatur

Abkürzung und Indizes

<u>Bezeichnung</u>	<u>Bedeutung</u>
Inflow	Ladevorgang
Outflow	Entladevorgang
Bal	Bilanz (Balance)
WT	Wärmetauscher
KV	Kontrollvolumen (Control Volume)
mess	Daten aus Messung
sim	Daten aus Simulation
PCM	Phasenwechselmaterial (phase change material)
kum	kumulativ
WK	Wärzgekühler
FW	Frischwasser
Circ	Zirkulation
SH	Sudhaus
Netz	Fernwärme
BMK	Biomassekessel
Solar	Solaranlage
Kond	Rauchgaskondensation
VL	Vorlauf
RL	Rücklauf
<i>dp</i>	Doppelanschluss (double port)
<i>vert</i>	Vertikale Wärmeleitung
<i>loss</i>	Verluste an Umgebung
<i>hx</i>	Wärmetauscher (heat exchanger)
<i>aux</i>	Elektrische Zusatzheizung (auxiliary heating)
<i>flow</i>	Wärmefluss durch Konvektion
<i>in</i>	Eintretend
<i>out</i>	Austretend
<i>const</i>	konstant
<i>amb</i>	Umgebung (ambient)
<i>res</i>	resultierend
<i>i</i>	Knoten des Eintrittsmassenstroms
<i>o</i>	Knoten des Austrittsmassenstroms
<i>j</i>	Knoten
<i>c</i>	Anschlussnummer (Port)
<i>d</i>	Richtung des Wärmeflusses
δ	Schalter für den Knoten des Einlasses
ξ	Schalter für die Berechnung der Konvektion (konstante Füllhöhe)
ζ	Schalter für die Berechnung der Konvektion (variable Füllhöhe)

Formelzeichen

<u>Symbol</u>	<u>Einheit</u>	<u>Bedeutung</u>
A	m^2	Fläche
Δz	m	Höhe des Knotens
T_j	$^{\circ}C$	Temperatur des Knotenelements
m_j	kg	Masse eines Knotenelements
h_j	$kJ / (kg \cdot K)$	Enthalpie eines Knotenelements
N	-	Knotenanzahl
λ_{eff}	$W / (m \cdot K)$	Effektive Wärmeleitfähigkeit
T_{init}	$^{\circ}C$	Initialisierungstemperatur
V	m^3	Speichervolumen
h	m	Höhe
cp	$kJ / (kg \cdot K)$	spezifische Wärmekapazität
η	$kg / (m \cdot s)$	Dynamische Zähigkeit
ν	m^2 / s	Kinematische Zähigkeit
λ	$W / (m \cdot K)$	Wärmeleitfähigkeit
ρ	kg / m^3	Dichte
U	$W / (m^2 \cdot K)$	Wärmedurchgangszahl
α	$W / (m^2 \cdot K)$	Wärmeübergangszahl
\dot{Q}	kW	Wärmeleistung
P	kW	Leistung durch elektrische Wärmequellen
Q	kWh	Energie
E'_f	kWh	Energieinhalt des KV zum Simulationsende
E'_i	kWh	Energieinhalt des KV zum Simulationsstart
ΔE	kWh	Änderung der inneren Energie im KV
η^D	-	Effizienz

Begriffserläuterungen

Doppelanschluss	Zwei zusammenhängende Anschlüsse - ein Einlass und ein Auslass – an einem Speicher. Der Massenstrom ist in beiden Anschlüssen gleich, er unterscheidet sich nur in der Strömungsrichtung.
Inverse Temperaturschichtung	Die Temperatur im Speicher ist in einer höheren Schicht kleiner als in der darunterliegenden. Bedingt durch den Dichteunterschied führt dies beim Speichermedium Wasser zu einer Durchmischung der betroffenen Knoten.
Klörperboden	Beim Klörperboden handelt es sich um einen gewölbten Boden, der als Abschlusselement von meist zylindrischen (rohrförmigen) Druckbehältern oder anderen prozesstechnischen Behältern verwendet wird.

1 EINLEITUNG

1.1 Motivation

Industrielle Produktionsprozesse zeichnen sich durch eine hohe Energiedichte, unterschiedliche Temperaturniveaus und diskontinuierlichen Energiebedarf aus. Um derartige Prozesse optimal hinsichtlich des Energiebedarfs betreiben zu können, ist es neben einer Vielzahl weiterer Aspekte von Interesse in welchem Ausmaß Unterschiede zwischen Wärmeeinfall und Wärmebedarf ausgeglichen werden können. Hierzu ist es unumgänglich, neben einem Wärmetauscher Netzwerk, auch entsprechende Energiespeicher zu integrieren um auch die zeitlichen Unterschiede ausgleichen zu können.

Die vorliegende Diplomarbeit hat die Aufgabe, bestehende Simulationsmodelle von Energiespeichern hinsichtlich ihrer Verwendbarkeit zur Analyse von industriellen Produktionsprozessen zu validieren und ein einfaches Speichermodell mit guter Genauigkeit zu erstellen, welches die Integration von Energiespeichern in eine dynamische Bedarfssimulation ermöglicht.

1.2 Stand der Technik

Der Stand der Technik in der Prozessintegration ist die Verwendung der Pinch Analyse (Klemeš et al., 2010). Durch die Nutzung der Pinch Analyse können Energieströme analysiert und hinsichtlich ihres Energieeinsparpotentials bewertet werden. Da damit zusätzliche Investitionen in die Anlage (Wärmetauscher, Verrohrung) notwendig werden, muss eine ganzheitliche Kostenbetrachtung durchgeführt werden. Somit ist die Ausführung und Gestaltung dieser Wärmetauschernetzwerke von großer Bedeutung. Durch den immer wichtiger werdenden Einsatz von erneuerbarer Energie kommt der Optimierung dieser Prozesse eine noch größere Bedeutung zu. Diese geänderten Rahmenbedingungen machen die Ausdehnung der Optimierung, auf die im Prozess eingesetzten thermischen Energiespeicher, notwendig. Diese Fragestellung wurde bereits aufgegriffen (Varbanov et al. 2011, Chen and Ciou, 2008; Foo et al., 2008; Atkins et al., 2010; Schnitzer et al., 2007; Majozi, 2009), weist jedoch noch ein großes Optimierungspotential auf. Für die Integration von erneuerbaren Energieträgern (Solarthermie, Biomasse) ist es essentiell, dass thermische Energiespeicher hinsichtlich der Anzahl, Art, Größe und des Temperaturniveaus optimiert werden. Derzeit (Stand: 2012) gibt es noch kein Planungstool, welches die Möglichkeit bietet bei der Optimierungsanalyse von Industrieprozessen auch die Einbindung von Energiespeichern zu optimieren. Bisherige Planungstools wie PE² (Pinch Energy Efficiency) können nur statische Betriebsdaten (durchschnittliche Last, welche ein- und ausgeschaltet wird) verarbeiten.

PE² ist ein Planungstool, welches im Rahmen gemeinsamer Projekte der AEE INTEC und der JOANNEUM RESEARCH (SolProBat; IEA Task 33, Promise Application) entwickelt wurde. Mit PE² konnte mit dem Ergebnis einer Pinch Analyse der aufzuwärmenden und abzukühlenden Energieströme auf Basis eines kombinatorischen Algorithmus ein Wärmetauschernetzwerk erstellt werden. Jedoch konnte mit PE² durch den Einsatz der statischen Prozessdaten keine korrekte Aussage über die Auslegung benötigter Speicher getroffen werden.

1.3 Vorgehensweise

An das Planungstool PE² knüpft das Projekt SOCO (Storage Optimisation Concepts – Optimierte thermische Speichersysteme für Industrie, Gewerbe und Fernwärmenetze, Projektnummer 829856) - in dessen Rahmen die vorliegende Arbeit erstellt wurde- an, welches die Optimierung von Energiespeichersystemen und Wärmetauschernetzwerken, auf Basis realer Prozessdaten, ermöglichen soll. Dadurch wird die Erweiterung der Pinch Analyse, sowie der Einsatz eines Energiespeichermodells notwendig. Durch diesen gesamtheitlichen Ansatz kann der Speicher auch zusätzlich auf die Anforderungen der erneuerbaren Energien optimiert werden. Das eingesetzte thermische Energiespeichermodell sollte mit möglichst wenigen Eingabedaten sein Auslangen finden und eine gute Genauigkeit bei schnellstmöglicher Ausführungszeit verfügen. Eine Übersicht über die möglichen Ansätze zur Berechnung eines thermischen Energiespeichers sind in Han et al. (2009) zu finden.

Zur Erstellung und Validierung des Speichermodells standen Messdaten von verschiedenen thermischen Energiespeichern zur Verfügung, die von kleinen Fernwärmespeichern mit 18 m³ bis zu großen Speichern mit 2050 m³ reichen. In dieser Auswahl sind neben der Anwendung in Wärmenetzen auch zwei typische Industriespeicher enthalten. Die zehn Energiespeicher stellen eine repräsentative Auswahl zur Validierung der betrachteten Modelle dar. Die Datenerhebung der Größen wie die Temperaturen und die Volumenströme erfolgte entweder durch Messung (Industriespeicher) oder durch Auslesen der Anlagensteuerung. Für die weiteren relevanten Parameter, welche für die Simulation benötigt werden, wurde ein Fragebogen erstellt. Diesen Fragebogen erhielten die Betreiber der Anlagen, in welchen die zeitinvarianten Daten wie Abmessungen, Geometrie, Höhen, Dämmung und die Stoffdaten der Energiespeicher anzugeben waren.

Die Wahl für die Kontrolle der Messdaten fiel auf das Tabellenkalkulationsprogramm Microsoft EXCEL, da es mit „Visual Basic for Applications“ (VBA) auch die Möglichkeit bietet, eigene Programme zu erstellen. Dadurch ist es möglich die Messwerte zu überprüfen und Abweichungen aufzuzeigen. Auch ist es mit EXCEL möglich, große Datenmengen zu verarbeiten und diese auch entsprechend in Diagrammen zu visualisieren. In EXCEL wurden die Eingabegrößen der Simulation aufbereitet und an TRNSYS (Transient System Simulation Tool), mittels einer Textdatei, übergeben.

TRNSYS (2011) bietet die Möglichkeit der dynamischen (instationären) Simulation von Anlagen und Gebäuden. Es können durch Verknüpfungen von Modellen komplexe Anlagen abgebildet werden. TRNSYS wird von einer weltweiten wissenschaftlich orientierten Community zur Abbildung von komplexen thermischen Energiesystemen unter der Einbindung von Gebäuden genutzt. TRNSYS wird unter anderem am Institut für Thermodynamik und Wärmetechnik (ITW) der Universität Stuttgart, dem Institut für Solartechnik (SPF) der Fachhochschulen der Ostschweiz sowie dem Institut für Wärmetechnik der TU Graz zur Validierung von Kollektoren, Speichern und Solaranlagen erfolgreich eingesetzt.

Bevor der Aufbau der Simulationen stattfand wurden die vorhandenen Speichermodelle in TRNSYS getestet, um zu analysieren, welches physikalische Modell den vorliegenden Anforderungen am besten entspricht. Nach der Wahl des entsprechenden Modells wurden in der Simulationsumgebung TRNSYS die realen Speicherkonzepte inklusive der zur Betrachtung notwendigen Peripherie abgebildet. Fehlende Parameter wurden durch eine entsprechende Berechnung oder auch durch Vorsimulationen ermittelt. Dies wurde mit allen gemessenen Energiespeichern durchgeführt. Die Ergebnisse dieser Simulationen wurden mit den Messungen validiert. Abschließend erfolgte in TRNSYS der Vergleich der Effizienzen der gemessenen Energiespeicher anhand der realen Prozessdaten. Um eine Vergleichbarkeit der Energiespeicher untereinander zu ermöglichen, wurde zusätzlich zu den realen Be- und Entladungsstrategien eine Effizienz anhand eines fiktiven Vergleichsprozesses berechnet.

Basierend auf den Ergebnissen der vorangegangenen Simulationen und Validierungen wurde ein thermisches Speichermodell in der VBA - Umgebung von Microsoft EXCEL entwickelt, da bereits während der Aufbereitung der Messdaten umfangreiche Erfahrungen gesammelt wurden. Auch konnten in EXCEL die Parameter und Eingangsgrößen rasch modifiziert werden und somit die Berechnungsergebnisse unter Zuhilfenahme der vorliegenden Energiespeichersimulationen in TRNSYS überprüft und anhand der Messdaten validiert werden.

2 GRUNDLAGEN

2.1 Speichermodelle

Die mathematischen Modelle zur Beschreibung eines Speichers sollten die tatsächlichen physikalischen Vorgänge, möglichst genau, wiedergeben können. Da jedoch eine Erhöhung der Komplexität auch eine Erhöhung der Rechenzeit bedingt, ist es oft nötig Vereinfachungen zu treffen, um die Simulationsdauer zu reduzieren. Die Modelle reichen von eindimensionalen Modellen für Jahressimulationen bis zu mehrdimensionalen Modellen welche auch die reale Geometrie und die Strömungen im Inneren eines Speichers abbilden können.

2.1.1 Eindimensionales Modell

Die folgenden Gleichungen sind der Vorlesung „Numerische Methoden in der Energietechnik“, (Schranzhofer, 2010), entnommen.

Nach dem Ansatz eines Knotenmodells, wird der Speicher in N vollkommen gemischte Segmente unterteilt. Die Anzahl der Segmente stellt zugleich den Grad der Schichtung dar, welcher erreicht werden kann. Für den Sonderfall $N = 1$ ergibt sich ein vollkommen gemischter Speicher. Die Segmente besitzen die Grundfläche (A) und die Höhe (Δz). Jeder Knoten besitzt eine Temperatur (T_j), eine Masse (m_j) und die Enthalpie (h_j), siehe Abbildung 2-1.

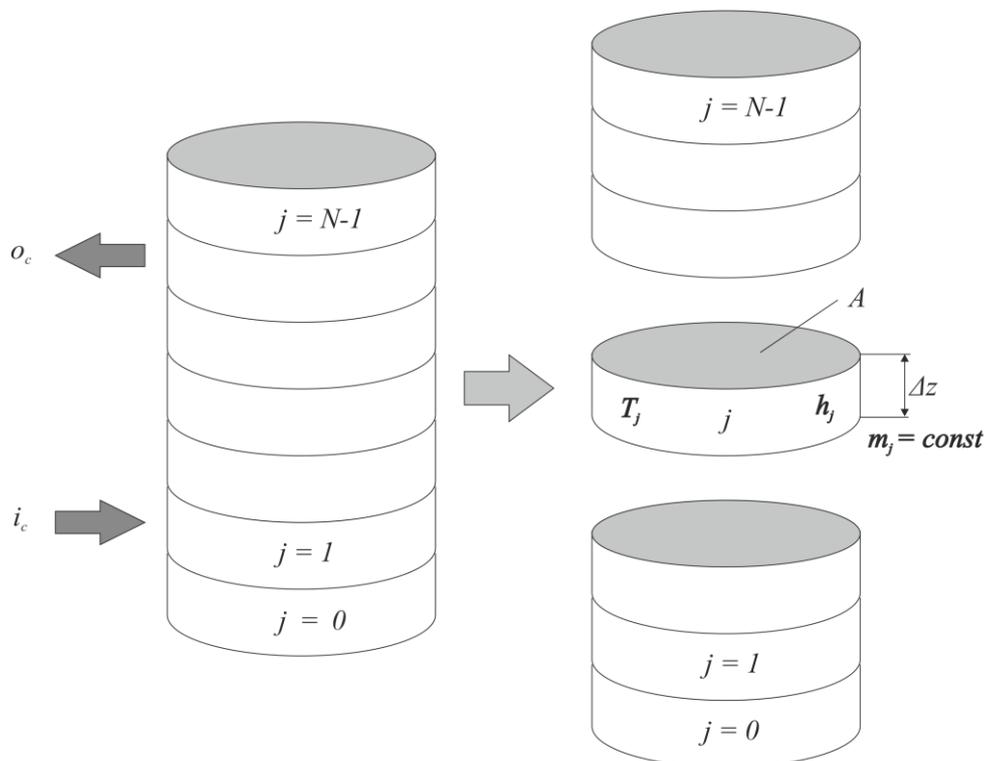


Abbildung 2-1: Eindimensionales Speichermodell

Die Energiebilanz lautet somit für jeden j -ten Knoten.

$$\sum \dot{Q}_j = m_j \cdot \frac{dh_j}{dt} \quad \text{Gl. 2-1}$$

Diese Differentialgleichung (Gl. 2-1) kann mit der expliziten Methode gelöst werden. Daraus ergibt sich für jeden Knoten folgende Differenzengleichung:

$$\sum \dot{Q}_j(t) = m_j \cdot \frac{h_j(t + \Delta t) - h_j(t)}{\Delta t} \quad \text{Gl. 2-2}$$

Die linke Seite von Gl. 2-2 beschreibt die Summe aller auftretenden Energieströme, die rechte Seite beschreibt die Änderung der inneren Energie im Knoten j .

Im Wesentlichen können am Knoten j fünf Energieströme auftreten:

$$\sum \dot{Q}_j = \dot{Q}_j^{(dp)} + \dot{Q}_j^{(vert)} + \dot{Q}_j^{(loss)} + \dot{Q}_j^{(hx)} + \dot{Q}_j^{(aux)} \quad \text{Gl. 2-3}$$

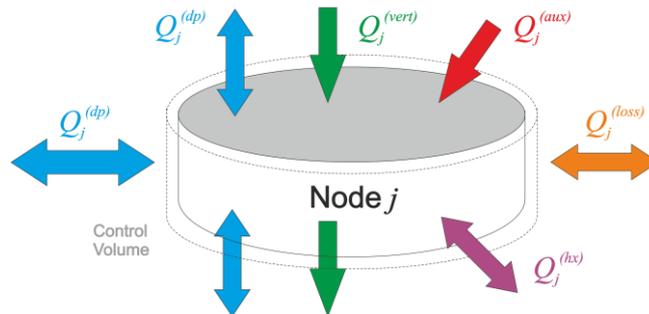


Abbildung 2-2: Energieströme am Knoten j , konstante Füllhöhe

Die in Gl. 2-3 angegebenen Energieströme sind in Abbildung 2-2 schematisch dargestellt und werden nachfolgend im Detail beschrieben.

Beschreibung der einzelnen Energieströme im Detail

a) Energie durch Doppelanschluss

Dieser Term beschreibt den Eintrag von thermischer Energie in den jeweiligen Speicherknoten, sowie den Wärmetransport durch Konvektion im Speicher (Gl. 2-4).

Der Schalter $\delta_{ic,j}$ weist die eintretende Energie nur dem Einlassknoten zu. Die Richtung des Wärmeflusses (nach oben oder unten) wird vom Schalter d_c bestimmt. Der Schalter $\epsilon_{ic,j,oc}$ begrenzt den Wärmefluss auf die vom Doppelanschluss betroffenen Knoten.

$$\dot{Q}_j^{(dp)} = \sum_c \dot{m}_{c,in} \cdot \left[(h_{c,in} - h_j) \cdot \delta_{i_c,j} + (h_{j+dc} - h_j) \cdot \varepsilon_{i_c,j,o_c} \right] \quad \text{Gl. 2-4}$$

$$\delta_{i_c,j} = \begin{cases} 1 & \text{wenn } i_c = j \\ 0 & \text{wenn } i_c \neq j \end{cases}, \quad d_c = \begin{cases} +1 & \text{wenn } i_c \geq o_c \\ -1 & \text{wenn } i_c < o_c \end{cases}$$

$$\varepsilon_{i_c,j,o_c} = \begin{cases} 1 & \text{wenn } 0 < d_c \cdot (i_c - j) \leq d_c \cdot (i_c - o_c) \\ 0 & \text{sonst} \end{cases}$$

b) Energiestrom durch vertikale Wärmeleitung (Gl. 2-6)

Die Wärmeleitung in einem Speicher kann durch die effektive Wärmeleitfähigkeit (λ_{eff}) beschrieben werden. Sie hat einen wesentlichen Einfluss auf die Schichtung in einem Speicher. λ_{eff} ist abhängig von der Geometrie, vom Material der Speicherwand, der Wärmedämmung und den zusätzlichen Einbauten im Speicher. Nach Drück (2006) ergibt sich für einen Speicher mit guter Wärmedämmung λ_{eff} näherungsweise zu (Gl. 2-5)

$$\lambda_{eff} = \frac{A_{Wasser} \cdot \lambda_{Wasser} + A_{Wand} \cdot \lambda_{Wand}}{A_{Wasser} + A_{Wand}} \quad \text{Gl. 2-5}$$

und der Energiestrom durch vertikale Wärmeleitung für den Speicherknoten j zu

$$\dot{Q}_j^{(vert)} = \lambda_{eff} \cdot \frac{A}{\Delta z} \cdot (T_{j+1} + T_{j-1} - 2 \cdot T_j) \quad \text{Gl. 2-6}$$

c) Wärmeverluste an Umgebung (Gl. 2-7)

Berücksichtigt den Wärmedurchgang (U) durch die Speicherhülle und die Dämmung.

$$\dot{Q}_j^{(loss)} = (U \cdot A)_{loss,j} \cdot (T_{amb} - T_j) \quad \text{Gl. 2-7}$$

d) Energiestrom durch Wärmetauscher (Gl. 2-8)

Beschreibt die Energieübertragung durch Leitung der Wärmetauscherknoten mit dem jeweiligen Knoten des Speichers.

$$\dot{Q}_j^{(hx)} = \sum_c (U \cdot A)_{c,j} \cdot (T_{c,j} - T_j) \quad \text{Gl. 2-8}$$

e) Elektrische Wärmequellen

Gl. 2-9 beschreibt den Energiestrom durch elektrische Wärmequellen (Heizstab).

$$\dot{Q}_j^{(aux)} = P_j \quad \text{Gl. 2-9}$$

Adaptierung der Gleichungen für eine variable Füllhöhe

Durch die Annahme, dass die Masse jedes Knoten konstant ist ($m_j = const$) und somit die Be- und Entladung durch Doppelanschlüsse (Gl. 2-4) erfolgt, können thermische Speicher mit einer variablen Füllhöhe ($m_j \neq const$) nicht abgebildet werden.

Die erste Modifikation für eine variable Füllhöhe betrifft den internen Zeitschritt des Modells. Da im Simulationszeitschritt ein Energieeintrag in den Knoten erfolgen kann, welcher den möglichen Energieinhalt ($m_j \cdot h_j$) eines Knotens (j) übersteigt, ist es notwendig den internen Zeitschritt des Modells zu reduzieren. Somit wird sichergestellt, dass die Änderung des Füllvolumens keinen Einfluss auf das Simulationsergebnis des aktuellen Zeitschritts ausübt. Nach der Berechnung des internen Zeitschritts muss eine Anpassung der Masse (m_j) und der Knotenhöhe (Δz) nach jedem internen Zeitschritt erfolgen.

Die zweite Anpassung erfolgt bei $\dot{Q}^{(dp)}$. (Gl. 2-4) berücksichtigt sowohl den eintretenden Wärmestrom durch den jeweiligen Doppelanschluss, als auch den daraus resultierenden Wärmestrom im Speicher (in den Knoten welche vom Doppelanschluss betroffen sind). Bedingt durch die variable Füllhöhe kann der eintretende Massenstrom (\dot{m}_{in}) vom austretenden Massenstrom (\dot{m}_{out}) abweichen. Somit müssen die beiden Wärmeströme im Speicher aufgeteilt werden (Gl. 2-10).

$$\dot{Q}_j^{(dp)} = \dot{Q}_j^{(in)} + \dot{Q}_j^{(flow)} \quad \text{Gl. 2-10}$$

Der eintretende Massenstrom wird $\dot{Q}^{(in)}$ zugeordnet. Dieser führt, im Falle der Beladung eines Speichers, zu einer Temperaturerhöhung im jeweiligen Eintrittsknoten (Gl. 2-11).

$$\dot{Q}_j^{(in)} = \sum_c \dot{m}_{c,in} \cdot \left[(h_{c,in} - h_j) \cdot \delta_{i_c,j} \right] \quad \text{Gl. 2-11}$$

$$\delta_{i_c,j} = \begin{cases} 1 & \text{wenn } i_c = j \\ 0 & \text{wenn } i_c \neq j \end{cases}$$

$\dot{Q}^{(flow)}$ beschreibt den Effekt des Wärmeflusses im Speicher in vertikaler Richtung sowie den Wärmefluss durch den Austritt. Für $\dot{Q}^{(flow)}$ muss jedoch der resultierende Massenstrom aller Eintritte und Austritte im gesamten Speicher berechnet werden (

Gl. 2-12)

$$\dot{Q}_j^{(flow)} = \left| \dot{m}_{j,res} \right| \cdot (h_{j-1} - h_j) \cdot \zeta_j + \left| \dot{m}_{j,res} \right| \cdot (h_{j+1} - h_j) \cdot (1 - \zeta_j) \quad \text{Gl. 2-12}$$

$$\dot{m}_{j,res} = \sum_c \dot{m}_{c,j,i_c} + \sum_c \dot{m}_{c,j,o_c}$$

$$\zeta_j = \begin{cases} 1 & \text{wenn } \dot{m}_{j,res} > 0 \\ 0 & \text{wenn } \dot{m}_{j,res} \leq 0 \end{cases}$$

Somit ergeben sich die Energieströme, für ein variables Füllvolumen, zu Gl. 2-13.

$$\sum \dot{Q}_j = \dot{Q}_j^{(in)} + \dot{Q}_j^{(flow)} + \dot{Q}_j^{(cond)} + \dot{Q}_j^{(loss)} + \dot{Q}_j^{(hx)} + \dot{Q}_j^{(aux)} \quad \text{Gl. 2-13}$$

Für die weiteren Energieströme ergeben sich keine weiteren Adaptierungen. In Abbildung 2-3 sind die Energieströme für den Fall einer variablen Füllhöhe schematisch dargestellt.

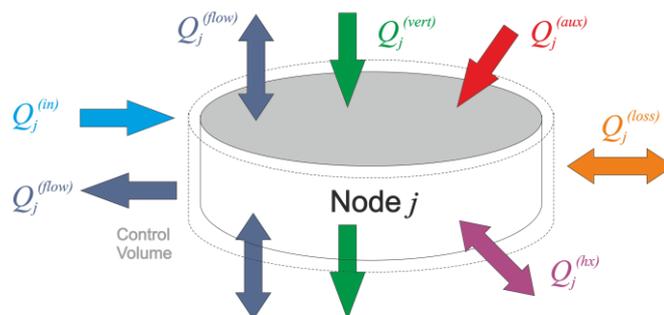


Abbildung 2-3: Energieströme am Knoten j , variable Füllhöhe

Modellbedingte Annahmen

Bedingt durch den Ansatz eines Knotenmodells mit N vollkommen gemischten Knoten wird ein eintretender Massenstrom immer den Eintrittsknoten zugeordnet. Angewendet auf einen Speicher und unter der Annahme, dass sich der Einlass an der Decke befindet, entspricht dies einer Prallplatte welche den Impuls des eintretenden Massenstromes um 90 Grad ablenkt.

Der große Nachteil der eindimensionalen Modelle ist, dass keine Strömungen im inneren berücksichtigt werden können. Ein Beispiel hierfür wäre die Fallströmung, welche sich durch den Wärmeverlust an der Wand ausbildet (Huhn, 2007) und somit zu einer rascheren Auskühlung im unteren Speicherbereich führt.

Eine weitere Schwäche ist die Vereinfachung der Geometrie. Die äußere Hülle kann in der Realität von der Form eines Zylinders abweichen, wie durch einen geneigten Deckel, oder durch gewölbte Böden (Klöpperböden). Dies kann in einem vereinfachten Modell nur durch Anpassung der Dämmstärke des Bodens berücksichtigt werden. Die aber in der Realität auftretenden Strömungsverhältnisse (Abbildung 2-4) müssen unberücksichtigt bleiben. Huhn (2007) beschrieb diesen Effekt bei der Einströmung in den Speicher, bei dem das Fluid über Prallplatten einströmt. Bei einem Zylinder sorgen die Ecken für eine turbulente Mischung, dadurch ist das Temperaturfeld bereits nach 2 min stationär, bei einem gewölbten Boden konnte es bis zu 10 min dauern (Modelle mit 162 Liter und 4370 Liter).

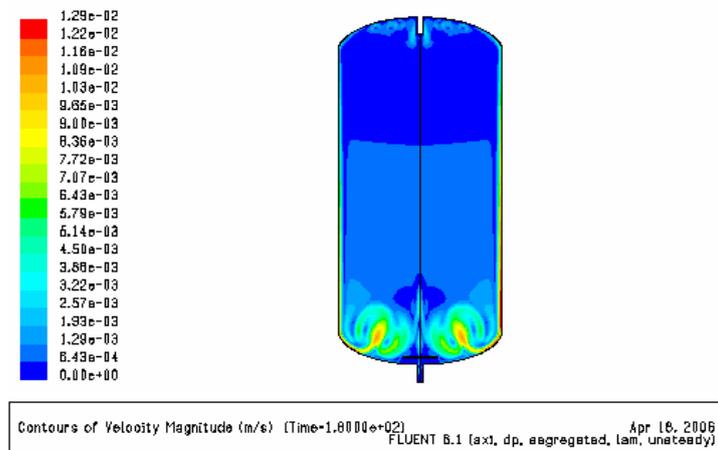


Abbildung 2-4: Geschwindigkeitsfeld (2D) im Speicher mit gewölbten Böden und Deckel (Quelle: Huhn, 2007, S104)

Auch Speichereinbauten wie Ringe und auch ihre Anordnung (Panthalookaran et al., 2007) haben in der Realität Einfluss auf die Strömung und die effektive Wärmeleitfähigkeit. Wie Panthalookaran beschreibt, bewirken Ringe, welche eine Neigung in Richtung Speicherdecke aufweisen, eine Steigerung der Effektivität des Speichers, da sie die abfallende Strömung unterbindet und für eine lokale Vermischung der kälteren Strömung mit dem wärmeren Speichermedium sorgen (Abbildung 2-5).

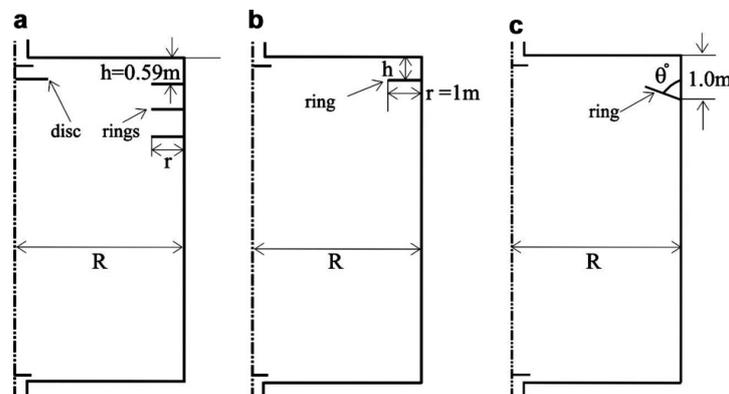


Abbildung 2-5: Übersicht über die internen Anordnungen der Experimente (Quelle: Panthalookaran et al., 2007, S1052)

Jedoch bieten eindimensionale Modelle den wesentlichen Vorteil, Jahressimulationen mit einem vertretbaren Rechenaufwand, zu simulieren. Dadurch lässt sich der Einfluss von optimierten Energiespeichern in thermischen Systemen studieren, um zum Beispiel für einen gesamten Industrieprozess Verbesserungsvorschläge durch Simulationsuntersuchungen auszuarbeiten.

2.1.2 Mehrdimensionale Modelle

Mit mehrdimensionalen Modellen lassen sich auch Strömungsvorgänge und Turbulenzen im Speicher simulieren. Dadurch lassen sich Veränderungen in der Einlassgeometrie, sowie Speichereinbauten und ihre Auswirkungen auf den Speicher beobachten. In Abbildung 2-6 ist eine Fallströmung durch ein Rohr von 25 mm Durchmesser abgebildet. Das im Rohr auskühlende Wasser sinkt bis zu 100 mm tief in einen homogen geschichteten Speicher ab (Huhn, 2007).

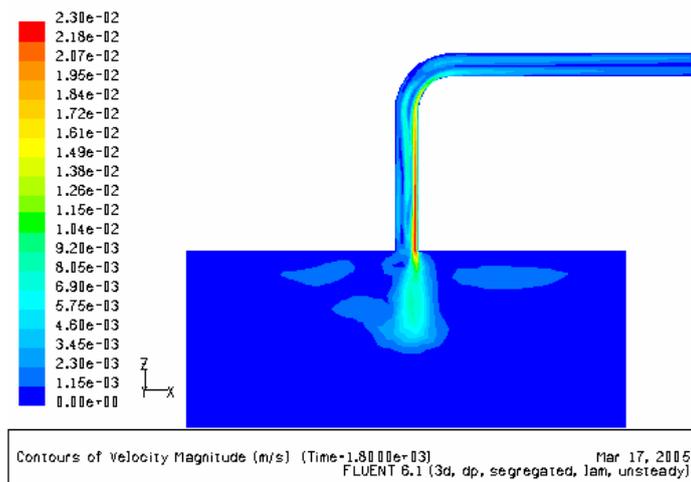


Abbildung 2-6: Geschwindigkeitsfeld im Speicher mit Anschlussrohr im Deckel, Wärmeverlust am Anschlussrohr $3,5 \text{ W} / (\text{m}^2 \cdot \text{K})$, Simulation

(Quelle: Huhn, 2007, S108)

Die Betrachtung von mehrdimensionalen Modellen findet in der vorliegenden Arbeit keine weitere Beachtung, da eine derart detaillierte Modellierung nicht den Vorgaben eines einfachen physikalischen Modells zur Berechnung eines thermischen Energiespeichers im Rahmen einer Monats- oder Jahressimulation entspricht.

2.2 Simulationsumgebung TRNSYS

Die Simulationsumgebung TRNSYS ein Programmpaket zur Simulation von instationären Energiesystemen. Der Einsatzbereich von TRNSYS umfasst sowohl thermische, als auch elektrische Energiesysteme und erlaubt die Simulation von dynamischen Prozessen (www.trnsys.com, 26.10.2012).

„TRNSYS wurde ursprünglich am Solar Energy Laboratory (SEL) der Universität von Wisconsin entwickelt. Es diente dort zur dynamischen Simulation von thermischen Solarenergie-Nutzungssystemen in der Programmiersprache FORTRAN 77. Durch die regelmäßige Weiterentwicklung und Aktualisierung wird TRNSYS international für wissenschaftliche und kommerzielle Projekte eingesetzt.“ [Altenburger, 2010, S32].

Die klassischen Anwendungsgebiete von TRNSYS liegen in der dynamischen Gebäude- und Anlagensimulation. So stehen Mehrzonengebäude mit integrierten Modellen für thermoaktive Bauelemente wie Betonkernaktivierung, Fußbodenheizung und Kühldecken zur Verfügung, welche mit der umfangreichen Standard-Komponenten Bibliothek, zur einer Anlagensimulation, kombiniert werden können (http://www.trnsys.de/docs/trnsys/trnsys_uebersicht_de.htm, 26.10.2012). Da sich diese Arbeit im Speziellen mit einer reinen Anlagensimulation beschäftigt, wurde das Gebäudemodell nicht näher betrachtet.

Die grafische Benutzeroberfläche von TRNSYS - das sogenannte Simulation Studio - gibt dem Anwender die Möglichkeit grafisch orientiert mit der Simulationsumgebung zu arbeiten. Die verschiedenen physikalischen Anlagenkomponenten die in TRNSYS Types genannt werden, können hier zu einer gesamten Anlagensimulation zusammengestellt werden (siehe auch Abbildung 2-13). Jeder Type ist durch ein mathematisches Modell beschrieben. Der Programmcode der Types ist in FORTRAN 90 (TRNSYS 17) geschrieben welcher im Rahmen des Programmes frei verfügbar ist. Jeder Type besitzt ein Proforma, welches quasi die Eingabemaske in der Benutzeroberfläche darstellt. Ein Proforma besteht zumindest aus drei Teilen, Eingabegrößen, Ausgabegrößen und den Parametern. In der Simulationsumgebung können die verschiedenen physikalischen Modelle miteinander verknüpft werden.

„Durch den modularen Ansatz wird die Lösungsstrategie der klassischen Handrechnung auf das Simulationsprogramm übertragen. Infolgedessen, ist der Lösungsweg für den Benutzer gedanklich besser nachvollziehbar. Aus der Übersetzung des Eingabefiles in die Programmiersprache und die anschließende Kompilierung ergibt sich das Hauptprogramm. Von diesem werden die Unterprogramme der Komponentenmodelle verwaltet und die Berechnungsreihenfolge mit den zugehörigen Konvergenzkriterien festgelegt.“ [Altenburger, 2010, S32].

Vor dem Beginn jeder Simulation wird im Simulation Studio eine Text-Datei mit der Endung .dck erstellt, welche im Folgenden Deck genannt wird. In dieser Datei werden alle wesentlichen Informationen einer Simulation zusammengefasst. Diese umfassen die allgemeinen Simulationsbedingungen, sowie die Eigenschaften und Verknüpfungen der Types (Abbildung 2-7).

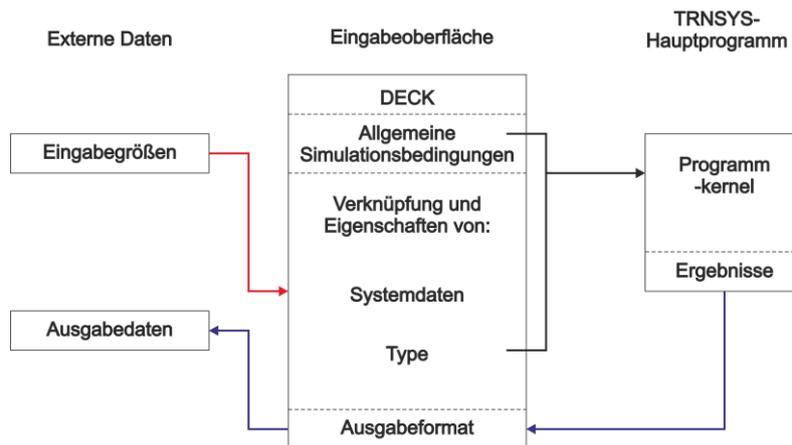


Abbildung 2-7: TRNSYS – Struktur (angelehnt an Heimrath 2004)

„Das Lösungsverfahren in TRNSYS ist ein iterativer Gleichungslöser. Dabei wird versucht, für jeden Zeitschritt eine Konvergenz aller Komponentengrößen und Differentialgleichungen zu erreichen, wobei die Abarbeitung der Komponenten in einer festgelegten Reihenfolge stattfindet. Eine Iteration innerhalb eines Zeitschrittes erfolgt solange, bis sich Eingangs- und Ausgangsgrößen aller Komponenten nur noch unwesentlich von einer Iteration auf die Nächste ändern. Hierbei werden aber nur jene Komponenten neu berechnet die nicht konvergiert sind. Nach einer erfolgten Konvergenz, d.h. Übergang zum nächsten Zeitschritt, werden die Ausgangsgrößen der vorherigen Iteration zu Eingangsgrößen der aktuellen Iteration.“ [Altenburger, 2010, S33].

2.2.1 Mathematisch- physikalisches Modell

Ein Type wird in TRNSYS im Wesentlichen durch drei Informationsflüsse beschrieben (siehe Abbildung 2-8). Parameter stellen Werte dar, die über die Simulationszeit konstant sind, bzw. zur Initialisierung dienen. Werte die sich mit jedem Zeitschritt ändern können, werden als Input bezeichnet. Die Vorgabe dieser Werte kann durch Verknüpfen anderer Types (z.B.: Datenleser, Pumpen) erfolgen. Die Initialisierung des Wertes (Temperatur, Massenstrom) erfolgt mit dem Wert der im Proforma angegeben wird. Sollte keine Verknüpfung vorliegen, so wird der Wert im Proforma als konstant (über die gesamte Simulationszeit) angenommen. Die Outputs stellen die

Rückgabewerte des Types dar und können ausgegeben werden (Printer oder Plotter) oder als Input für einen weiteren Type dienen.

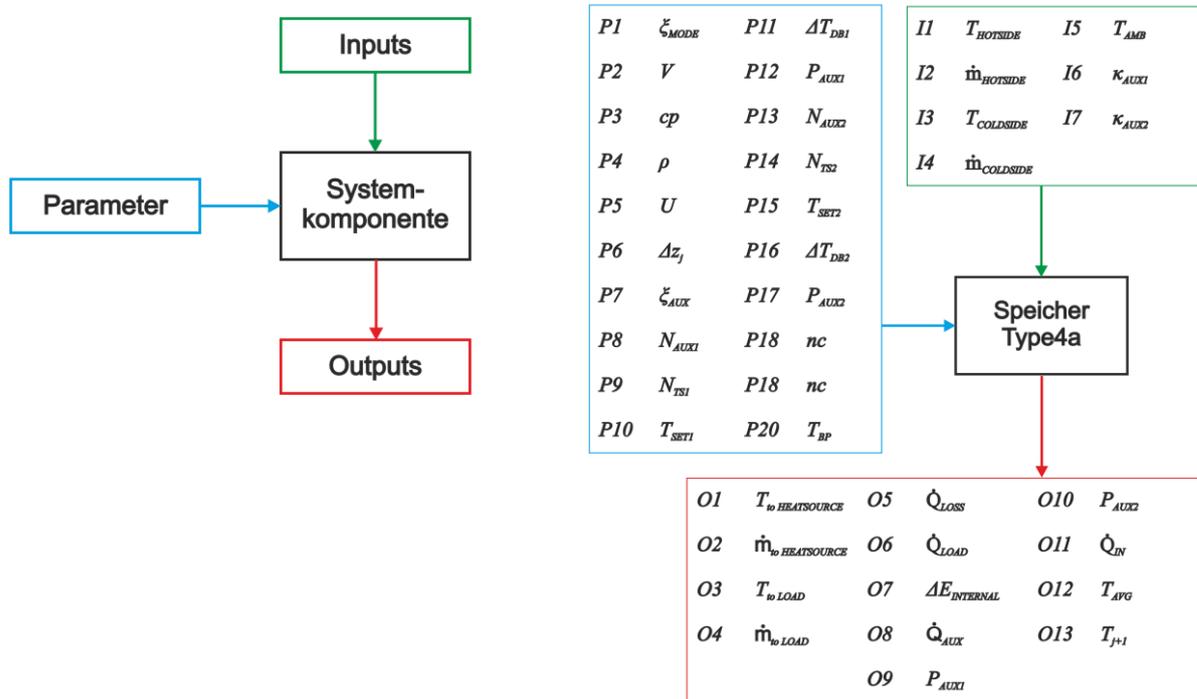


Abbildung 2-8: Beispielhafter Informationsfluss für ein einfaches Speichermodell

In Abbildung 2-8 sind die Informationsflüsse eines einfachen Speicher-Types (Type4a) dargestellt. Die Parameter dieses Types bestehen aus allgemeinen Parametern wie der Betriebsweise (ξ_{MODE}), dem Volumen (V), der Höhe des Knotens (Δz), den Stoffgrößen Dichte (ρ) und Wärmekapazität (cp) des Speichermediums (in der Regel ist das Wasser) und der Wärmedurchgangszahl (U) der Speicherhülle. Die weiteren Parameter (7 – 17) dienen zur Parametrierung der Heizstäbe. Parameter 20 ermöglicht die Angabe einer Siedetemperatur (T_{BP}) und stellt den Wert dar, wo im Regelfall ein Überdruckventil öffnen würde.

Die Inputs bestehen aus den eintretenden Temperaturen und Massenströmen auf Seite der Wärmequelle ($T_{HEATSOURCE}$, $\dot{m}_{HEATSOURCE}$), sowie auf Seite der Wärmesenke (T_{LOAD} , \dot{m}_{LOAD}). Weiters ist die Vorgabe einer Kontrollfunktion für beide Heizstäbe möglich (κ_{AUX1} , κ_{AUX2}).

Die Outputs bestehen aus den austretenden Größen wie Temperatur ($T_{to HEATSOURCE}$, $T_{to LOAD}$), Massenstrom ($\dot{m}_{to HEATSOURCE}$, $\dot{m}_{to LOAD}$) und Leistung (\dot{Q}_{LOAD} , \dot{Q}_{IN}). Zur genaueren Analyse und Betrachtung des Speichers kann die Temperatur der Speicherknoten (T_{j+1}), die übertragene Leistung der Heizstäbe (\dot{Q}_{AUX} , P_{AUX1} , P_{AUX2}), die mittlere Speichertemperatur (T_{AVG}) und die Änderung der inneren Energie ($\Delta E_{INTERNAL}$) ausgegeben werden.

2.2.2 Informationsflüsse im Simulationsaufbau

Für den Simulationsaufbau der zum Testen der Speichermodelle verwendet wurde (siehe Abbildung 2-13) ergibt sich der in Abbildung 2-9 gezeigte Informationsfluss (hier als Speichermodell der Type 4e). Der zuvor in Abbildung 2-8 dargestellte Type ist nun ein Teil des Gesamtsystems. Er erhält seine Inputs ($T_{in,DP1}$, $T_{in,DP2}$, $\dot{m}_{in,DP1}$, $\dot{m}_{in,DP2}$) aus der Equation (Unit 5) und gibt wiederum die Outputs ($T_{out,DP2}$, \dot{Q}_{LOAD} , \dot{Q}_{IN} , \dot{Q}_{LOSS} , ΔE_{CHANGE}) an die Equation zurück. Dieser Testaufbau wird später in Kapitel 2.5 näher beschrieben.

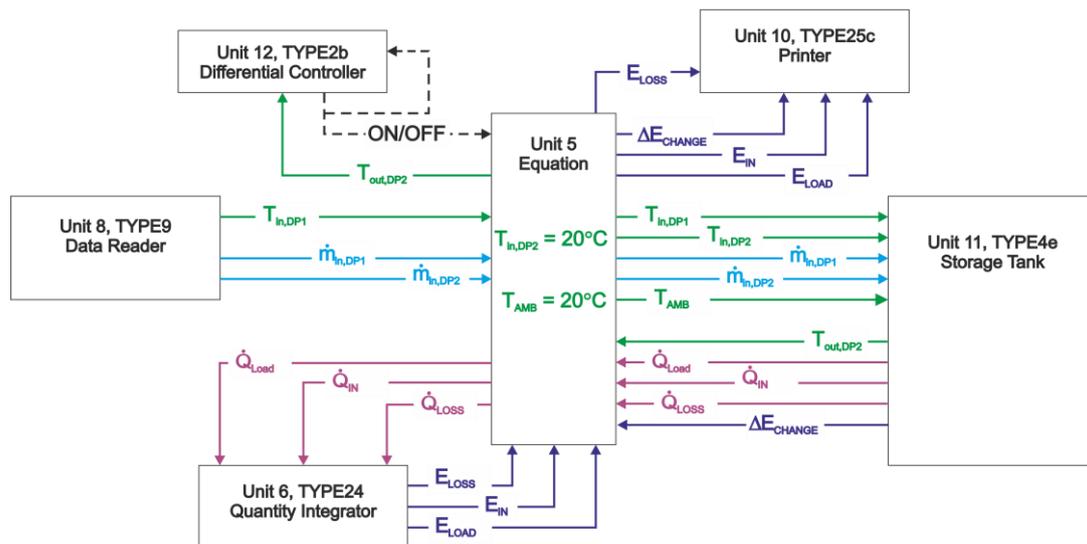


Abbildung 2-9: Informationsfluss für den Testaufbau des Prüfzyklus (Type4e)

2.3 Speichermodelle in TRNSYS

Alle Types in TRNSYS besitzen für die Einbindung und Parametrierung in der grafischen Benutzeroberfläche Simulation Studio einen Platzhalter, Proforma, genannt. Es ist auch möglich, dass ein Modell mehrere Proforma hat, die unterschiedliche Betriebsweisen ermöglichen. Erkennbar ist dies durch Kleinbuchstaben oder Ziffern, welche hinter der Modellbezeichnung angefügt werden (z.B. Type4b oder Type15-2). Jedes Proforma kann somit Parameter sperren oder zusätzliche Parameter freigeben (Abbildung 2-10).

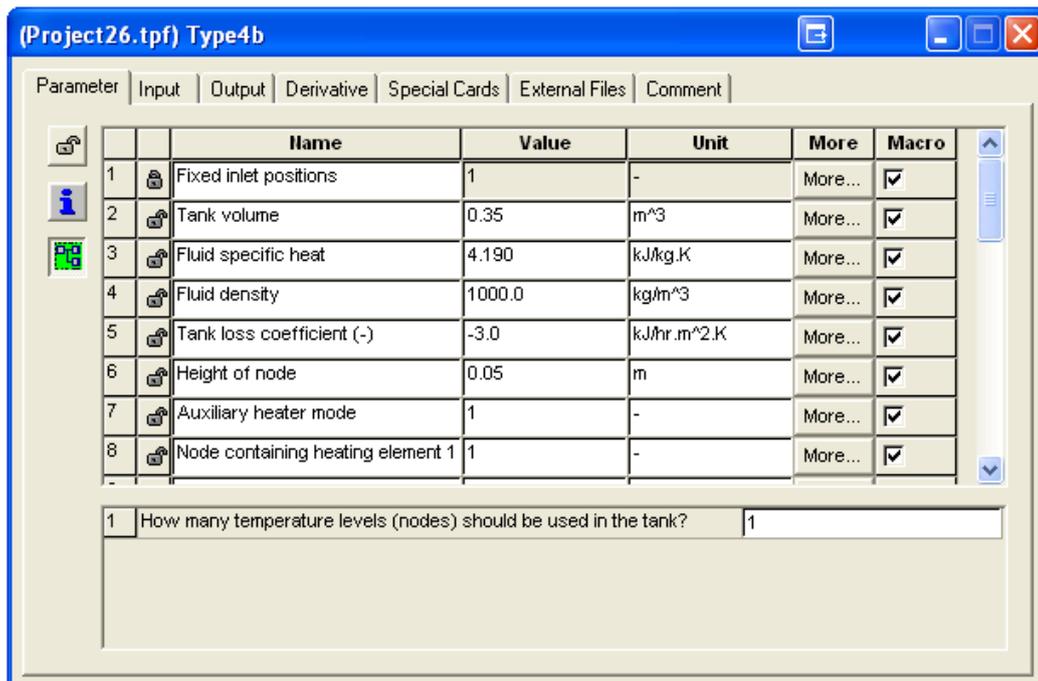


Abbildung 2-10: Proforma vom TRNSYS-Type4b (Schichtspeicher)

Ein Proforma besteht aus mehreren Registerkarten, siehe Tabelle 2-1.

Tabelle 2-1: Registerkarten des Proformas von TRNSYS -Types4

Name	Beschreibung
Parameter	Parameter, konstant über Simulationsdauer
Input	Eingabegrößen, Änderung mit jedem Simulationszeitschritt möglich
Output	Ausgabegrößen, für jeden Simulationszeitschritt
Derivative	Zusätzliche Parameter
Special Cards	Zusätzliche Eingaben, wenn notwendig
External Files	Angabe zur Steuerung durch externe Dateien (Textdateien)
Comments	Kommentare zum Modell

Auf den folgenden Seiten werden die verschiedenen Speichermodelle, welche im Rahmen dieser Arbeit betrachtet wurden genauer beschrieben. Modelle die nicht im Standardpaket von TRNSYS vorkommen werden als „NonStandardTypes“ bezeichnet.

2.3.1 Schichtspeicher (Type4)

Der Type4 ist ein geschichteter Speicher, welcher einen vollständig gefüllten Speicher darstellt. Für jeden Knoten kann eine Initialisierungstemperatur (in den zusätzlichen Parametern) angegeben werden. Die Angabe der Dichte, der Wärmekapazität und der Siedetemperatur des Speichermediums ist möglich. Der Speicher besitzt zwei Doppelanschlüsse, wobei die Höhe der Auslässe nicht veränderbar ist. Wärmequellenseitig ist der Anschluss im untersten Knoten,

wärmesenkenseitig befindet er sich im obersten Knoten. Die Einlasshöhen sind je nach Proforma veränderbar und werden in drei Betriebsweisen (Modes) eingeteilt (TRNSYS, 2011).

Mode1 (Type4a, b) rechnet mit fixen Einlasshöhen am oberen Ende des Speichers (Wärmequelle) und auf der Unterseite des Speichers (Wärmesenke).

Mode2 (Type4c, d) entspricht einer Schichtladungseinheit, welche den Massenstrom immer auf der Höhe einschichtet, in der der geringste Unterschied der Temperaturen vorliegt.

Mode3 (Type4e, f) lässt dem Anwender die Möglichkeit die Einlasshöhen frei zu wählen.

Die Knotenhöhe lässt sich für jeden Knoten individuell angeben, wobei die Gesamthöhe aus der Summe der Knotenhöhen gebildet wird. Sollten durch den Beladungsprozess inverse Temperaturschichten entstehen, so werden diese nach jedem Zeitschritt durch Mischen der betreffenden Knoten, ausgeglichen. Der Verlustbeiwert (Tank loss coefficient) kann für den gesamten Speicher (Type4a, c, e), oder für jeden einzelnen Knoten (Type4b, d, f) angegeben werden. Es ist möglich zwei Heizelemente anzugeben, welche noch zusätzliche Parameter wie die Position, die Leistung und Temperaturen zur Regelung der Nachheizung benötigen.

Die Knotenanzahl einer Instanz ist auf Hundert begrenzt. Wenn mehrere Instanzen des Type4 in einer Simulation eingesetzt werden darf auch die Summe aller Knoten der verwendeten Instanzen, Hundert nicht übersteigen. Dies ist begründet in der maximalen Anzahl von zusätzlichen Parametern (Derivative) die in einer TRNSYS – Simulation verwendet werden dürfen. Die vertikale Wärmeleitung ($\dot{Q}^{(vert)}$) wird im Type4 im Gegensatz zu allen folgenden Speichermodellen nicht berücksichtigt.

2.3.2 Detaillierter Flüssigkeitsspeicher (Type60)

Der Type60 lässt eine umfangreichere Parametrierung als der Type4 zu, welche sich auch in der Anzahl der verschiedenen Proformas äußert (Type60a – Type60x). Es ist möglich zwischen einem horizontalen und einem vertikalen Zylinder, sowie einem Speicher mit gleichbleibendem Querschnitt (z.B. Ovale Grundfläche, dann muss der Umfang angegeben werden) zu wählen. Der Speicher wird als vollständig gefüllt angenommen, die Knotenhöhen und die Initialisierungstemperaturen können für jeden Knoten individuell oder für alle Knoten einheitlich angegeben werden, jedoch bedingen variable Knotenhöhen auch die Angabe des Verlustbeiwertes für jeden Knoten. Die Dichte, die Wärmekapazität und die Siedetemperatur des Speichermediums können ebenso angegeben werden (TRNSYS, 2011).

Die Angabe zweier Doppelanschlüsse ist möglich, wobei ein Doppelanschluss auch aus zwei Einlässen und einem Auslass, oder zwei Auslässen und einem Einlass bestehen kann. Der Speicher lässt zwei Modes zu. Mode1 erlaubt die geschichtete Beladung des Speichers, während Mode2 die Ladung auf, der vom Anwender, definierten Höhe, durchführt. Sollte sich dabei eine inverse Temperaturschichtung ergeben, so wird diese am Ende des Zeitschritts durch Mischen der betreffenden Knoten ausgeglichen. Neben den zwei möglichen

Heizelementen und des eingebauten Gasbrenners können bis zu drei Wärmetauscher integriert werden.

Auch lässt sich der interne Zeitschritt des Modells ändern, was sich auf die Genauigkeit auswirkt. Das mathematische Modell ähnelt dem, welches bereits im Kapitel 2.1.1 erläutert wurde.

Die Anzahl der verwendeten Instanzen in einer Simulation ist nicht begrenzt (Type60I mit vier Instanzen getestet), die Summe der Knoten muss kleiner gleich Hundert sein und die Knotenzahl einer einzelnen Instanz darf vierundsiebzig (getestet mit Type60I) nicht übersteigen. Es muss darauf geachtet werden, dass die Knotenzahl zweimal im Proforma angegeben werden muss.

2.3.3 Ideal geschichteter Speicher (Type38)

Der Type38 beruht im Gegensatz zu den anderen hier vorgestellten Speichern nicht auf dem Knotenmodell, sondern teilt den Speicher in Segmente, welche in ihrer Größe verschieden sind und sich zu jeder Simulationszeit verändern können (Plug Flow Ansatz). Der Speicher besitzt zwei Doppelanschlüsse wobei nur die Höhe am Eintritt auf der Seite der Wärmequelle beeinflusst werden kann. Zur Vorgabe dieser Höhe muss sich der Speicher in der Betriebsweise Mode1 befinden, mit Mode2 wäre die Einlasshöhe auf der Höhe, auf der ohne Temperaturinversion eingeschichtet werden kann. Da im Gegensatz zu einem Knotenmodell keine Mischvorgänge im jeweiligen Knoten notwendig sind, entspricht dies einer idealen Schichtung im Speicher. Der Austritt befindet sich am untersten Punkt des Speichers. Auf der Seite der Wärmesenke sind die Eintrittshöhe am untersten Punkt und die Austrittshöhe am obersten Punkt (TRNSYS, 2011).

Der Speicher ist immer vollständig gefüllt, die Dichte und Wärmekapazität des Speichermediums können angegeben werden. Die Abbildung eines Heizelements mit den nötigen Regelparametern ist möglich. Es kann zwischen einem vertikalen und einem horizontalen Speicher gewählt werden, wobei auch die Möglichkeit einer ungleichmäßigen Dämmung besteht. Bei einem stehenden Speicher kann das Verhältnis zwischen der Dämmstärke des Deckels und des Mantels angegeben werden. Für einen liegenden Zylinder ist die Angabe eines Verhältnisses, zwischen der Dämmstärke an der Oberseite und an der Unterseite, möglich. Der Verlustbeiwert kann nur für den gesamten Speicher angegeben werden. Es ist nur eine Instanz in einer Simulation möglich.

2.3.4 MULTIPORT – Speichermodell (Type340)

Der Type340, ein „NonStandardType“, wurde am Institut für Thermodynamik und Wärmetechnik (ITW) der Technischen Universität Stuttgart, entwickelt und wird auch für Speichertests am ITW verwendet (Drück, 2006a). Somit stellt der Type340 quasi eine Referenz für Speichermodelle dar. Das Volumen des Speichers entspricht auch hier dem Füllvolumen, die Dichte, die Wärmekapazität des Speichermediums und effektive Wärmeleitfähigkeit kann angegeben werden.

Die Doppelanschlüsse (maximal zehn Stück) können in der Höhe frei positioniert werden und sowohl normal als auch geschichtet Beladen und Entladen. Für geschichtetes Beladen muss jedoch der Einlass über dem Auslass liegen, für geschichtetes Entladen der Auslass über dem Einlass. In dem dazwischenliegenden Bereich kann geschichtet Beladen und Entladen werden. Zum Beladen und Entladen können, statt der Doppelanschlüsse, auch bis zu vier Wärmetauscher (interne oder externe) verwendet werden.

Die möglichen Parameter reichen von internen und externen Wärmetauschern (Mantelwärmetauscher), bis hin zu einem „Tank im Tank“- Ansatz, wie er in Solar- und Heizungsspeichern als Boiler eingesetzt wird (als Beispiel: <http://www.jenni.ch>). Jedoch dürfen sich die Wärmetauscher 1 und 4, sowie 2 und 3, nicht im selben Knoten befinden. Diese Einschränkung gilt nicht für 1 und 2, respektive 1 und 4. Für den Fall eines Mantelwärmetauschers kann auch geschichtet beladen und entladen werden, jedoch werden dann die Doppelanschlüsse 7-10 benützt (zum Beispiel Wärmetauscher 1 verwendet virtuell Doppelanschluss 7). Zur Beheizung ist auch ein Heizstab vorhanden, welcher in der Höhe, als auch in der Lage (vertikal und horizontal) verändert und durch die Vorgabe der momentanen Heizleistung geregelt werden kann.

Die Verluste können entweder durch einen Verlustbeiwert für den gesamten Speicher, oder für jeden Bereich einzeln (Decke, Zone 1- 4, Boden), angegeben werden. Bei der Verwendung eines einheitlichen Verlustbeiwerts muss dieser bei der Zone 1 angegeben werden. Bei einem nicht einheitlichen Verlustbeiwert kann dieser getrennt für Boden, Decke und Mantel erfolgen. Beim Mantel ist die Einteilung in 4 Bereiche über die relative Höhe möglich, jedoch muss die Summe der relativen Höhen gleich eins sein, wobei sich Zone 4 aus der Differenz zur Zahl 1 der relativen Höhen davor ergibt.

Das Modell besitzt Parameter, welche die Genauigkeit der Berechnung und die Temperaturabhängigkeit beeinflussen. Als wichtiger Parameter hat sich der Parameter 123, „precision of the mixing process in the store“, herausgestellt. Dieser stellt die Genauigkeit des Mischprozesses dar, falls inverse Temperaturschichten vorliegen. Dieser Mischprozess beeinflusst maßgeblich die Energieübertragung an den Wärmetauschern. In der Beschreibung (Drück, 2006a) sollten mögliche Werte zwischen 10 und 100 liegen. In der Realität hat sich aber herausgestellt, dass dieser Parameter äußerst relevant für eine erfolgreiche Simulation ist. Für einfache Simulationen macht es oft keinen Unterschied, welcher Wert hierbei gewählt wird.

Wenn mehrere Wärmetauscher zugleich in Betrieb sind, kann es aber vorkommen, dass die Simulation abbricht. Wenn der Parameter dann auf einen größeren Wert (1000 und mehr) eingestellt wird kann das Gesamtsystem simuliert werden. Die Knotenzahl ist laut Beschreibung auf 200 Knoten limitiert, jedoch kann dieser Wert nicht mit jedem Gesamtsystem voll genutzt werden.

In Abhängigkeit vom Gesamtsystem variiert die maximale Knotenanzahl zwischen 198 und 200 Knoten.

2.3.5 Speicher mit Phasenwechselmaterial (Type840)

Der Type840, ebenfalls ein „NonStandardType“, wurde am Institut für Wärmetechnik (IWT) für die Simulation von Speichern mit integrierten Phasenwechselmaterialien (PCM) entwickelt (Schranzhofer et al., 2006). Es stellt einen vollständig gefüllten Speicher (maximal 200 Knoten) dar, der mit PCM- Modulen bestückt werden kann. Die Angabe von Wasser als Speichermedium ist möglich, jedoch kann die Dichte und die Wärmekapazität nicht explizit angegeben werden (die Stoffdaten für Wasser sind derzeit temperaturabhängig hardcodiert im Modell implementiert), jedoch sind auch Phasenwechselmaterialien (derzeit für drei Bereiche im Speicher) als Speichermedium möglich. Die effektive Wärmeleitfähigkeit umfasst hier das Speichermedium, die Speicherwand und alle Einbauten, wie interne Wärmetauscher und PCM-Module.

Es ist möglich bis zu zehn Doppelanschlüsse anzugeben, welche auch als interne Wärmetauscher fungieren können. Eine geschichtete Beladung ist im Falle von Doppelanschlüssen möglich, wobei die Ein- und Auslasshöhen variabel sind. Bei der Angabe eines internen Wärmetauschers kann die Geometrie, wie Höhe und Durchmesser, sowie die Wärmeleitfähigkeit und Wärmeübergangskoeffizient angegeben werden. Ein Heizstab ist in der Höhe frei positionierbar und kann über zugeführte Leistung geregelt werden. Der Verlustbeiwert an die Umgebung kann nur für den gesamten Speicher angegeben werden.

Die Besonderheit dieses Speichers ist, dass man im Inneren des Speichers PCM- Module simulieren kann. Dazu kann man drei verschiedene Geometrien (Zylinder, Platten und Kugeln), sowie unterschiedliche Phasenwechselmaterialien frei definieren. Die Anzahl der Module, als auch ihre Parameter (Durchmesser oder Dicke und Wandstärke) müssen angegeben werden. Mehrere Instanzen dieses Modells, innerhalb einer TRNSYS - Simulation, sind möglich (bis zu vier Speicher in einer Simulation wurden getestet).

2.3.6 Speicher mit variabler Füllhöhe (Type39)

Type39 ist ein vollständig gemischter Speicher (1 Knoten, keine Wärmeleitung) und besitzt ein variables Volumen. Zur Regelung kann ein minimales und ein maximales Füllvolumen vorgegeben werden, die Überlaufstrategie hängt von der Betriebsweise ab. In Mode1 tritt das Volumen in den Speicher ein und tritt mit der Mischtemperatur des Speichers wieder aus. In Mode2 wird nur das Volumen angenommen, das im Speicher noch Platz hat und der restliche Anteil wird am Speicher vorbeigeleitet. Die Angabe der Dichte, der Wärmekapazität und des Startvolumen des Speichermediums ist möglich und es kann ein Verlustbeiwert getrennt für den Wasserbereich und den Luftbereich angegeben werden (TRNSYS, 2011).

Das Modell kann eine vollständige Entleerung nicht abbilden (getestet mit einem Volumen von 10 m³ und einem austretenden Massenstrom von 10000 kg/h, wobei eine Restmenge von 0,01 m³ nicht unterschritten wird).

2.3.7 Speicher mit variabler Füllhöhe (Type844)

Der Type844 ist eine Neuentwicklung am IWT (Heinz et al., 2012). Dieser „NonStandardType“ stellt einen in der Füllhöhe variablen Speicher dar. Er entspricht im wesentlichen dem in Kapitel 2.1.1 vorgestellten Speichermodell mit variabler Füllhöhe. Die Eingabe von Dichte, Wärmekapazität des Speichermediums und einer effektiven Wärmeleitfähigkeit ist möglich. Im Gegensatz zu allen hier vorgestellten Speichern ist es möglich mittels Textdatei die Starttemperatur jedes Knotens vorzugeben. Das Modell bietet 4 Anschlüsse zum Beladen und Entladen, wobei der Massenstrom an den Eingängen unabhängig vom Massenstrom an den Ausgängen ist (keine Doppelanschlüsse).

Ein interner Wärmetauscher lässt sich hinsichtlich seiner Parameter (Position, Dichte, Wärmekapazität und Wärmeübergangszahl) definieren. Ein Heizelement ist nicht im Modell vorgesehen. Zur Berechnung der Verluste lässt sich die Dicke der Dämmung für die Speicherdecke, Speicherboden und Speichermantel einzeln angeben. Die Angabe der Wärmeleitfähigkeit der Dämmung ist möglich. Im Modell wird bei der Berechnung auch zwischen dem mit Wasser und Luft gefüllten Bereich unterschieden. Zusätzlich können die Anschlussverluste für jeden Anschluss extra angegeben werden. Somit lassen sich die Verluste durch ungedämmte Flansche und Rohrleitungen berücksichtigen.

Mehrere Instanzen dieses Typs sind innerhalb einer Simulation möglich (zwei getestet).

Das Modell beinhaltet keine Überlaufstrategie, jedoch kann eine minimale und eine maximale Füllhöhe (relative Höhe) angegeben werden. Sollte der obere Grenzwert überschritten, oder der untere Grenzwert unterschritten werden, wird eine Warnung generiert. Die TRNSYS-Simulation bricht nach einer in der Simulationsumgebung definierten Zahl von Warnungen, die Simulation ab.

2.4 Verwendete Modelle in TRNSYS

2.4.1 Regler (Type2b)

Der Regler Type2b generiert ein Kontrollsignal (0 oder 1), in Abhängigkeit von zwei Temperaturen (obere Eingangstemperatur T_H , untere Eingangstemperatur T_L) und den Differenzen der Grenztemperaturen zueinander (obere Grenztemperaturdifferenz ΔT_H , untere Grenztemperaturdifferenz ΔT_L). Dieser Regler besitzt eine Hysterese, wodurch dieser Type eine Rückkopplung seines Kontrollsignals benötigt (Input control function), d.h. das Kontrollsignal des letzten Zeitschritts dient im darauffolgenden als Eingangssignal. Durch seine Parametrierung kann der Type2b sowohl zum Einschalten, als auch zum Ausschalten verwendet werden. Die Bedingungen hierfür sind im Mathematical Reference von TRNSYS 17 zu finden.

Der Regler besitzt auch die Möglichkeit eine Überwachungstemperatur (T_{in}) als Eingangsgröße anzugeben, welche das Ausschalten des Kontrollsignal (Wert 0) bei einer definierten Temperatur (Übertemperatur, High limit cut-off) ermöglicht. Die maximale Anzahl der Iterationen innerhalb eines Zeitschritts bevor das Kontrollsignal seinen Wert ändert, kann angegeben werden, jedoch sollte diese Zahl eine ungerade (Rest) sein (TRNSYS, 2011).

Dieses Modell kann neben der Temperatur, auch für jegliche andere Werte verwendet werden (z.B. Strahlung,...).

2.4.2 Einfacher Massenstromteiler (Type11f)

Der Type11f stellt das Modell eines einfachen Massenstromteilers (Flow Diverter) dar. Die Eingangsgrößen dieses Modells bestehen aus der Temperatur und dem Massenstrom eines Fluides (Wasser, Glykol) und der Kontrollfunktion, welche Werte zwischen Null und Eins annehmen kann. In Abhängigkeit dieser Kontrollfunktion wird der eingehende Massenstrom zwischen zwei Abgängen aufgeteilt. Wird der Wert der Kontrollfunktion zu Null, dann wird der gesamte Massenstrom durch den Abgang 1 geleitet. Bei einem Wert von 1 fließt der gesamte Massenstrom durch Abgang 2. Die Temperatur an den Austritten entspricht der Eintrittstemperatur (TRNSYS, 2011).

2.4.3 Rohrleitungsmodell (Type709)

Das Rohrleitungsmodell Type709 bildet eine gedämmte Rohrleitung nach dem Plug-Flow-Ansatz ab. Nach diesem Ansatz wird die Rohrleitung in maximal 25 Segmente eingeteilt, wobei keine Wärmeleitung zwischen den Segmenten berücksichtigt wird. Das eintretende Fluid bildet ein neues Segment und schiebt die bereits vorhandenen Segmente in Strömungsrichtung weiter. Es erfolgt keine Durchmischung innerhalb der Segmente. Sollte die Anzahl von 25 Segmenten überschritten werden, so werden zwei benachbarte Segmente, welche den

geringsten Temperaturunterschied aufweisen, zusammengefasst. Die Stoffgrößen (Dichte (ρ), Wärmekapazität (c_p),...) für das Fluid können in den Parametern angegeben werden. Zur Berechnung der Wärmedurchgangszahl (U) wird die Dicke und die Wärmeleitfähigkeit (λ_{ins}) der Dämmung benötigt. Die Rohrleitung ist definiert durch den Innendurchmesser, den Außendurchmesser, der Länge und der Wärmeleitfähigkeit (λ_{st}) des Rohrleitungsmaterials. Dieses Modell benötigt als Input neben der Eintrittstemperatur, dem Massenstrom und der Außentemperatur zusätzlich die Angabe des Wärmeübergangskoeffizienten (α_a) an der Außenseite der Dämmung (TRNSYS, 2011).

2.4.4 Rohrleitungsmodell (Type604a)

Das Rohrleitungsmodell Type604a arbeitet im Gegensatz zu Type709 nach dem Knotenprinzip, d.h. die Rohrleitung wird als eine Serie gleich großer, vollständig gemischter Volumina abgebildet. Die Rohrleitung und die Dämmung werden als eigenständige Knoten (Anzahl ist ident) in der jeweiligen Energiebilanz (massebehaftet) betrachtet. Die Anzahl der Fluidknoten muss mindestens ein ganzzahliges Vielfaches der Rohrleitungsknoten sein. Die Abbildung von horizontalen und vertikalen Rohrleitungen ist möglich. Ein weiterer Unterschied zum Type 709 besteht in der Möglichkeit die Strömung in beide Richtungen (jedoch nicht gleichzeitig) abzubilden (TRNSYS, 2011).

Bedingt durch die genauere Betrachtung des Rohres und der Dämmung ist die Angabe von einer Vielzahl von Parametern (z.B. die Temperatur der Rohrleitung und der Dämmung) nötig. Zur Bestimmung dieser Parameter sind mehrere Iterationsschritte erforderlich.

2.4.5 Erweiterter Massenstromteiler (Type647)

Der Type647 bildet, wie Type11f, einen Massenstromteiler ab, erlaubt jedoch bis zu 100 Abgänge. Ein weiterer Unterschied besteht in der Angabe der Aufteilung der Massenströme, es muss für jeden Massenstrom ein Anteil angegeben werden. Die Summe der Anteile muss gleich Eins sein. Die Temperatur an den Ausgängen entspricht der Eintrittstemperatur.

2.4.6 Mischer (Type649)

Der Mischer Type649 erlaubt bis zu 100 Eingänge und gibt die Summe der Massenströme dieser Eingänge an einem Austritt aus. Die Temperatur am Austritt entspricht der Mischtemperatur der Eingänge, welche aus der Energiebilanz der Zu- und Abflüsse des Mixers berechnet wird.

2.5 Vergleich der Speichermodelle

In Kapitel 2.3 wurde ein kurzer Überblick über Speichermodelle in TRNSYS gegeben. Nun folgt ein Vergleich der einzelnen Speichermodelle im Detail. Dazu wurde ein Be- und Entladungsvorgang mittels eines vorgegebenen Lastprofils (

Abbildung 2-12: Lastprofil zum Testen der einzelnen Speichermodelle in TRNSYS

) simuliert. Es wurden, sofern die Eingabe möglich war, dieselben Randbedingungen (

Abbildung 2-11: Aufbau des Test- Speichers

, sowie Tabelle 2-2) gewählt. Mittels dieses Lastprofils wurde auch ein Speichermodell mit konstanter Füllhöhe getestet, welches in der „Visual Basic for Application“- Umgebung von Excel (VBA), im Rahmen dieser vorliegenden Arbeit entwickelt wurde.

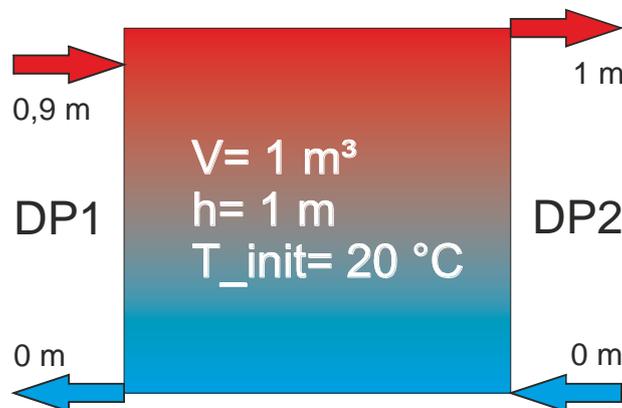


Abbildung 2-11: Aufbau des Test- Speichers

2.5.1 Beschreibung des Lastprofils

Das Lastprofil, welches in

Abbildung 2-12: Lastprofil zum Testen der einzelnen Speichermodelle in TRNSYS

abgebildet ist, stellt einen fiktiven Be- und Entladevorgang dar. Er enthält zwei Temperaturverläufe mit zugehörigen Massenströmen, für die zwei Doppelanschlüsse eines Speichers. Der erste Temperaturverlauf stellt den Temperaturverlauf eines Solarkreiswärmeüberträgers dar, welcher angelehnt ist an den Prüfbericht (Nr.:04STO97) vom Institut für Thermodynamik und Wärmetechnik, Universität Stuttgart (ITW). Der Massenstrom wurde über die ersten sieben Stunden mit 450 kg/h als konstant angenommen. Bedingt durch die Höhe des Doppelanschluss1 (DP1), welche in

Abbildung 2-11: Aufbau des Test- Speichers

ersichtlich ist, muss während des Beladens eine inverse Temperaturschicht ausgeglichen werden.

Nach dem Beladeprozess erfolgt eine Ruhephase von einer Stunde, gefolgt vom Entladeprozess (Stunde 8 bis 10). Dazu wird mit einem konstanten Massenstrom eine Stunde lang entladen. Nach einer kurzen Ruhephase (5 Minuten) wird der Speicher entladen, bis die Temperatur am Austritt des Doppelanschluss2 (DP2) 30 °C unterschreitet. Bedingt durch den unterschiedlichen Energieinhalt der Speicher kann die Dauer der Entladung sich unterscheiden.

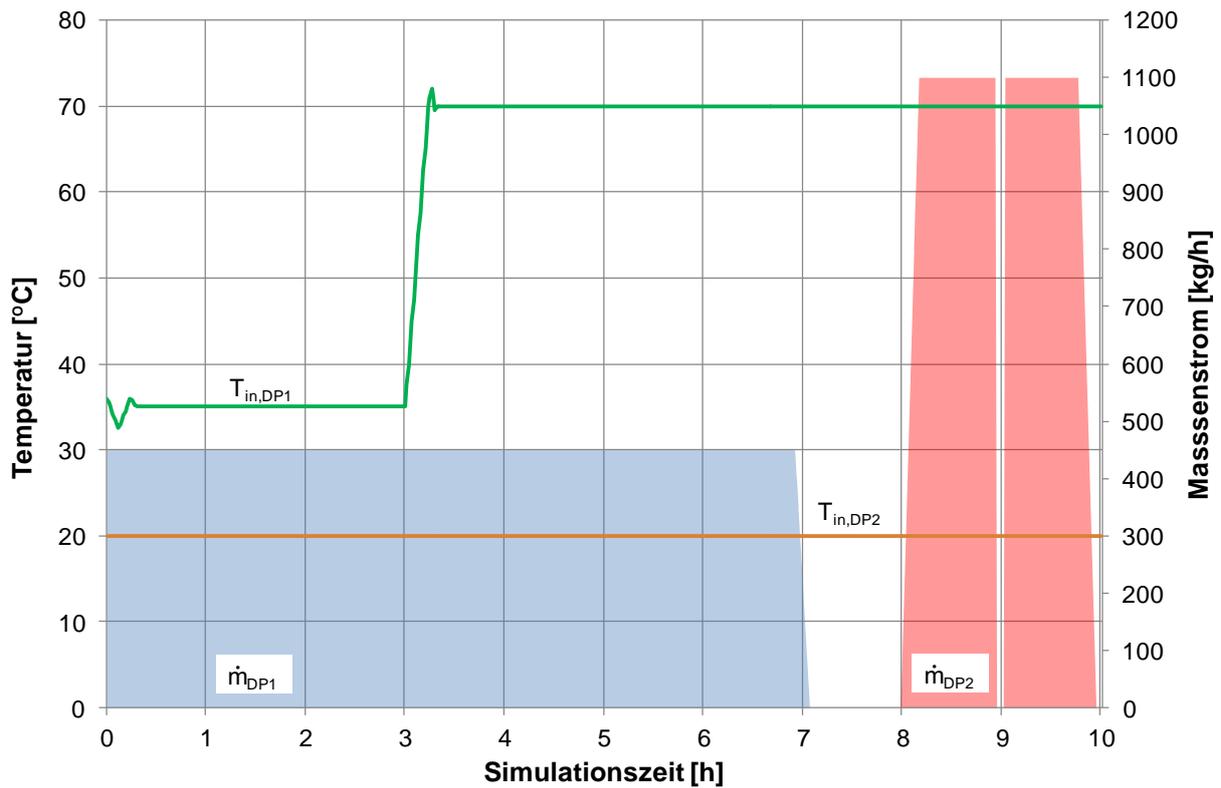


Abbildung 2-12: Lastprofil zum Testen der einzelnen Speichermodelle in TRNSYS

Zusätzlich zu den Angaben in der
Abbildung 2-11: Aufbau des Test- Speichers
sind für die Simulation noch folgende Parameter notwendig (Tabelle 2-2).

Tabelle 2-2: Parameter für den Speichertest

Beschreibung	Wert	Einheit
Anzahl der Speicherknoten	50	-
Simulationszeitschritt	1	min
Spezifische Wärmekapazität	4,19	kJ/kgK
Dichte	1000	kg/m ³
Wärmedurchgangszahl (U-Wert)	0,6967	W/m ² K
Effektive Wärmeleitfähigkeit (sofern möglich)	1,3233	W/mK
Siedepunkt	100	°C
Initialisierungstemperatur der Speicherknoten	20	°C

2.5.2 Testaufbau des Prüfzyklus

In Abbildung 2-13 ist schematisch ein Testaufbau, in Simulation Studio (TRNSYS), für den Type844 (Speicher mit variabler Füllhöhe) abgebildet. Erkennbar auf der linken Seite ist der Type9c (Data Reader), in welchem eine Textdatei mit Inputs definiert werden kann. Der Type2b (Differential Controller) stellt einen Regler dar, welcher das Ausschalten des Massenstrom bei

unterschreiten der Mindesttemperatur, über eine Hysterese regelt (siehe dazu Kapitel 2.4.1). Der Type24 (Quantity Integrator) dient zum Integrieren der Leistungen über die Zeit. Den Mittelpunkt dieses System bildet eine Sammlung von Gleichungen (Equation) die sich hinter dem Icon eines „Taschenrechners“ verbergen. Er nimmt alle Größen wie Temperatur oder Massenstrom auf und kombiniert sie, wenn nötig, mit dem Signal des Reglers. Ebenso werden alle Ausgangsgrößen wie Austrittstemperaturen und Leistungen, welche vom Type844 übergeben werden, erfasst und gegebenenfalls umgerechnet oder an den Type24 weitergereicht. Im unteren Bereich ist der Type25c (Printer) erkennbar, er überträgt die Daten, welche übergeben werden, in eine vom Benutzer spezifizierte Textdatei. Die weiteren Types, welche im Bild als Monitore (Plotter, Type65d) dargestellt sind, dienen zur Visualisierung der Ergebnisse während der Simulation. Durch den modularen Aufbau, und der Zusammenfassung der wesentlichen Berechnungen im Taschenrechner, können die unterschiedlichen Speichertypes relativ einfach in die Simulation integriert werden. Der Informationsfluss für diesen Simulationsaufbau ist schematisch in Abbildung 2-9 abgebildet.

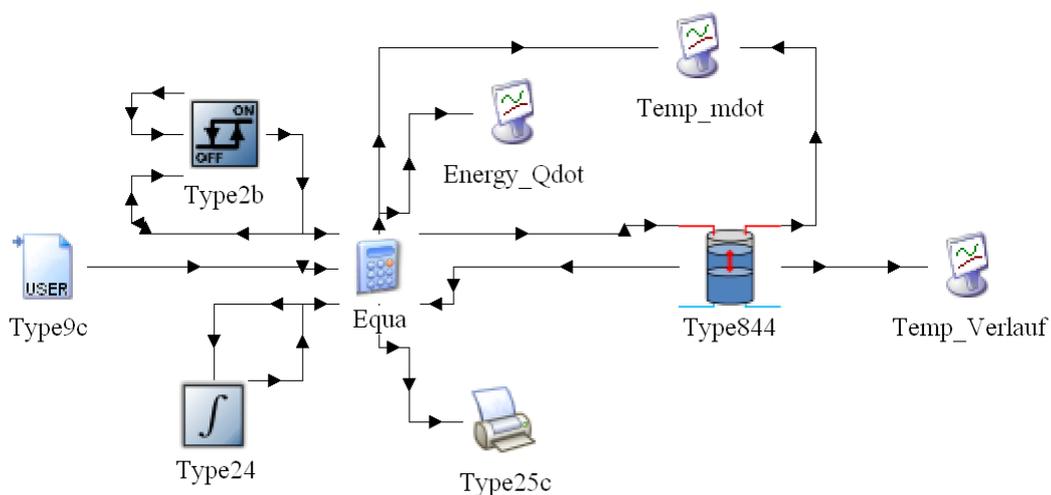


Abbildung 2-13: Testaufbau für die Speichermodelle in TRNSYS

In einer Vorversion des Testaufbaues (Abbildung 2-13) wurde auf den Regler (Type2b) verzichtet und die Regelung über Abfragen der Temperatur in einer Gleichung (Equation) realisiert. Es stellte sich aber heraus, dass eine solche Regelung zum Schwingen neigt und es somit zu Fehlern in der Energiebilanz kommen kann.

Aus diesem Grund wurde der Simulationsaufbau, durch einen Regler (Type2b), erweitert. In der Abbildung 2-14 ist die Abweichung der Energiebilanz zwischen den Zeitschritten des Testaufbaues (Abbildung 2-13) abgebildet.

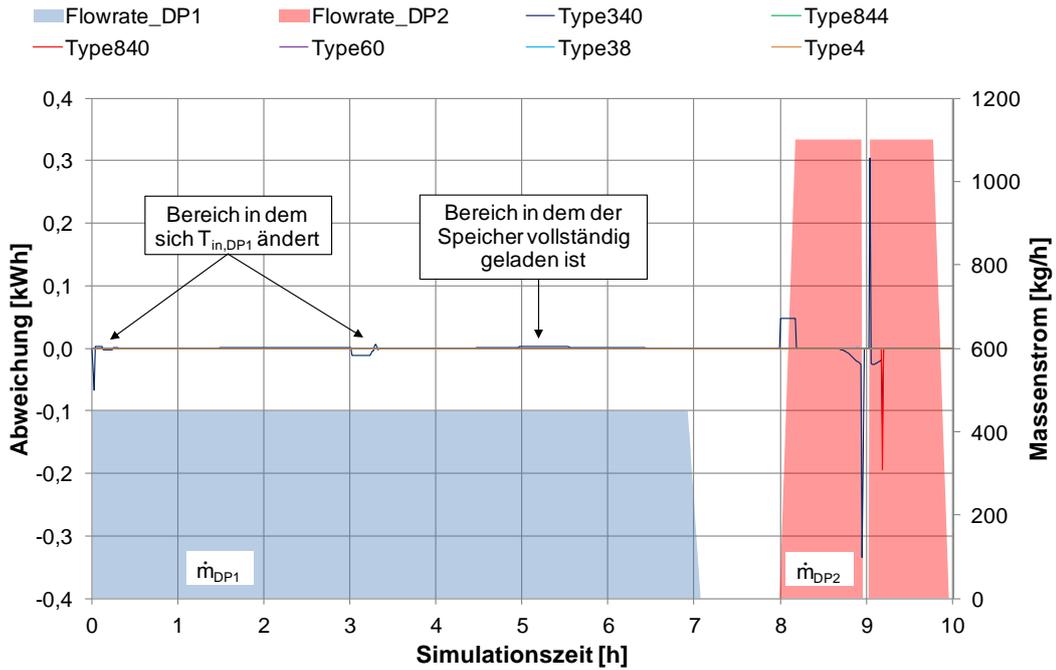


Abbildung 2-14: Abweichung der Energiebilanz zwischen den Zeitschritten

Wie in der Abbildung 2-14 erkennbar, weicht der Type340 deutlich von der Nulllinie ab, der Type840 weicht im Bereich der Abschaltung des Massenstroms (\dot{m}_{DP2}) kurzfristig ab.

Die Abweichung der Energiebilanz der Types 844, 60, 39, 4 ist in Abbildung 2-15: Abweichung der Energiebilanz zwischen den Zeitschritten (skaliert) dargestellt. Da diese Abweichungen um ein Vielfaches geringer sind, wurde hierzu die Abbildung 2-14 skaliert.

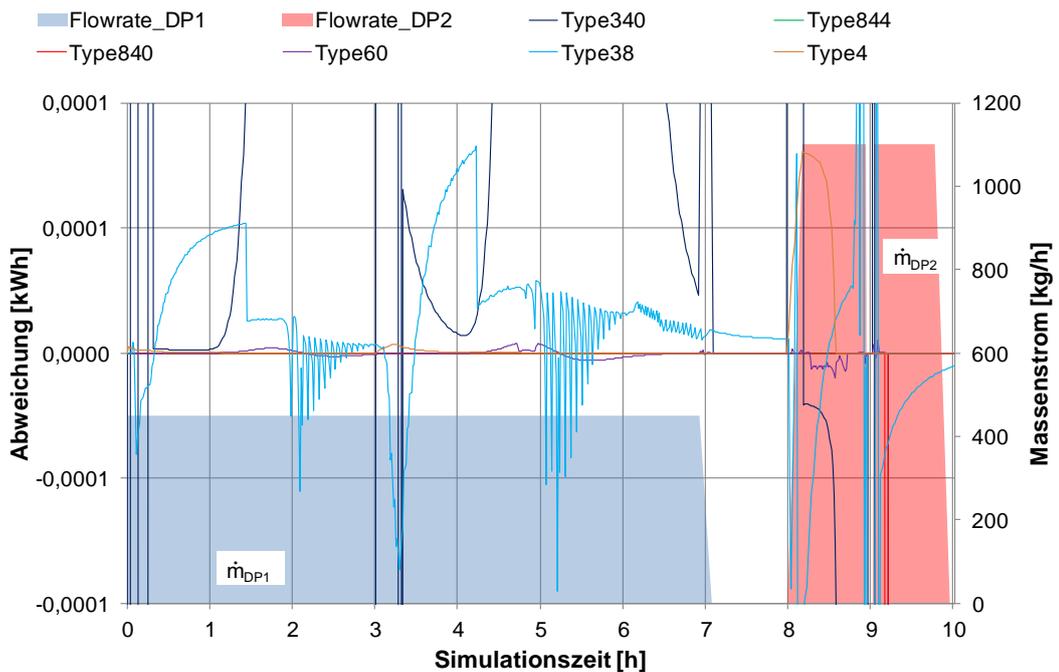


Abbildung 2-15: Abweichung der Energiebilanz zwischen den Zeitschritten (skaliert)

In

Abbildung 2-16: Kumulative Abweichung der Energiebilanz

ist die kumulative Abweichung der Energiebilanz der einzelnen Types ersichtlich. Der Type340 weicht im Bereich der Beladung und Entladung ab, am Ende der Simulationszeit erreicht seine kumulative Abweichung das sehr geringe Niveau der anderen Types ($\sim 3 \cdot 10^{-5}$ kWh). Lediglich der Type840 zeigt am Simulationsende eine deutlich höhere kumulative Abweichung von -0,2 kWh.

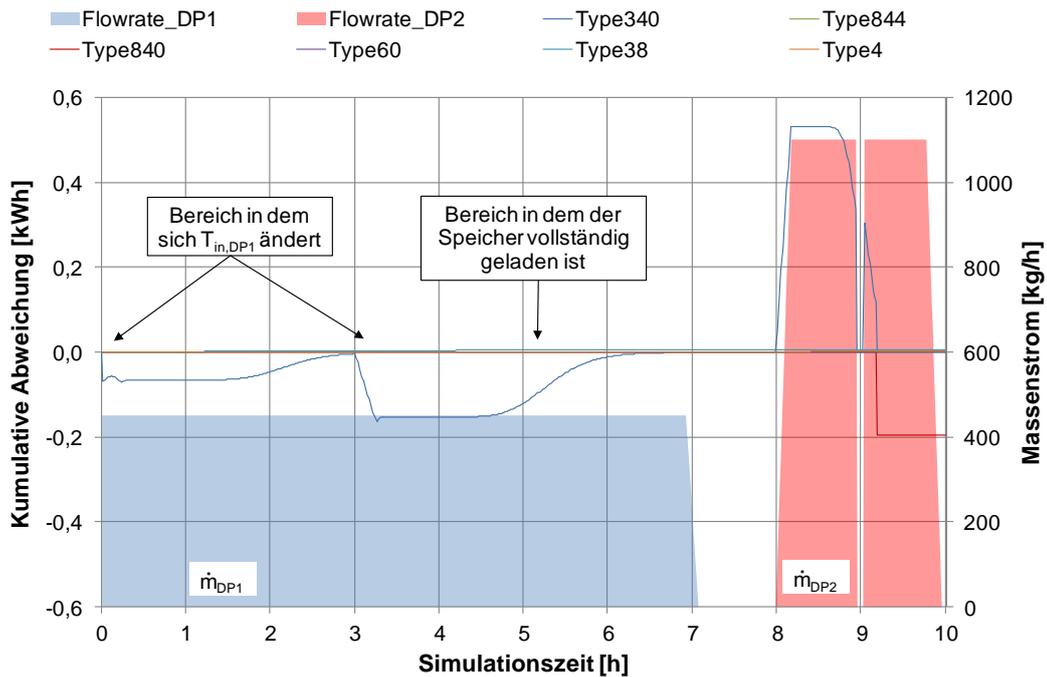


Abbildung 2-16: Kumulative Abweichung der Energiebilanz

2.5.3 Ergebnisse des Speichertest

Die Ergebnisse aus den Simulationen wurden in Microsoft EXCEL ausgewertet. In der Tabelle 2-3 sind die kumulativen Endwerte der Energien jedes Speichers angegeben. Bedingt durch die geringe Simulationszeit von 10 h, ergeben sich nur minimale Abweichungen zwischen den einzelnen Speichern.

Tabelle 2-3: Ergebnisse des Speichertest

	Type340	Type844	Type840	Type60	Type38	Type4
Eintritt [kWh]	58,388	58,395	58,293	58,386	58,391	58,391
Austritt [kWh]	-56,834	-56,786	-56,810	-56,866	-57,686	-56,830
Verluste [kWh]	-0,4249	-0,4255	-0,4251	-0,4247	-0,4240	-0,4256

Da die absoluten Abweichungen bei den Verlusten sehr klein sind, wurden sie ins Verhältnis zum Type340 gestellt, welcher ein sehr gut validiertes Speichermodell darstellt.

Wie in Abbildung 2-17 ersichtlich, weist die eintretende Energie (über den DP1) nur eine sehr geringe Differenz auf.

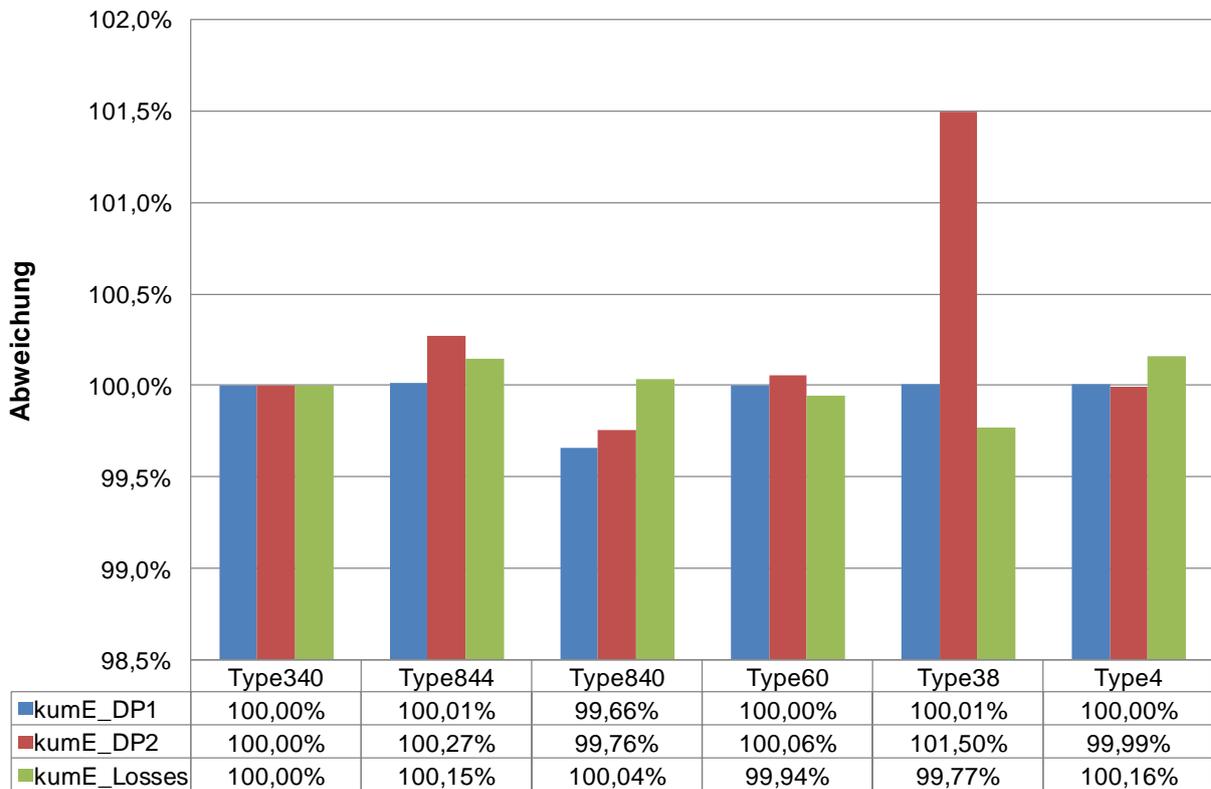


Abbildung 2-17: Relative Abweichung der Simulationsergebnisse

Die austretende Energie (über den DP2) hängt im großen Maße vom Zeitpunkt ab, an dem der Regler die Entladung unterbindet. Bedingt durch den großen Massenstrom ergibt sich eine relativ große Differenz zwischen den einzelnen Speicher. Der Type38 nimmt hier eine Sonderstellung ein, da es durch den Plug- Flow- Ansatz (Ideale Schichtung) zu keiner Durchmischung, infolge der Beladung, kommt. Allerdings kann auch der Type340, welcher als Referenz angenommen wurde, die Realität in einem Speicher nicht vollständig abbilden, da auch dieses physikalische Modell auf Annahmen und Vereinfachungen beruht.

Da der Speicher – Type844 in Abbildung 2-16: Kumulative Abweichung der Energiebilanz nur eine geringe Abweichung aufweist, die Vorgabe eines Temperaturprofils (zum Simulationsstart) ermöglicht und das Füllvolumen eines Speichers berücksichtigen kann wurde dieses Speichermodell für die weiteren Simulationen in TRNSYS im Rahmen dieser Arbeit verwendet.

2.6 Datenerhebung - Messdaten

Die Daten welche im Rahmen des Projektes SOCO für diese Arbeit zur Verfügung gestellt wurden, bestanden aus gemessenen Daten (Industriespeicher), gespeicherten Daten von Anlagensteuerungen, sowie Fragebögen, welche von den Anlagenbetreibern auszufüllen waren (Fragebogen Warmwasserspeicher Graz, Anhang - C). Bei der Auswertung dieser Fragebögen zeigte sich, dass meist nicht alle erforderlichen Daten ausgefüllt wurden. Somit mussten fehlende Daten aus den vorliegenden Unterlagen (Fotos, Zeichnungen) abgeschätzt, oder erneut angefordert werden.

Die Messdaten konnten unterschiedliche Zeitstempel aufweisen und variierten im Umfang der gemessenen Temperaturen im Speicher, den Temperaturen in den Rohrleitungen, den Volumen- und Massenströme (in Verbindung mit der Stellung des Regelventils) sowie der Außenlufttemperatur. Durch die gemessenen Temperaturen in den Rohrleitungen wurden weitere Parameter wie die Länge der Rohrleitung, die Gestaltung der Rohrleitung und die Informationen über die Messfühlerpositionen nötig. Diese Informationen waren meist nicht zu bekommen.

Die Messdaten wurden vor der Simulation mittels EXCEL analysiert und gegebenenfalls aufbereitet. Aus der Analyse der Messdaten ergab sich, dass nicht alle Datensätze fehlerfrei waren. Folgende Fehler konnten erkannt werden:

- Fehlende Zeitschritte
- Überzählige Zeitschritte (zusätzliche Zeitschritte vorhanden aber zeitlich begrenzt)
- Falsche Zeitstempel (innerhalb eines Zeitschritts - verschobener Zeitschritt)

Diese Fehler konnten durch EXCEL-Makros (Anhang - F) erkannt und durch Einfügen des fehlenden Zeitschritts (Datum und Uhrzeit), sowie durch die Vorgabe des Wertes Null (keine Beeinflussung der Simulation) bei den jeweiligen Messgrößen behoben werden.

Wie wichtig die Kontrolle der Messwerte war, zeigte sich am Speicher Knittelfeld. Hier konnte ein defektes Sicherheitsventil bereits in der Analysephase erkannt werden. Der Fehler wurde behoben und die Messwerte nach einer weiteren Messperiode erneut angefordert.

2.7 Effizienz

Zum Beschreiben der Effizienz eines Speichers sind zwei Ansätze möglich.

Der Erste ist der Ansatz nach dem ersten Hauptsatz der Thermodynamik, der Energieansatz. Hierzu ist nur die Kenntnis der Energie (Q) notwendig. Der Zweite ist der Ansatz nach dem

zweiten Hauptsatz der Thermodynamik, der Exergieansatz. Dieser berücksichtigt die Energie (Q) und auch die Temperatur (T), bei der die Energie gespeichert wird.

Dincer et al. (2002) teilte die Effizienzen eines thermischen Energiespeichers in vier Fälle (A, B, C, D) ein, welche sich in ihrer Definition unterscheiden. Die folgenden Berechnungsansätze gelten inhaltlich sowohl für den Energieansatz, als auch den Exergieansatz welcher hier aber nicht explizit angeführt wurde.

2.7.1 Gesamteffizienz

Die Gesamteffizienz kann anhand einer Energiebilanz (Abbildung 2-18) ermittelt werden. In der Mitte ist die gespeicherte Energie (ΔE), auf der linken Seite ist die zugeführte Energie (Q_{in}), auf der rechten Seite die entnommene Energie aus dem Speicher (Q_{out}) und die Speicherverluste (Q_{loss}) abgebildet.

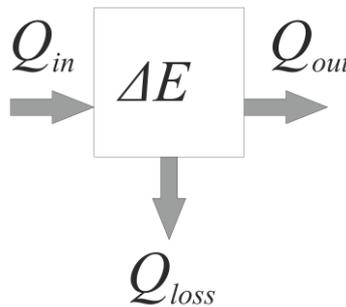


Abbildung 2-18: Energiebilanz eines thermischen Energiespeichers

Der Fall A (Gl. 2-14) berücksichtigt den Energieinhalt des Speichers zum Startzeitpunkt (E_i) nicht und rechnet eine Energieerhöhung im Speicher (ΔE) den Speicherverlusten (Q_{loss}) zu.

$$\eta^A = \frac{\text{entnommene Energie}}{\text{zugeführte Energie}} = \frac{Q_{out}}{Q_{in}} \quad \text{Gl. 2-14}$$

Der Fall B (Gl. 2-15) berücksichtigt den Energieinhalt des Speichers zum Startzeitpunkt (E_i) nicht, jedoch wird eine Energieerhöhung (ΔE) im Speicher berücksichtigt.

$$\eta^B = \frac{\text{entnommene Energie} + \text{Energieerhöhung}}{\text{zugeführte Energie}} = \frac{Q_{out} + \Delta E}{Q_{in}} \quad \text{Gl. 2-15}$$

Der Fall C (Gl. 2-16) lässt eine Energiezunahme (ΔE) im Speicher außer Acht, berücksichtigt jedoch die Energie im Speicher zum Startzeitpunkt (E_i).

$$\eta^C = \frac{\text{entnommene Energie}}{\text{zugeführte Energie} + \text{Energieinhalt}^{\text{Startzeitpunkt}}} = \frac{Q_{out}}{Q_{in} + E_i'} \quad \text{Gl. 2-16}$$

Der Fall D (Gl. 2-17) berücksichtigt sowohl den Energieinhalt zum Startzeitpunkt (E_i'), als auch zum Endzeitpunkt (E_f').

$$\eta^D = \frac{\text{entnommene Energie} + \text{Energieinhalt}^{\text{Endzeitpunkt}}}{\text{zugeführte Energie} + \text{Energieinhalt}^{\text{Startzeitpunkt}}} = \frac{Q_{out} + E_f'}{Q_{in} + E_i'} \quad \text{Gl. 2-17}$$

2.7.2 Periodeneffizienz

Ausgehend von diesen Definitionen hat Dincer et al. (2002) auch Periodeneffizienzen für die verschiedenen Betriebszustände eines thermischen Energiespeichers abgeleitet. Aufgrund der unter Kapitel 2.7.1 gewählten Definitionen können diese nicht für alle Fälle bestimmt werden. Im Fall A und C ist die austretende Energie (Q_{out}), im Ladezyklus und im Speicherzyklus gleich Null. Für die Fälle B und D gilt zusätzlich, dass das Produkt der drei Periodeneffizienzen (Ladezyklus, Entladezyklus und Speicherzyklus) gleich der Gesamteffizienz ist (Gl. 2-18).

$$\eta^B = \prod_{i=1}^3 \eta_i^B, \quad \eta^D = \prod_{i=1}^3 \eta_i^D \quad \text{Gl. 2-18}$$

Die Definition des Falles B, dass der Energieinhalt des Speichers zum Startzeitpunkt nicht berücksichtigt wird, beschränkt diesen auf Vergleichsfälle von Speichern mit gleichen Energieinhalten. Würde ein Speicher vor der Messung und Beladung bereits zur Hälfte geladen sein, so käme der Entladezyklus auf Effizienzen größer eins.

Da in weiterer Folge auch die Effizienz realer Speicher bestimmt werden soll, kann somit nur die Definition D zum Einsatz kommen.

Die folgenden Effizienzen leiten sich von der Gesamteffizienz im Fall D ab.

a) Effizienz des Ladezyklus

Da die austretende Energie (Q_{out}) im Falle einer Beladung gleich Null ist, vereinfacht sich Gl. 2-17 zu Gl. 2-19.

$$\eta_1^D = \frac{\text{Energieinhalt}^{\text{Endzeitpunkt des Beladezyklus}}}{\text{zugeführte Energie} + \text{Energieinhalt}^{\text{Startzeitpunkt des Beladezyklus}}} = \frac{E_{f,1}'}{Q_{in} + E_{i,1}'} \quad \text{Gl. 2-19}$$

b) Effizienz des Speicherzyklus

Im Falle der Speicherung treten keine Energieströme auf, sondern nur Speicherverluste (Q_{loss}), auf (Gl. 2-20).

$$\eta_2^D = \frac{\text{Energieinhalt}_{\text{Endzeitpunkt des Speicherzyklus}}}{\text{Energieinhalt}_{\text{Startzeitpunkt des Speicherzyklus}}} = \frac{E'_{f,2}}{E'_{i,2}} \quad \text{Gl. 2-20}$$

c) Effizienz des Entladezyklus

Die eintretende Energie (Q_{in}) wird im Fall der Entladung gleich Null (Gl. 2-21).

$$\eta_3^D = \frac{\text{entnommene Energie} + \text{Energieinhalt}_{\text{Endzeitpunkt des Entladezyklus}}}{\text{Energieinhalt}_{\text{Startzeitpunkt des Entladezyklus}}} = \frac{Q_{out} + E'_{f,3}}{E'_{i,3}} \quad \text{Gl. 2-21}$$

d) Effizienz des Be- und Entladezyklus

Im realen Zyklus kann bedingt durch den gleichzeitigen ein- und austretenden Massenstrom eine Unterscheidung zwischen Beladung und Entladung nicht durchgeführt werden. Somit wird die Effizienz der Beladezyklus (η_1^D) und die Effizienz der Entladezyklus (η_3^D) zur Effizienz des Be- und Entladezyklus (η_4^D) zusammengefasst (Gl. 2-22).

$$\eta_4^D = \eta_1^D \cdot \eta_3^D \quad \text{Gl. 2-22}$$

3 SIMULATIONS-AUFBAU UND ERGEBNISSE

Nun folgt eine kurze Beschreibung der einzelnen realen Speicher, welche in TRNSYS abgebildet wurden. Es werden die getroffenen Annahmen und Vereinfachungen, sowie die daraus resultierenden Abweichungen erläutert. Die Wärmeverluste der Anschlussleitungen am Speicher wurden anhand einer Studie der Hochschule Rapperswil – Institut für Solartechnik (Lauber, 2007) abgeschätzt. Zusätzliche Verluste durch die Geometrie des Speichers (Klörperboden) werden durch eine entsprechende Verkleinerung der Dämmstärke berücksichtigt. Das dem Speicher aufgeprägte Temperaturprofil konnte, mittels einer Textdatei (T_init.dat), berücksichtigt werden. In dieser Textdatei müssen sich die Initialisierungstemperaturen jedes Knotens befinden. Sind die Speichertemperaturen zum Simulationsstart konstant, ist die Angabe einer Initialisierungstemperatur im Proforma (Registerkarte Parameter, Parameter 4 – Initial temperature of the store), möglich.

3.1 Energiespeicher (93 m³) – STIEGL

Der Energiespeicher STIEGL (Abbildung 3-1) dient zur Energierückgewinnung in einem Brauprozess. Die beim Kochen der Würze freiwerdende Energie, wird im Pfannendunstkondensator (Pfaduko), welcher als Wärmetauscher fungiert, an das Speichermedium übertragen und dem Speicher zugeführt. Die gespeicherte Energie dient in einem späteren Prozessschritt, zum Vorwärmen der Würze, vor dem erneuten Kochprozess (<http://www.krones.com/de/produkte/wuerzekocher.php>, 17.10.2012).

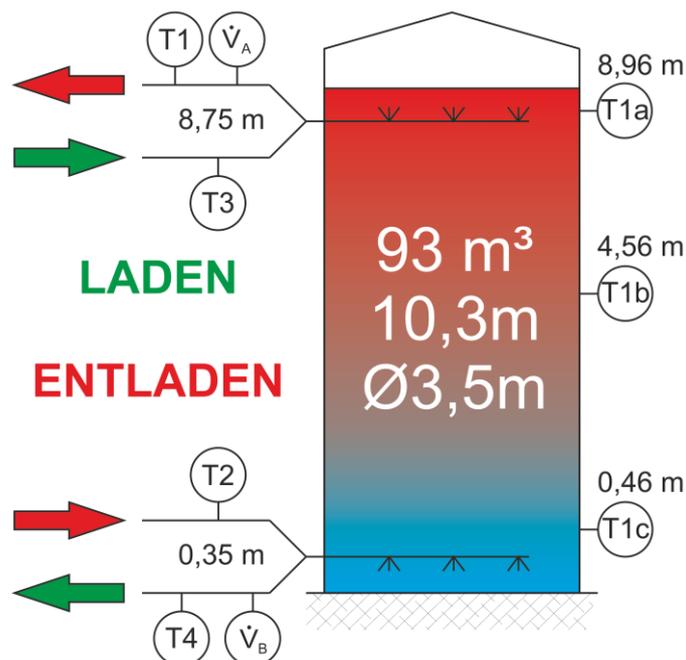


Abbildung 3-1: Schema Energiespeicher STIEGL

Der Energiespeicher ist in einer Halle positioniert, seine Anschlüsse (Doppelanschlüsse) werden sowohl zum Laden, als auch Entladen benutzt. Der Behältermantel und der Deckel sind mit Mineralwolle, der Boden mit einer Schaumglasdämmung (Foamglas) gedämmt.

Tabelle 3-1 gibt einen kurzen Überblick auf die Eckdaten der Simulation.

Tabelle 3-1: Eckdaten – Energiespeicher STIEGL

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	164,2	h
Anzahl der Messpunkte im Speicher	3	-
Füllvolumen des Speichers	89,8	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	0,7525	W/mK
Dämmstärke – Mantel	10	cm
Dämmstärke – Deckel	9,55	cm
Dämmstärke – Boden	10	cm

a) Annahmen und Vereinfachungen.

Die Geometrie der beiden Speichereinlässe im Inneren des Speichers entsprechen dem eines Rohres, welches 38 tangentielle Bohrungen aufweist, die von der Horizontalen um 20 Grad geneigt sind (Abbildung 3-2). Dadurch strömt das Speichermedium nicht, wie im idealen Fall, in tangentialer Richtung ein, sondern bewirkt durch den Eintrittsimpuls eine Durchmischung in vertikaler Richtung. Im Falle der Beladung wird das Speichermedium über dem Rohr durchmischt. Bei der Entladung, hier sind die Bohrungen in Richtung des Bodens geneigt, wird das Speichermedium unterhalb durchmischt.

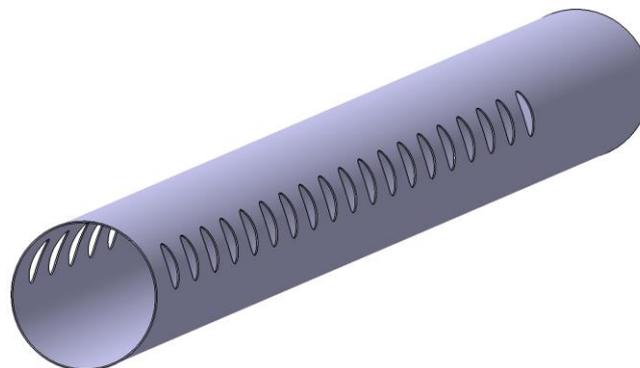


Abbildung 3-2: 3D- Ansicht des oberen Speichereinlaufs (8,75 m)

Simulationsmodelle können Mischvorgänge nur dann beschreiben, wenn sie eine inverse Temperaturschichtung hervorrufen. Da dies auch zu einer Änderung der Dichte führt, sinken die kälteren Temperaturschichten ab und durchmischen sich mit den wärmeren Schichten.

Eine Durchmischung in umgekehrter Richtung wäre in der Simulation nur dann möglich, wenn die Knotenzahl stark reduziert werden würde. Dies hätte aber eine schlechtere Schichtung im gesamten Speicher zufolge. Der Einfluss des Rohres selbst, welches zu einer lokalen Erhöhung der effektiven Wärmeleitfähigkeit im Speicher führt, muss unberücksichtigt bleiben ($\lambda_{eff} = const$). Da in TRNSYS zu Simulationsbeginn angegeben werden muss, ob es sich um einen Einlass oder Auslass handelt, werden die gemeinsamen Anschlüsse getrennt behandelt. Das Füllvolumen ($89,8 \text{ m}^3$) war aus den Messdaten der Regelung bekannt.

b) Modellierung in TRNSYS

Abbildung 3-3 zeigt die Modellierung in der grafischen Benutzeroberfläche von TRNSYS, dem Simulation Studio. Die Massenströme und die Temperaturen, aus den Messdaten, wurden mittels Textdatei vorgegeben und in Gleichungen (Equation im „Taschenrechner“ mit der Bezeichnung Filter in Abbildung 3-3) gefiltert. Da diese gemessenen Daten in den Rohrleitungen ermittelt wurden, musste dies in der Simulation berücksichtigt werden. Zum Einsatz kam hier das Rohrleitungsmodell Type709 (siehe Kapitel 2.4.3). Wie aus den Übersichtszeichnungen der Verrohrung des Energiespeichers hervorgeht, ist die gemeinsame Länge der Rohrleitung des Doppelanschluss geringer als $0,75 \text{ m}$. Da die gemessene Temperatur in großen Abständen ($8,12$ bis $21,24 \text{ m}$ Rohrlänge) ermittelt worden sind, wurde auf die zusätzliche Modellierung dieses kurzen Einlaufstücks ($0,75 \text{ m}$) verzichtet.

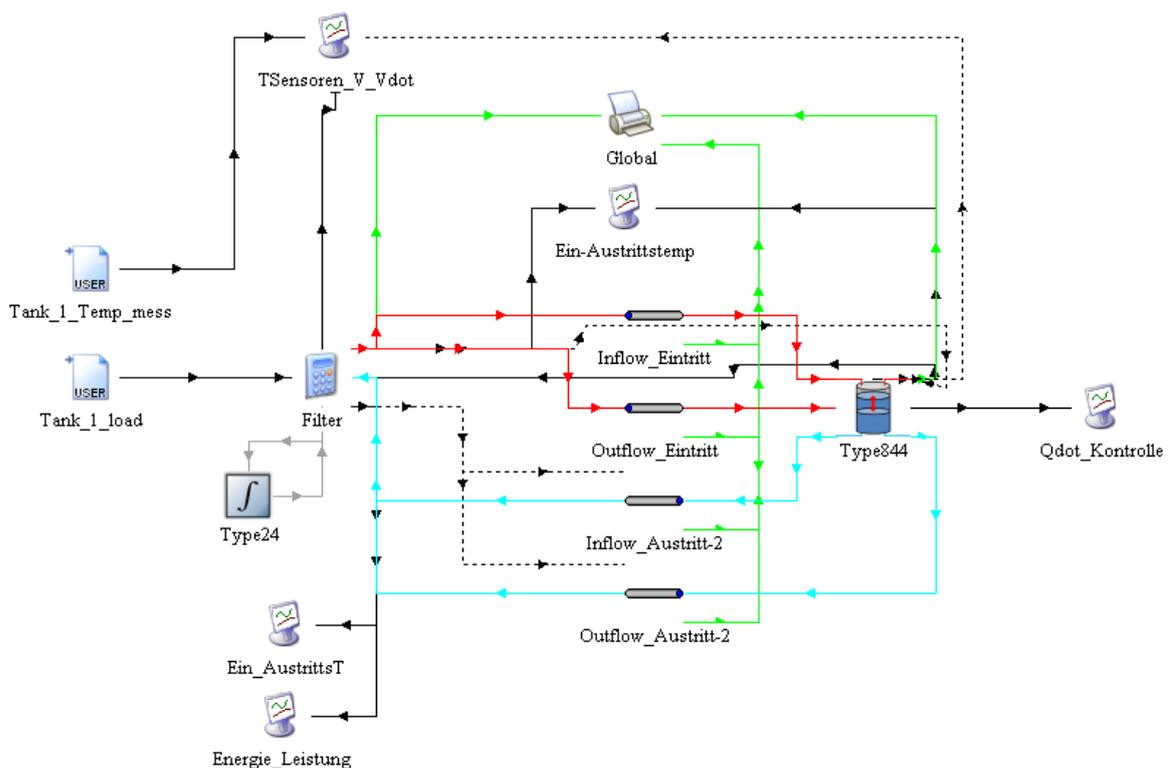


Abbildung 3-3: Simulationsaufbau Energiespeicher STIEGL

Die gefilterten Daten wurden den Rohrleitungen (Type709) übergeben, welche an das Speichermodell angeschlossen sind. Die Temperaturen, der aus dem Speicher austretenden Massenströme, wurden nach den Rohrleitungen im Taschenrechner eingelesen und daraus die Leistungen und die Energien berechnet. Die Speicherung der Ergebnisse erfolgte in einer Textdatei.

c) Vergleich Messung – Simulation

In der Abbildung 3-4 ist der Temperaturverlauf der drei Temperatursensoren im Speicher abgebildet. Erkennbar ist, dass der oberste Temperatursensor ($T_{1a,mess}$) Sprünge im Falle der Beladung (Stunde 0 bis 22 und 113 bis 163) aufweist. Diese Temperaturspitzen kann die Simulation $T_{1a,sim}$ nachbilden, jedoch den darauffolgenden Einbruch der gemessenen Temperatur $T_{1a,mess}$ nicht. Da es, wie schon zuvor beschrieben, durch die Einlass- und Auslassgeometrie zur Ausbildung von Strömungen im Speicher kommt, führt dies zu einer lokalen Durchmischung im oberen Bereich des Speichers. Darin könnte auch die Ursache der Abweichung des mittleren Temperatursensoren ($T_{1b,mess}$) begründet sein, liegt dieser doch deutlich über dem Temperaturniveau von $T_{1b,sim}$. Den untersten Temperatursensoren ($T_{1c,mess}$) konnte die Simulation ($T_{1c,sim}$) gut nachbilden.

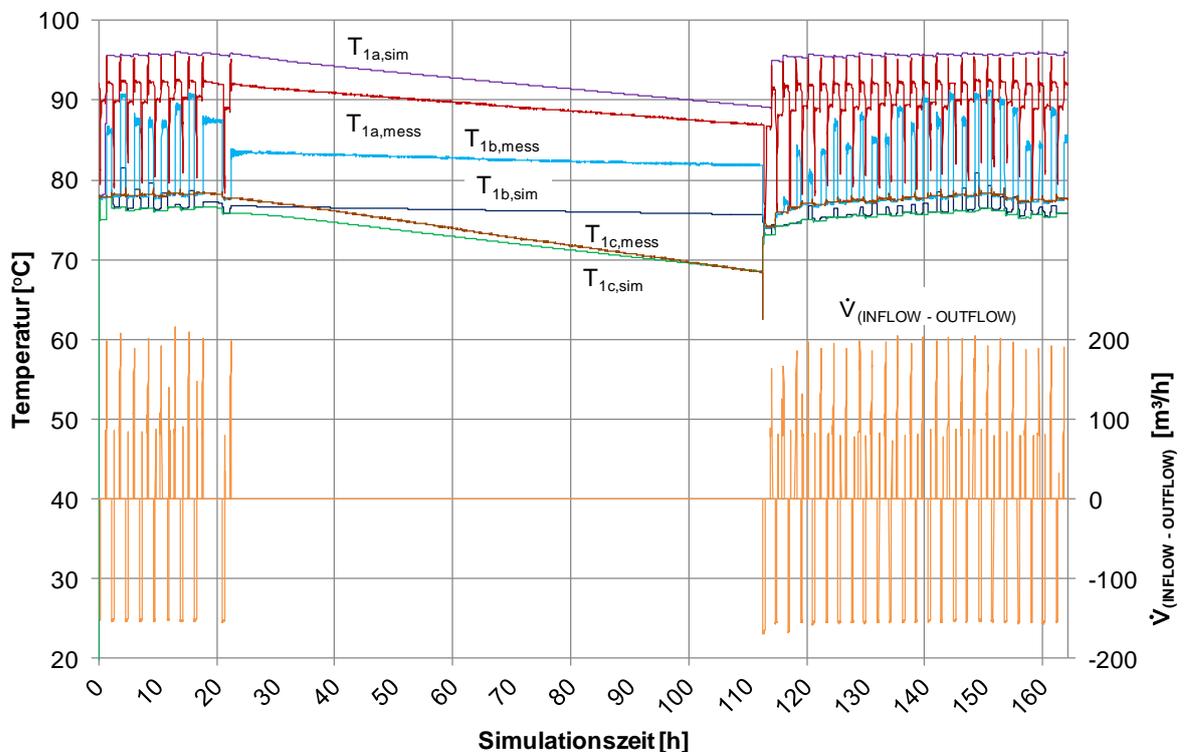


Abbildung 3-4: Temperaturverlauf – Sensoren

Die Verluste des Speichers in der Ruhephase (Stunde 22 bis 112), wiesen ein annähernd gleiches Niveau auf, was durch den nahezu parallelen Verlauf der gemessenen und simulierten Temperaturen in Abbildung 3-4 bedingt ist.

Die Abbildung 3-5 zeigt den Temperaturverlauf der Beladung. Die Temperatur T_3 wurde in einem Nebengebäude gemessen, wobei die Länge der Rohrleitung, sowie die Einbauten unbekannt waren. Somit kann auch die Spitze, welche während der Ruhephase (Stunde 37) auftrat, nicht näher betrachtet werden. Die Temperatur T_4 ist an einer fallenden Rohrleitung von etwa sieben Metern Länge gemessen worden. Die Verbindung zum Speicher führt neben einer Pumpe über drei Ventile und einem Massenstromteiler in den Speicher. Die Verrohrung wurde in der Simulation mittels eines Modells (Type709) berücksichtigt. Der Abfall der gemessene Temperatur ($T_{4,mess}$) war außerhalb der Ruhephase (Stunde 22 bis 112) größer, innerhalb der Ruhephase wies jedoch die simulierte Temperatur den größeren Abfall auf ($T_{4,sim}$).

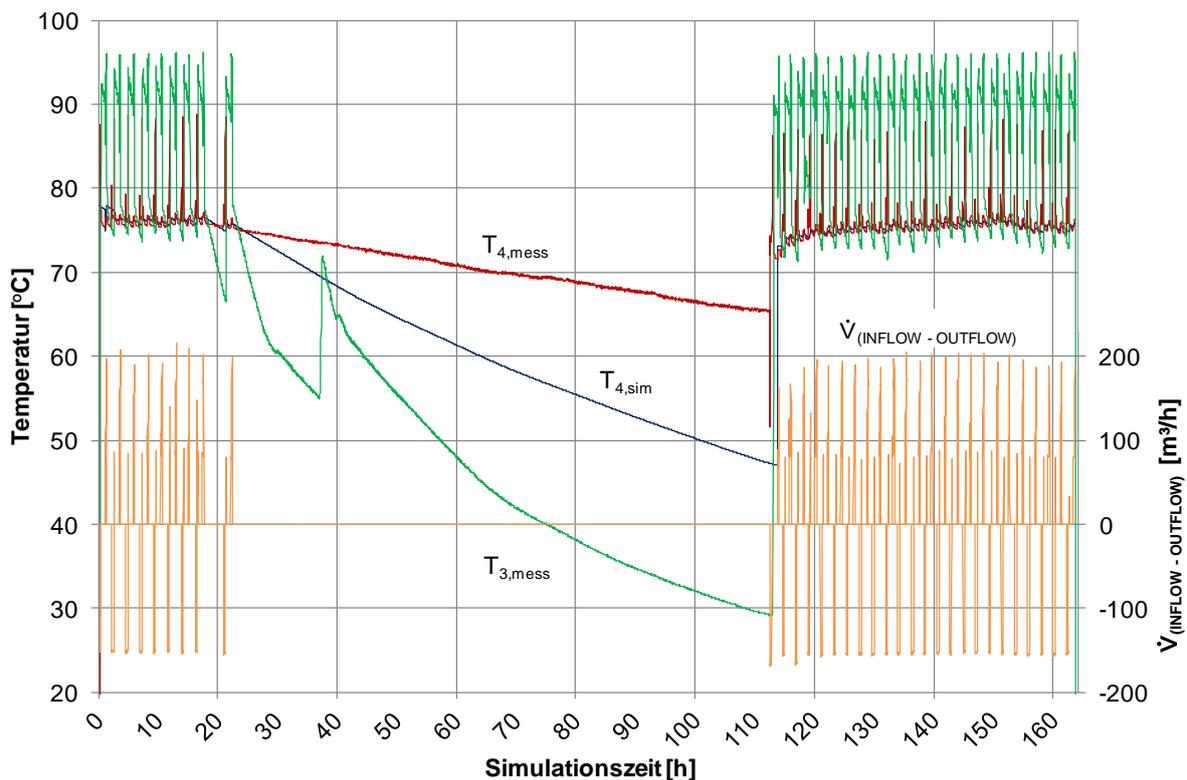


Abbildung 3-5: Temperaturverlauf beim Beladen (Inflow)

In der Abbildung 3-6 ist der Temperaturverlauf der Entladung erkennbar. Die Temperatur $T_{1,mess}$ wurde nach zwei Ventilen und einer Pumpe gemessen. Die Temperaturspitzen der Simulation ($T_{1,sim}$) erreichten ihr Maximum bevor die Messung ($T_{1,mess}$) ihr Maximum erreichte. Da die Rohrleitungslängen hier für beide Temperaturen bekannt waren, bedeutet

dies für die Messung, dass die eingesetzten Temperaturfühler ($T_{1,mess}$) ein träges Ansprechverhalten besitzen.

Da die Simulation im oberen Bereich des Speicher höhere Temperaturen aufwies ($T_{1a,sim}$ in Abbildung 3-4) führte dies zu einer höheren Temperatur in der Rohrleitung, was sich an der Höhe der Temperaturspitzen bemerkbar machte ($T_{1,sim}$). Außerhalb des Entladevorgang war der Temperaturabfall der Messung ($T_{1,mess}$) größer als der Abfall in der Simulation ($T_{1,sim}$), was sich wiederum in der Ruhephase (Stunde 22 bis 112) änderte.

Der Messpunkt der eintretenden Temperatur T_2 (Abbildung 3-1), befand in 1,5 Meter Entfernung an einer fallenden Rohrleitung. Da keine weiteren Information über den Messaufbau bekannt waren, kann auch keine Aussage, über die rasche Auskühlung auf knapp über der Umgebungstemperatur (T_{amb}), getroffen werden.

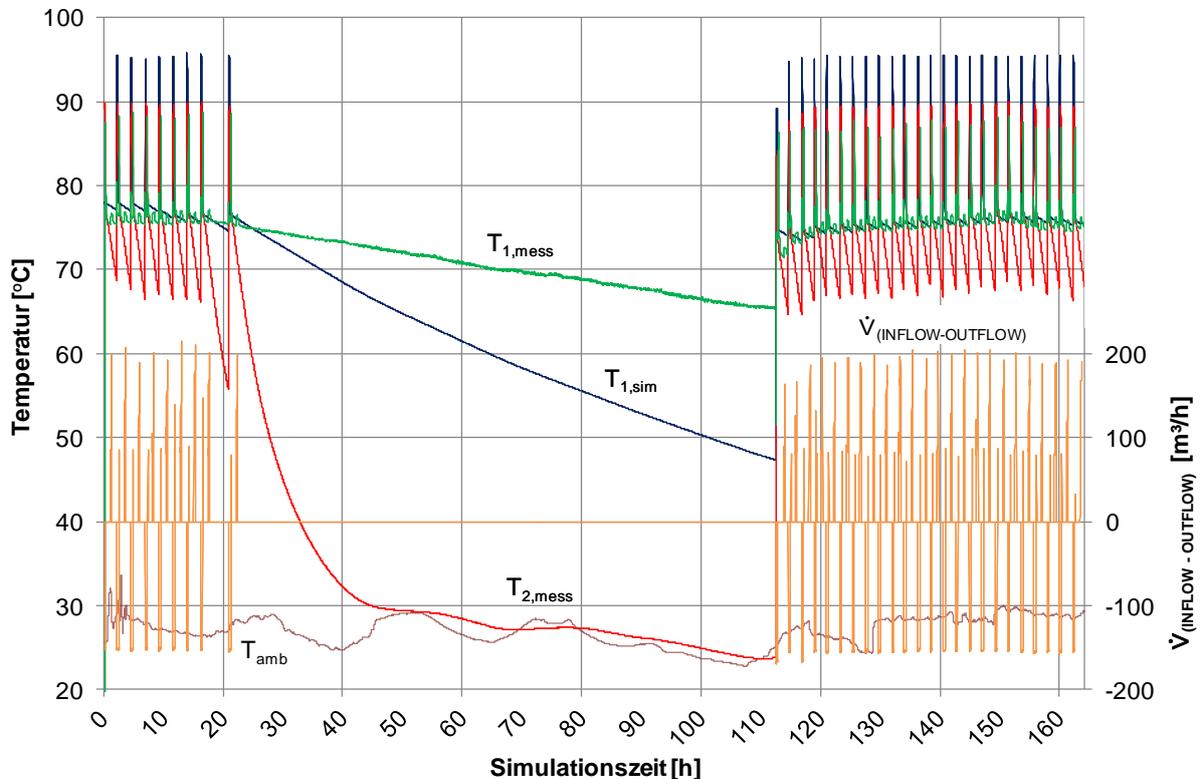


Abbildung 3-6: Temperaturverlauf beim Entladen (Outflow)

In Abbildung 3-7 ist eine Leistungsbilanz, sowie der Verlauf des kumulativen Energiebedarf, aus der Messung und der Simulation, gegenübergestellt. Die Leistungen wurden aus den jeweiligen Temperaturen der Rohrleitungen ermittelt und stellen eine Bilanz (eintretend minus austretend) im jeweiligen Zeitschritt eines Doppelanschlusses dar. Positiv aufgetragen ist der Wärmestrom bei der Beladung, negativ der Wärmestrom bei der Entladung.

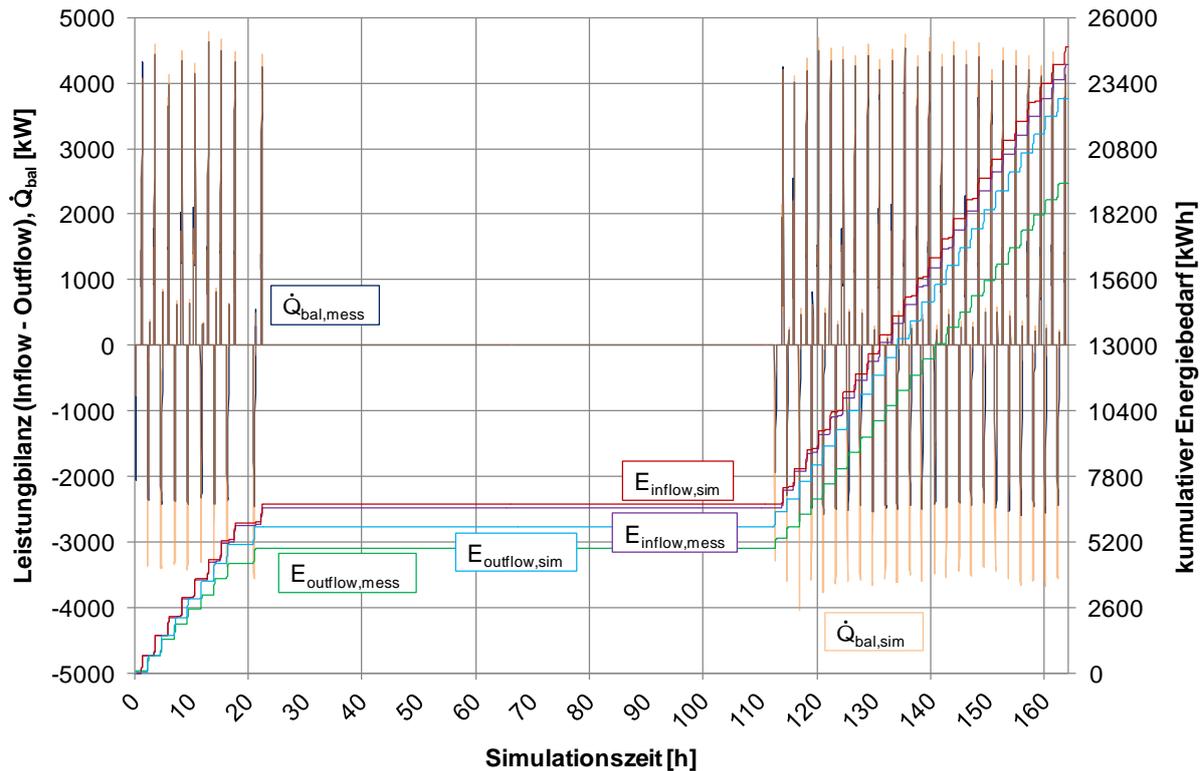


Abbildung 3-7: Leistung und kumulativer Energiebedarf

Der Wärmestrom bei der Beladung wies eine geringere Abweichung auf, als der Wärmestrom bei der Entladung, welcher in Abbildung 3-7 negativ aufgetragen worden ist. Da beide Wärmeströme in der Simulation größer als die der Messung waren (ausgenommen des ersten Zyklus der Beladung und Entladung) ist davon auszugehen, dass die Verluste der Speicher und der Verrohrung, in der Simulation, unterschätzt wurden.

Der kumulative Energiebedarf sind aus den Wärmeströmen ermittelt worden und stellen das Integral der Wärmeströme über die Zeit dar.

d) Fazit

Der kumulative Energiebedarf wies am Ende der Simulation eine Abweichung gegenüber der Messung von 2,7 % bei der Beladung und 17,4 % bei der Entladung auf. Diese Abweichung konnte mangels detaillierter Informationen über den Messaufbau nicht näher bestimmt werden. Auch die signifikante Abweichung des oberen Temperaturfühlers (T_{1a}) und des mittleren Temperaturfühlers (T_{1b}) konnte in der Simulation nicht wiedergegeben werden. Jedoch ist davon auszugehen, dass sich durch die Durchmischung während der Be- und Entladung ein Temperatenausgleich zwischen den mittleren und dem oberen Bereich des Speichers stattfindet.

Die Angabe von Temperaturen, welche in Rohrleitungen gemessen wurden, ist als ein kritischer Punkt anzumerken, da damit zusätzliche Angaben über den Messaufbau, wie Durchmesser, Längen und Dämmstärke der Rohrleitungen, und die eingesetzten

Messinstrumente notwendig sind. Da die Simulation generell höhere Werte lieferte als die Messung, ist davon auszugehen, dass zusätzliche Verluste noch unberücksichtigt geblieben sind.

3.2 Warmwasserspeicher (220 m³) – STIEGL

Der Warmwasserspeicher STIEGL repräsentiert einen Speicher mit einer variablen Füllhöhe. Er wird, wie der Energiespeicher STIEGL, zur Energierückgewinnung in einem Brauprozess eingesetzt. Der in Abbildung 3-8 abgebildete Wärmetauscher stellt den Kondensatkühler dar, in welchem das Kondensat (99 °C) aus dem Pfannendunstkühler (Pfaduko), weiter abkühlt wird. Auf der Gegenseite wird dieser Kondensatkühler von einem Zirkulationsstrom aus dem Speicher und einem Frischwasserstrom durchströmt, welcher in den Speicher eintritt (80 °C). Ein zusätzlicher Energieeintrag in das System ergibt sich durch den Würzekühler (Wärmetauscher). Der Würzekühler hat die Aufgabe die Würze nach dem Kochvorgang von 99 °C auf 3 °C abzukühlen, bevor diese der Gärung zugeführt wird. Zur Kühlung der Würze wird der Würzekühler von einem Eiswassertank (3 °C) gespeist (in Abbildung 3-8 nicht dargestellt). Das erwärmte Wasser (80 °C) tritt vom Würzekühler in den Speicher ein. Die gespeicherte Energie kann, in darauffolgenden Prozessschritten der Produktion (Warmwasser Sudhaus und Netz), entnommen werden (<http://www.krones.com/de/produkte/wuerzekocher.php>, 17.10.2012).

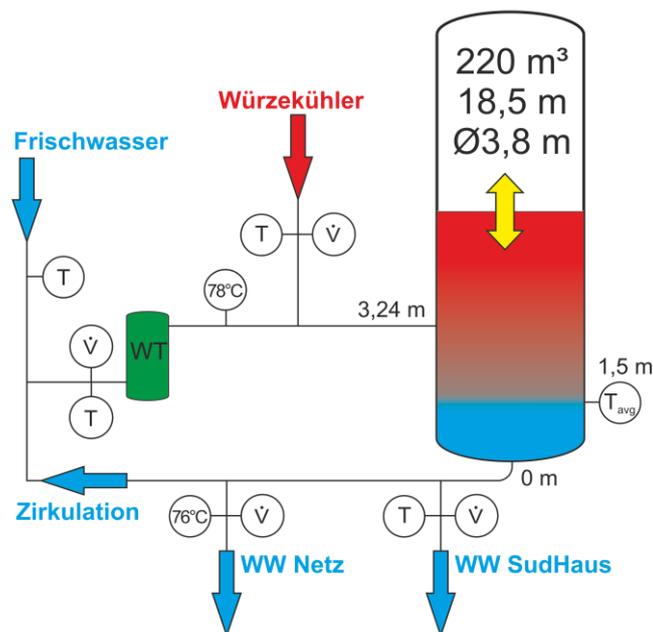


Abbildung 3-8: Schema Warmwasserspeicher STIEGL

Der Energiespeicher ist im Freien positioniert, und besteht, entgegen der Darstellung in Abbildung 3-8 aus zwei Speichern, welche als kommunizierende Gefäße betrachtet werden können. Die Behälter besitzen eine durchgängige Dämmung mit Mineralwolle. Die Anschlüsse sind bei beiden Speichern für die Beladung in einer Höhe von 3,24 Meter und für die Entladung am Speicherboden. Die wichtigsten Eckdaten der Simulation wurden in Tabelle 3-2 zusammengefasst.

Tabelle 3-2: Eckdaten – Warmwasserspeicher STIEGL

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	159,2	h
Anzahl der Messpunkte im Speicher	1	-
Füllvolumen des Speichers	179,38	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	0,7541	W/mK
Dämmstärke – Mantel	12	cm
Dämmstärke – Deckel	9,1	cm
Dämmstärke – Boden	9,1	cm

a) Annahmen und Vereinfachungen.

Die beiden Speicher wiesen aus den Simulationsdaten dasselbe Füllvolumen und Temperaturniveau auf. Somit musste für die Simulation nur ein Speicher betrachtet werden, welcher mit der Hälfte des gemessenen Volumenstromes beaufschlagt wurde. Weiters wurde die Dämmung des Bodens und Decke aufgrund der größeren Oberfläche, welche in der Simulation nicht berücksichtigt werden kann, angepasst. Die Umgebungstemperatur wurde für die Rohrleitungen, welche sich im Inneren eines Gebäudes befinden mit einer mittleren Temperatur von 25 °C angenommen. Für den Speicher wurde die mittlere Außentemperatur auf 10 °C angesetzt. Die Ermittlung erfolgte auf Basis der Temperaturen in der Auskühlphase. Da nicht alle Rohrleitungslängen bekannt waren, mussten diese, auf Basis der gemessenen Temperaturen, abgeschätzt werden.

Für den Austritt des Warmwasser Netzes lagen keine Messdaten über die Temperatur vor, es wurde eine mittlere Temperatur von 76 °C angenommen. Der Zirkulationsstrom, welcher mit dem Frischwasserstrom gemischt wird, konnte, bedingt durch die hydraulische Verschaltung, am Austritt des Wärmetauschers nicht gemessen werden. Da im Wärmetauscher eine Nachheizung erfolgt, wird angenommen, dass die Zieltemperatur von 78 °C am Austritt erreicht wird.

b) Modellierung in TRNSYS

Die Abbildung 3-9 zeigt den Aufbau der Simulation. Wie auch im schematischen Aufbau (Abbildung 3-8) ersichtlich ist, mussten, bedingt durch die hydraulische Verschaltung, zwei Mischer (Type649) und zwei Massenstromteiler (Type11f) eingesetzt werden. Weiters war dadurch auch der Aufbau einer Regelung der Massenstromteiler notwendig. In Rot dargestellt ist der eintretende Volumenstrom, in Blau eingezeichnet, der austretende Volumenstrom aus dem Speicher.

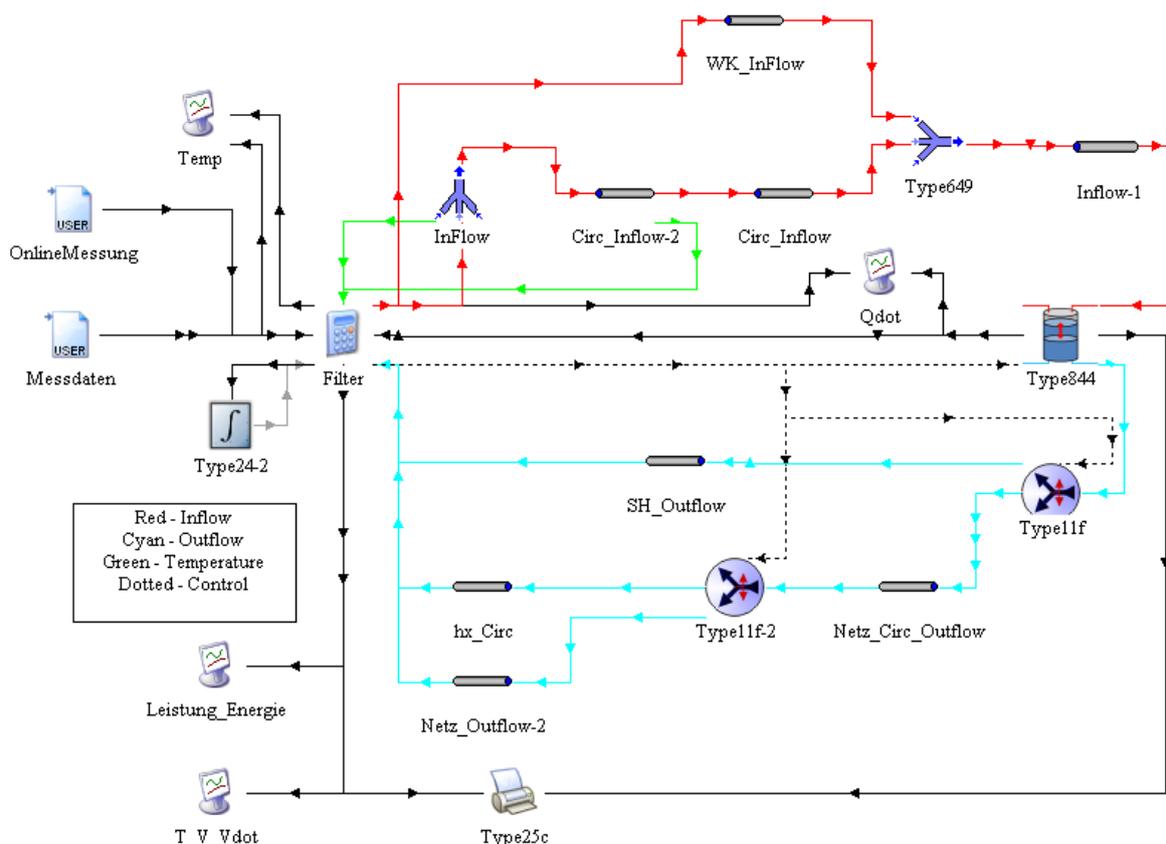


Abbildung 3-9: Simulationsaufbau des Warmwasserspeicher STIEGL

c) Vergleich Messung – Simulation

In der Abbildung 3-10 ist die Änderung der Füllhöhe V_{avg} und der Temperatur T_{avg} im Speicher dargestellt. Es wurde festgestellt, dass während der Simulationszeit das berechnete Volumen, vom gemessenen, abwich. Am Ende der Simulation stimmte die Füllmenge der Simulation mit der Berechnung überein. Die größte Abweichung trat auf, wenn nach einem austretenden Massenstrom (Outflow), kein eintretender Massenstrom (Inflow) folgte. Da auch während der Stillstandsphase die Füllmenge kontinuierlich abnahm, wäre zu klären, ob dies durch fehlerhafte Ventile bedingt ist. Der Temperaturverlauf am Temperaturfühler konnte während des Betriebs wiedergegeben werden, jedoch kam es im Bereich der Stillstandsphase (Stunde 44 bis 104) zu einer erhöhten Abweichung.

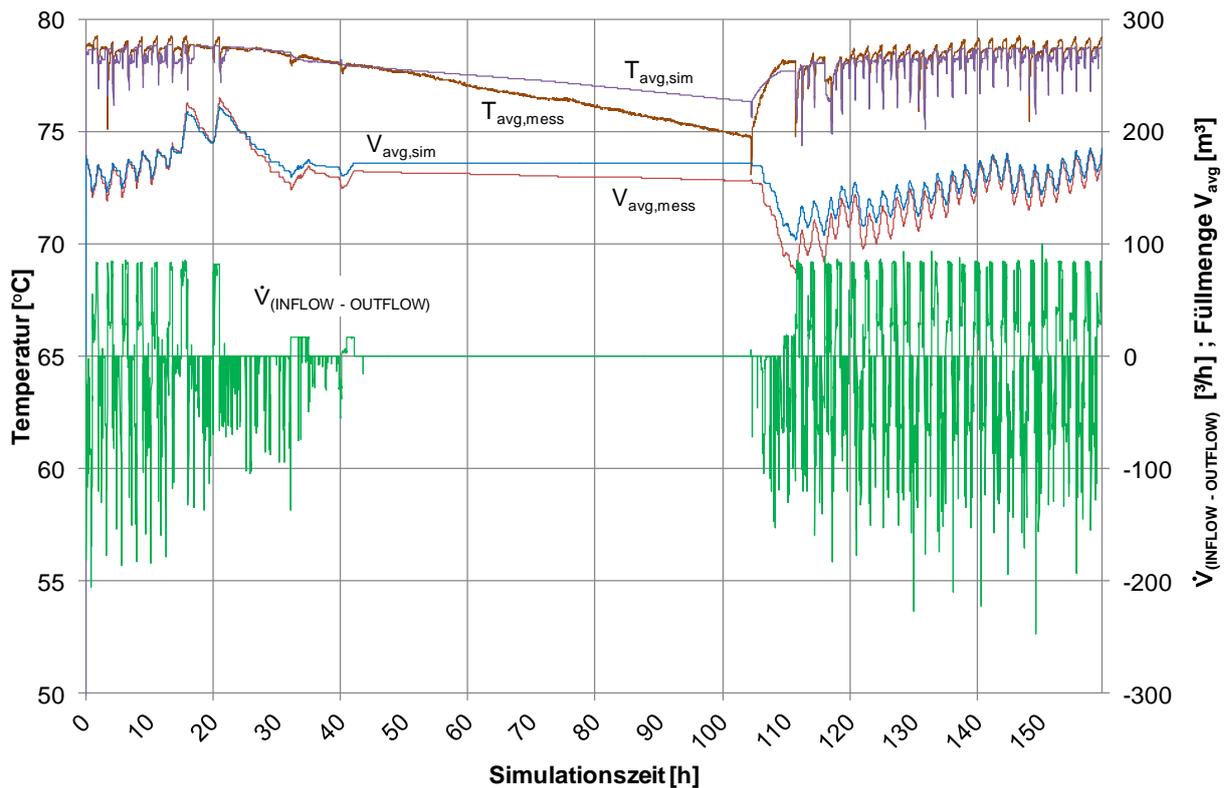


Abbildung 3-10: Änderung des Füllvolumens und Temperaturverlauf

Die eintretenden Temperaturen in den Speicher ist in Abbildung 3-11 abgebildet. Der Volumenstrom der Zirkulation (\dot{V}_{Circ}) nahm eine Sonderrolle ein, er stellte sowohl einen Eintritt als auch einen Austritt aus dem Speicher dar. Wie in Abbildung 3-8 ersichtlich war, wurden die Temperatur und der Volumenstrom der Zirkulation gemeinsam mit dem des Frischwassers ermittelt. Die Aufteilung in Frischwasser und Zirkulation erfolgte mittels eines Filters. Der Filter rechnete den Volumenstrom bis zum Grenzwert von 20 m³/h dem Frischwasser (\dot{V}_{FW}), darüber der Zirkulation (\dot{V}_{Circ}) zu. Diese Annahme bedingt aber, dass die beiden Volumenströme nicht gemeinsam auftreten dürfen. Diese Bedingung konnte auf Basis der vorhandenen Daten nicht überprüft werden.

Die Mischtemperatur, welche $T_{\text{Circ,mess}}$ darstellte, wies Abweichungen auf, welche nicht durch die Messdaten erklärt werden konnten. Zum Simulationsbeginn wies $T_{\text{Circ,mess}}$ gleichbleibende Schwankungen auf, jedoch ist der Betrag zu klein, als das der Volumenstrom nur aus dem Speicher gedeckt werden konnte. Vergleichsrechnungen zeigten, dass unter der Annahme eines zusätzlich vorhandenen Frischwasservolumenstroms (\dot{V}_{FW}), sich eine Mischtemperatur $T_{\text{Circ,sim}}$ ergab, welche in der Größenordnung von $T_{\text{Circ,mess}}$ lag. Erkennbar ist dies in den Bereichen zur Stunde 0 bis 20 h und 135 bis 159 h. Der Bereich nach der Ruhephase (Stunde 104 bis 110) ließ sich damit nicht erklären. Die Voraussetzung, damit dieses Temperaturniveau erreicht werden kann, ist, dass der gesamte Volumenstrom durch Frischwasser gedeckt werden muss. Durch die Nachfrage beim Anlagenbetreiber konnte in

Erfahrung gebracht werden, dass der Frischwasservolumenstrom (\dot{V}_{FW}) eingedrosselt ($\sim 17 \text{ m}^3/\text{h}$) und der Volumenstrom der Zirkulation (\dot{V}_{Circ}) ungedrosselt ($\sim 90 \text{ m}^3/\text{h}$) erfolgt. Somit konnte ausgeschlossen werden, dass der gesamte Volumenstrom durch Frischwasser gedeckt wird.

Da keine Daten über den Messaufbau vorliegen, konnte der Einfluss von T_{FW} auf $T_{Circ,mess}$ nicht bestimmt werden. Erkennbar ist auch, dass die Austrittstemperatur der Zirkulation aus der Rohrleitung, $T_{CircOut,sim}$ nicht mit $T_{Circ,mess}$ übereinstimmt.

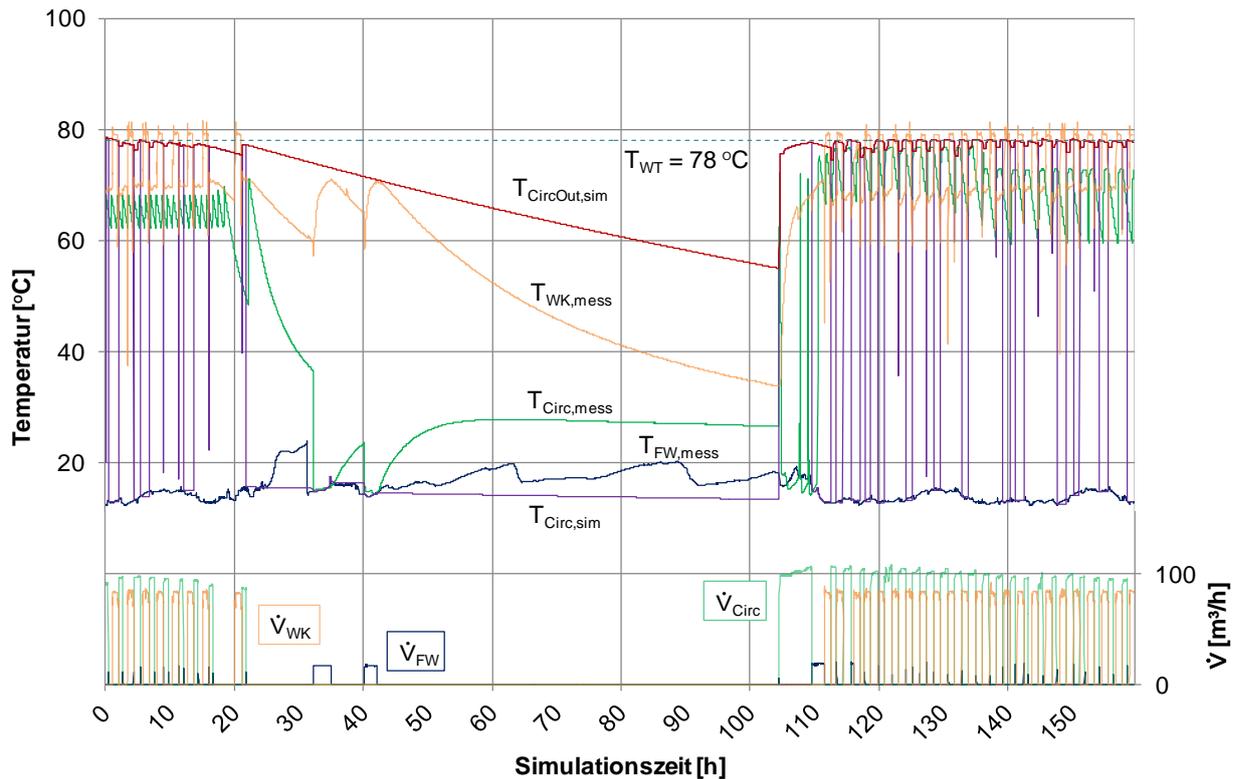


Abbildung 3-11: Temperaturen und Volumenströme (Inflow)

Bei den austretenden Temperaturen (Abbildung 3-12) aus den Speichern zeigte sich das die Annahme einer konstanten Austrittstemperatur für die Temperatur Warmwasser Netz (T_{Netz}) ihre Berechtigung hatte. Der Temperaturverlauf vom Warmwasser Sudhaus (T_{SH}) konnte, außer die kurzzeitigen Auskühlungen, wiedergegeben werden.

3 Simulationsaufbau und Ergebnisse

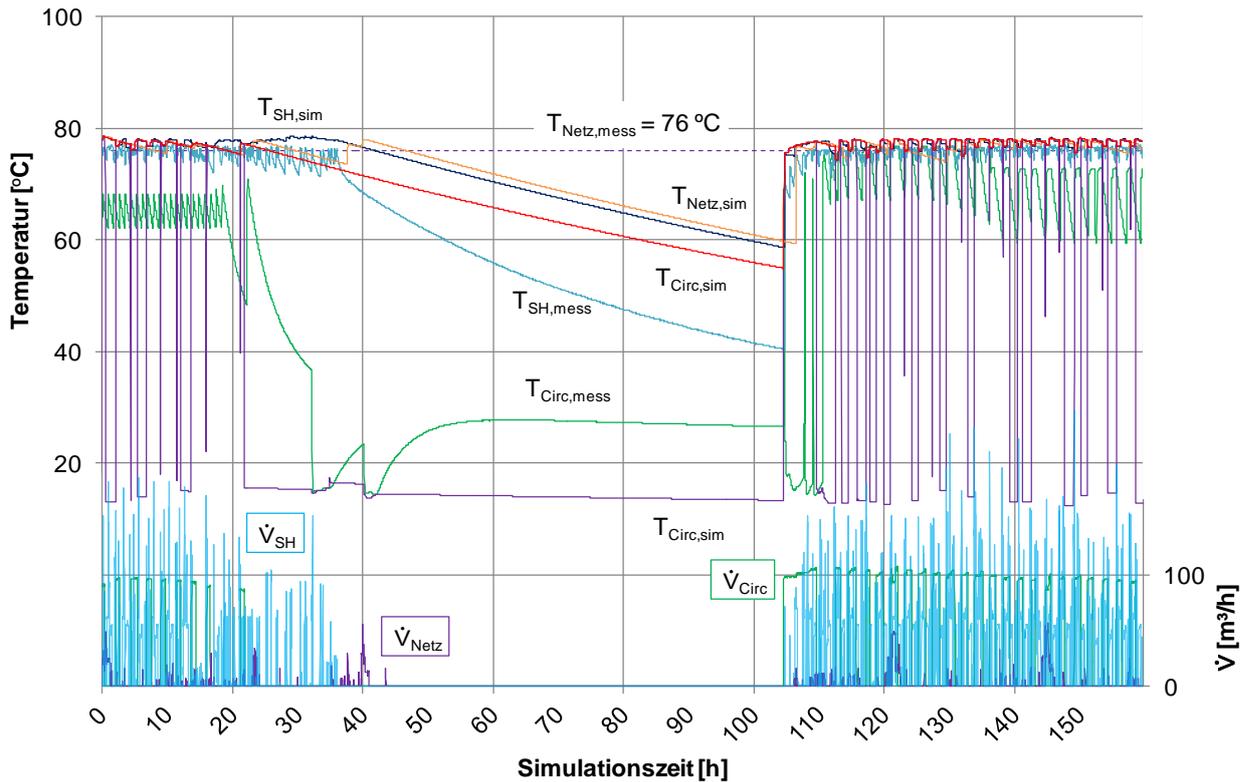


Abbildung 3-12: Temperaturen und Volumenströme (Outflow)

Der Vergleich der eintretenden und austretenden Leistungen (Abbildung 3-13) zeigte den Effekt der unterschiedlichen Temperaturen ($T_{Circ,mess}$, $T_{Circ,sim}$) deutlicher.

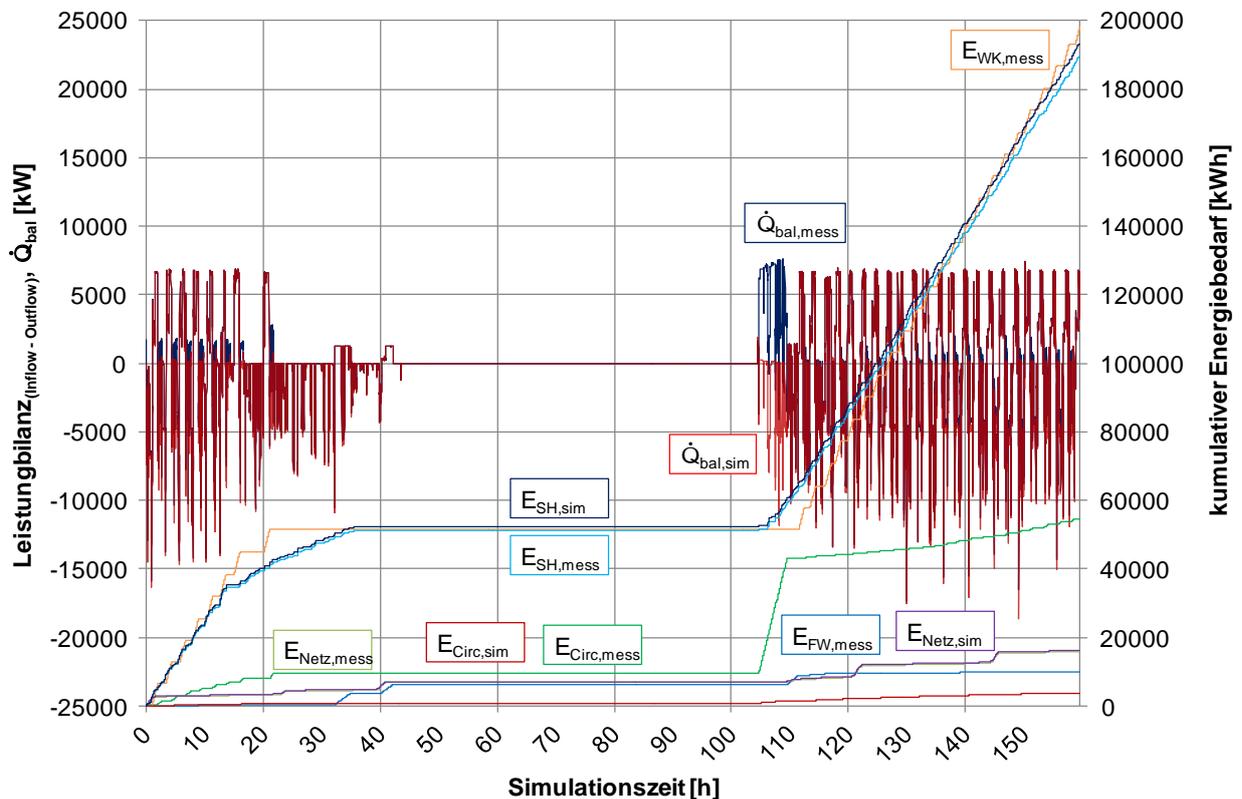


Abbildung 3-13: Leistungen und kumulativer Energiebedarf

Durch die Annahme einer konstanten Austrittstemperatur aus dem Wärmetauscher, ergaben sich für die austretenden Energien (E_{Netz} , E_{SH}) nur geringe Abweichungen. Da E_{Netz} nur ein Zehntel von E_{SH} entsprach, wurde der entsprechende Wertebereich skaliert in Abbildung 3-14 dargestellt.

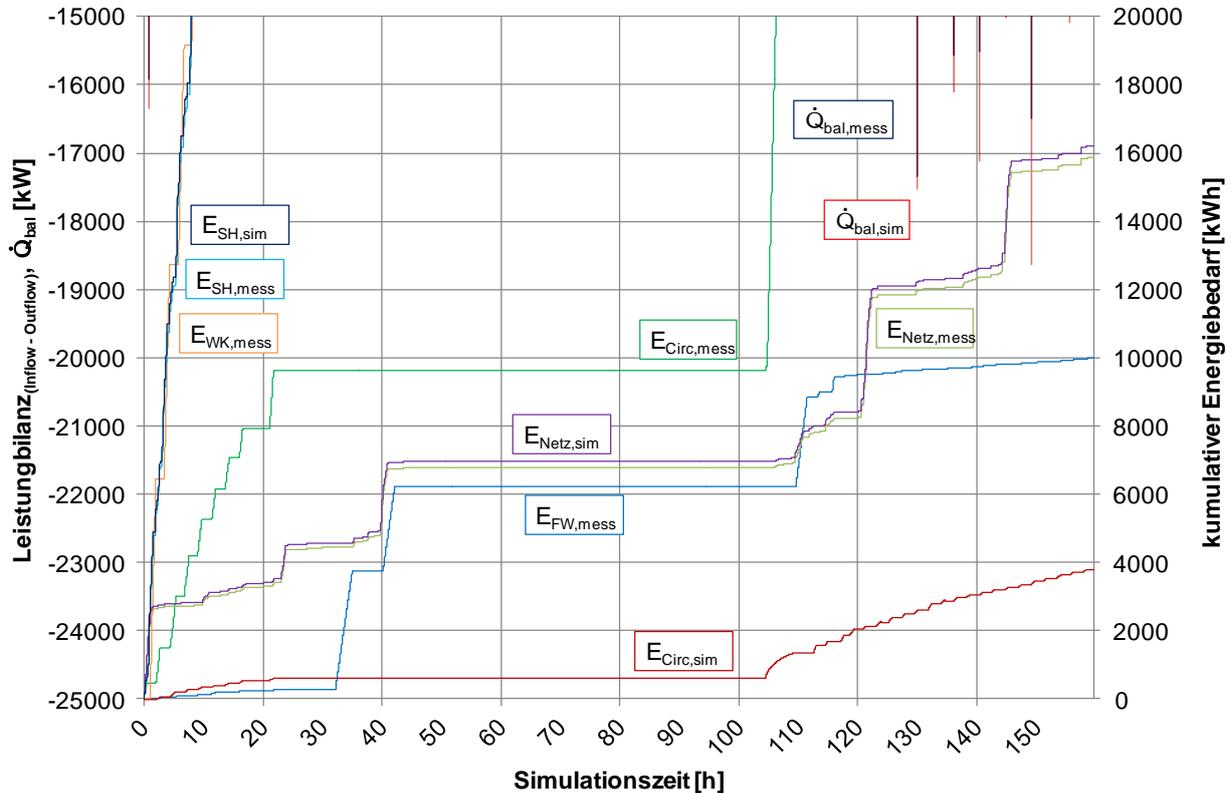


Abbildung 3-14: Leistungen und kumulativer Energiebedarf (skaliert)

d) Fazit

Bei Speichern mit einer variablen Füllhöhe kann die Leistung nicht durch die ein- und die austretende Temperatur, sowie dem Massenstrom durch den Doppelanschluss ($\dot{m}_{\text{in,DP1}} = \dot{m}_{\text{out,DP1}}$) bestimmt werden. Durch die Annahme der variablen Füllhöhe darf der eintretende Massenstrom vom austretenden Massenstrom abweichen. Dies bedingt eine getrennte Betrachtung des eintretenden und des austretenden kumulativen Energiebedarfs. Der eintretende kumulative Energiebedarf stellt für die Simulation einen Eingangsgröße dar, wohingegen der austretende kumulative Energiebedarf eine Ausgangsgröße der Simulation ist. Somit konnten nur die austretenden Energien verglichen werden.

Hier konnte die Simulation, obwohl viele Annahmen getroffen wurden, die Realität wiedergeben. Die Abweichung des kumulativen Energiebedarfs am Ende der Simulation betrug bei $E_{\text{SH}} = -1,96 \%$ und bei $E_{\text{Netz}} = -2,15 \%$.

Der berechnete kumulativen Energiebedarf aus den eintretenden Wärmeströmen von der Zirkulation (E_{Circ}) und dem Frischwasser (E_{FW}) können nur in der Summe genau beschrieben

werden. Die Aufteilung ist, durch die Annahmen bei der Aufteilung der Volumenströme, problematisch.

Obwohl sich herausgestellt hat, dass die zugrundeliegende Annahme, bei der Berechnung des Volumenstromes, nicht in allen Simulationsschritten erfüllt war, stimmte sie dennoch mit der Realität überein. Dies ist bedingt durch die Annahme einer konstanten Austrittstemperatur am Wärmetauscher ($T_{WT} = 78 \text{ °C}$) und das der gesamte Volumenstrom (Frischwasser und Zirkulation) aus der Messung bekannt war.

3.3 Graz (2150 m³) – Energie Steiermark

Der Warmwasserspeicher Graz (Abbildung 3-15) dient als Pufferspeicher im Fernwärmenetz der Steirischen Fernwärme GmbH. Er ist im Freien positioniert und besitzt nur am Behältermantel und an der Decke eine Dämmung mit Mineralwolle. Zusätzlich besitzt der obere Bereich des Speichers eine Dampfhaube, für die aber keine Messdaten vorliegen. Der Boden weist konstruktiv keine Dämmung auf. Die Anschlüsse zum Laden und Entladen befinden sich auf unterschiedlichen Höhen und sind als Doppelanschlüsse ausgeführt. Der austretende Massenstrom aus beiden Anschlüssen des Speichers wird zusammengefasst und an einem Messpunkt vermessen.

In Tabelle 3-3 sind die wichtigsten Daten der Simulation zusammengefasst worden.

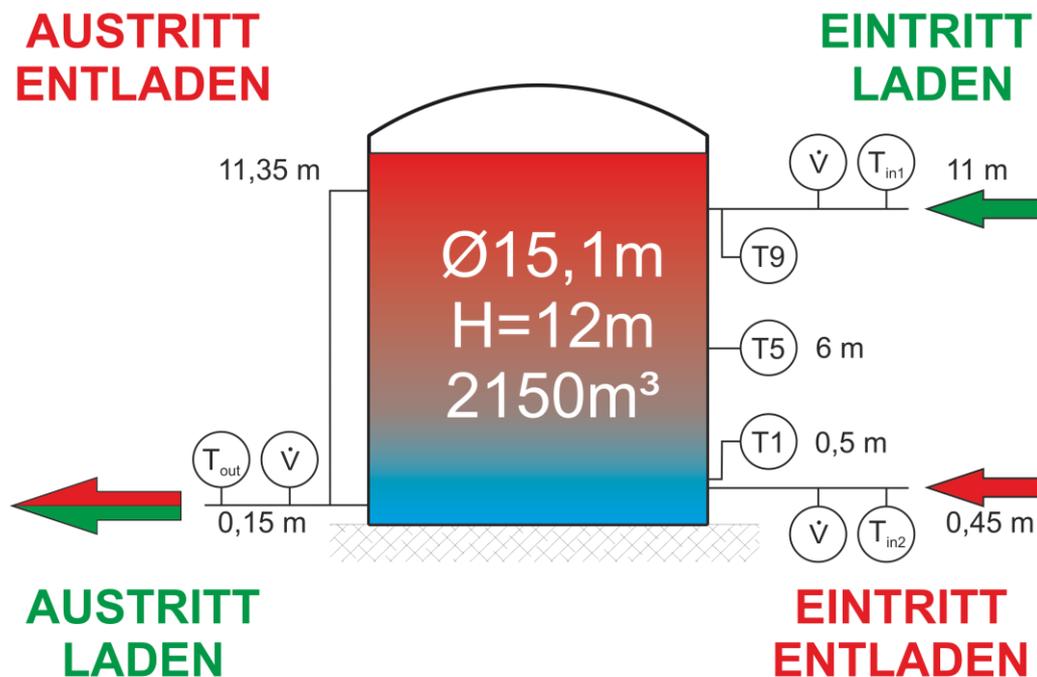


Abbildung 3-15: Schema Warmwasserspeicher Graz – Energie Steiermark

Tabelle 3-3: Eckdaten – Warmwasserspeicher Graz

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	144,2	h
Anzahl der Messpunkte im Speicher	9	-
Füllvolumen des Speichers	2050	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	0,7647	W/mK
Dämmstärke – Mantel	10	cm
Dämmstärke – Deckel	10	cm
Dämmstärke – Boden	2	cm

a) Annahmen und Vereinfachungen.

Da der Boden des Speichers keine Dämmung aufweist, musste die Bodentemperatur des umliegenden Erdreichs mitberücksichtigt werden. Die ungestörte Temperatur im Boden folgt im Jahresverlauf einer Sinusschwingung. Dies trifft auf den Warmwasserspeicher Graz nur bedingt zu, da er bereits seit mehreren Jahren laufend in Betrieb ist. Dadurch hat er das umliegende Erdreich bereits merklich erwärmt. Diese Wechselwirkung des Speichers mit dem Erdreich wurde auch in der Simulation festgestellt. Durch Abschätzung und Vorsimulationen ergab sich eine Bodentemperatur in der Umgebung des Speichers von etwa 35 °C. Mit dieser Temperatur konnte durch Berücksichtigung der Wärmeleitung eine fiktive Dämmstärke des Erdreichs berechnet werden.

Die Dampfhaut dient dazu die Wärmeverluste an der Oberseite des Speichers zu minimieren. In Abhängigkeit von den obersten Speichertemperaturen kann dadurch auch ein Energieeintrag in den Speicher erfolgen. Die Speichereinbauten, welche in den Übersichtsplänen ersichtlich sind, sind in Hinsicht auf die Geometrie, nicht im Detail bekannt. Das Füllvolumen wurde auf Basis der Übersichtspläne und der Dokumentation des Speichers abgeschätzt. Die Rohrlängen wurden anhand der Temperaturen und der Energieströme (eintretend und austretend) abgeschätzt.

b) Modellierung in TRNSYS

Die Abbildung des Warmwasserspeicher Graz in der Simulation ist in Abbildung 3-16 erkennbar. Die Massenströme und die Temperaturen wurden der Simulationsumgebung mittels einer Textdatei übergeben. Über die Modelle der Rohrleitungen (Type709) erfolgte die Be- und Entladung des Speichers. Der Austritt aus dem Speicher wurde mittels eines Mischers (Type649) zusammengefasst.

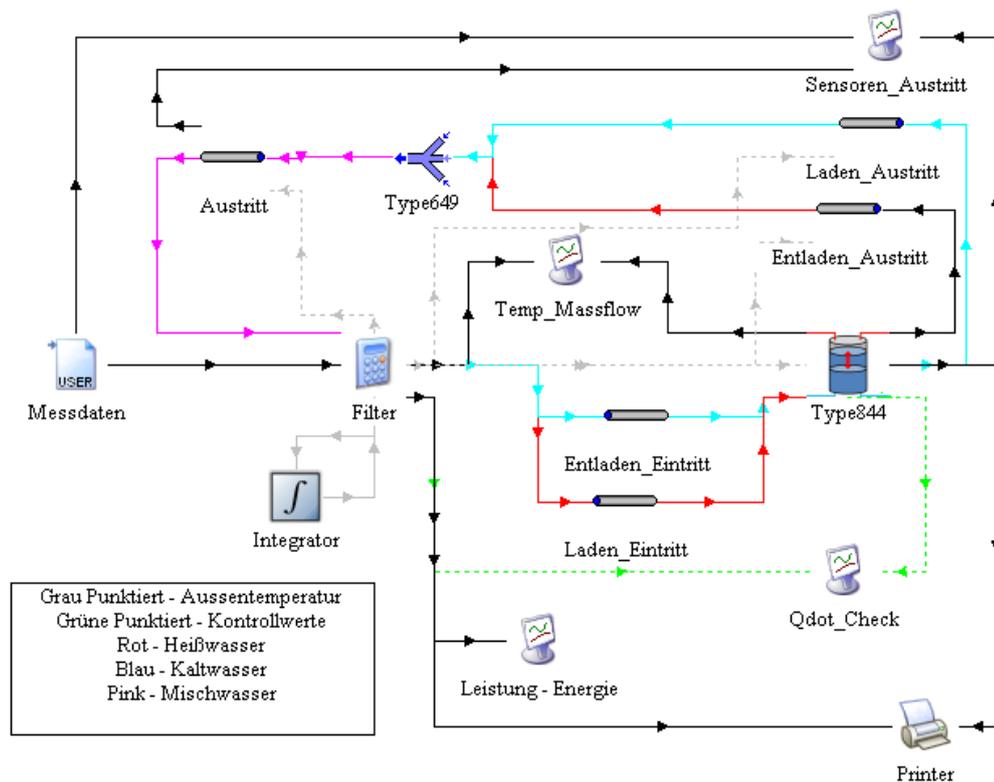


Abbildung 3-16: Simulationsaufbau Warmwasserspeicher Graz – Energie Steiermark

c) Vergleich Messung – Simulation

Abbildung 3-17 zeigt den Verlauf der Temperaturen (oben Messung, linke y-Achse; unten Simulation, rechte y-Achse) der im Speicher eingebauten Temperaturfühler. Im oberen Bereich der Abbildung 3-17 befinden sich die gemessenen Temperaturen, im unteren Bereich die dazugehörigen Temperaturen aus der Simulation. Die realen Messdaten wiesen eine Temperaturerhöhung (T_9) im Bereich zwischen der Stunde 10 und 20 auf. Dies könnte durch die Dampfauflage begründet sein, durch welche ein Energieeintrag in den realen Speicher erfolgte. Die Temperaturen in der Simulation waren geringfügig höher als die Messdaten. Am Behälterboden (T_1 , T_2) wies die Simulation eine geringfügig bessere Schichtung auf. Durch die fehlende Kenntnis über die Einlaufgeometrie in den Speicher und die der Temperaturfühler kann dies nicht näher betrachtet werden. Auch der Temperaturabfall im Bereich zwischen der Stunde 110 und 125 wies einen größeren Abfall (Messung 1,1 [K], Simulation 0,4 [K]) auf, obwohl der reale Speicher während der Beladung geringere Verluste zeigte.

3 Simulationsaufbau und Ergebnisse

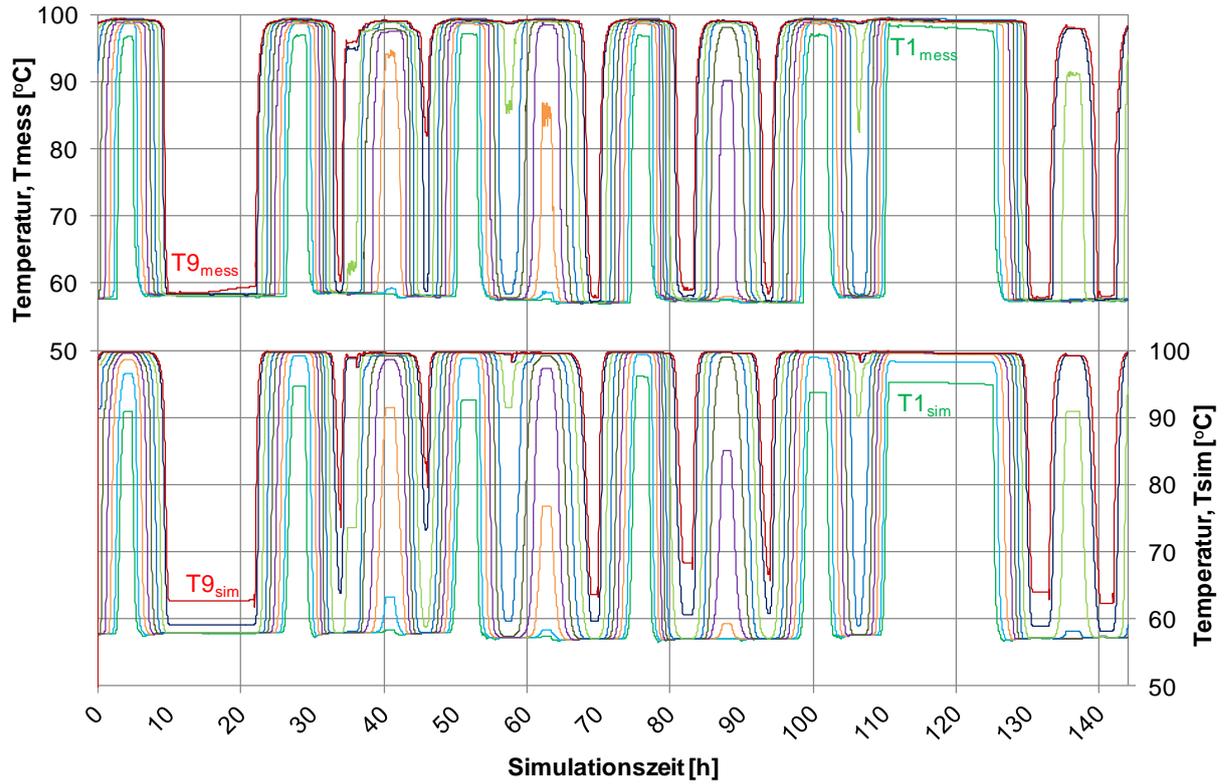


Abbildung 3-17: Temperaturverlauf im Speicher

Abbildung 3-18 zeigt den Verlauf der gemessenen Temperaturen der Rohrleitungen (linke y-Achse) sowie eine Bilanz des Volumenstroms (unterer Bereich, rechte y-Achse).

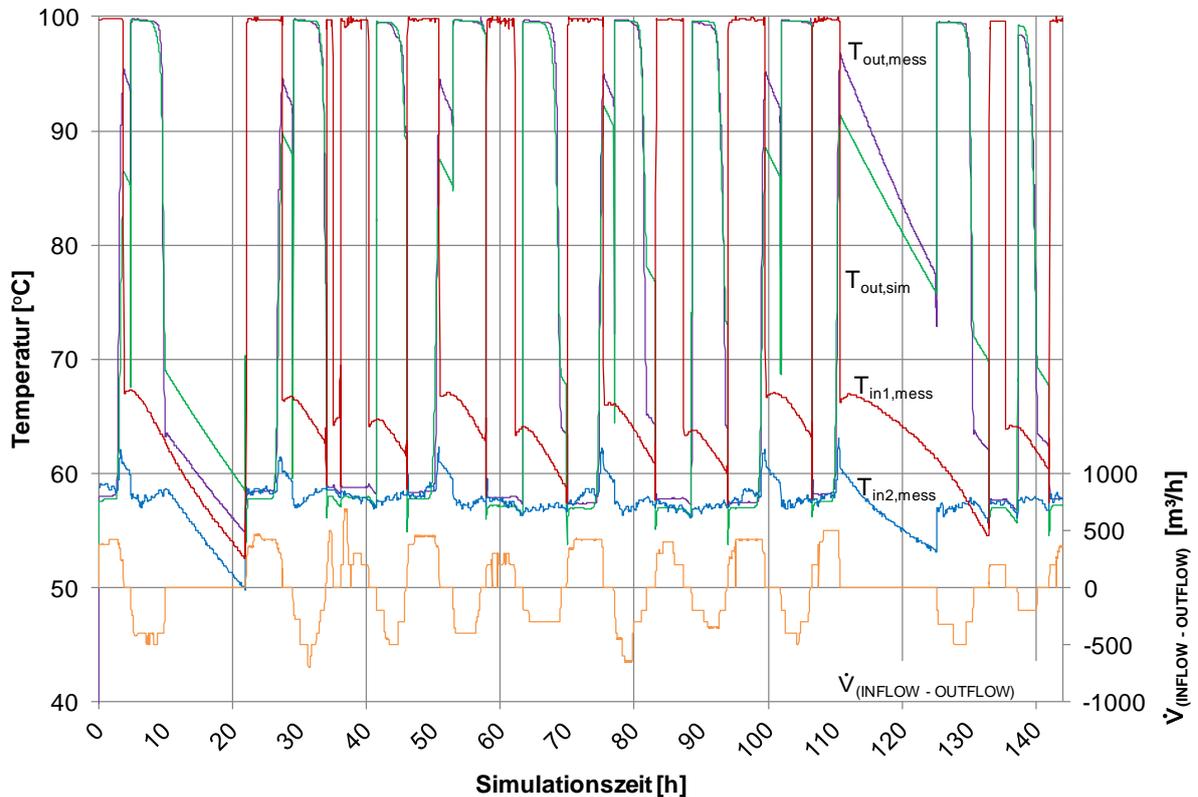


Abbildung 3-18: Temperaturverlauf der Rohrleitungen

Der Temperaturverlauf der Simulation wich, im Falle der Beladung (z.B. Stunde 70 bis 75) und des Ruhezustands (z.B. Stunde 110 bis 125), von der Messung ab. Der Temperaturverlauf während der Entladung (z.B. Stunde 63 bis 69) zeigte kleinere Abweichungen. Die Ursache für die Umkehrung der Abweichung der Austrittstemperaturen ($T_{out,mess}$, $T_{out,sim}$) könnte in den Randbedingungen der Simulation zu finden sein. Es musste jedem Rohrleitungsmodell, eine Initialisierungstemperatur, sowie konstante Stoffdaten, wie Dichte (ρ), Wärmeleitfähigkeit (λ) und Viskosität (η), zugewiesen werden.

Die eintretenden und austretenden Leistungen ($Q_{bal,mess}$, $Q_{bal,sim}$) sind in Abbildung 3-19 zu erkennen. Wie schon in der Abbildung 3-18 beschrieben war, weisen die Temperaturen nur geringe Abweichungen auf. Diese minimalen Abweichungen sind auch bei den kumulativen Energien erkennbar.

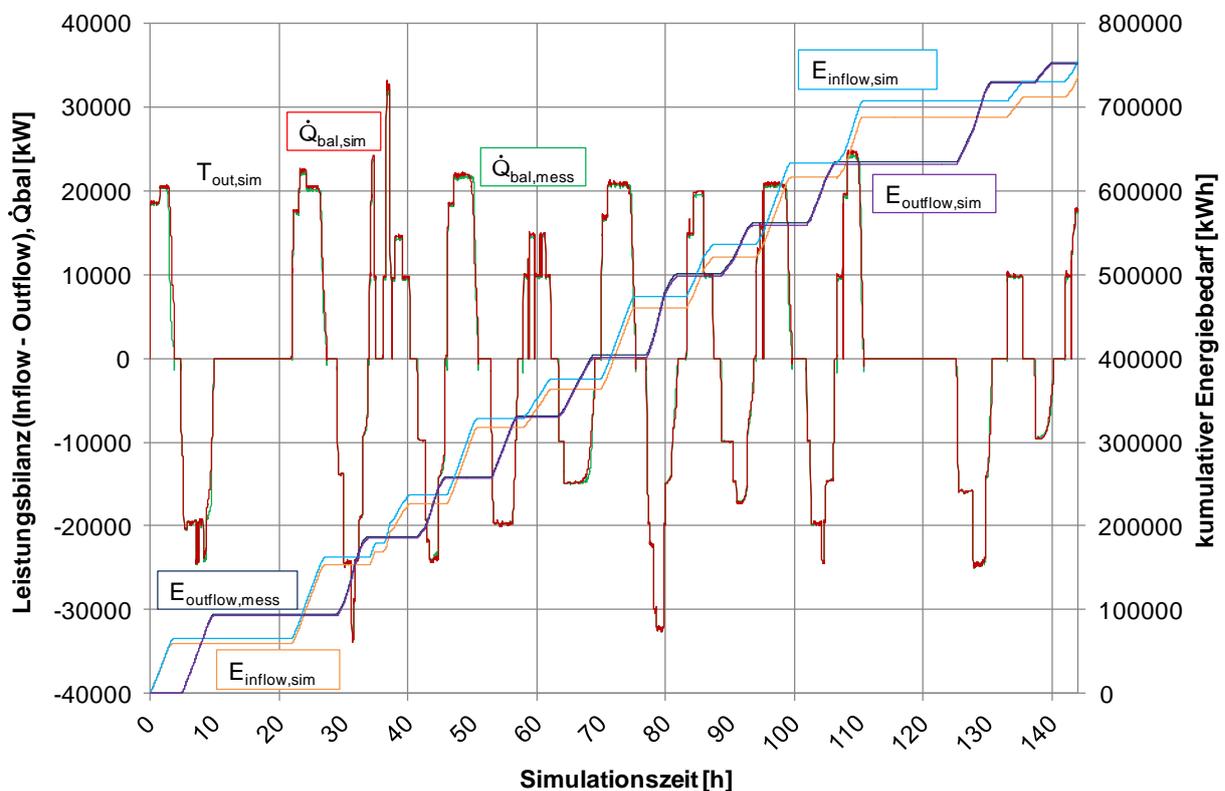


Abbildung 3-19: Leistung und kumulativer Energiebedarf

d) Fazit

Der kumulative Energiebedarf wies, am Ende der Simulation, eine Abweichung von -2,2 % bei der Beladung und -0,1 % bei der Entladung auf. Obwohl die Rohrlängen unbekannt waren und keine Information über die Messung selbst bekannt war, stimmten diese mit der Realität gut überein. Das unbekannte Füllvolumen stellte in der Simulation kein Problem dar.

Den Einfluss der Dampfhaube, im Sinne einer Energieerhöhung, war in den Messdaten ($T_{9, \text{mess}}$) nur zwischen der Stunde 10 und 20 zu erkennen. Da in der Simulation die weiteren Einbauten, wie das Be- und Entlüftungsventil und der Überlauftrichter nicht berücksichtigt wurden, ist davon auszugehen, dass diese Verluste durch die Dampfhaube kompensiert werden. Die fehlende Dämmung des Behälterbodens (T_1) wurde durch die Berechnung einer äquivalenten Dämmstärke berücksichtigt und stellte für diesen, bereits länger in Betrieb stehenden Speicher (stationärer Zustand), eine hinreichend gute Näherung dar.

3.4 Rottenmann (54 m³) – Energie Steiermark

Der Speicher Rottenmann 54 (Abbildung 3-20) ist einer der beiden Pufferspeicher im Motorheizkraftwerk (MHKW) Rottenmann. Der zweite, Rottenmann 250 wird in Kapitel 3.5 behandelt. Die Beladung und Entladung der beiden Speicher erfolgt mittels zweier Regelklappen, wobei immer nur ein Speicher in Betrieb sein darf. Beide Speicher sind im Freien positioniert, die Außenlufttemperatur ist aus den Daten der Steuerung bekannt. Der Speicher besitzt zwei Doppelanschlüsse die zum Laden, als auch zum Entladen, benutzt werden können. Der Behälterboden und der Deckel sind als Klöpperboden ausgeführt, der Behälter weist eine durchgehende Dämmung mit Mineralwolle auf. Die wichtigsten Daten sind in Tabelle 3-4 zusammenfasst.

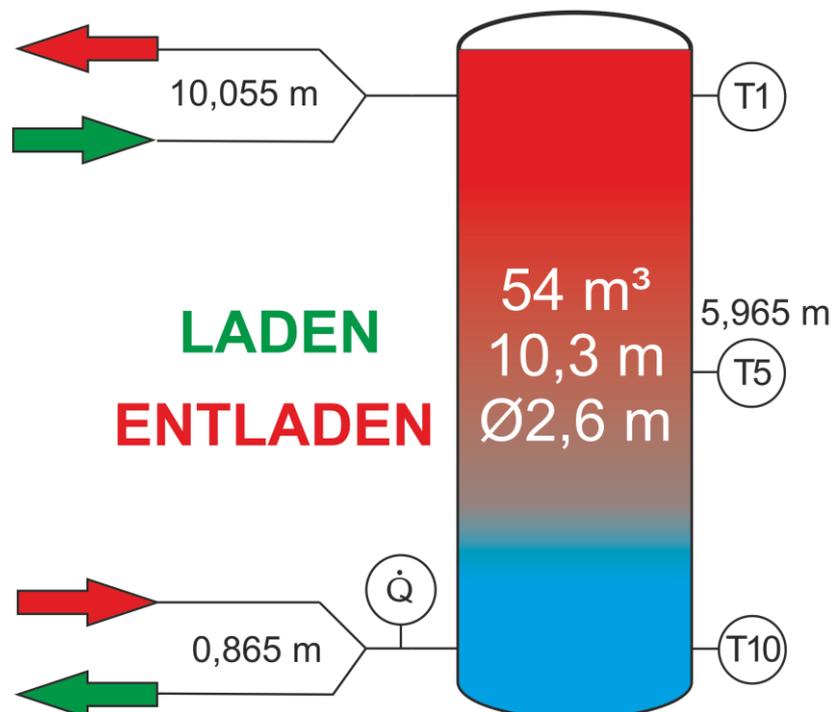


Abbildung 3-20: Schema Rottenmann 54

Tabelle 3-4: Eckdaten – Rottenmann 54

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	144	h
Anzahl der Messpunkte im Speicher	10	-
Füllvolumen des Speichers	52	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	1,8135	W/mK
Dämmstärke – Mantel	20	cm
Dämmstärke – Deckel	14,35	cm
Dämmstärke – Boden	14,35	cm

a) Annahmen und Vereinfachungen.

Als Datensatz standen die momentan übertragene Wärmeleistung, die Stellung der Regelklappen sowie die Temperaturfühler der beiden Speicher (Rottenmann 54 und Rottenmann 250) zur Verfügung. Der Volumenstrom wurde aus der übertragenen Leistung (\dot{Q}), der Stellung der jeweiligen Regelklappe, sowie den beiden Temperaturen ($T_{1_{mess}}$, $T_{10_{mess}}$) im Speicher bestimmt. Die Aufteilung der Be- und Entladung auf die beiden Speicher erfolgt über die Stellung der Regelklappen. Das Füllvolumen wurde auf Basis der vorliegenden Zeichnungen abgeschätzt.

b) Modellierung in TRNSYS

Beide Speicher wurden in einer gemeinsamen Simulation (Abbildung 3-21) abgebildet, um mögliche Fehler oder Einflüsse der Regelung, der realen Anlage, herauszufinden.

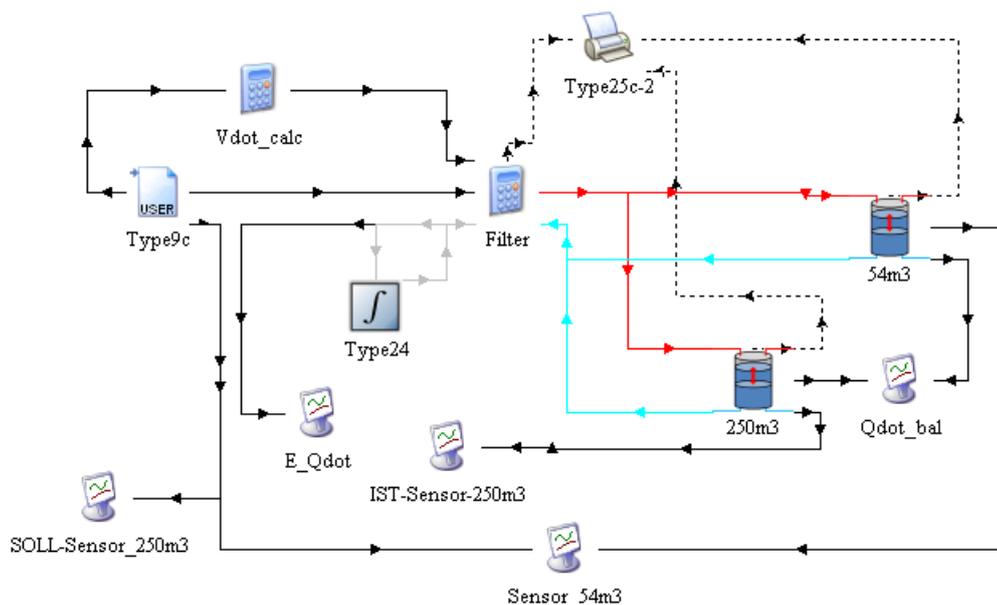


Abbildung 3-21: Simulationsaufbau MHKW Rottenmann (54 und 250)

Die Berechnung des Volumenstromes aus den Messdaten erfolgte in TRNSYS in einer eigenen Equation (Vdot_calc in Abbildung 3-21). Die Regelung der Beladung und Entladung wurde in der Equation (Filter in Abbildung 3-21) durchgeführt. Da keine Eintrittstemperaturen und Austrittstemperaturen vorlagen wurde auf eine Modellierung von allfälligen Rohrleitungen verzichtet. Die Anschlussverluste wurden jedoch berücksichtigt.

c) Vergleich Messung – Simulation

In der folgenden Abbildung ist im oberen Bereich die gemessenen Temperaturen (linke y-Achse) und im unteren Bereich die Temperaturen der Simulation (rechte y-Achse) dargestellt (Abbildung 3-22).

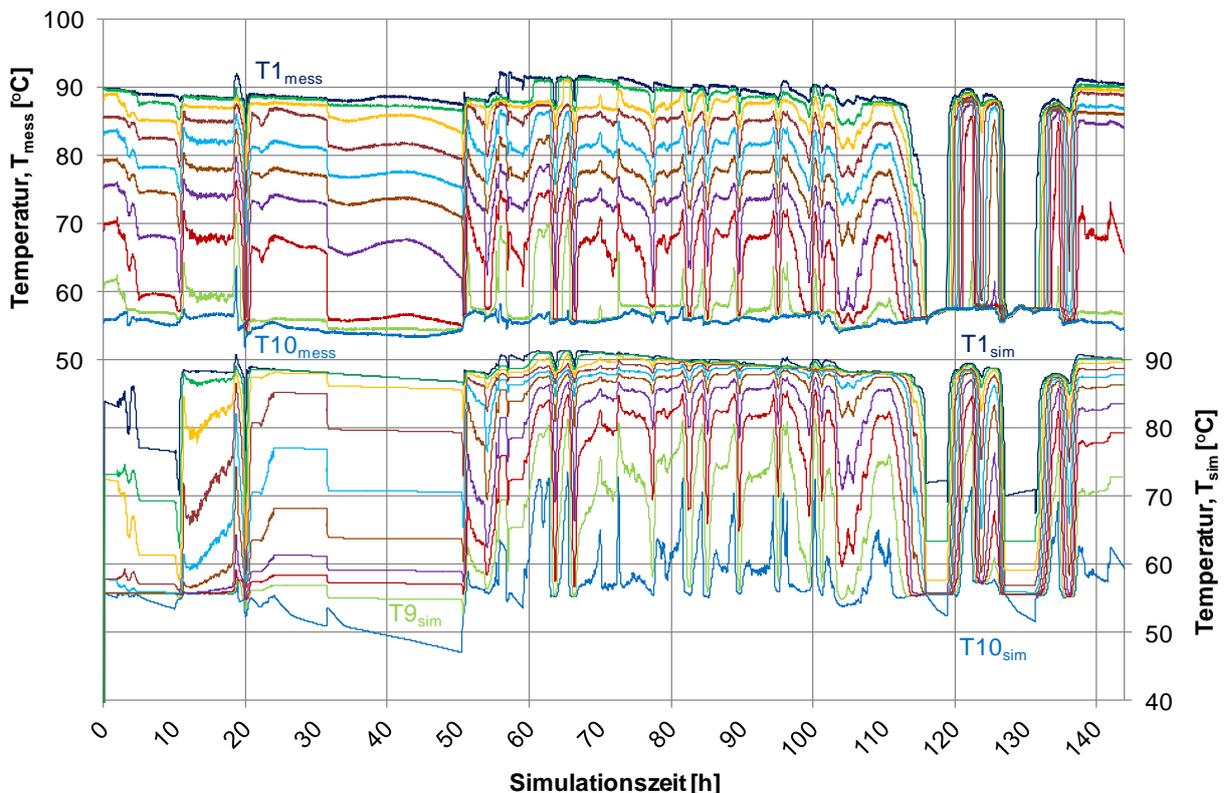


Abbildung 3-22: Temperaturverlauf – Sensoren Rottenmann 54

Die Temperaturschichtung ist in der Simulation besser, obwohl das Volumen eines Knotens auf 1 m^3 angepasst wurde (54 Knoten). Vorsimulationen mit 100 Knoten zeigten im Bereich von der Stunde 70 bis 95, dass sich in der Simulation nur der Temperaturfühler ($T_{9_{sim}}$) merklich änderte. Deshalb erfolgte eine Reduzierung der Anzahl der Knoten auf 54 im Speichermodell (Type844) um die schlechtere Schichtung im realen Speicher nachzubilden. Dadurch konnte der Temperaturverlauf der Simulation an den der Messung angepasst werden.

Im Bereich zwischen der Stunde 10 und 20 weist der reale Speicher keine Änderung auf, obwohl der Speicher in dieser Zeit eigentlich Beladen wird. Aus dem Temperaturverlauf

(Abbildung 3-23) erkennt man, dass die gemessene Eintrittstemperatur $T_{10_{\text{sim}}}$ in den Bereichen eines Volumenstromes hohe Temperaturspitzen aufweist. Dies resultierte aus der Beladung, welche in der Messung nicht zu sehen war. Die Aufteilung des Massenstromes auf die beiden Speicher erfolgte durch die Stellung der beiden Regelklappen. Die gemessenen Werte der Stellung der Regelklappen mussten zur Berücksichtigung der Messunsicherheiten gefiltert werden. Dieser Filter berücksichtigte Werte größer gleich 100 % als Regelklappe „offen“ (Wert = 1) und allen anderen Werte mit Regelklappe „zu“ (Wert = 0). Da auch die gemessene Leistung (\dot{Q}) Messunsicherheiten unterliegt, wurde der berechnete Volumenstrom gefiltert. Werte kleiner als $1 \text{ m}^3/\text{h}$ und Werte größer als $100 \text{ m}^3/\text{h}$ wurden nicht berücksichtigt.

Mangels Daten über die Messung, die eingesetzten Messinstrumente und ihrer Genauigkeit, konnten die eingesetzten Filter nicht genauer parametrisiert werden.

Da als Eintrittstemperaturen die Speichertemperatur angenommen wurde, bedeutet dies, dass für die Änderung von $T_{1_{\text{mess}}}$ eine höhere Eintrittstemperatur notwendig war. $T_{1_{\text{sim}}}$ weist im Bereich der Sprünge immer kleinere Beträge als $T_{1_{\text{mess}}}$ auf.

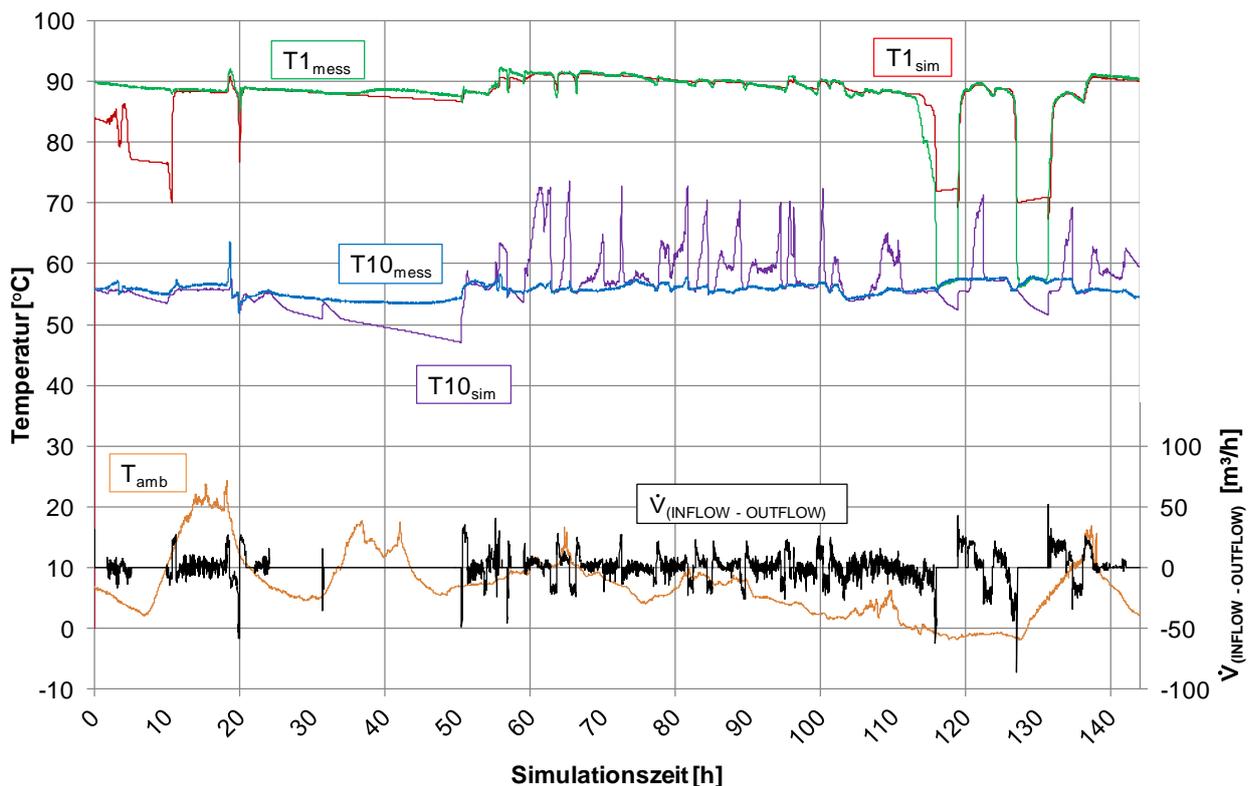


Abbildung 3-23: Temperaturverlauf beim Beladen und Entladen

Die Leistungsbilanz (Abbildung 3-24), welche aus der eintretenden und der austretenden Temperatur berechnet worden ist, wies bedingt durch den Unterschied von T_{10} erhebliche Abweichungen auf, welche sich auch am kumulativen Energiebedarf äußerte.

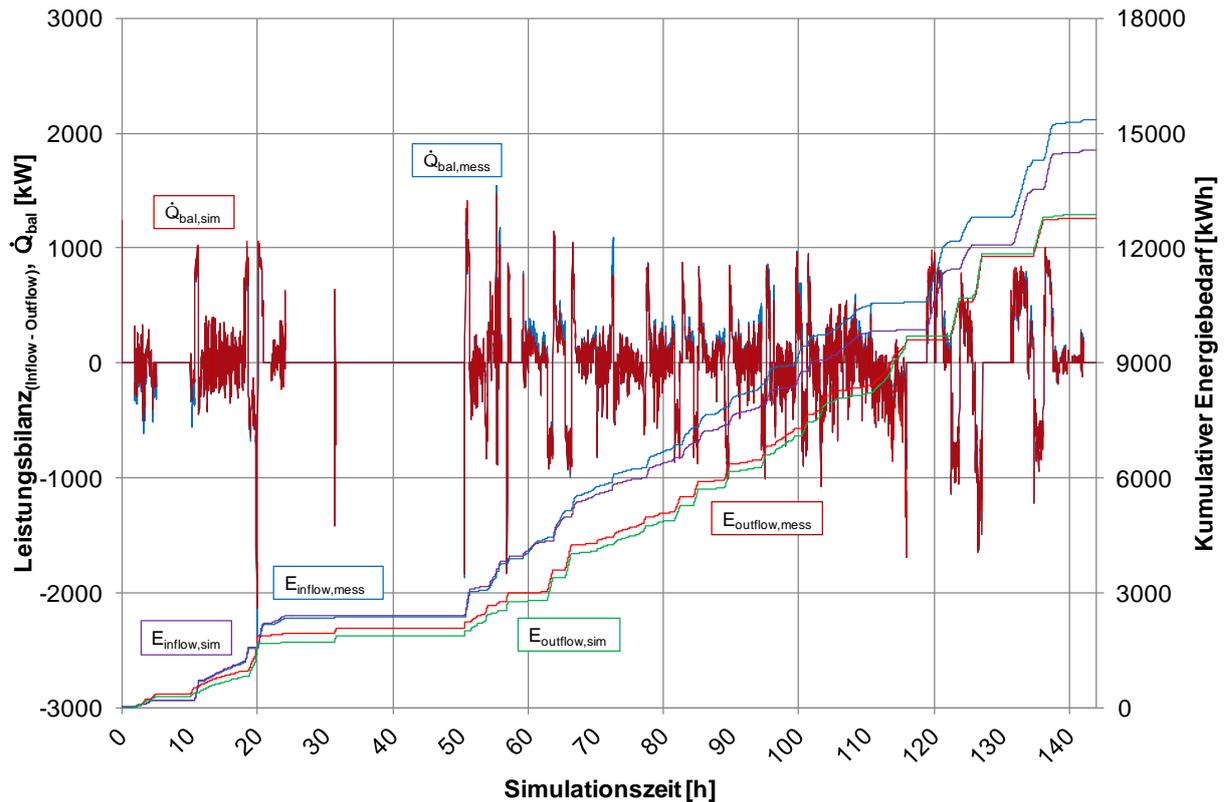


Abbildung 3-24: Leistung und kumulativer Energiebedarf

d) Fazit

Bedingt durch die Annahmen der Speichertemperatur als Eintrittstemperatur wurde der eintretende kumulative Energiebedarf um 9,4 % unterschätzt. Der austretende kumulative Energiebedarf, welche von der Temperatur T10 abhängt, wies eine Überschätzung von 2,4 % auf. Die Berechnung des Volumenstroms aus der Leistung und den Temperaturen im Speicher ist mit Unsicherheiten behaftet. Zur genaueren Bestimmung sollten auch die Messgenauigkeit der eingesetzten Messgeräte bekannt sein, um zwischen einem geänderten Messwert und einem Rauschen unterscheiden zu können. Weiters konnte festgestellt werden, dass die Knotenanzahl nicht zu hoch gewählt werden sollte, da dies einer zu guten Schichtung entspricht. Dies hat einen großen Einfluss auf das Temperaturniveau in einem Speicher und wirkt sich direkt auf die Austrittstemperaturen während der Entladung aus.

3.5 Rottenmann (250 m³) – Energie Steiermark

Der Speicher Rottenmann 250 (Abbildung 3-25) war der zweiten Pufferspeicher im MHKW Rottenmann, welcher analysiert wurde. Die Speicher unterscheiden sich in der Dämmstärke, sowie in der Anzahl der Temperatursensoren. Der Behälterboden und der Deckel wurden als Korbbogenboden (etwas höher als ein Klöpperboden) ausgeführt und der Behälter weist eine durchgehende Dämmung auf. Beide Speicher wurden gemeinsam abgebildet, es gelten die Annahmen und Vereinfachungen (a), sowie die Modellierung (b), aus Kapitel 3.4. Die wichtigsten Daten wurden in Tabelle 3-5 zusammengefasst.

Tabelle 3-5: Eckdaten – Rottenmann 250

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	144	h
Anzahl der Messpunkte im Speicher	20	-
Füllvolumen des Speichers	227	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	1,3204	W/mK
Dämmstärke – Mantel	25	cm
Dämmstärke – Deckel	17,2	cm
Dämmstärke – Boden	17,2	cm

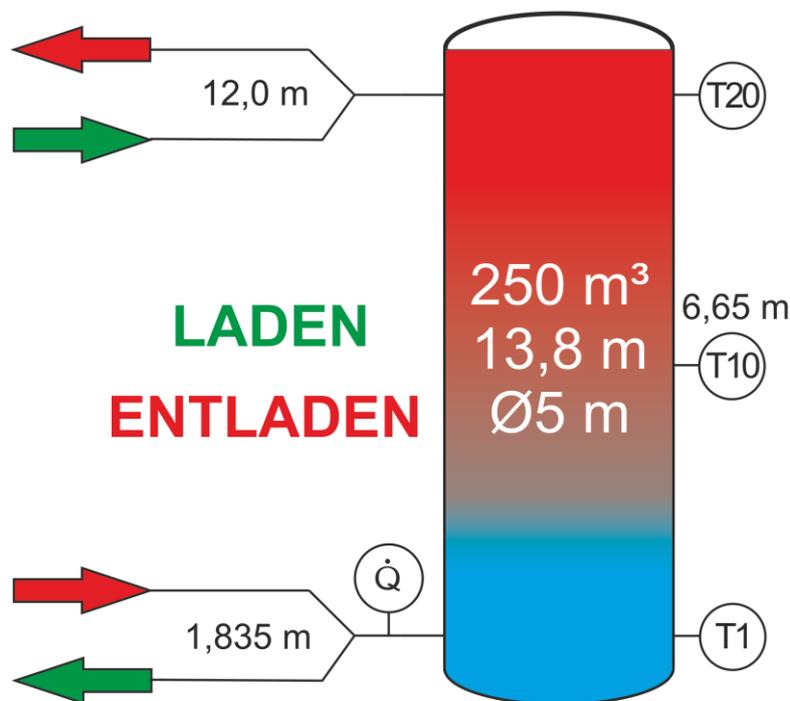


Abbildung 3-25: Schema Fernwärmespeicher MHKW Rottenmann 250

c) Vergleich Messung – Simulation

In der folgenden Abbildung ist im oberen Bereich die gemessenen Temperaturen (linke y-Achse) und im unteren Bereich die Temperaturen der Simulation (rechte y-Achse) dargestellt (Abbildung 3-26). Der Verlauf der Temperaturen konnte bis zur Simulationszeit von 40 Stunden wiedergegeben werden. Im darauffolgenden Bereich kam es zu kurzzeitigen Beladungen und Entladungen. Weiters stiegen die Temperaturen in diesen Bereich der Simulation an, und fielen in den Messdaten ab. Zeitgleich kam es in Rottenmann 54 (Abbildung 3-22) zu einer Erhöhung der Temperaturen im Speicher. Dies könnte auf fehlerhafte Ventile zurückzuführen sein.

Der Bereich nach 50 Stunden zeigte, dass der reale Speicher nahezu entladen war. Die Simulation hingegen wies einen erheblich größeren Energieinhalt auf. Der spätere Abfall der Temperatur $T_{20,sim}$ im Vergleich zu $T_{20,mess}$ war durch der Vorgabe der gemessenen Temperatur ($T_{20,mess}$) als Eintrittstemperatur für den Massenstrom in der Simulation bedingt. Dies ist in der Abbildung 3-26 ab der Stunde 55 zu erkennen, wo es im realen Speicher zu einer Erhöhung der Temperatur ($T_{20,mess}$) und in der Simulation gleichzeitig zu einem Abfall der Temperatur ($T_{20,sim}$) kommt. Im späteren Verlauf setzt sich diese Tendenz fort (z.B. Stunde 72 und Stunde 82).

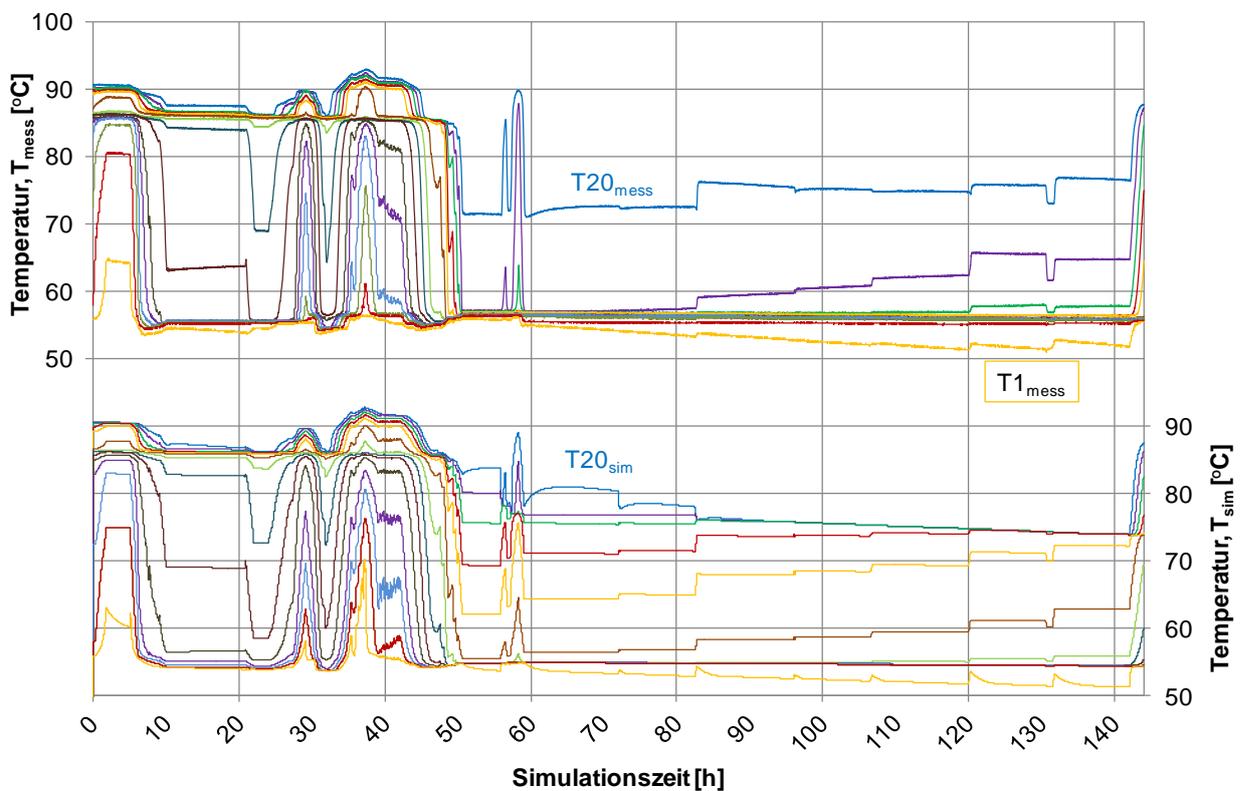


Abbildung 3-26: Temperaturverlauf der Sensoren

Wie beim Speicher Rottenmann 54 konnte auch hier festgestellt werden, dass die Temperaturspitzen nicht von der Simulation nachgebildet werden konnten (Abbildung 3-27).

3 Simulationsaufbau und Ergebnisse

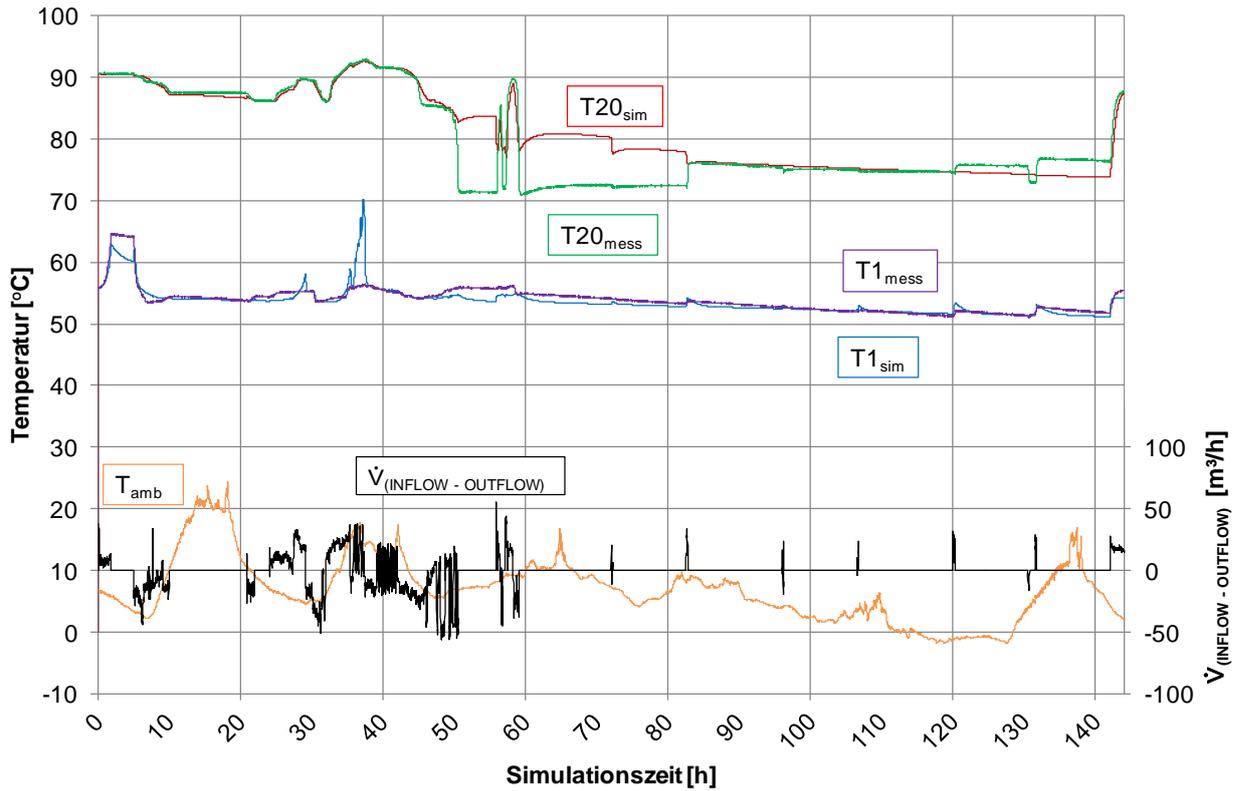


Abbildung 3-27: Temperaturverlauf beim Beladen und Entladen

In der Leistungsbilanz (Abbildung 3-28) zeigte sich, wie schon in den letzten Abbildungen, dass die Simulation bis zu einer Simulationszeit von 50 Stunden, die Realität widerspiegelt.

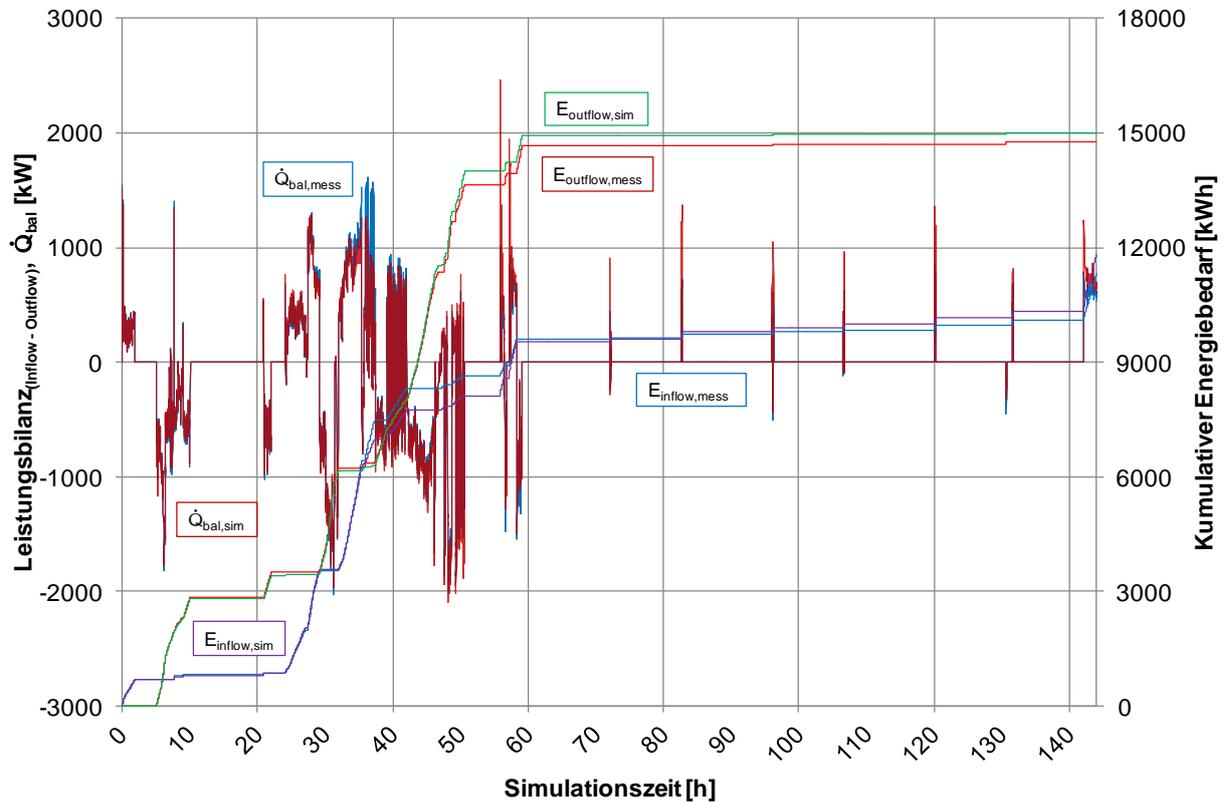


Abbildung 3-28: Leistung und kumulativer Energiebedarf

Im späteren Simulationsverlauf kam es durch die Annahme der gemessenen Temperatur im Speicher als Eintrittstemperatur der Simulation zu Abweichungen im Temperaturverlauf.

d) Fazit

Wie schon bei Rottenmann 54 weist auch dieser Speicher Abweichungen bei den Eintrittstemperaturen auf. Der ermittelte Volumenstrom stellte sich auch hier als ein Kernproblem heraus. Zusätzlich lässt sich die zeitgleiche Temperaturerhöhung zur Stunde 40 im Rottenmann 54 und Rottenmann 250 nicht, auf Basis der Messdaten, erklären.

Der kumulative Energiebedarf der Beladung ($E_{\text{inflow, sim}}$) wies am Simulationseende eine Abweichung von 4,5 % zu den Messwerten auf. Der kumulative Energiebedarf bei der Entladung ($E_{\text{outflow, sim}}$) wich am Ende der Simulation um 1,6 % ab.

3.6 Knittelfeld (105 m³) – Energie Steiermark

Das Motorheizkraftwerk (MHKW) Knittelfeld dient der Fernwärmeversorgung, sowie der Stromerzeugung. Durch eine Kraft- Wärme- Kopplung (KWK) wird die bei der Stromerzeugung anfallende Wärme für die Fernwärmeversorgung von über 900 Kunden genutzt. Das MHKW besteht aus zwei Gasmotoren (KWK-Anlage) zur Grundlastabdeckung, zwei Gaskessel zur Spitzenlastabdeckung und als Ausfallsreserve, sowie zwei Pufferspeicher (Knittelfeld 105, Knittelfeld 320) zur Speicherung der anfallenden Wärme.

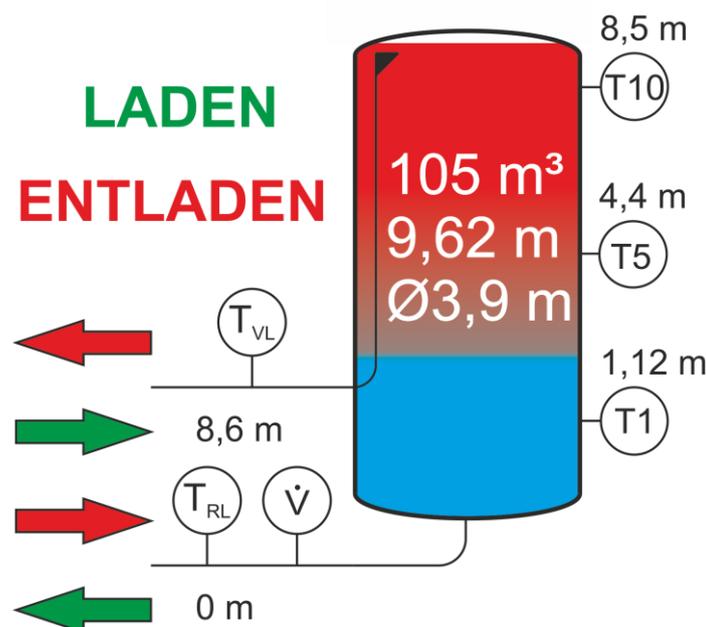


Abbildung 3-29: Schema Fernwärmespeicher MHKW Knittelfeld 105

Beide Speicher weisen zwei Doppelanschlüsse auf, über welche die Be- und Entladung erfolgt. Der Volumenstrom ist aus den Daten der Anlagensteuerung bekannt, die Aufteilung auf die beiden Speicher erfolgt mittels zweier Regelklappen, wobei immer nur ein Speicher geladen werden darf. Die beiden Speicher sind im Freien positioniert und die Außentemperatur der Außenluft war bekannt. Der Deckel und der Boden der beiden Speicherbehälter sind als Korbbogenboden ausgeführt (etwas höher als der Klöpperboden).

Der Speicher Knittelfeld 105 (Abbildung 3-29) weist eine durchgängige Dämmstärke von 20 cm Mineralwolle auf. In der Tabelle 3-6 sind die wichtigsten Daten zusammengefasst.

Tabelle 3-6: Eckdaten – Knittelfeld 105

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	120	h
Anzahl der Messpunkte	10	-
Volumen	103	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	1,2892	W/mK
Dämmstärke – Mantel	20	cm
Dämmstärke – Deckel	13,8	cm
Dämmstärke – Boden	13,8	cm

a) Annahmen und Vereinfachungen.

Für den Speicher Knittelfeld 105 und Knittelfeld 320 standen die Temperaturen der Messfühler, die Temperatur im Vorlauf des Kraftwerks, die Temperaturen des Rücklaufs jedes Speichers, die Stellung der Regelventile und der Volumenstrom zur Verfügung. Diese Daten wurden aus der Steuerung der Anlage ausgelesen. Die Temperaturen der Messfühler weisen am Start der Simulation eine maximale Abweichung von 3,2°C auf. Dadurch wurde der Speicher mit einer mittleren Temperatur von 64,7 °C initialisiert. Das Füllvolumen wurde auf Basis der Zeichnungen der Speicherbehälter abgeschätzt.

b) Modellierung in TRNSYS

Beide Speicher wurden in einer Simulation zusammengefasst, da sie während der Beladung aus einer gemeinsamen Rohrleitung (Vorlauf, T_{VL}) gespeist werden und im Falle der Entladung eine Rückeinspeisung stattfindet. In Abbildung 3-30 ist erkennbar, dass dadurch ein Mischer (Type649) und ein Stromteiler (Type647) notwendig wurden. Die Rücklaufleitungen wurden nicht zusammengefasst, da für jeden Speicher eine gemessene Rücklauftemperatur (T_{RL}) vorlag. Zur Modellierung der Rohrleitungen kam der Type 604a zum Einsatz. Dieser Type erlaubt Massenströme in beiden Richtungen, jedoch dürfen diese nicht gleichzeitig auftreten. Zur Abschätzung der Initialisierungstemperaturen des

Speichermediums, der Rohrwandung und der Isolierung wurde eine Vorsimulation durchgeführt (siehe auch Kapitel 2.4.4).

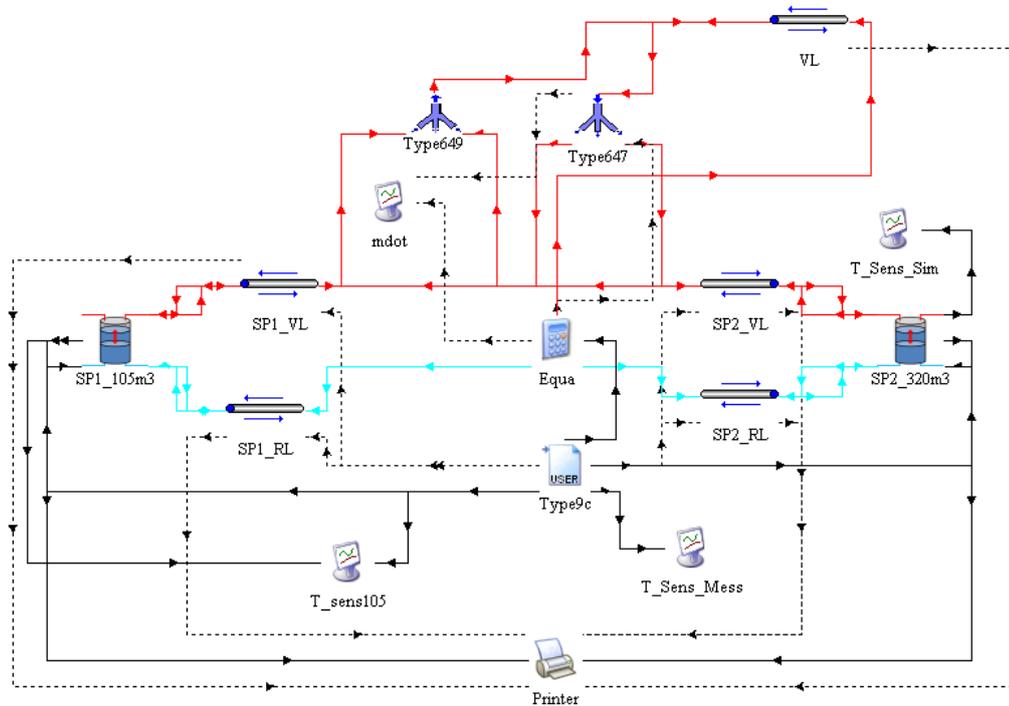


Abbildung 3-30: Simulationsaufbau MHKW Knittelfeld (105 und 320)

c) Vergleich Messung – Simulation

In der Abbildung 3-31 ist der Verlauf der gemessenen Temperaturen (linke y-Achse) und der simulierten Temperaturen (rechte y-Achse) zu erkennen. Es zeigte sich, dass die Temperatur des obersten Temperaturfühlers der Messung ($T_{10_{mess}}$) im Vergleich zur Simulation ($T_{10_{sim}}$) einen größeren Abfall zur Stunde 15 aufweist, obwohl der Speicher, bedingt durch die Annahme einer konstanten Initialisierungstemperatur zum Simulationsbeginn, eine geringere Temperatur ($T_{10_{sim}}$) hatte. Dieser Effekt kann auch beim Temperaturfühler $T_{4_{mess}}$ beobachtet werden. Zur Stunde 60 wies $T_{4_{mess}}$ eine Temperatur von $91,5\text{ °C}$ auf, während die Simulation ($T_{4_{sim}}$) nur eine Temperatur von $78,7\text{ °C}$ ergab. Jedoch war die Temperatur der Messung ($T_{4_{mess}}$) zur Stunde 105 erheblich kleiner als die der Simulation ($T_{4_{sim}}$). Der unterste Temperaturfühler $T_{1_{mess}}$ zeigte zum Simulationsbeginn eine geringfügig höhere Temperatur als die Simulation ($T_{1_{sim}}$) an, jedoch kehrte sich dies zum Simulationsende um.

Diese beschriebenen Vorgänge können nicht allein durch die Verluste begründet sein, da sie keine einheitliche Tendenz zeigen. Dies könnte durch eine fehlerhafte Aufteilung des Volumenstromes durch die Regelventile begründet sein.

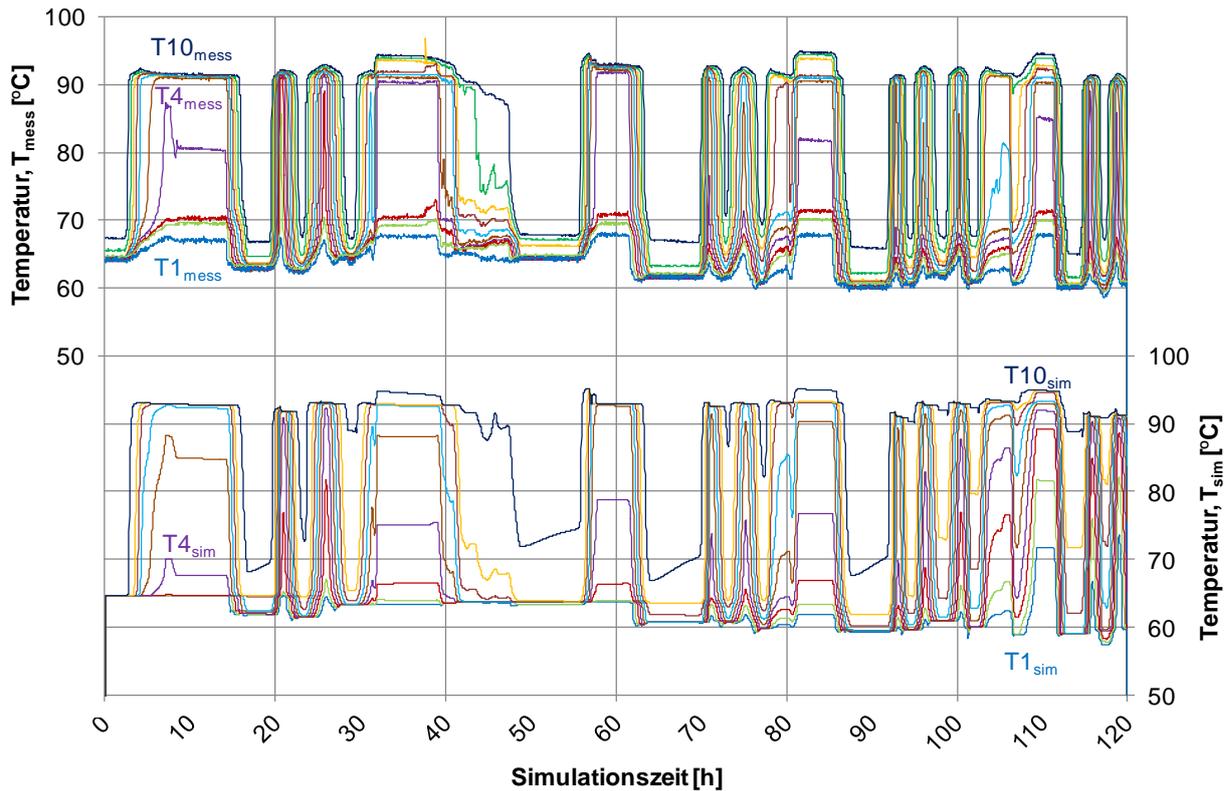


Abbildung 3-31: Temperaturverlauf – Sensoren Knittelfeld 105

Aufgrund dieser Abweichungen wurde der berechnete Volumenstrom näher betrachtet. In der Abbildung 3-32 ist der gemessene Volumenstrom (linke y-Achse) und der Volumenstrom der Simulation (rechte y-Achse) dargestellt. Bedingt durch den direkten Anschluss jedes Speichers am Vorlauf der Fernwärme werden bei beiden Speichern (Knittelfeld 105 und Knittelfeld 320) die Be- und Entladung mittels zweier Regelklappen gesteuert.

Für die Simulation wurde der Volumenstrom jedes Speichers mit der Stellung der Regelklappe kombiniert.

Für diese Regelklappe wurde ein Filter (Messunsicherheit) eingesetzt. Dieser Filter berücksichtigte Werte größer gleich 100 % als Regelklappe „offen“ (Wert = 1) und allen anderen Werte mit Regelklappe „zu“ (Wert = 0). Eine Überprüfung in Excel ergab, dass es während der gesamte Simulationszeit (7200 Zeitschritte) in nur 3 Zeitschritten zu einer Überschneidung (beide „offen“) kam. Auch beim Volumenstrom wurde ein Filter angelegt, um ein Rauschen der Messwerte auszuschließen. Dieser Filter beschränkte einen gültigen Volumenstrom auf 1 bis 100 m³/h. Bei den 3 zuvor erwähnten Zeitschritten in denen beide Regelklappen „offen“ waren, erfüllte 1 Zeitschritt (Stunde 23,1 ; 09.07.2012 23:04 ; - 15 m³/h Knittelfeld 105 ; 1,1 m³/h Knittelfeld 320) beide Kriterien. Dieser Wert wurde dann in der Simulation herausgefiltert.

In der Abbildung 3-32 sind im oberen Bereich (linke y-Achse) die gemessenen Volumenströme der Speicher Knittelfeld 105 ($\dot{V}_{\text{bal1,mess}}$) und Knittelfeld 320 ($\dot{V}_{\text{bal2,mess}}$) und im unteren Bereich (rechte y-Achse) die Volumenströme der Simulation (Knittelfeld 105, $\dot{V}_{\text{bal1,sim}}$

und Knittelfeld 320, $\dot{V}_{\text{bal2,sim}}$) abgebildet. Die Volumenströme der Simulation ergaben sich aus der Stellung der Regelklappen und den gemessenen Volumenströmen unter Berücksichtigung des Filters. In der Abbildung ist an den gemessenen Volumenströmen deutlich zu erkennen, dass Überschneidungen (z.B. blaue Kreise in Abbildung 3-32) auftraten.

Da aber über die Messungen des Volumenstroms keine Daten bezüglich der Messpositionen und der hydraulischen Verschaltungen (Rohrlängen) bekannt sind, kann hier keine genauere Analyse erfolgen.

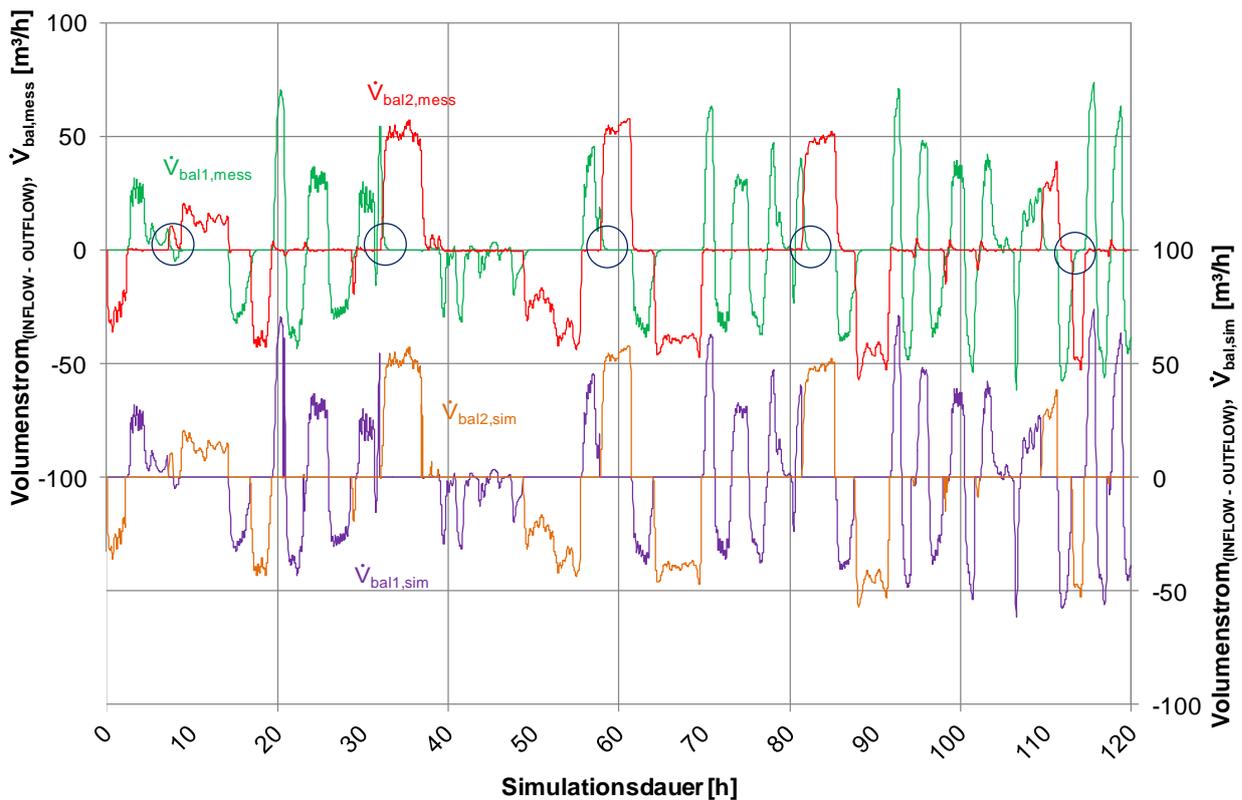


Abbildung 3-32: Gemessener Volumenstrom vs. Volumenstrom Simulation (beide Speicher)

Die eintretenden Temperaturen, die gemessene Außentemperatur, sowie der Volumenstrom während der Be- und Entladung ist in Abbildung 3-33 abgebildet. Bedingt durch die Modellierung einer gemeinsamen Versorgungsleitung durch den Type604a stellt die Vorlauftemperatur $T_{\text{VL,mess}}$ eine Mischtemperatur aus beiden Speichern (Knittelfeld 105 und Knittelfeld 320) dar. Die gemessene Vorlauftemperatur $T_{\text{VL,mess}}$ konnte von der Simulation gut wiedergegeben werden. Die Abweichungen welche sich zur Stunde 30, 75, 95 können auf die geringe gemessene Temperatur ($T_{10\text{mess}}$ in Abbildung 3-31) zurückgeführt werden. Hier weist der simulierte Speicher ($T_{10\text{sim}}$ in Abbildung 3-31) ein höheres Temperaturniveau auf. Der Temperaturabfall in der Rohrleitung des Vorlaufs konnte am Messpunkt nachgebildet werden (Stunde 3, 18, 31, 65).

Die gemessene Temperatur des Rücklaufs am Messpunkt konnte bei einem vorhandenen Volumenstrom gut nachgebildet werden. Die Temperaturabfälle in der Ruhephasen am Messpunkt zeigten in der Messung ($T_{RL1,mess}$) Sprünge auf in Zeiten an denen kein Volumenstrom vorlag. Dieser Effekt trat während der Simulation (Stunde 10, 35, 69) und kann nicht nur auf zu gering angenommenen Verlusten beruhen. Dies könnte auf die Messpositionen zurückzuführen sein, welche nicht bekannt war.

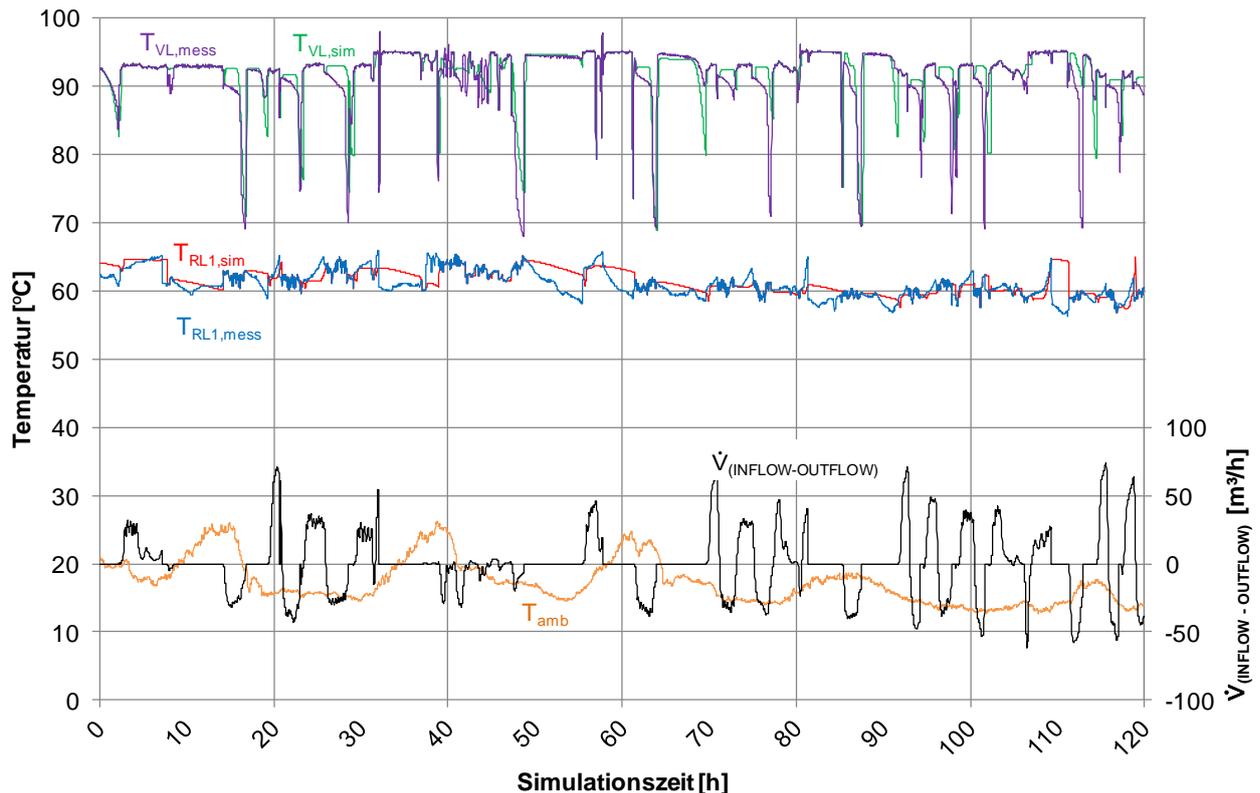


Abbildung 3-33: Temperaturverlauf beim Beladen und Entladen

Die Leistungsbilanz (Abbildung 3-34), zeigte eine geringe Abweichungen im Falle der Entladung (\dot{Q}_{bal} negativ) auf. Da die Leistung im Falle der Entladung aus der eintretenden Temperatur in den Speicher und der austretenden Temperatur aus dem Speicher berechnet wurde, führen höhere Temperaturen im Speicher hier zu einer Abweichung.

Dies konnte auch an den kumulativen Energiebedarf festgestellt werden. Während der kumulative Energiebedarf beim Beladen ($E_{inflow,sim}$) nur geringe Abweichung zur Messung ($E_{inflow,mess}$) aufwies, zeigte der kumulative Energiebedarf der Simulation beim Entladen ($E_{outflow,sim}$) eine große Abweichung zur Messung ($E_{outflow,mess}$).

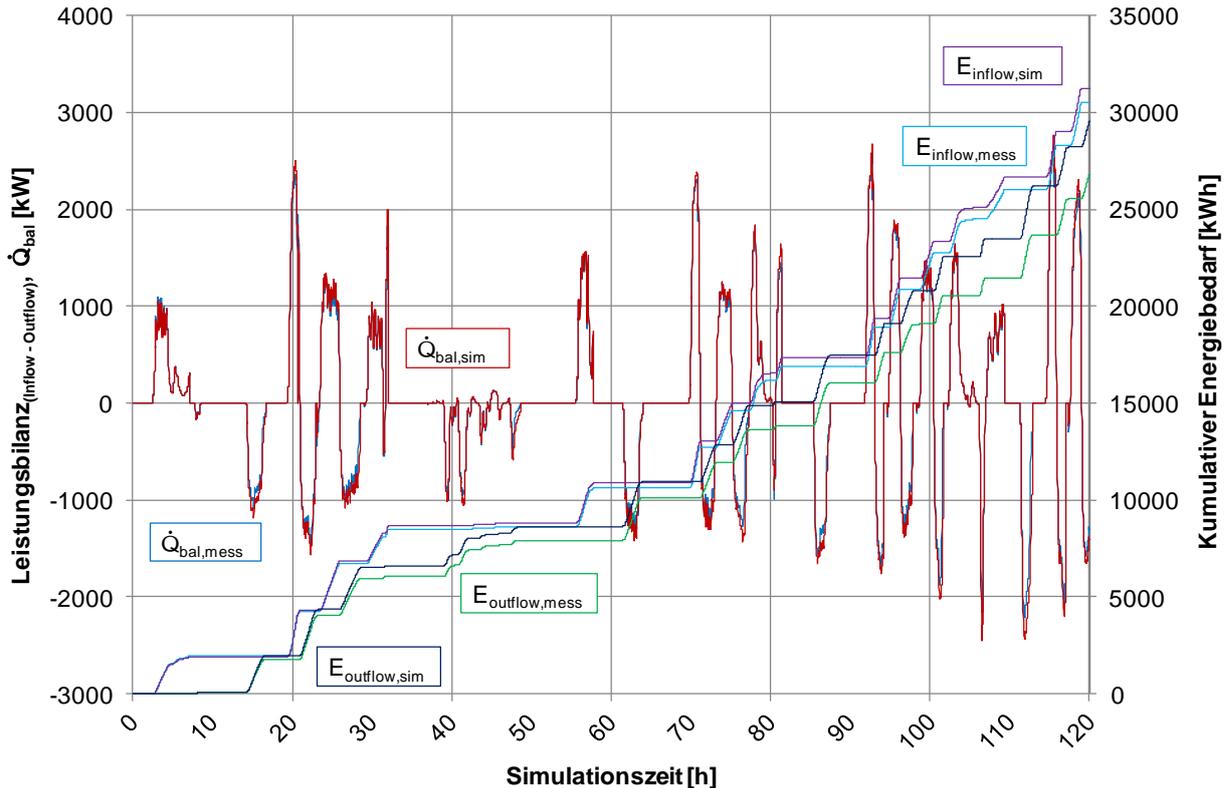


Abbildung 3-34: Leistung und kumulativer Energiebedarf

d) Fazit

Die Annahme einer Initialisierungstemperatur stellte kein Problem in der Simulation dar, was auch Simulationen mit höheren Initialisierungstemperaturen zeigten. Die Temperaturen in den Rohrleitungen des Vorlauf und des Rücklaufs konnten in den Betriebsphasen des Speichers gut wiedergegeben werden. Unklar ist warum die Messung Sprünge aufwies wenn keine Be- und Entladung stattfand.

Zu klären wäre ob der gleichzeitig gemessene Volumenstrom ($\dot{V}_{bal1,mess}, \dot{V}_{bal2,mess}$), welcher in der Simulation durch den Filter wegfällt, wirklich bei den Speichern auftritt, oder ob es sich um eine gegenseitige Beeinflussung der Messung handelt.

Der kumulative Energiebedarf der Beladung am Ende der Simulation wich um 2,5 % von der Messung ab. Bei der Entladung ergab sich am Ende der Simulation eine Abweichung von 10,2 %, was durch das höhere Temperaturniveau des obersten Speicherfühlers ($T_{10,sim}$) begründet ist.

3.7 Knittelfeld (320 m³) – Energie Steiermark

Der Speicher Knittelfeld 320 (Abbildung 3-35) war der zweite Speicher im Motorheizkraftwerk (MHKW) Knittelfeld welcher betrachtet wurde. Die Speicher unterscheiden sich in der Kubatur, der Dämmstärke, der Anzahl der Temperaturfühler und der Gestaltung der Anschlüsse.

Da beide Speicher gemeinsam abgebildet wurden, gelten abgesehen der konstanten Initialisierungstemperatur des Speichers Knittelfeld 105, alle Annahmen und Vereinfachungen (a), sowie die Angaben zur Modellierung (b) des Kapitels 3.6.

Eine Übersicht über die wichtigsten Daten gibt die Tabelle 3-7.

Tabelle 3-7: Eckdaten – Knittelfeld 320

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/1	min
Simulationsdauer	120	h
Anzahl der Messpunkte im Speicher	20	-
Füllvolumen des Speichers	315	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	1,2094	W/mK
Dämmstärke – Mantel	25	cm
Dämmstärke – Deckel	17,2	cm
Dämmstärke – Boden	17,2	cm

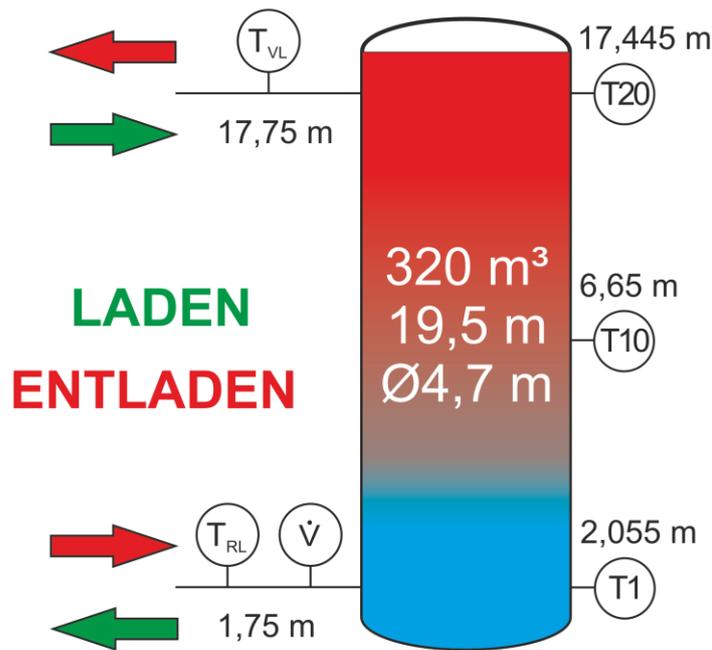


Abbildung 3-35: Schema Fernwärmespeicher MHKW Knittelfeld 320

c) Vergleich Messung – Simulation

In der folgenden Abbildung ist der Verlauf der gemessenen Temperaturen (linke y-Achse) und der simulierten Temperaturen (rechte y-Achse) der Temperaturfühler des Speichers dargestellt. Die gemessenen Temperaturen im Speicher konnten hinsichtlich der von der Änderung der Temperatur betroffenen Knoten (Schichtung) gut wiedergegeben werden. Jedoch zeigen die Messwerte eine höhere Temperatur im oberen Speicherbereich ($T_{20_{mess}}$), als die Simulation ($T_{20_{sim}}$).

Auch waren Temperatursprünge in den Messdaten zu erkennen (Stunde 19 bis 30), in der für den Speicher Knittelfeld kein Volumenstrom (laut dem Regelventil) vorlag. Der Temperaturabfall zur Stunde 40 bis 49 der Temperaturfühler 3 (grün), 4 (rot) und 5 (orange) ist in der Simulation nicht nachvollziehbar. Dieser Abfall konnte auch an den obersten Temperaturfühlern zur Stunde 70 bis 81 beobachtet werden.

Der Verlauf des untersten Messfühlers ($T_{1, \text{mess}}$) konnte in der Simulation ($T_{1, \text{sim}}$) gut wiedergegeben werden.

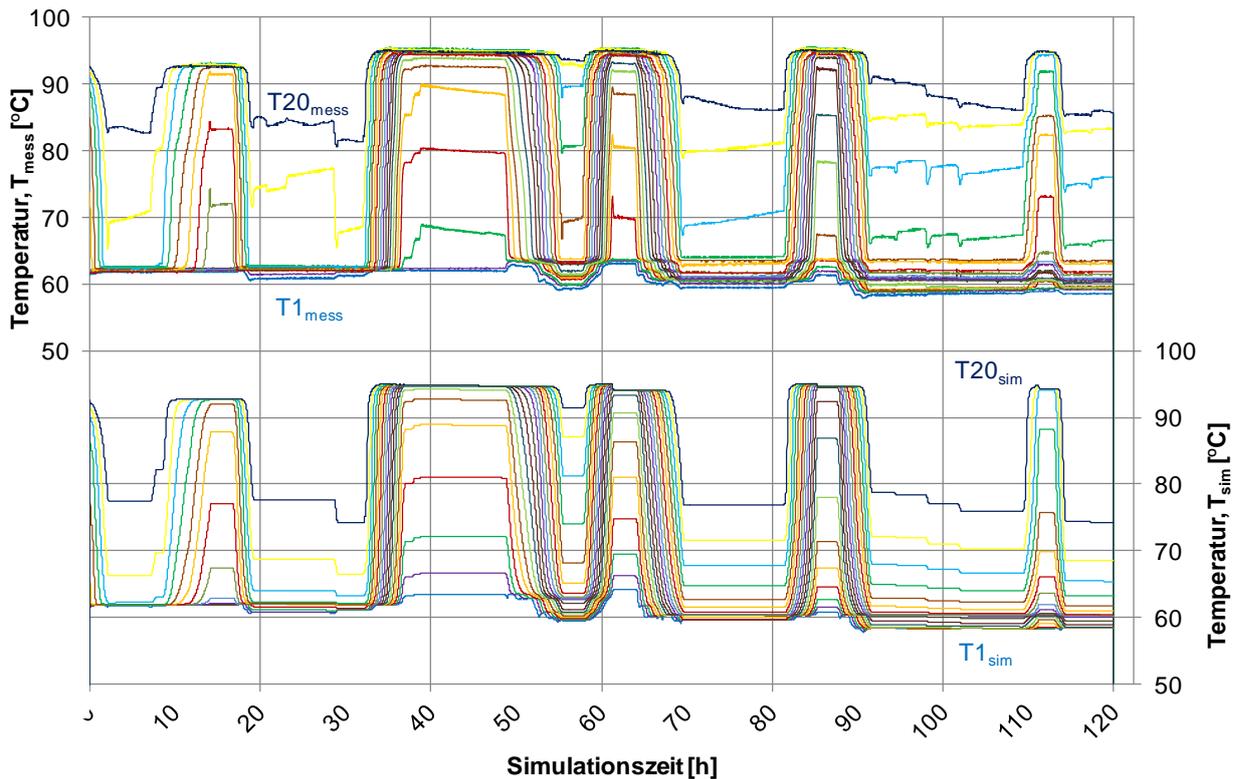


Abbildung 3-36: Temperaturverlauf – Sensoren

Wie schon zuvor in Kapitel 3.6 beschrieben ist die abgebildete Vorlauf­temperatur in Abbildung 3-37 eine Mischtemperatur aus den oberen Anschlüssen der beiden Speicher Knittelfeld 105 und Knittelfeld 320. Der Verlauf der Vorlauf­temperatur ($T_{VL, \text{mess}}$) im Falle eines vorhandenen Volumenstroms, konnte durch die Simulation ($T_{VL, \text{sim}}$) gut wiedergegeben werden. Die Rücklauf­temperatur ($T_{RL2, \text{mess}}$) wies bei jeder Entladung (negativer Volumenstrom in Abbildung 3-44 unten) starke, kurzzeitige Schwankungen im Temperaturverlauf auf, die aufgrund der unzureichenden Datenlage nicht erklärbar sind. Jedoch wurde die Tendenz von der Simulation ($T_{RL2, \text{sim}}$) gut nachgebildet.

3 Simulationsaufbau und Ergebnisse

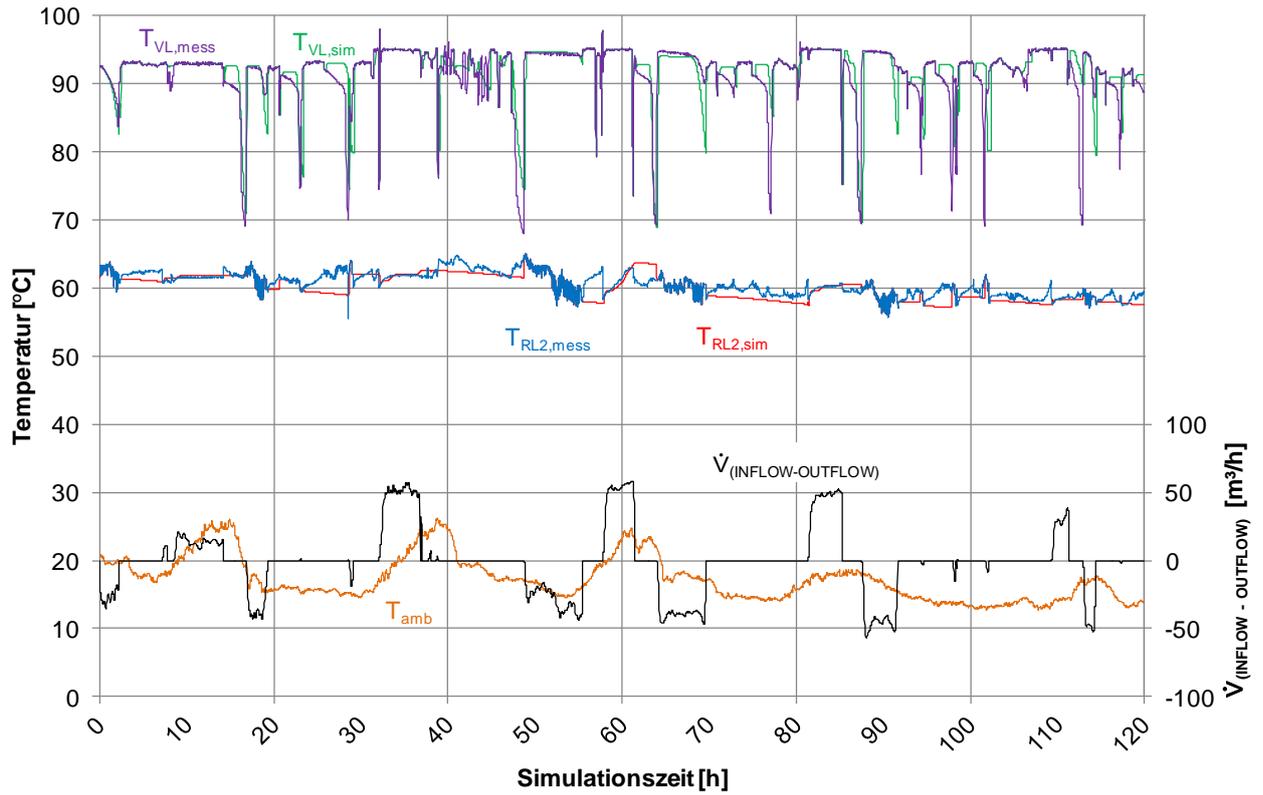


Abbildung 3-37: Temperaturverlauf beim Beladen und Entladen

In Abbildung 3-38 ist die Leistungsbilanz des Speichers Knittelfeld 320 abgebildet.

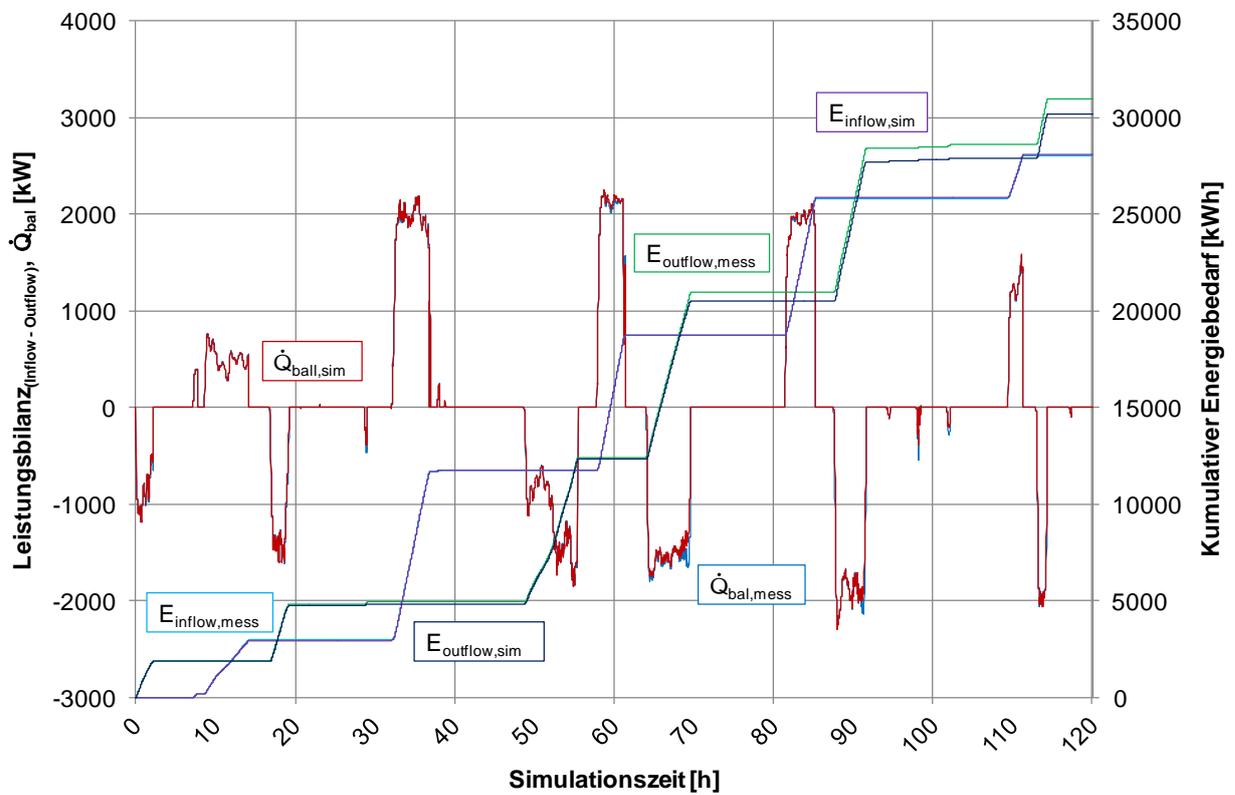


Abbildung 3-38: Leistung und kumulativer Energiebedarf

Die Abweichung fiel bedingt durch die geringe Temperaturdifferenz zwischen der Messung und der Simulation sehr gering aus. Nur die Leistungsbilanz während der Entladung wies zur Stunde 70 und 92 eine merkliche Abweichung aus.

Der kumulative Energiebedarf am Ende der Beladung der Simulation ($E_{\text{inflow,sim}}$) fiel nur unmerklich höher aus als die Messung ($E_{\text{inflow,mess}}$). Die geringe Abweichung des kumulativen Energiebedarfs beim Entladen war hauptsächlich durch die Abweichungen der gemessenen Temperatur der Rohrleitung am Messpunkt (in Abbildung 3-44 zur Stunde 30, 70, 92 und 105) bedingt.

d) Fazit

Der Speicher Knittelfeld konnte in der Simulation gut wiedergegeben werden, was sich auch am geringen Unterschied von Messung und Simulation zeigt. Der kumulative Energiebedarf der Beladung war um 0,2 % größer als der kumulative Energiebedarf der Messung. Beim kumulativen Energiebedarf der Entladung zeigte sich eine Abweichung von -2,5 %.

Dies stellt in Hinblick auf die Abschätzung aller Rohrleitungen und Verluste ein gutes Ergebnis dar.

Da keine Daten über die Verrohrung bekannt waren, können keine genaueren Aussagen über die Abweichungen der beiden Speicher des MHKW Knittelfeld getroffen werden.

3.8 Gleinstätten (44 m³) – Nahwärme

Der Speicher Gleinstätten, besteht, entgegen der Darstellung in Abbildung 3-39, aus zwei Speichern mit einem Füllvolumen von jeweils 44,15 m³. Sie sind Teil des Heizwerkes Gleinstätten welches die Fernwärmeversorgung von 80 Kunden (darunter private, kommunale und gewerbliche Objekte) sicherstellt. Die Wärmeversorgung übernimmt ein Biomassekessel mit einer Anschlussleistung von 2500 kW, an dem eine Rauchgaskondensation (300 kW) angeschlossen ist. Diese dient im Heizwerk zur Rücklaufanhebung des Fernwärmenetzes, sowie zur Hackgutrocknung. Auf dem Dach des Heizwerkes ist eine Solaranlage mit einer Fläche von 1315 m² montiert, welche auch die Grundlastabdeckung im Sommer übernimmt. Überschüssige Wärme kann im Heizwerk zur Hackgutrocknung verwendet werden (<http://www.nahwaerme-gleinstaetten.at>, 30.10.2012).

Die beiden Speicher befinden sich auf der nordwestlichen Ecke des Heizwerkes und sind durch Paneelwände eingehaust. Die beiden Speicher sind miteinander verbunden (Verbindung in 1 m, 4,5 m und 7,5 m Höhe) und der Raum dazwischen wurde mittels Steinwolle aufgefüllt. Der Außentemperaturfühler ist an der Außenseite des Heizwerkes angebracht (Orientierung nach Nordnordwest).

Die Beladung erfolgt, je nach Betriebsweise (Sommer, Winter) in entsprechenden Höhen (siehe Abbildung 3-39). Die Solaranlage und der Rücklauf des Netzes sind am ersten der beiden Speicher angeschlossen. Der Biomassekessel und die fünf Temperatursensoren (T1 - T5) sind am zweiten Speicher angeschlossen. Die Fernwärme wird aus beiden Speichern (am obersten Punkt) entnommen.

Die wichtigsten Daten der Simulation wurden in Tabelle 3-8 zusammengefasst.

Tabelle 3-8: Eckdaten – Speicher Gleinstätten

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	2,5/5	min
Simulationsdauer	744	h
Anzahl der Messpunkte im Speicher	5	-
Füllvolumen des Speichers	88,3	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	0,9587	W/mK
Dämmstärke – Mantel	28	cm
Dämmstärke – Deckel	34,1	cm
Dämmstärke – Boden	20,4	cm

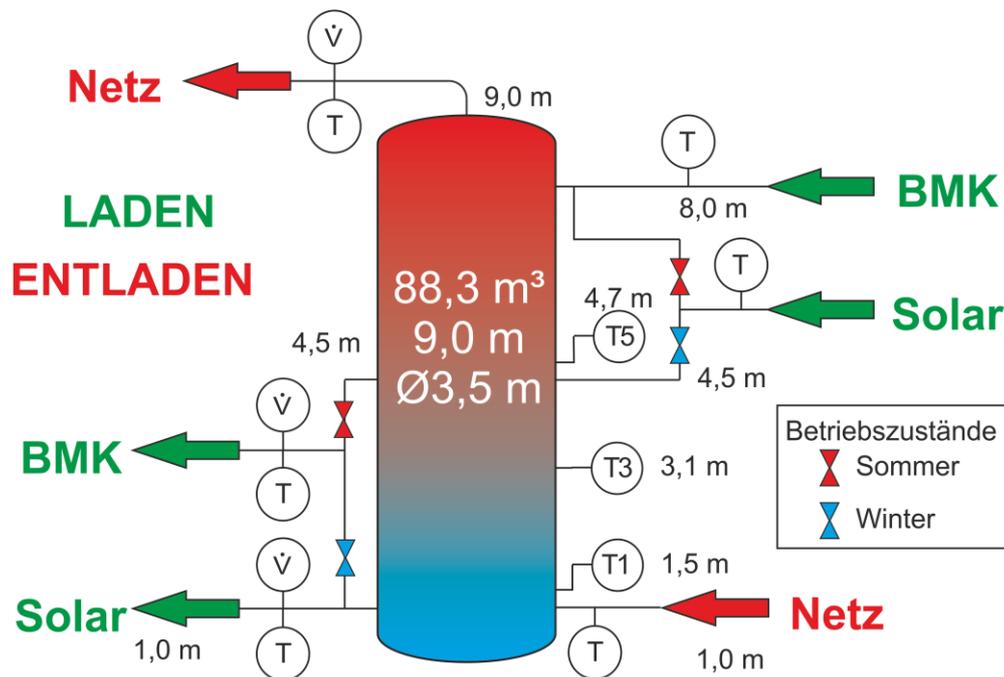


Abbildung 3-39: Schema Gleinstätten (Sommer, Winter)

a) Annahmen und Vereinfachungen

Wie bereits erwähnt wurde, besteht das Heizwerk aus zwei Speichern welche Verbindungen aufweisen. Zurzeit gibt es kein Modell in TRNSYS das die Ausgleichsvorgänge, welche sich durch die Temperaturunterschiede bei der Beladung einstellen werden, berücksichtigen

kann. Somit wurden die beiden Speicher zusammengefasst zu einem Speicher mit derselben Höhe. Die damit verbundene Reduzierung der Oberfläche wurde durch eine Korrektur der Dämmstärke berücksichtigt. Ebenso wie die Verbindungen, welche als zusätzliche Anschlussverluste modelliert wurden.

Die Rauchgaskondensation, welche einen Wärmeeintrag für das System darstellt, wurde durch die Einbindung mit der jeweilig gemessenen Temperatur in den Rücklauf des Netzes durchgeführt.

Eine weitere Annahme betrifft die Außentemperatur. Wie bereits beschrieben wurde, ist die Außentemperatur an der Außenseite des Heizwerks gemessen worden. Die Rohrleitungen welche die Verbindungen zwischen den einzelnen Bestandteilen des Heizwerks darstellen (z.B. Vorlauf Biomassekessel), befinden sich im Heizwerk (abgeschlossenes Gebäude). Somit konnte die gemessene Außentemperatur (T_{amb}) nicht zur Berechnung der Wärmeverluste dieser Rohrleitungen herangezogen werden.

In Vorsimulationen war erkennbar, dass die Solaranlage im Jänner längere Stillstandszeiten aufwies. Die gemessenen Temperaturen der Rohrleitungen der Solaranlage, im speziellen die des Rücklaufs ($T_{RL,Solar,mess}$), zeigten eine ausgeprägte Abhängigkeit von der Außentemperatur (T_{amb}) auf. Die Abweichung zwischen der gemessene Rücklauftemperatur ($T_{RL,Solar,mess}$) und der gemessenen Außentemperatur (T_{amb}) war innerhalb der Stillstandszeiten der Solaranlage nahezu konstant (unter Vernachlässigung des Zeitverzugs). Somit wurde aus der Differenz von $T_{RL,Solar,mess}$ und T_{amb} eine mittlerer Temperaturdifferenz (ΔT) berechnet. Die Außentemperatur erhöht um diese Temperaturdifferenz (ΔT) wurde als Bezugstemperatur (Gebäudetemperatur) für die Verlustberechnung (Temperatur im Heizwerkgebäude) herangezogen.

b) Modellierung in TRNSYS

Der Aufbau der Simulation ist in Abbildung 3-40 dargestellt. Die Vorgabe der Eintrittstemperaturen in den Speicher (Vorlauf Biomassekessel, Rücklauf Netz, Vorlauf Kondensation, Vorlauf Solaranlage) erfolgt mittels Type9c (Datareader). Die Längen der Rohrleitungen wurden anhand von Fotos des Heizwerks abgeschätzt. Das verwendete Rohrleitungsmodell war der Type709, da immer von einer Strömungsrichtung auszugehen war. Der Massenstrom der Rauchgaskondensation, welche in den Rücklauf des Netzes eingebunden ist, kann maximal die Größe des Rücklaufs des Netzes betragen. Dies wurde durch den Einsatz von Equations berücksichtigt, welche den Anteil der Aufteilung des Massenstroms (Netz oder Kondensation) für den Massenstromteiler (Type647) berechnete. Der Rohrleitung „Kond_VL“ wurde die Temperatur der Rauchgaskondensation sowie der Massenstrom des Massenstromteilers aufgeprägt. Nach den Rohrleitungen treten der Restmassenstrom des Rücklaufs Netz und der Massestrom der Kondensation gemeinsam

durch die Rohrleitung „Netz_RL_Mix“ in den Speicher ein. Die Leistungen und Energien wurden durch den Type25c (Printer) ausgegeben.

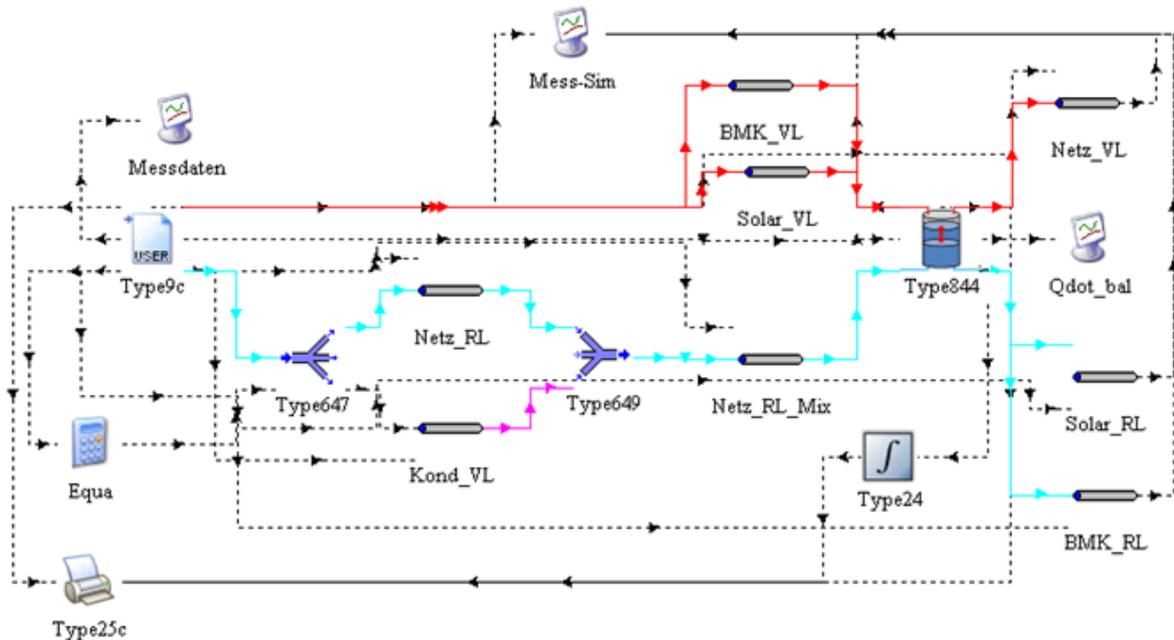


Abbildung 3-40: Simulationsaufbau Speicher Gleinstätten

c) Vergleich Messung – Simulation

In der folgenden Abbildung 3-41 ist im oberen Bereich der Temperaturverlauf der gemessenen Temperaturen (linke y-Achse) und im unteren Bereich der Temperaturverlauf der Simulation (rechte y-Achse) dargestellt. Wie sich schon bei einer ersten Sichtung der Messdaten in EXCEL zeigte, wiesen die gemessenen Daten fehlende Zeitschritte auf. Diese fehlenden Zeitschritte wurden durch ein EXCEL-Makro eingefügt, wobei der Wert zu Null gesetzt wurde. Somit konnten fehlende Messwerte die Simulation nicht beeinflussen. Zur besseren Darstellung im Diagramm wurden die fehlenden Zeitschritte der Temperatur durch die Zahl 100 ersetzt (erkennbar in Abbildung 3-41 durch vertikale Linien, z.B. Stunde 80 bis 89).

Erkennbar in Abbildung 3-41 ist, dass der oberste Temperaturfühler ($T_{5_{\text{mess}}}$), welcher sich auf einer Höhe von 4,5 m (im mittleren Bereich des Speichers) befindet eine um ca. 40 K höhere Temperatur aufweist als die Simulation ($T_{5_{\text{sim}}}$).

Durch die Zusammenfassung der beiden Speicher (gleiches Füllvolumen) konnte die reale Temperaturerhöhung durch die Beladung (nur eines Speichers) nicht abgebildet werden. Der simulierte Speicher wies infolge der Beladung auch eine Temperaturerhöhung auf, jedoch erreichte diese Temperaturerhöhung im Speicher bedingt durch das größere beteiligte Volumen nicht den mittleren Speicherbereich ($T_{5_{\text{mess}}}$, 4,7 m).

Da keine Temperaturen im oberen Bereich des Speichers gemessen wurden, ist ein Vergleich der Messung mit der Simulation nur im unteren Bereich des Speichers möglich. Der unterste Messfühler ($T_{1_{\text{mess}}}$) befindet sich auf einer Höhe von 1,5 m. Die Tendenz der Messung kann von der Simulation ($T_{1_{\text{sim}}}$) wiedergegeben werden, jedoch nicht die Höhe der Änderung.

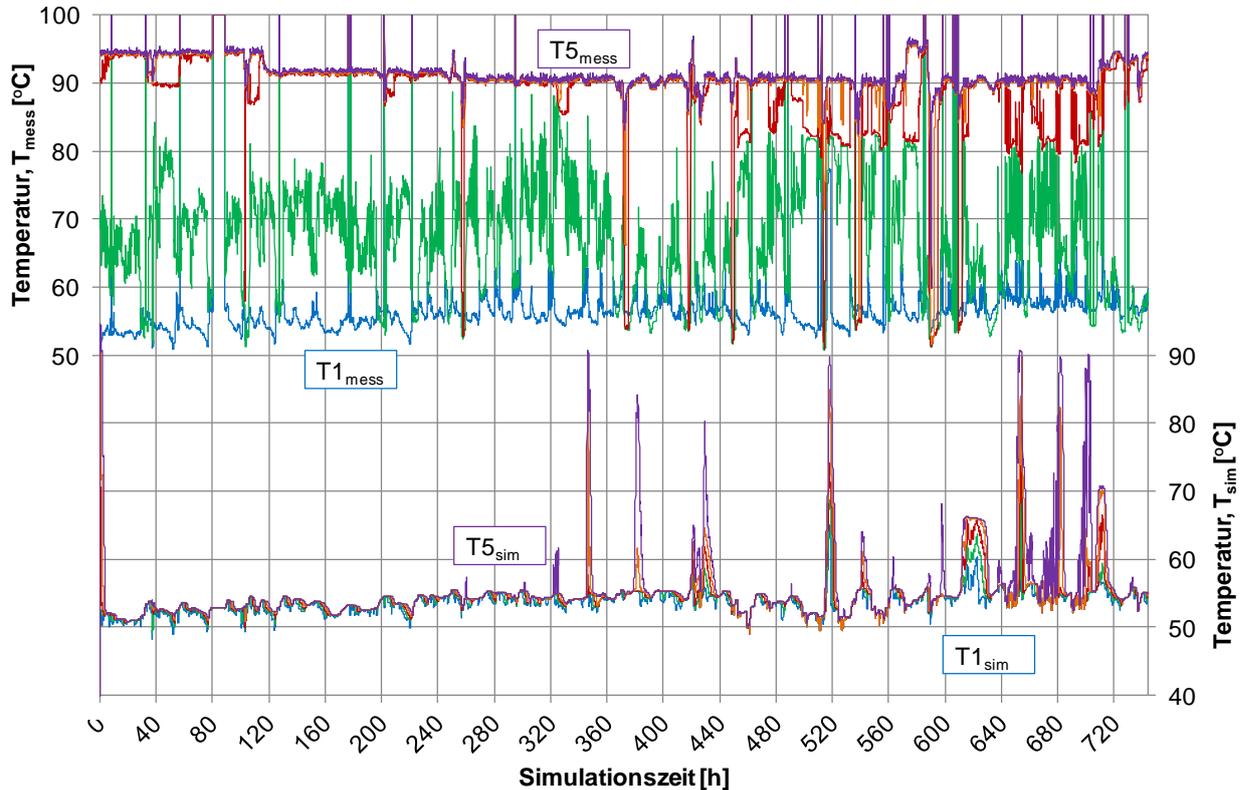


Abbildung 3-41: Temperaturverlauf – Sensoren Gleinstätten Jänner

Die Abbildung 3-42 zeigt die Verläufe der einzelnen Rohrtemperaturen. Die gemessene Vorlauftemperatur des Netzes ($T_{VL,Netz,mess}$) war kleiner als die simulierte ($T_{VL,Netz,sim}$). Ein Grund hierfür könnte sein, dass sich in der Realität eine Mischtemperatur aus beiden Speichern bildet. Die gemessene Rücklauftemperatur der Biomasse ($T_{RL,BMK,mess}$) stimmte mit der Temperatur der Simulation ($T_{RL,BMK,sim}$) gut überein. Am Beispiel der gemessenen Temperatur des Rücklaufs der Solaranlage ($T_{RL,Solar,mess}$) ist erkennbar, dass die Außentemperatur zwar einen Einfluss auf die Rohrleitung hat, die Temperatur in der Rohrleitung selbst aber nicht auf das Niveau der Außentemperatur absinkt. Dass die Annahme einer Gebäudetemperatur mit der Realität übereinstimmte zeigte die geringe Abweichung von ($T_{RL,Solar,sim}$). Am Verlauf der Vorlauftemperatur der Solaranlage ($T_{VL,Solar,mess}$) sind ab der Stunde 440 starke, kurzzeitige Temperaturschwankungen erkennbar. Obwohl hier meist eine negative Temperaturspreizung vorliegt, scheint die Anlage stark zu takten (häufiges Ein- und Ausschalten).

3 Simulationsaufbau und Ergebnisse

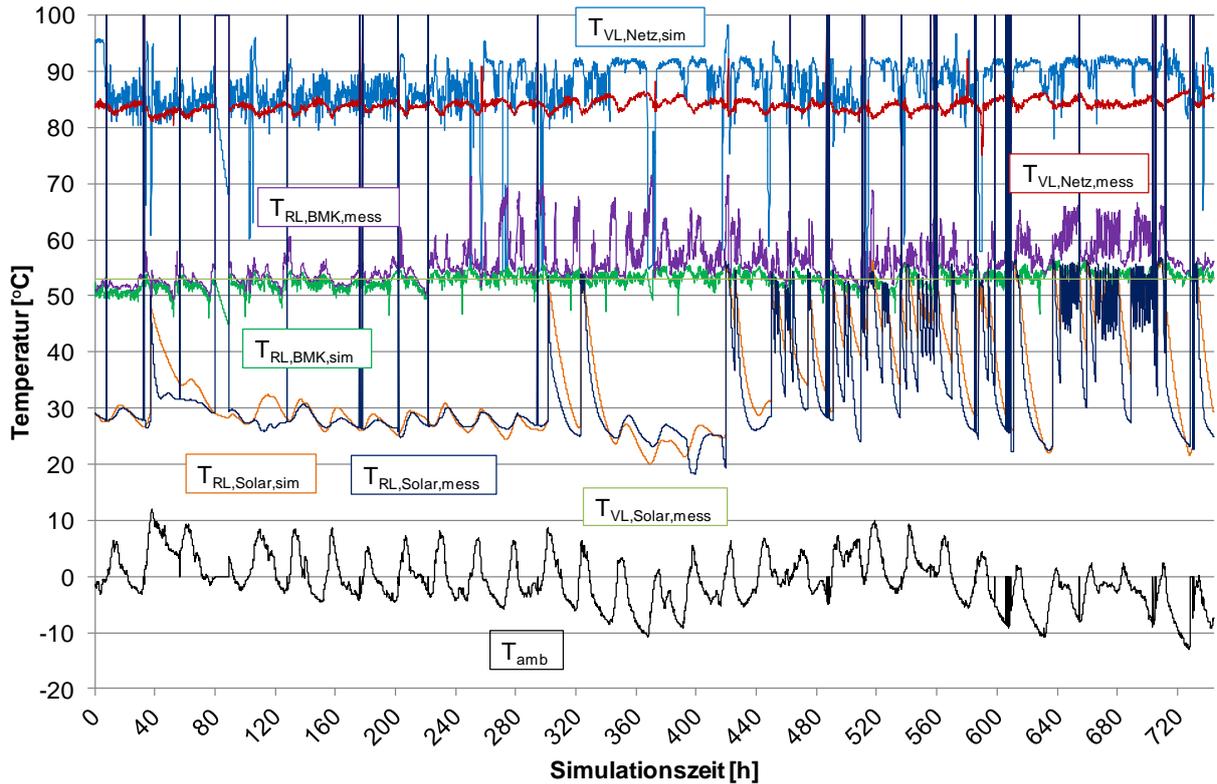


Abbildung 3-42: Temperaturverlauf der Eintrittstemperaturen

Bedingt durch die Abweichungen der Temperaturen im simulierten Speicher, wies die Leistungsbilanz eine Abweichung zwischen Messung ($\dot{Q}_{bal,mess}$) und Simulation ($\dot{Q}_{bal,sim}$) auf. Dies setzte sich auch beim kumulativen Energiebedarf fort (siehe Abbildung 3-43).

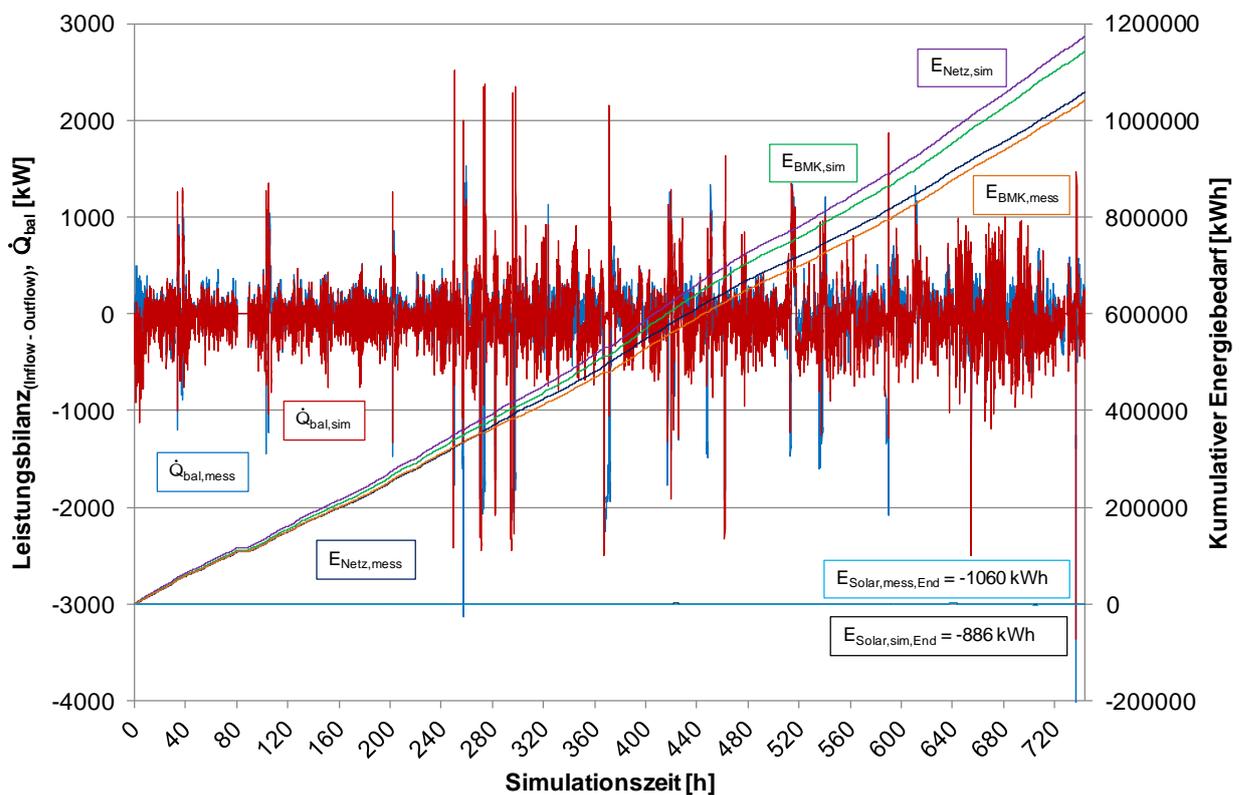


Abbildung 3-43: Leistung und kumulativer Energiebedarf Jänner

Zur besseren Darstellung des kumulativen Energiebedarfs der Solaranlage wurde Abbildung 3-43 skaliert (Abbildung 3-44). Es ist erkennbar, dass der kumulative Energiebedarf zur Stunde 280 zunimmt. Jedoch im Bereich ab Stunde 440 kommt es zu häufigen Einschaltvorgängen an denen die Vorlauftemperatur schlagartig absinkt, was sich an einen negativen kumulativen Energiebedarf äußert (siehe dazu auch Abbildung 3-42, ab Stunde 440).

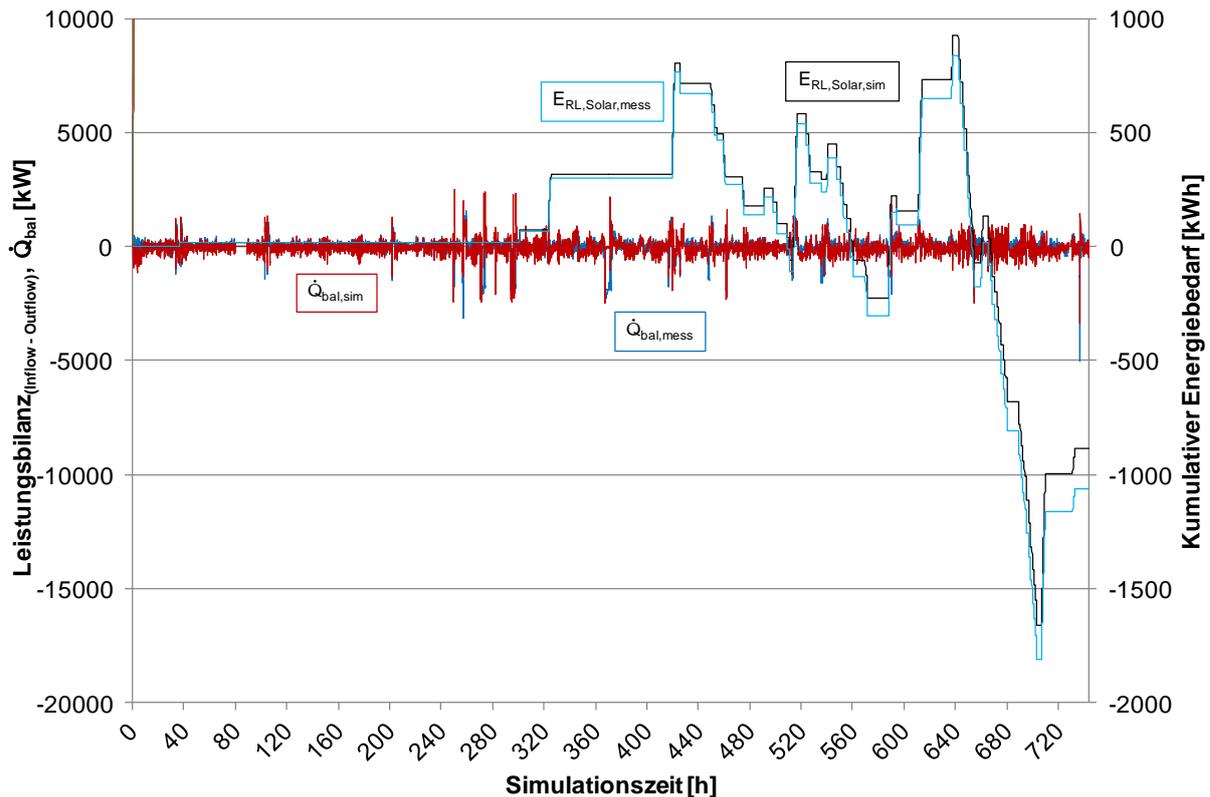


Abbildung 3-44: Leistung und kumulativer Energiebedarf Jänner (skaliert)

d) Fazit

Für die Abbildung des Speichers Gleinstätten mussten eine Vielzahl von Annahmen getroffen werden. Jedoch zeigte sich, dass auch Speicher welche eine Verbindung miteinander aufweisen abgebildet werden können. Da die Simulationszeit hier (744 h = 1 Monat) deutlich höher ist, als bei den bisherigen Speichern (Graz 144 h, Knittelfeld 120 h), wurde für die Berechnung der Abweichung des kumulativen Energiebedarfs nur der Bereich bis 144 Stunden herangezogen.

Hier ergab sich für den kumulativen Energiebedarf des Netzes eine Abweichung von 8,3 % gegenüber der Messung, was durch die höhere Austrittstemperatur $T_{VL,Netz,sim}$ bestimmt war. Für den kumulativen Energiebedarf des Biomassekessels ergab sich eine Abweichung von 3,6 % zur Messung. Der kumulative Energiebedarf der Solaranlage wich in diesem Bezugszeitraum um 14,8 % gegenüber der Messung ab.

Erwähnt werden muss, dass die Solaranlage am Simulationende (Stunde 744) einen negativen kumulativen Energiebedarf aufwies. Hier wäre zu klären ob das Einschalten der Solaranlage vom Benutzer initiiert wurde, oder die Regelung Optimierungsbedarf hat.

3.9 Mattsee (40 m³) – Nahwärme

Der Speicher Mattsee (Abbildung 3-45) dient als Fernwärmespeicher im Heizwerk Mattsee. Das Heizwerk Mattsee versorgt 40 Kunden (darunter private, kommunale und gewerbliche Objekte) mit Fernwärme. Die Wärmeversorgung des Heizwerks wird durch einen Biomassekessel (BMK) mit einer Anschlussleistung von 1500 kW sichergestellt. Zusätzlich ist am Dach des Heizwerkes eine Solaranlage mit einer Kollektorfläche von 260 m² montiert (<http://www.nahwärme.net>, 31.10.2012).

Der Speicher Mattsee (40 m³ Füllvolumen) ist im Freien positioniert und der Bereich zwischen dem Behälter und der Verkleidung ist mit Zellulose ausgefüllt. Der Fernwärmespeicher verfügt über mehrere Anschlüsse zum Be- und Entladen. Die Beladung durch die Solaranlage erfolgt je nach Betriebsweise (Sommer über rotes Ventil, 8,7 m; Winter über blaues Ventil, 4,1 m) in den entsprechenden Höhen (Abbildung 3-45). Der Rücklauf des BMK wird je nach Betriebsweise (Sommer über rotes Ventil, 4,1 m; Winter über blaues Ventil, 0,5 m) aus unterschiedlichen Höhen entnommen.

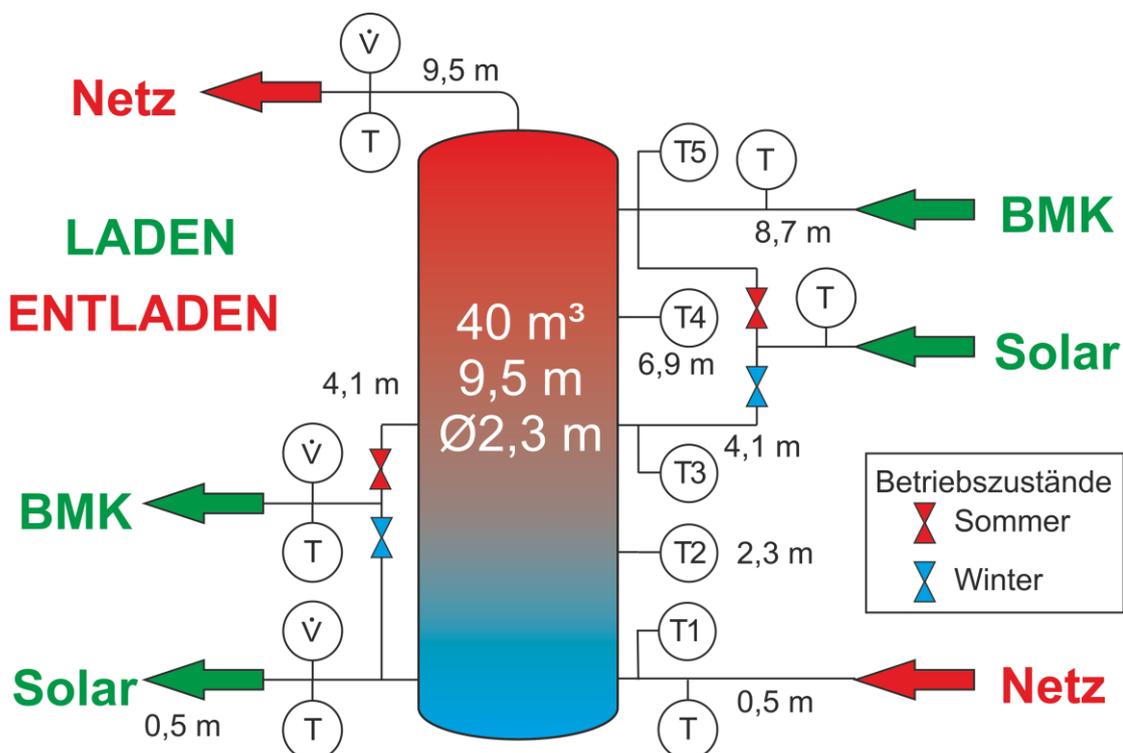


Abbildung 3-45: Schema Mattsee

Tabelle 3-9: Eckdaten – Speicher Mattsee

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	1/2	min
Simulationsdauer	610,3	h
Anzahl der Messpunkte im Speicher	5	-
Füllvolumen des Speichers	39	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	1,4446	W/mK
Dämmstärke – Mantel	50	cm
Dämmstärke – Deckel	69,8	cm
Dämmstärke – Boden	53,3	cm

a) Annahmen und Vereinfachungen

Für den Speicher Mattsee war keine Außenlufttemperatur (September) bekannt und musste somit ermittelt werden. Die Rohrleitungen sind im Heizwerk Mattsee (abgeschlossenes Gebäude) positioniert. Dadurch war für die Berechnung der Wärmeverluste der Rohrleitungen eine Bezugstemperatur zu finden. Im Gegensatz zum Speicher Gleinstätten war die Solaranlage, bis auf die Stunde 310 bis 355, kontinuierlich in Betrieb. In diesem Zeitraum fällt die Temperatur dieser Rohrleitung auf 32,5 °C ab. Aufgrund des relativ kurzen Zeitraumes, der zur Betrachtung herangezogen werden konnte, wurde für die Simulation von einer konstanten Umgebungstemperatur von 30 °C, sowohl für das Gebäude als auch für den Speicher, ausgegangen.

Aus der Beschreibung des Heizwerks Mattsee ging nicht hervor, dass sich im Heizwerk ein Ölbrenner befand. Jedoch wurde ein Massenstrom, sowie eine Vorlauf- und Rücklauftemperatur eines Ölbrenners gemessen. Durch Vorsimulationen zeigte sich, dass dieser Ölbrenner einen Einfluss auf die gemessene Netztemperatur ($T_{VL,Netz,mess}$) hatte (siehe Abbildung 3-48 ab Stunde 330).

Jedoch konnte der Massenstrom des Ölbrenners ($\dot{m}_{Netz,mess}$), welcher im Maximum ein Vielfaches des Massenstrom des Netzes ($\dot{m}_{Netz,mess}$) betrug, nicht durch den Speicher (und somit vom Rücklauf des Netzes) gedeckt werden (siehe Abbildung 3-48 ab Stunde 330). Auch konnte eine Erhöhung der Temperaturen im Speicher ($T_5 - T_2$) während der Betriebszeit des Ölbrenners nicht festgestellt werden (siehe Abbildung 3-47 ab Stunde 330). Der Ölbrenner wurde somit im Simulationsaufbau als Nachheizung des Vorlaufmassenstroms der Fernwärme berücksichtigt.

Auch über die Rauchgaskondensation des BMK waren außer den Messdaten, keine Informationen vorhanden. Die Rauchgaskondensation wurde, wie zuvor im Speicher Gleinstätten, als Rücklaufanhebung berücksichtigt.

b) Modellierung in TRNSYS

Die Eingabegrößen der Simulation werden mittels Type9c (Datareader) den Modellen der Rohrleitungen (Type709) übergeben. Die Kondensation wurde als Rücklaufanhebung der Fernwärme berücksichtigt. Die Aufteilung des Rücklaufs in Restmassenstrom Netz und Vorlauf Kondensation erfolgt mittels des Type647. Die Anteile der Massenströme wurden mittels einer Equation berechnet. Nach dem Massenstromteiler wurde dem Massenstrom der Kondensation die Temperatur der Kondensation aufgeprägt. Nach den Rohrleitungen wurden die beiden Massenströme mittels Mischer (Type649) wieder vermischt und traten durch eine Rohrleitung in den Speicher ein.

Die Berücksichtigung der Nachheizung der Vorlauftemperatur der Fernwärme erfolgt durch eine Equation mit der Bezeichnung „Öl“. Wenn ein gemessener Massenstrom des Ölbrenner vorliegt, wird die Vorlauftemperatur der Fernwärme durch die Vorlauftemperatur des Ölbrenners ersetzt. Für den Speicher Mattsee mussten alle Dämmstärken und Längen der Rohrleitungen abgeschätzt werden, da hier keine Daten vorlagen (siehe Abbildung 3-46).

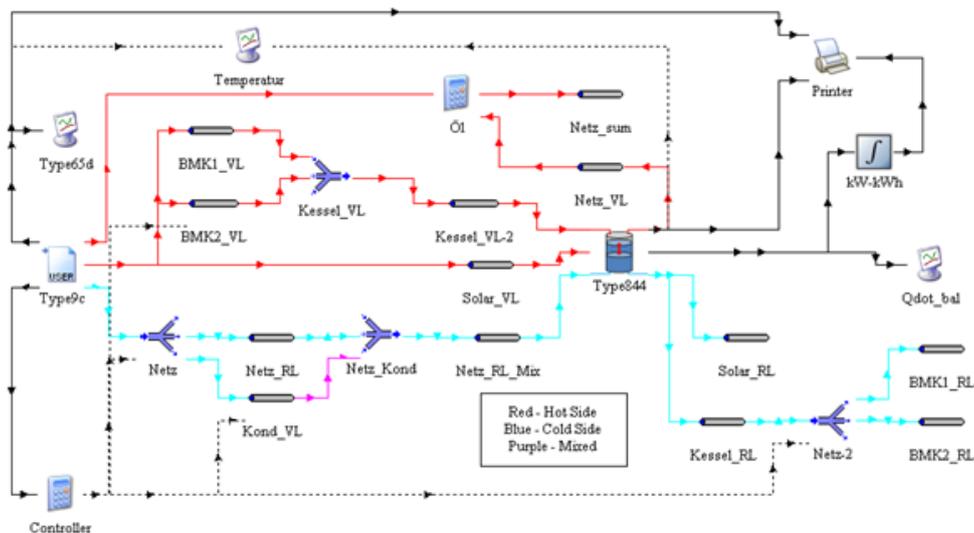


Abbildung 3-46: Simulationsaufbau Speicher Mattsee

c) Vergleich Messung – Simulation

In der Abbildung 3-47 ist im oberen Bereich (linke y-Achse) der gemessene Temperaturverlauf im Speicher Mattsee abgebildet, im unteren Bereich (rechte y-Achse) der Temperaturverlauf der Simulation (rechte y-Achse). Wie schon zuvor beim Speicher Gleinstätten wies auch der Speicher Mattsee fehlende Zeitschritte auf. Die fehlenden Zeitschritte wurden mittels eines EXCEL- Makros eingefügt.

Für die Simulation wurde als Wert für die fehlenden Zeitschritte der Wert 0 gewählt, jedoch zur übersichtlicheren Darstellung im Diagramm (EXCEL) wurde der Wert 0 bei den Temperaturen durch den Wert 100 ersetzt (siehe Abbildung 3-47, senkrechte Linien, z.B. zur Stunde 280 bis 320).

Die Temperatur des obersten Temperatursensors ($T_{8_{\text{mess}}}$) konnte von der Simulation ($T_{8_{\text{sim}}}$) bis auf den Bereich zwischen der Stunde 335 bis 500 wiedergegeben werden. Die Temperatur des untersten Fühlers ($T_{1_{\text{mess}}}$) wies in der Messung erhebliche Schwankungen auf, welche in Simulation ($T_{1_{\text{sim}}}$) nicht auftraten.

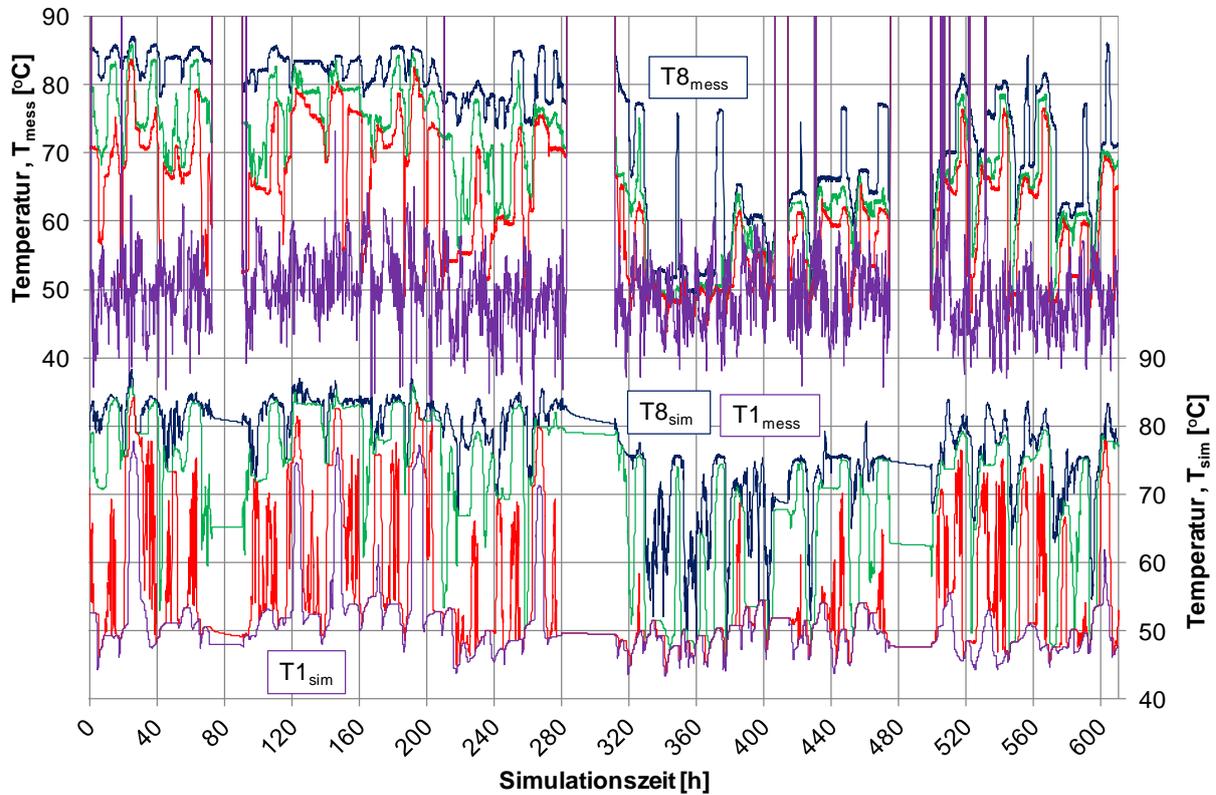


Abbildung 3-47: Temperaturverlauf – Sensoren Mattsee

Wie schon zu Beginn dieses Kapitels erwähnt waren bis auf die Messdaten keine weiteren Informationen über den Ölbrenner verfügbar. In Abbildung 3-48 ist die gemessene Netztemperatur ($T_{\text{VL,Netz,mess}}$) zu erkennen. $T_{\text{VL,Netz,mess}}$ wies bei einem vorhandenen Massenstrom des Ölbrenners ($\dot{m}_{\text{Öl,mess}}$) eine wesentliche Erhöhung der Temperatur im Vergleich zur Simulation ($T_{\text{VL,Netz,sim}}$) auf (in Abbildung 3-48 nur zur Stunde 330 bis 400 erkennbar). Durch die Berücksichtigung des Ölbrenners als Nachheizung der Rücklaufumlauftemperatur konnte die Vorlaufumlauftemperatur der Messung ($T_{\text{VL,Netz,mess}}$) von der Simulation ($T_{\text{VL,NetzÖl,sim}}$) gut wiedergegeben werden.

3 Simulationsaufbau und Ergebnisse

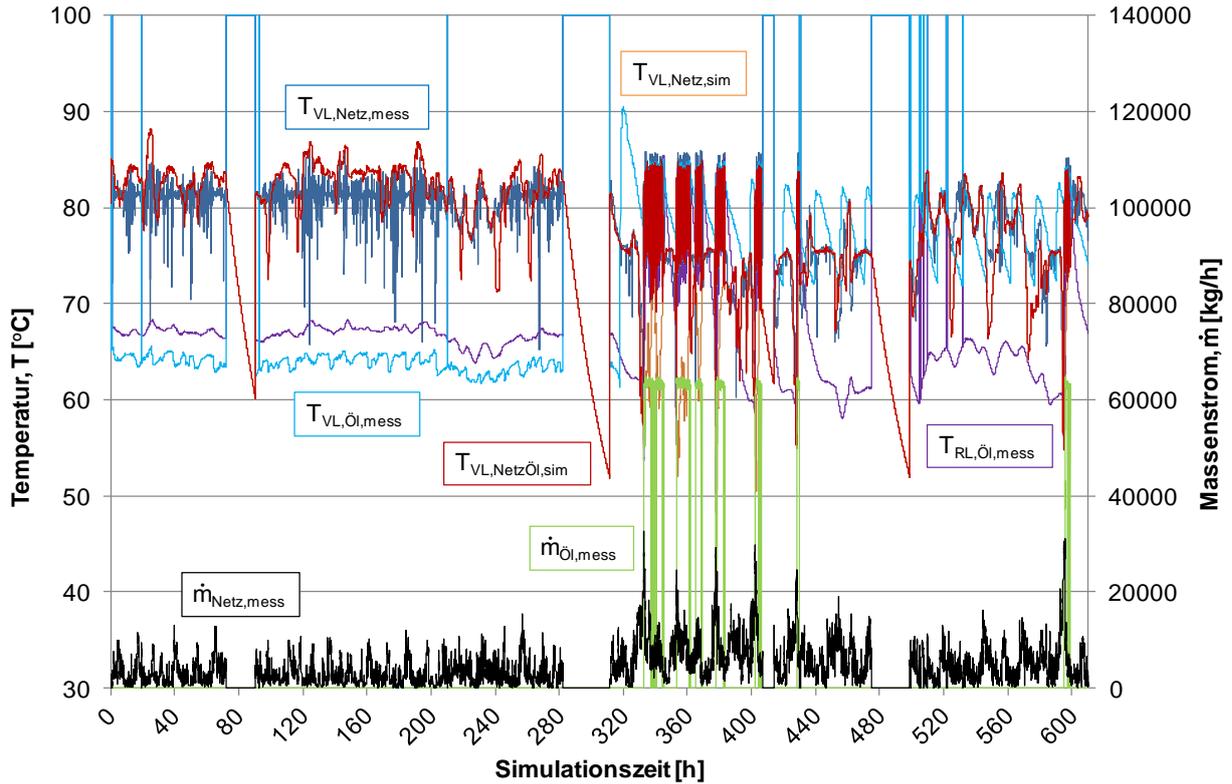


Abbildung 3-48: Temperaturverlauf der Vorlauftemperaturen des Netzes

Die Temperaturen in den Rohrleitungen des Rücklaufs von der Solaranlage und des Biomassekessels sind in Abbildung 3-49 abgebildet.

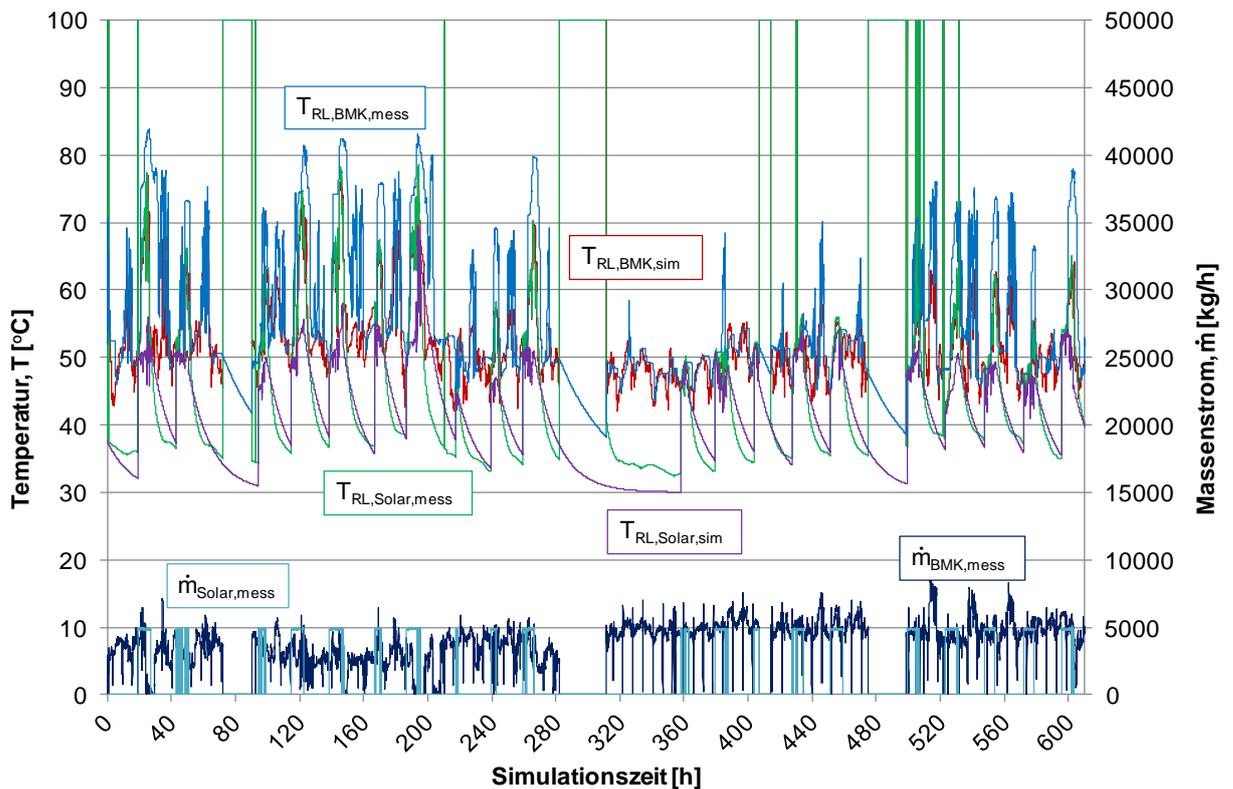


Abbildung 3-49: Temperaturverlauf der Rücklaufemperaturen (Solar, BMK)

Die gemessene Rücklauftemperatur der Solaranlage ($T_{RL,Solar,mess}$) konnte zu Simulationsbeginn nicht von der Simulation ($T_{RL,Solar,sim}$) abgebildet werden. Jedoch änderte sich dies ab der Stunde 360. Die gemessene Rücklauftemperatur des Biomassekessels ($T_{RL,BMK,mess}$) zeigte große Schwankungen, die in der Simulation ($T_{RL,BMK,sim}$) nicht zu sehen waren. Diese Abweichungen konnten aufgrund fehlender Informationen nicht näher bestimmt werden.

Durch die Annahme der konstanten Umgebungstemperatur (30 °C) konnte die Auskühlung im Rücklauf der Solaranlage in der Messung ($T_{RL,Solar,mess}$) von der Simulation ($T_{RL,Solar,sim}$) wiedergegeben werden (z.B. in Abbildung 3-49 zur Stunde 240).

In der Leistungsbilanz (Abbildung 3-50) kam es bedingt durch die niedrigeren Rücklauftemperaturen vom Biomassekessel und der Solaranlage zu Abweichungen. Dies führte zu einer Abweichung des kumulativen Energiebedarfs. Während der kumulative Energiebedarf des Netzes gut wiedergegeben werden konnte, kam es bei dem kumulativen Energiebedarf des Biomassekessels in der Simulation ($E_{BMK,sim}$) zu einer größeren Abweichung. Auch der kumulative Energiebedarf der Solaranlage ($E_{Solar,sim}$) wies bedingt durch den großen Unterschied der Rücklauftemperatur (bis zur Stunde 360), einen großen Unterschied auf.

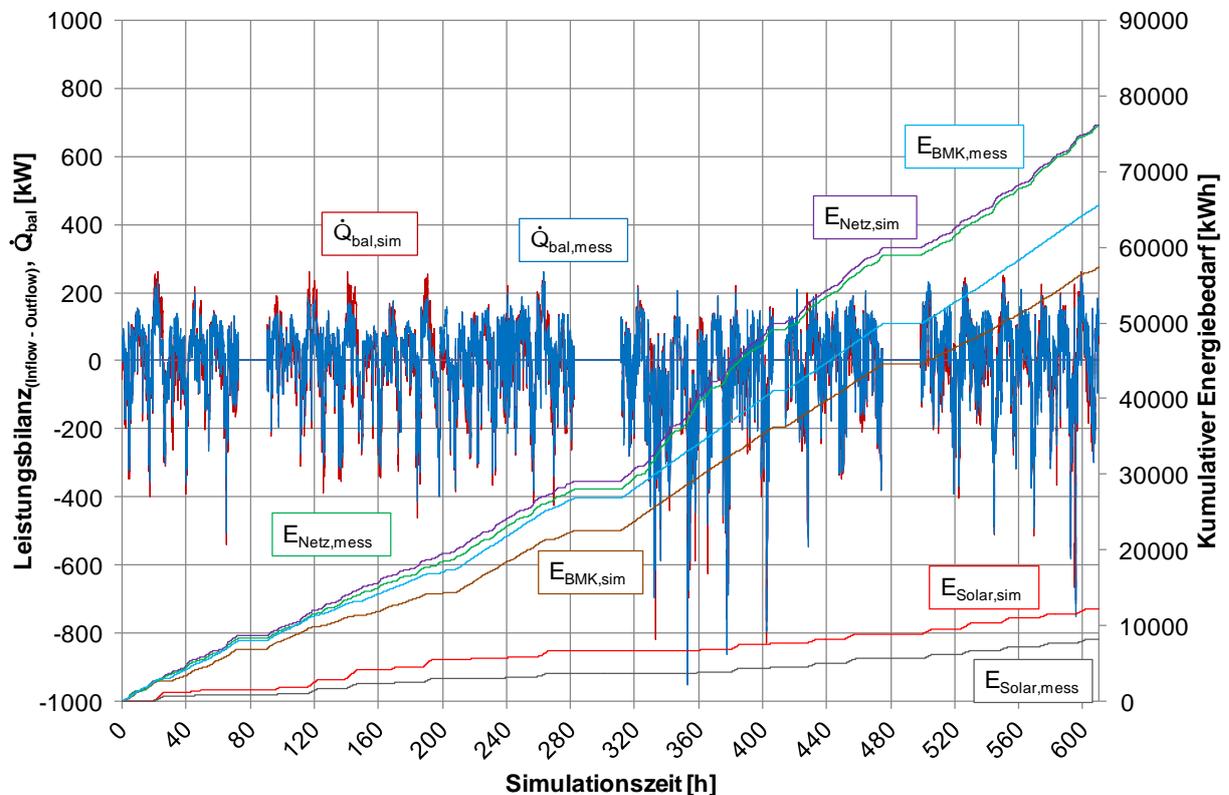


Abbildung 3-50: Leistung und kumulativer Energiebedarf

d) Fazit

Die Annahme einer konstanten Umgebungstemperatur von 30 °C, sowohl für das Gebäude, als auch für den Speicher, kann die Realität nur bedingt wiedergeben. Auch zeigte sich, dass die Annahmen die zur Parametrierung der Rohrleitungen getroffen wurden (Dämmung, Rohrlänge) die Temperaturspitzen an den Messpunkten der Rohrleitungen nur unzureichend abbilden.

Wie schon beim Speicher Gleinstätten (Simulationsdauer 744 h) wurde auch beim Speicher Mattsee (Simulationsdauer 610,3 h) nur der Bereich bis Stunde 144 zur Berechnung der Abweichungen des kumulativen Energiebedarfs herangezogen. Der kumulative Energiebedarf des Netzes ($E_{\text{Netz, sim}}$) wich in diesem Zeitraum um 4,7 % von der Messung ($E_{\text{Netz, mess}}$) ab. Der kumulative Energiebedarf des Biomassekessels ($E_{\text{BMK, sim}}$) wies zur Messung ($E_{\text{BMK, mess}}$) eine Abweichung von 14,4 % auf. Der kumulative Energiebedarf der Solaranlage ($E_{\text{Solar, sim}}$) wies bedingt durch den großen Unterschied der Temperaturen (in der ersten Hälfte der Simulationszeit) eine sehr große Abweichung von 75,6 % zur Messung ($E_{\text{Solar, mess}}$) auf.

Bedingt durch die fehlende Datenlage (Außenlufttemperatur, Rohrlängen, Rohrparameter) konnten die Abweichungen nicht näher eruiert werden.

3.10 Golling (18 m³) – Nahwärme

Der Speicher Golling (18 m³) (Abbildung 3-51) stellt einen Fernwärmespeicher mit einem Füllvolumen von 18 m³ dar. Dieser Speicher ist Teil des Heizwerks Golling, welches die örtliche Fernwärmeversorgung (private, kommunale und gewerbliche Objekte) sicherstellt. Das Heizwerk besteht aus zwei Biomassekesseln (150 und 240 kW Leistung) und besitzt kein Hackschnitzellager. Die Brennstoffversorgung wurde am Heizwerk Golling durch austauschbare Container (jeweils 70 m³) realisiert.

Tabelle 3-10: Eckdaten – Speicher Golling

Benennung	Wert	Einheit
Zeitschritt Simulation / Messdaten	2,5/5	min
Simulationsdauer	461,17	h
Anzahl der Messpunkte im Speicher	8	-
Füllvolumen des Speichers	17,8	m ³
Effektive Wärmeleitfähigkeit (λ_{eff})	2,0023	W/mK
Dämmstärke – Mantel	15	cm
Dämmstärke – Deckel	10,5	cm
Dämmstärke – Boden	14,35	cm

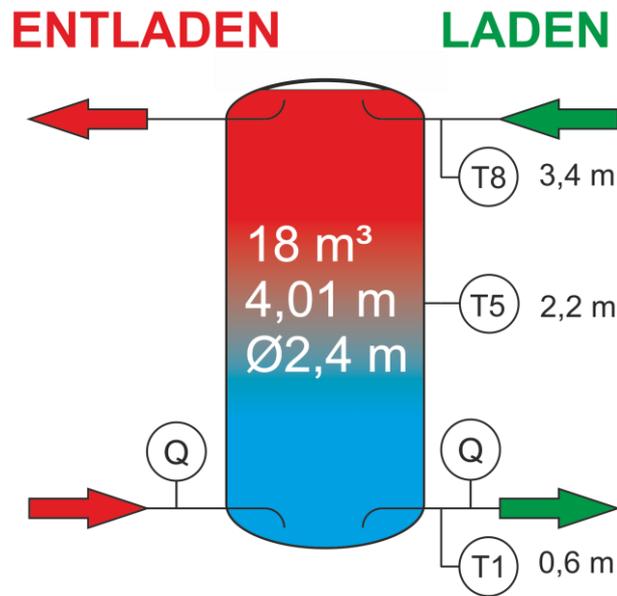


Abbildung 3-51: Schema Golling

Der Speicher Golling besitzt im oberen und im unteren Bereich des Speichers zwei Anschlüsse, welche zum Laden und Entladen benutzt werden. Der eingehauste Speicher weist eine durchgehende Dämmung (Steinwolle) auf und besitzt acht Temperatursensoren (T1 – T8). Die Eckdaten der Simulation wurden in Tabelle 3-10 zusammengefasst.

a) Annahmen und Vereinfachungen

Bedingt durch die im oberen Bereich nach oben und im unteren Bereich nach unten zeigenden Rohrendungen (Abbildung 3-51) musste die Einlasshöhe und die Auslasshöhe anhand von Zeichnungen abgeschätzt werden. Durch die fehlende Messung von Eintritts- und Austrittstemperaturen mussten die nächstliegenden Temperaturen im Speicher (oben T8, unten T1) als aktuelle Eintritts- und Austrittstemperaturen herangezogen werden. Mittels der gemessenen Energiemengen der beiden Biomassekessel sowie der übertragenen Energiemenge an die Fernwärme konnten die Massenströme berechnet werden. Als Außentemperatur wurde für die Simulation 15 °C angenommen.

b) Modellierung in TRNSYS

Der Aufbau der Simulation besteht aus einer Equation welche die Berechnung der Leistung und Energie in der Simulation durchführte. Der Type649 (Mischer) verbindet den Massenstrom der beiden Biomassekessel und übergibt sie dem Speicher. Der Massenstrom des Netzes wird direkt von Type9c (Datareader) dem Speicher übergeben.

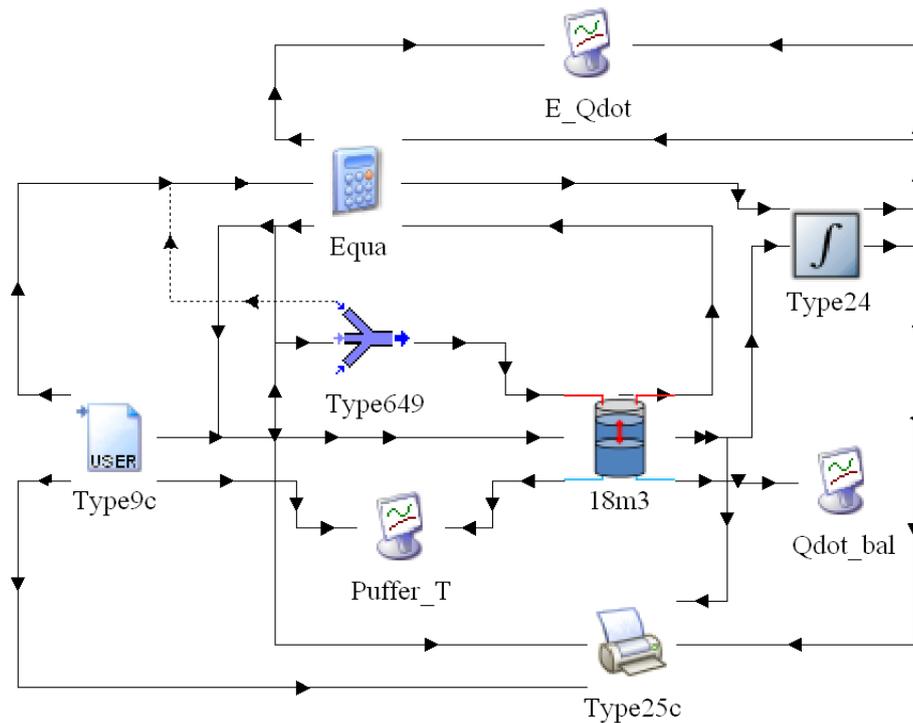


Abbildung 3-52: Simulationsaufbau Speicher Golling

c) Vergleich Messung – Simulation

In der Vorsimulation zeigte sich, dass der Speicher Golling zu bestimmten Zeitschritten geringe Temperaturdifferenzen zwischen dem obersten Temperaturfühler (T8) und dem untersten Temperaturfühler (T1) aufwies. So wies die Temperaturdifferenz (T8-T1, ΔT) innerhalb der Simulationszeit von 461,21 h nur innerhalb von 55 h einen Wert größer gleich 20 °C auf.

Bedingt durch die Berechnung des Massenstromes aus der Energiemenge und der Temperaturdifferenz ΔT ergaben sich sehr große Massenströme. In Abbildung 3-54 ist der Volumenstrom aufgetragen welcher im Maximum 3100 m³/h erreichte.

Die Abbildung 3-53 zeigt im oberen Bereich (linke y-Achse) die gemessenen Temperaturen im Speicher Golling und im unteren Bereich (rechte y-Achse) die Temperaturen der Simulation. Da die Messdaten Fehler aufwiesen wurden die fehlenden Zeitschritte mittels EXCEL- Makro eingefügt. Als Wert für fehlende Zeitschritte wurde für die Simulation Null gewählt um diese Zeiträume des Datenausfalls in der Simulation darstellen zu können. Zur besseren Darstellung im Diagramm wurde der Wert der fehlenden Temperaturen von Null auf 100 gesetzt, um Überlappungen mit dem darunterliegenden Temperaturverläufen zu vermeiden (in Abbildung 3-53 senkrechte Linien).

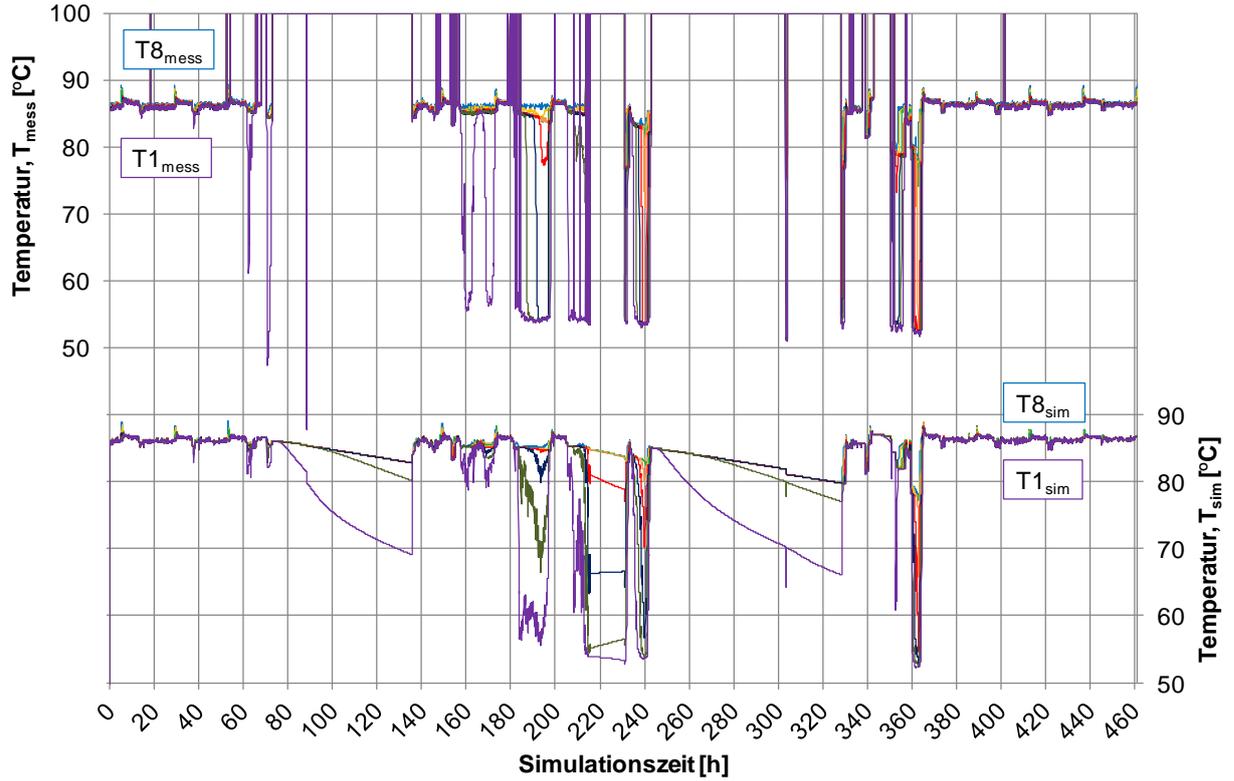


Abbildung 3-53: Temperaturverlauf – Sensoren

Die Abbildung 3-54 zeigt den obersten Temperaturfühler ($T_{8_{\text{mess}}}$) sowie den untersten Temperaturfühler ($T_{1_{\text{mess}}}$).

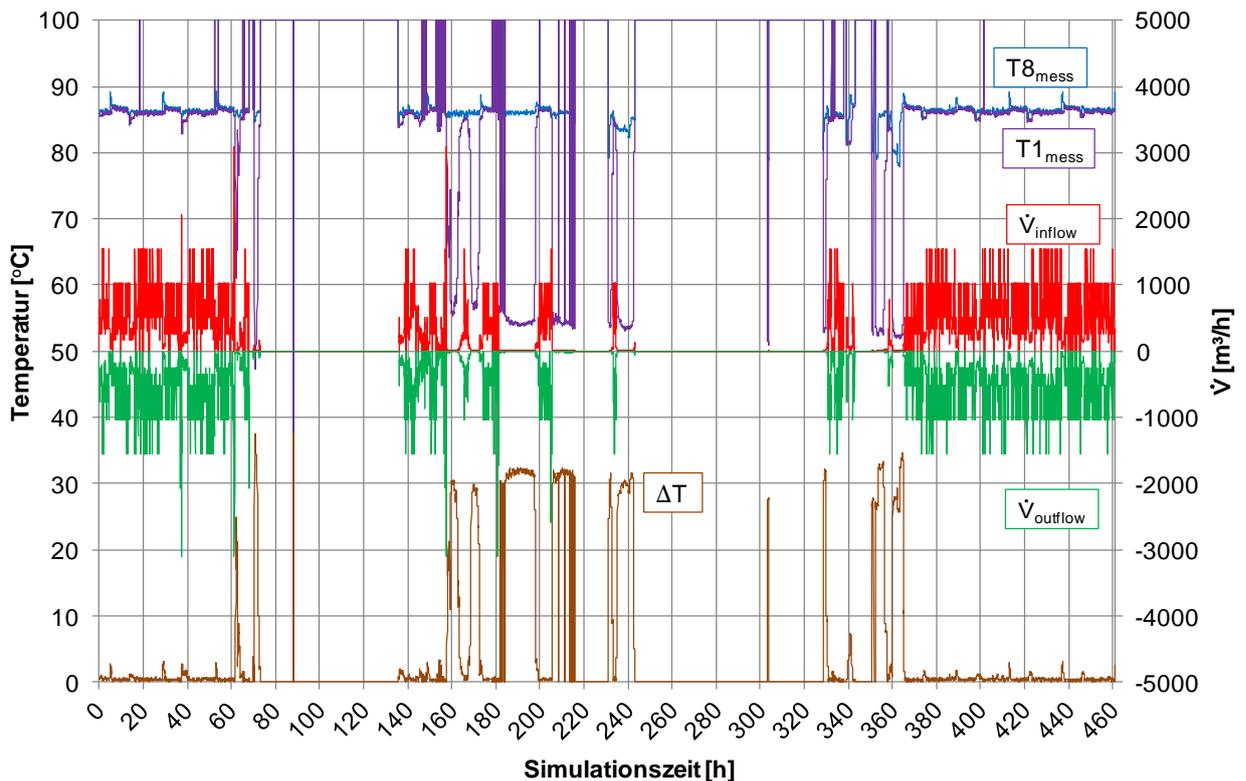


Abbildung 3-54: Temperaturverlauf beim Beladen und Entladen

Im unteren Bereich der Abbildung ist die Temperaturdifferenz (ΔT) dargestellt. Es ist zu erkennen, dass die Temperaturdifferenz (ΔT) der Simulation in großen Bereichen sehr klein ist. Dadurch erhöht sich, bedingt durch die Vorgabe der übertragenen Energie, der Volumenstrom (Maximum 3100 m³/h) beträchtlich.

d) Fazit

Für den Speicher Golling waren acht Temperaturfühler vorhanden. Jedoch wurden die Temperaturen nicht auf der Höhe des Einlasses und des Auslasses ermittelt (gemessene Differenz aus der Zeichnung 0,26 m zwischen dem Temperaturfühler und dem Rohrende im Speicher). Zugleich wird dieser Speicher auf derselben Höhe Beladen (Vorlauf der Biomassekessel) und Entladen (Vorlauf der Fernwärme). Am unteren Anschluss befinden sich der Rücklauf der Biomassekessel, sowie der Fernwärme (ebenfalls auf derselben Höhe). Dadurch könnte es zu einem direkten Austausch der Wärme der Biomassekessel und der Fernwärme gekommen sein. Auch wies die Messung nur eine geringe Genauigkeit (10 kWh Schritte im 5 min Zeitschritt) auf.

Dadurch ergab sich über eine Zeitdauer von 406,21 h (gesamte Simulationszeit 461,21 h) eine Temperaturdifferenz von Vor- und Rücklauf die kleiner als 20 °C war. Die Berechnung des Massenstromes ergab somit unrealistisch hohe Massenströme.

Aufgrund der unzureichenden Datenlage (Ein- und Austrittstemperatur, Messausfälle, ...), konnte dieser Speicher nicht in entsprechender Qualität abgebildet werden.

Daher wurde auf eine detailliertere Simulation verzichtet, und somit keine Energiebilanz über den Speicher Golling aufgestellt.

4 EFFIZIENZ

Ein Vergleich der Speicher war aufgrund ihrer unterschiedlichen Betriebsweisen (Fernwärme, Energiespeicher) nur bedingt möglich (Kapitel 3). Um diese Speicher nun untereinander vergleichen zu können wurde ein Prüfzyklus generiert, welcher in Anlehnung an die prEN 12977-3:2008 (E) erstellt wurde. Mittels dieses Prüfzyklus ist es möglich die Speicher hinsichtlich ihrer Parameter (z.B. Dämmung, Kubatur) zu vergleichen.

Wie sich aber schon zuvor in Kapitel 3 zeigte, hat die Betriebsweise der einzelnen Speicher einen entscheidenden Einfluss auf die Verluste, welche während der Speicherung entstehen. Deshalb sollte die reale Betriebsweise nicht vernachlässigt werden. Auch das Optimierungspotential hinsichtlich einer geschichteten Beladung der Speicher soll hier nun näher untersucht werden.

4.1 Effizienz des Modellprozesses

4.1.1 Beschreibung der prEN 12977-3:2008 (E)

Die in prEN 12977-3:2008 (E) vorgestellten Prüfverfahren dient zur Bewertung von Wasserspeicher thermischer Solaranlagen. Mit diesen Prüfverfahren ist es möglich Parameter zu generieren, welche zur Simulation eines Wasserspeichers ($0,05 \text{ m}^3 - 3 \text{ m}^3$) benötigt werden. Diese Parameter umfassen das effektive Speichervolumen, die Wärmeverlustrate des gesamten Speichers, den Abbau der Temperaturschichtung während der Ruhephase und die Temperaturschichtung bei der Entladung. Diese zu vermessenden Speicher weisen meist eine Vielzahl an Anschlüssen auf, wie Solarkreis (Wärmetauscher), Raumheizung, Trinkwasser und Kessel, wodurch eine Abschätzung schwierig ist. Das Prüfungsverfahren teilt Speicher hinsichtlich ihres Be- und Entladungsverfahrens in fünf Gruppen ein (Tabelle 4-1).

Tabelle 4-1: Einteilung der Speicher nach prEN 12977-3:2008 (E)

Gruppe	Beladeverfahren	Entladeverfahren
1	Direkt	Direkt
2	Indirekt	Direkt
3	Direkt	Indirekt
4	Indirekt	Indirekt
5	Sonstige	

Thermische Energiespeicher fallen durch die meist direkte Beladung durch Doppelanschlüsse der Gruppe 1 zu. Der Prüfzyklus (Test V) der Gruppe 1 besteht aus 7 Phasen. Die erste Phase dient zur Konditionierung des Speichers auf eine konstante Temperatur von 20 °C . Danach

erfolgt eine erste Beladung mit einer Temperatur von 40 °C, bis am Austritt eine Temperatur von 35 °C erreicht wird. Der Volumenstrom sollte der Hälfte des Speichervolumens pro Stunde entsprechen. Danach erfolgt eine Entladung bis das Integral des Volumenstroms (Dreiviertel des Speichervolumens pro Stunde) der Hälfte des Speichervolumens entspricht. Auf diese Entladung erfolgt eine erneute Beladungsphase mit 60 °C, mit demselben Volumenstrom wie die Beladung zuvor. Darauf folgt eine Ruhephase welche 16 Stunden dauert. Im Anschluss daran wird der Speicher entladen (Dreiviertel des Speichervolumens pro Stunde) bis die Austrittstemperatur aus dem Speicher eine konstante Temperatur von 20 °C erreicht (prEN 12977-3:2008 (E), 2008).

4.1.2 Beschreibung des Prüfzyklus

Der in der prEN 12977-3:2008 (E) beschriebene Prüfzyklus (Test V, Gruppe 1) ist zur Ermittlung von Parametern von Wasserspeicher von thermischen Solaranlagen entwickelt worden.

Das Volumen der im Rahmen dieser Arbeit betrachteten Speicher (18 m³ bis 2050 m³) ist um ein Vielfaches höher als das in der prEN 12977-3:2008 (E) angegebene (0,05 m³ - 3 m³).

Die Prüfergebnisse, welche die prEN 12977-3:2008 (E) liefern kann wurden durch die Randbedingungen der Simulation bereits vorweggenommen. Die Wärmeverlustrate ist bestimmt durch die Größen welche für die Dämmung des Speichers und die Anschlussverluste (Angabe nur beim Type844 möglich) angegeben werden. Die Festlegung der Knotenzahl in der Simulation hat einen direkten Einfluss auf die Schichtung im Speicher (Knotenmodell, ein Knoten ist ein vollständig gemischtes Volumen) und damit auf den Temperaturverlauf der Ruhephase und der Entladung.

Der in Kapitel 4.1.1 beschriebene Prüfzyklus wurde in Richtung industriell genutzer Energiespeicher erweitert, sodass der Speicher ($T_{\text{init}} = 20 \text{ °C}$) bereits bei der ersten Beladung (DP1) mit 90 °C beladen wird bis am Austritt des DP1 ($T_{\text{out,DP1}}$) eine Temperatur von 85 °C erreicht wird (Abbildung 4-1). Nach einer kurzen Ruhephase von vier Stunden erfolgt die Entladung über DP2 (Eintrittstemperatur 20 °C). Die Entladung wird beendet, wenn die Austrittstemperatur ($T_{\text{out,DP2}}$) 25 °C unterschreitet. Nach diesem ersten Zyklus (Be- und Entladung) erfolgt eine Ruhephase von 11 Stunden.

Nach der Ruhephase erfolgt ein weiterer Be- und Entladungszyklus, wobei bei diesem Zyklus die Dauer der Ruhephase 16 Stunden beträgt. Wenn dieser Zyklus beendet ist, beginnt eine Auskühlphase, welche aber in der Berechnung der Effizienzen nicht berücksichtigt wird.

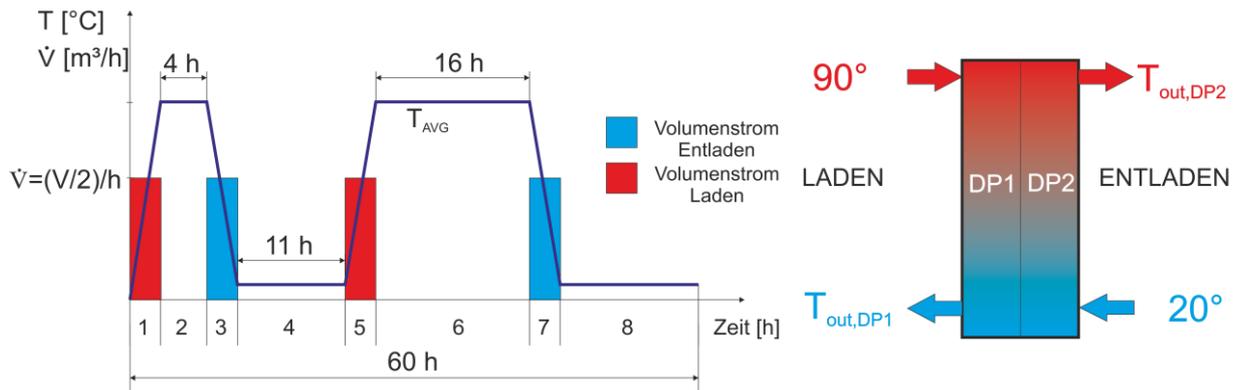


Abbildung 4-1: Prüfzyklus Speichertest für industriell genutzte Speicher (in Anlehnung an prEN 12977-3:2008 (E))

Um das mögliche Optimierungspotential eines Speichers aufzuzeigen, wurde bei der Berechnung der Effizienz zwischen einem realen Speicher und einem idealen Speicher unterschieden. Der reale Speicher entspricht einem Speicher mit den Parametern welche denen der Simulation aus dem Kapitel 3 entsprechen. Der ideale Speicher weist folgende Änderungen auf:

- Schichtladeeinheit
- optimale Entnahmehöhen
- keine Anschlussverluste
- Dämmstärke festgelegt auf 20 cm

4.1.3 Modellierung in TRNSYS

In Abbildung 4-2 ist der Testaufbau für den Prüfzyklus Speichertest abgebildet, welcher bereits in 4.1.2 beschrieben wurde. Dieser Testaufbau gliedert sich in zwei Teile, dem Regelkreis (Bereich innerhalb der grünen Linie) und dem Testkreis (Bereich innerhalb der roten Linie). Beide Kreise werden über eine Sammlung von Gleichungen (Equations) beschrieben. Die Informationsflüsse zwischen den beiden Bereichen bestehen aus den Austrittstemperaturen (in der Abbildung 4-2 Quadrat „b“ - $T_{out,DP2}$, Quadrat „c“ $T_{out,DP1}$) des Type844 (Speicher mit variabler Füllhöhe) und den Regelgrößen (Quadrat „a“ ; inflow, outflow) welche von der Equation „Regler“ an die Equation „Control“ weitergegeben werden. In „Control“ erfolgte die Multiplikation der Regelsignale mit den Massenströmen, welche an den Type844 übergeben werden. Die wesentlichen Berechnungen wurden in „Control“ zusammengefasst und über den Type25c (Printer) ausgegeben.

Im Regelkreis dienen die Austrittstemperaturen $T_{out,DP1}$ und $T_{out,DP2}$ als Input für die drei Regler (Type2b, Differential Controller). Sie geben ihre jeweiligen Kontrollsignale an den Type93 (Input Value Recall) weiter, welcher in diesem Testaufbau den Wert des Kontrollsignal des derzeitigen Zeitschritts, sowie den Wert des Kontrollsignals des vorherigen Zeitschritts an den „Regler“ übergibt. In den Equations „Regler“ befindet sich eine Sammlung von Gleichungen, welche die

verschiedenen Phasen des Prüfzyklus beschreiben. Dazu werden drei Modelle des Type980 (Run-time Calculator) benötigt, welche die Laufzeit eines Signals ausgeben können. Der Type93 (Quadrat „d“) dient der Speicherung des Signals outflow (Massenstrom DP2, Entladung) vom vorherigen Zeitschritt. Der Integrator (Type24) hat die Aufgabe die Veränderung von Signalen, welche er vom „Regler“ erhält, aufzusummieren.

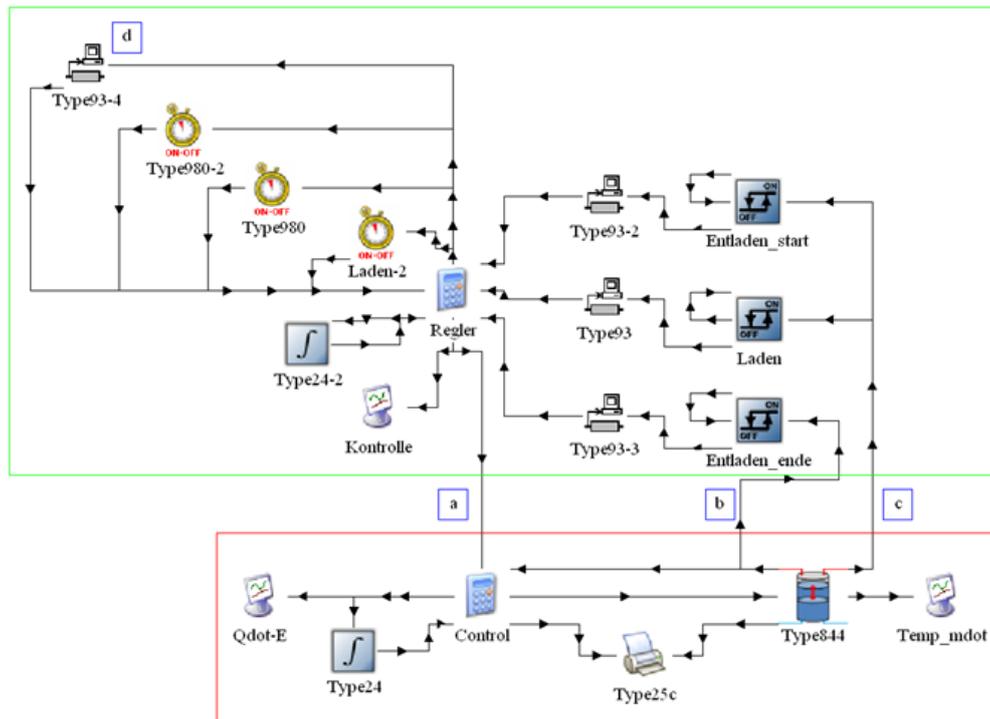


Abbildung 4-2: Simulationsaufbau Prüfzyklus Speichertest

4.1.4 Ergebnisse

Die Ergebnisse aus der Simulation wurden mittels EXCEL ausgewertet. Dazu wurde im ersten Schritt die Änderung der Energie im Speicher (ΔE) mit den kumulativen Energien des Eintritts, des Austritts und der Verluste am Ende der Simulation überprüft, um eine etwaige Abweichung der Energiebilanz festzustellen. Durch die Ausgabe der austretenden Massenströme des Speichers in der Simulation konnte eine Aufteilung in Beladung, Entladung und Ruhephase durchgeführt werden. Daraus wurden die einzelnen Periodeneffizienzen (siehe auch Kapitel 2.7.2) gebildet. Ihr Produkt ergab die Gesamteffizienz (siehe auch Kapitel 2.7.1).

In der Abbildung 4-3 ist zu erkennen, dass die im idealen Fall (Schichtladung, optimale Entnahmhöhen, keine Anschlussverluste, einheitliche Dämmstärke 20 cm) die Gesamteffizienz in großem Maße vom Volumen bestimmt ist („Graz“ größtes Volumen (2050 m³, „Golling“ kleinstes Volumen (18 m³)). Die Speicher Knittelfeld (KNI320, 25 cm Dämmstärke), Gleinstätten (GL_Jan, ~40 cm Dämmstärke) und Mattsee (~60 cm Dämmstärke) weisen größere Dämmstärken auf als die angenommene Dämmstärke (20 cm) für den idealen

Fall. Der Energiespeicher STIEGL (STIEGL_ESP, 10 cm Dämmstärke) weist bedingt durch die Annahmen in der Simulation (Anschlussverluste) eine große Differenz zum idealen Fall auf. Die Anschlussverluste sind bei den Speichern Rottenmann 54 (ROT54, 20 cm Dämmstärke) und Rottenmann 250 (ROT250, 25 cm Dämmstärke) besonders ausgeprägt. Auch der Warmwasserspeicher Graz weist eine Differenz auf, welche hauptsächlich auf die geringe Dämmstärke (12 cm) zurückzuführen ist. Der Speicher Golling (15 cm Dämmstärke) konnte in der Simulation nicht nachgebildet werden (siehe Kapitel 3.10). Seine Parameter wurden zwar ermittelt, konnten jedoch nicht mit der Messung validiert werden.

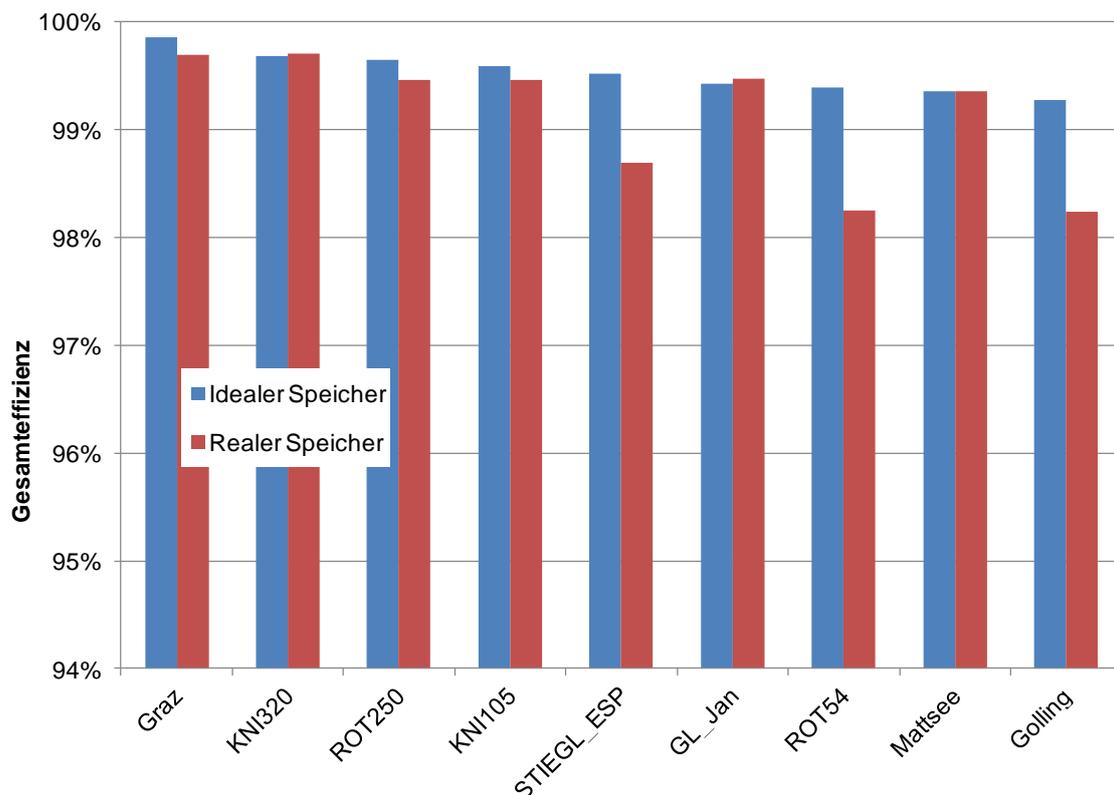


Abbildung 4-3: Gesamteffizienz Prüfprozess ; reale – ideale Speicher

Die Gesamteffizienz in Abhängigkeit vom Füllvolumen zeigt die Abbildung 4-4. Es lässt sich die Tendenz erkennen, dass mit steigendem Füllvolumen die Effizienz steigt. Dies kann auf die Verluste zurückgeführt werden.

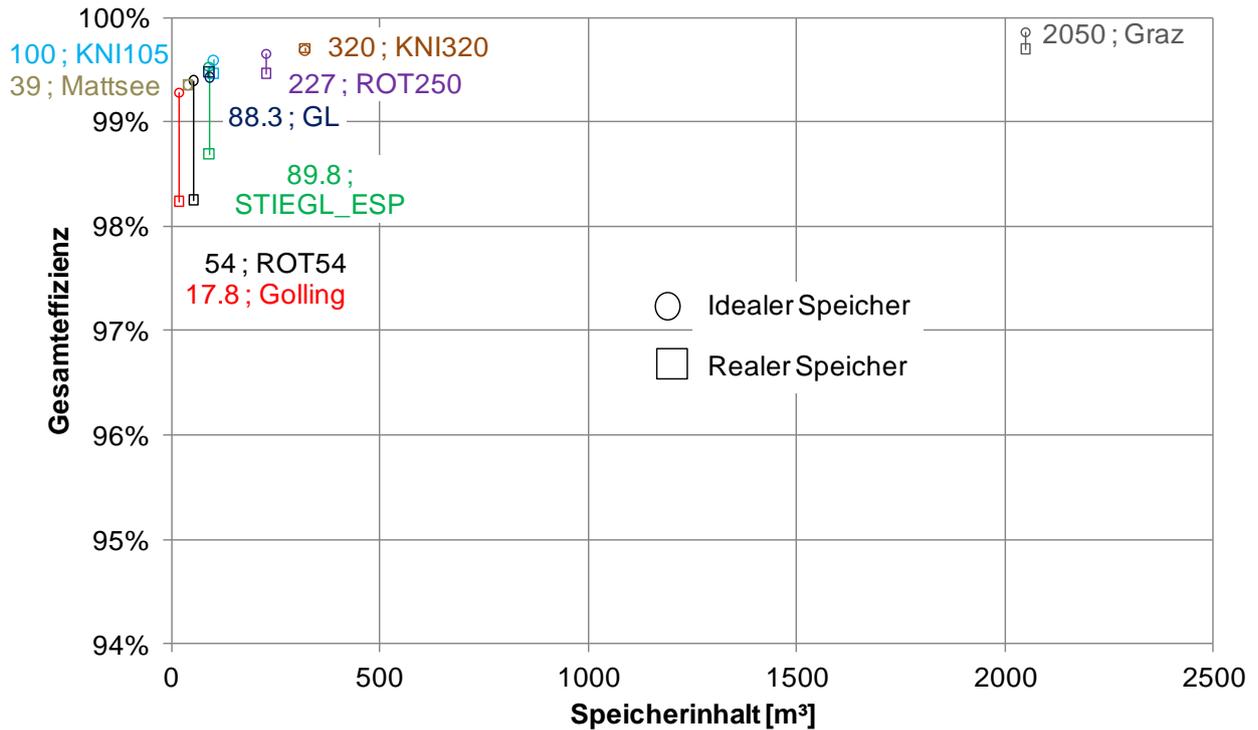


Abbildung 4-4: Gesamteffizienz Prüfprozess ; reale – ideale Speicher vs. Füllvolumen

Die Periodeneffizienzen für den Prüfzyklus sind im Anhang - A zu finden.

4.2 Effizienz des realen Prozesses

4.2.1 Ermittlung der Effizienz im realen Prozess

Der in Abbildung 4-5 schematische Aufbau wurde bereits in Kapitel 3.3 erläutert. Ergänzt wird der Simulationsaufbau durch die Equation „Calc_SC“, welche die wesentlichen Berechnungen für die Effizienz beinhaltet. Die dafür notwendigen Verknüpfungen sind mit einer rot punktierten Linie eingezeichnet. Im Unterschied zur Simulation werden nur die eintretenden und die austretenden Leistungen des Speichers, die Verluste des Speichers und die Änderung der inneren Energie im Speicher ausgewertet. Diese werden von „Calc_SC“ an den Printer (Type25c) übergeben. Diese Modifikation wurde an allen Speichern durchgeführt.

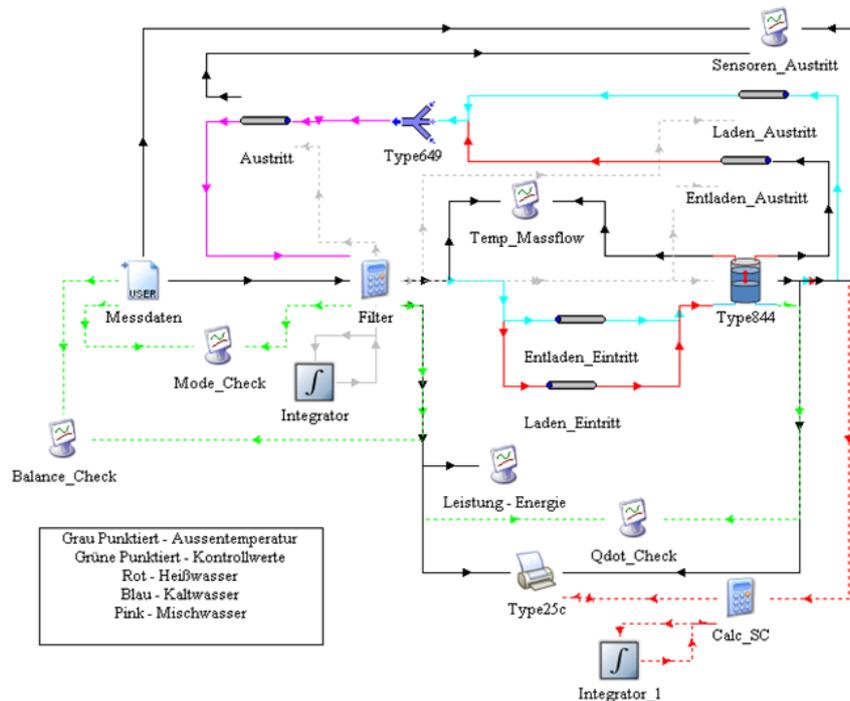


Abbildung 4-5: Simulationsaufbau des realen Prozess – Warmwasserspeicher Graz – Energie Steiermark

Die Berechnung der Periodeneffizienzen musste nun für jeden Be- und Entladungsvorgang, sowie für jede Speicherung durchgeführt werden. Da aber eine kürzere Speicherdauer auch geringere Verluste aufweist, würde bei einer großen Anzahl von kurzen Speichervorgängen und eines längeren Speichervorgangs, dieser längere Speichervorgang die Effizienz nur wenig beeinflussen. Dies wurde durch die Gewichtung der Periodeneffizienzen mit ihrer Dauer sehr gut berücksichtigt. Somit können kurze Speicherzyklen die Effizienz nur über ihren Anteil an der Gesamtzeit beeinflussen. Die Ermittlung der gewichteten Periodeneffizienzen erfolgte mittels eines Makros, welches in der VBA – Umgebung von Microsoft EXCEL geschrieben wurde.

Eine weitere Änderung betraf die Speicher Gleinstätten und Mattsee. In ihrem Betrieb weisen diese Speicher eine gleichzeitige Be- und Entladung auf. Somit kann nicht zwischen der Periodeneffizienz der Beladung und der Entladung unterschieden werden. Dadurch konnte bei diesen Speichern nur die Periodeneffizienz des Be- und Entladungszyklus (Kapitel 2.7.2, η^D_4), sowie die Periodeneffizienz des Speicherzyklus (η^D_2) ermittelt werden.

Für den Speicher Golling konnte bedingt durch die unzureichende Datenlage (Ein- und Austrittstemperaturen, Volumenströme) keine Berechnung der Effizienzen anhand der realen Prozessdaten erfolgen.

4.2.2 Ergebnisse

Der betrachtete Simulationszeitraum war durch die kürzeste Dauer der Messdaten, 120 Stunden (Messzyklus der Speicher Knittelfeld) begrenzt. Die Ergebnisse wurden mittels EXCEL ausgewertet und die Effizienzen bestimmt. Wie schon zuvor im Kapitel 4.1 wird auch hier der reale Speicher mit einem idealen Speicher (Schichtladung, optimale Entnahmehöhen, keine Anschlussverluste, einheitliche Dämmstärke 20 cm) verglichen. Da nun die reale Betriebsweise in die Berechnung der Gesamteffizienz einfließt, kann nur die Differenz zwischen dem realen und dem idealen Speicher verglichen werden und nicht die Differenz der Speicher untereinander. Die Ergebnisse der Anlagensimulation sind in Abbildung 4-6 dargestellt. Es zeigte sich, dass bei dem Energiespeicher STIEGL (Stiegl_ESP) die Verluste während der Auskühlphase die Effizienz maßgeblich beeinflussten. Beim Speicher Rottenmann 54 (ROT54) zeigte sich, dass durch die häufige Beladung die Verluste eine geringere Rolle spielten im Vergleich zum idealen synthetischen Prüfzyklus (Abbildung 4-3) davor. Der Effekt der Be- und Entladung ist auch am Speicher Mattsee zu erkennen, wies er doch nur eine Stillstandszeit von zwei Minuten über die gesamte Simulationszeit (120 Stunden) auf. Die Verluste der Speicherphase konnten somit die Gesamteffizienz nur unwesentlich beeinflussen.

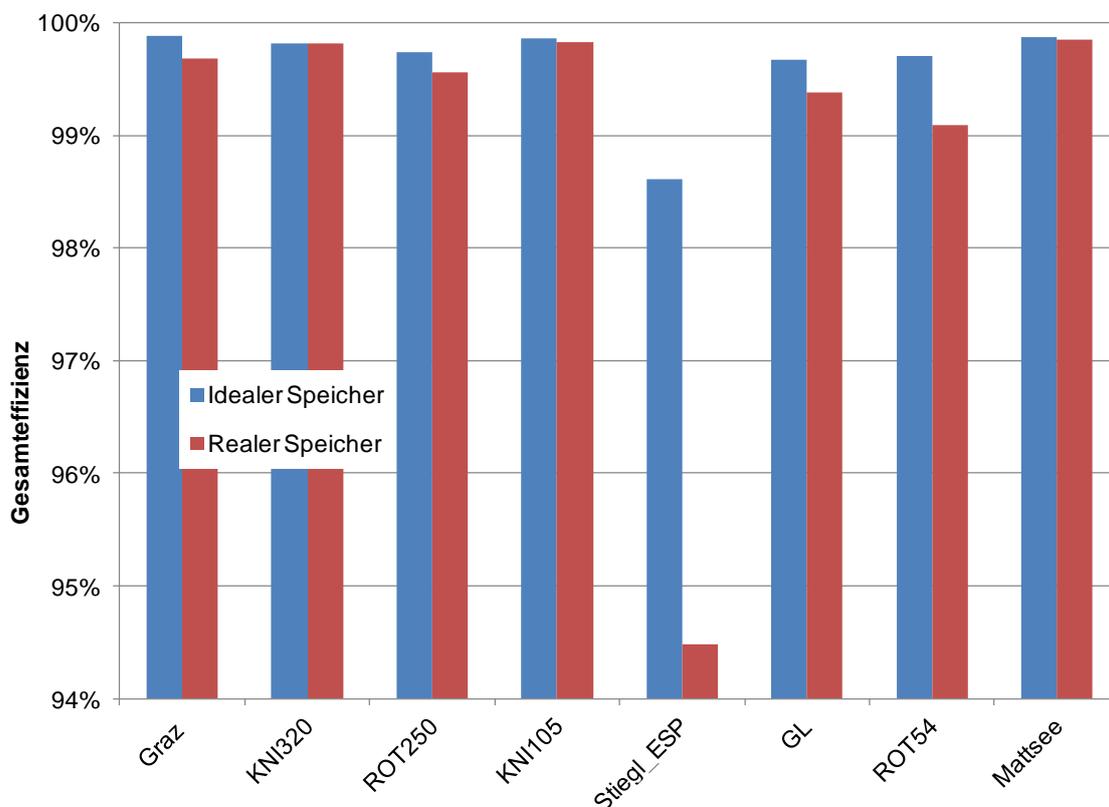


Abbildung 4-6: Gesamteffizienz Realer Prozess ; reale – ideale Speicher

Die Gesamteffizienz in Abhängigkeit vom Füllvolumen (Abbildung 4-7) gibt ein ähnliches Ergebnis, wie schon zuvor der Prüfzyklus (Abbildung 4-4) wieder. Jedoch führt nun die reale Betriebsweise (Sommer- oder Winterbetrieb, kontinuierlicher – oder diskontinuierlicher Betrieb) bei den Speichern zu einer Verschiebung der Ergebnisse (vor allem für den Speicher Stiegl_ESP in Abbildung 4-7 deutlich zu erkennen).

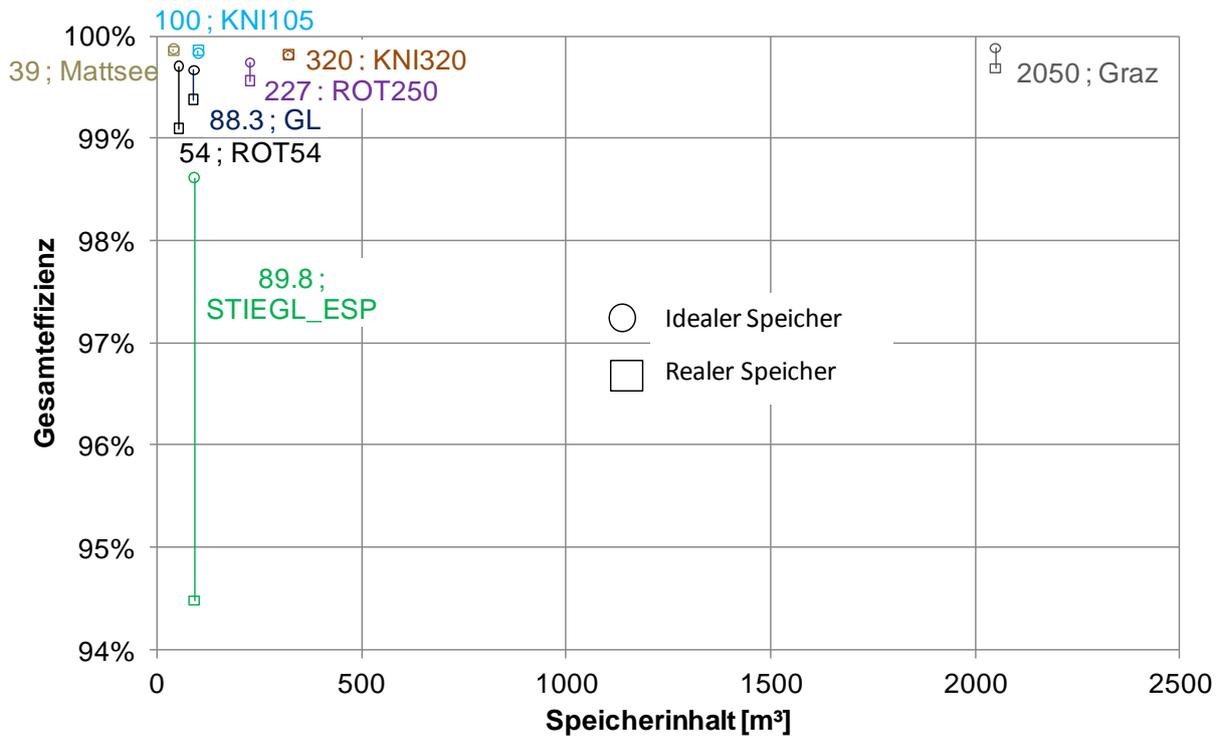


Abbildung 4-7: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen

Die Periodeneffizienzen für die realen Prozessdaten sind im Anhang - B zu finden.

5 SCHLUSSFOLGERUNGEN

Diese Diplomarbeit hatte die Aufgabe ein Speichermodell zu entwickeln welches die Anforderungen von thermischen Energiespeichersystemen, wie sie in Industrie, Gewerbe und Fernwärmenetzen auftreten, bestmöglich erfüllt. Dieses Speichermodell sollte mit einem Minimum an Eingabedaten bei schnellstmöglicher Ausführungszeit ein genaues Abbild der Realität geben können. Zur Validierung des Speichermodells standen Messdaten von verschiedenen thermischen Energiespeichern zur Verfügung, sowie ein Fragebogen welcher vom Anlagenbetreiber auszufüllen war.

Diese thermischen Energiespeicher konnten in TRNSYS unter der Voraussetzung einer ausreichenden Datenlage abgebildet werden. Als Beispiel wäre der Wärmespeicher Graz (2050 m³ Füllvolumen) und der Fernwärmespeicher Golling (18 m³ Füllvolumen) zu nennen. Die vorhandenen Messdaten des Wasserspeichers Graz beinhalteten neben den gemessenen Temperaturen im Speicher auch die Messdaten der ein- und austretenden Temperaturen, sowie der Massenströme. Im Gegensatz dazu waren für den Fernwärmespeicher Golling nur die Speichertemperaturen, welche sich nicht auf der gleichen Höhe wie die Anschlüsse befanden, und die übertragenen Energiemengen bekannt. Die Datenlage reichte hierbei nicht aus um den Fernwärmespeicher Golling in TRNSYS korrekt abzubilden. Als genereller, kritischer Parameter wurde in der Simulation die Datenlage bezüglich der Rohrleitungen erkannt. Während die Dämmstärke und der Durchmesser des Rohres aus dem Fragebogen meist bekannt waren, stellte die Länge der Rohrleitung, die Gestaltung der Rohrleitung und die unbekannt Position der Temperaturmessung die Unsicherheitsfaktoren dar. Hier zeigte sich, dass der direkte Kontakt mit dem Anlagenbetreiber von entscheidender Bedeutung für den Simulationsaufbau in TRNSYS ist.

Bei der Auswertung des kumulativen Energiebedarfs wurde festgestellt, dass die Abweichungen zwischen der Messung und der Simulation mit steigendem Speichervolumen sinken. Der kumulative Energiebedarf des größten betrachteten Energiespeicher, des Wärmespeichers Graz (2050 m³ Füllvolumen) wiesen eine Abweichung kleiner 2,5 % zur Messung auf. Hingegen betragen die Abweichungen des kumulativen Energiebedarfs des kleineren Speichers Rottenmann 54 (52 m³ Füllvolumen) 10 % zwischen der Messung und der Simulation. Der Speicher Rottenmann 54 wies im Gegensatz zum Wärmespeicher Graz eine wesentlich schlechtere Schichtung auf, was am Temperaturverlauf zu erkennen war. Bei kleineren Speichern (<100 m³ Füllvolumen) waren die Auswirkungen der unterschiedlichen Dämmstärken, der zusätzlichen Wärmeverluste der Anschlüsse und die unterschiedlichen Höhen der Anschlüsse beim Be- und Entladen deutlich zu erkennen. Die Verluste haben bei kleineren Energiespeichern auf die Berechnung des kumulativen Energiebedarfs einen bedeutenderen Einfluss als bei größeren Energiespeichern.

Die unterschiedlichen realen Betriebsweisen der einzelnen Energiespeicher erlaubten keine Bewertung der Effizienzen untereinander. Deshalb wurde ein idealer Prüfzyklus, in Anlehnung an die prEN 12977-3:2008 (E) erstellt, welcher nur den Speicher (ohne Verrohrung) berücksichtigt. Dadurch war es möglich einen Vergleich zwischen den einzelnen Speichern darzustellen. Es konnte die Abhängigkeit der Effizienz von der Dämmstärke, den Anschlusshöhen, sowie den Anschlussverlusten durch einen Vergleich mit einem optimal ausgeführten Speicher aufgezeigt werden. Daraus konnte für den jeweiligen Speicher auf das mögliche Optimierungspotential geschlossen werden. Welchen großen Einfluss die Dämmung auf die Effizienz hat, wurde am Beispiel des Energiespeichers STIEGL (10 cm Dämmung) deutlich. Im idealen Prüfprozess wich die Effizienz des Energiespeicher STIEGL bereits merklich vom optimalen Fall ab, was im realen Betrieb durch die längere Auskühlphase noch deutlicher zu erkennen war. Die berechneten Effizienzen im idealen Prüfprozess wiesen generell, im optimalen Fall, eine deutliche Verbesserung mit steigenden Speichervolumen auf.

Aus den Erkenntnissen der Simulationen wurde ein eindimensionales Speichermodell entwickelt, welches die Anforderungen der im Rahmen dieser Diplomarbeit betrachteten Speicher wiedergeben kann. Folgende Anforderungen konnten für das Speichermodell aus der Simulation abgeleitet werden:

- Vorgabe eines Temperaturprofils
- Vorgabe der Dämmstärken in den Bereichen Mantel, Deckel und Boden
- Angabe von zusätzlichen Wärmeverlusten an den Anschlüssen
- Abbildung einer konstanten oder einer variablen Füllhöhe

Ein Speichermodell, welches diese Anforderungen erfüllt, wurde in Rahmen dieser Arbeit in der entwickelt und befindet sich im Anhang - D.

Damit konnte in dieser Arbeit erfolgreich gezeigt werden, dass eindimensionale Speichermodelle die zur Verfügung stehenden Messdaten von den ausgewählten, industriellen, thermischen Energiespeichern, bei ausreichender Datenlage mit guter Genauigkeit wiedergeben können. Ein derartiges Speichermodell kann also für die Einbindung in Optimierungsalgorithmen für Industrieprozesse zum Einsatz kommen.

LITERATUR

Altenburger F., (2010), Simulation und Optimierung von Biomasse-Solar Systemen für Mikro-Wärmenetze mit einer thermischen Leistung von 20 bis 150 kW, Diplomarbeit, Fachhochschule Burgenland

Atkins M. J., Walmsey M., Neale J. R., (2010), The challenge of integration non continuous processes – milk powder plant case study. *Journal of Cleaner Production* 18, S927-934

Chen C.-L. and Ciou V.-J., (2008), Design and optimization of Indirect Energy Storage Systems for Batch Process Plants. *Ind. Eng. Chem. Res.* 47, S4817-4829

Dincer I., Rosen M., (2002), *Thermal energy storage: systems and applications*, Wiley, New York

Drück H., (2006), Wärmespeicher 1. Teil, Folien der Vorlesung „Solarenergienutzung WS 2006/07“, Institut für Thermodynamik und Wärmetechnik, Universität Stuttgart, Folie S29

Drück H., (2006a), Multiport Store - Model, Stratified Fluid Storage Tank with four internal heat exchangers, ten connections for direct charge and discharge and an internal electrical heater, Version 1.99F, Institut für Thermodynamik und Wärmetechnik, Universität Stuttgart

Foo F. C. Y., Chew Y. H., Lee C. T., (2008), Minimum units targeting and network evolution for batch heat exchanger network. *Applied Thermal Engineering* 28, S2089-2099

Haller M., Cruickshank C., Streicher W., Harrison S., Andersen E., Furbo S., (2009), Methods to determine stratification efficiency of thermal energy storage processes – Review and theoretical comparison. *Solar Energy* 83, S1847-1860

Han Y.M., Wang R.Z., Dai Y.J., (2009), Thermal stratification within the water tank. *Renewable and Sustainable Energy Reviews* 13, S1014-1026

Heimrath R., (2004), Simulation, Optimierung und Vergleich solarthermischer Anlagen zur Raumwärmeversorgung für Mehrfamilienhäuser, Dissertation, Institut für Wärmetechnik, Technische Universität Graz

Heinz A., Lerch W., Bredler J., Fink Ch., Wagner W. (2012): „WRGpot - Wärmerückgewinnung aus Abwasser im Niedrigenergie- und Passivhaus: Potenzial und Konzepte in Kombination mit Solarthermie und Wärmepumpe“; Projektbericht im Rahmen des Programms Haus der Zukunft plus, im Auftrag des Bundesministeriums für Verkehr, Innovation und Technologie (zur Veröffentlichung eingereicht)

Huhn R., (2007), Beitrag zur thermodynamischen Analyse und Bewertung von Wasserwärmespeichern in Energieumwandlungsketten. Dissertation, Technische Universität Dresden

IEA SHC Task 33, Weiss H., (2010), Solare Prozesswärme für industrielle und gewerbliche Anwendungen, Berichte aus der Energie- und Umweltforschung, 88/2010

John R., (2002), Validierung eines Rechenmodells für Großspeicher zur Auslegung solarer Heizsysteme, Dissertation, Philipps-Universität Marburg

Klemeš J., Friedler F., Bulatov I. Varbanov P., (2010), Sustainability in the process industry: Integration and optimization. McGraw-Hill Professional, New York

Lauber A., (2007), Wärmeverluste durch rohrinterne Gegenstromzirkulation in Speicheranschlussleitungen, und deren Verminderung mittels konvektionsbremsen, Konvektionssperren und Wärmesiphons – Quantifizierung der Effekte mittels Messung, SPF Solartechnik, Hochschule für Technik HSR Rapperswil

Majozi T., (2009), Minimization of energy use in multipurpose batch plants using hest storage: an aspect of cleaner production. Journal of Cleaner Production 17, S945-950

Panthalookaran V., Heidemann W., Müller-Steinhagen H., (2007), A new method of characterization for stratified thermal energy stores. Solar Energy 81, S1043-1054

prEN 12977-3:2008 (E), 2008, Thermal solar systems and components – Custom built systems – Part3: Performance test methods for solar hot water heater stores, FINAL DRAFT, Februar 2008

Promise Application,

<http://www.aeeintec.at/index.php?seitenName=projekteDetail&projekteld=128>, 25.10.2012

Schnitzer H., Brunner C., Gwehenberger G., (2007), Minimizing greenhouse gas emissions through the application of solar thermal energy in industrial processes. Journal of Cleaner Production 15, S1271-1286

Schranzhofer H., Heinz A., Puschnig P., Streicher W., (2006), Validation of a TRNSYS simulation model for PCM energy storages and PCM wall construction elements. Ecstock Conference, 31th May – 2nd June 2006, Stockton College, Pomona, USA

Schranzhofer H.,(2009), Thermische Anlagen- und Gebäudesimulation (TAGS) mit TRNSYS (TRaNsient System Simulation Program), Folien der Vorlesung „Numerische Verfahren in der Energietechnik“, Institut für Wärmetechnik, Technische Universität Graz, Folie S41-45

SolProBat, Schnitzer H., Brunner C., (2006), Weiterentwicklung einer Optimierungsmethode zur Integration von Solarwärme in gewerblichen und industriellen Produktionsprozessen. Berichte aus Energie- und Umweltforschung, 49/2006

TRNSYS (2011), TRNSYS 17, A Transient System Simulation Tool, V 17.00.0007, Solar Energy Lab, University of Wisconsin – Madison, USA, 2011

Varbanov P., Klemeš J., (2011), Integration and management of renewable into Total Sites with variable supply and demand. Computers & Chemical Engineering 35, S1815-1826

ANHANG

Anhang - A : Periodeneffizienzen des Prüfzyklus

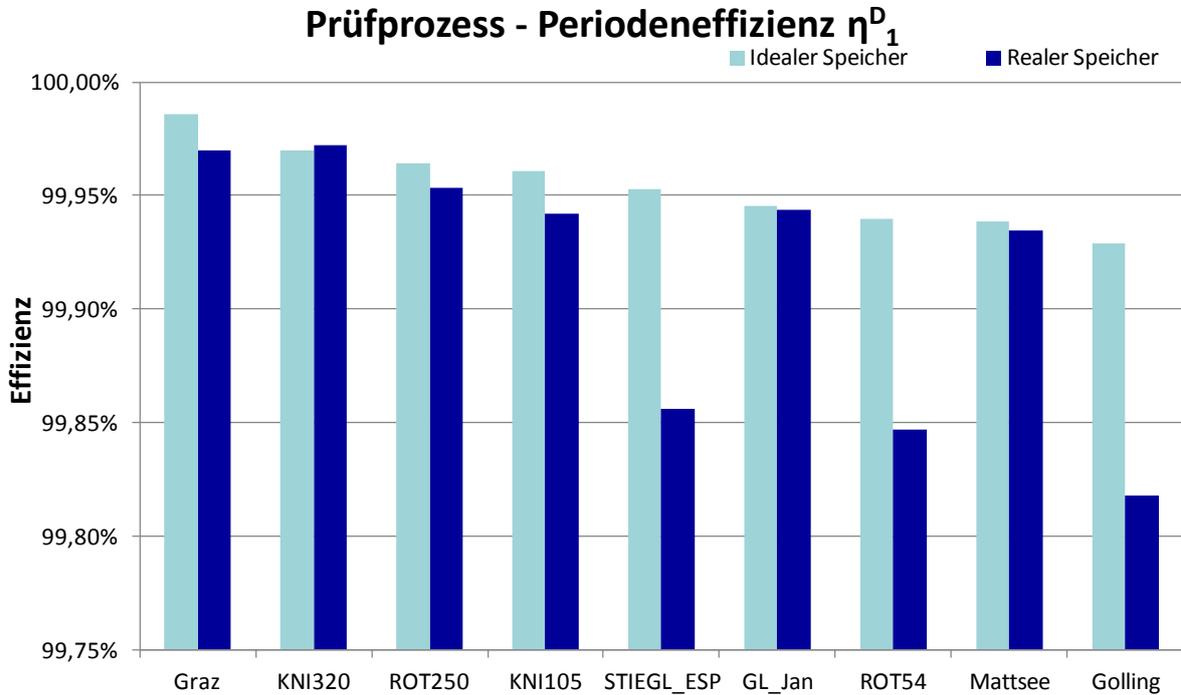


Abbildung A- 1: Periodeneffizienz Prüfprozess ; reale – ideale Speicher – η^D_1

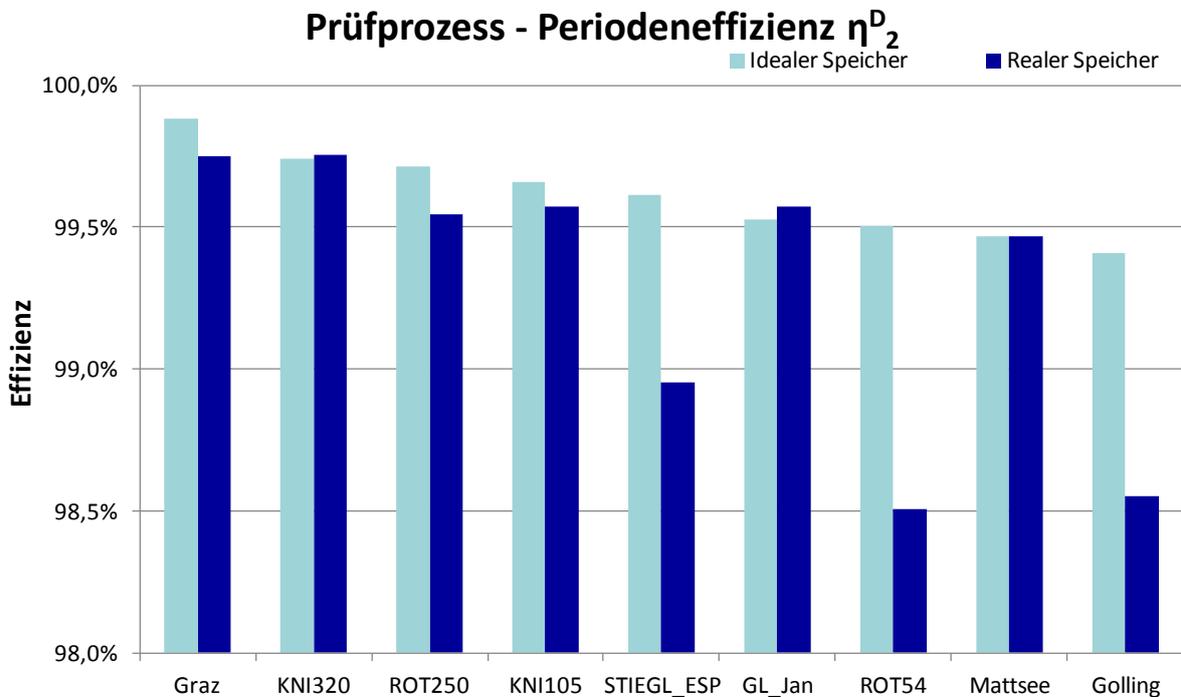


Abbildung A- 2: Periodeneffizienz ; reale – ideale Speicher - η^D_2

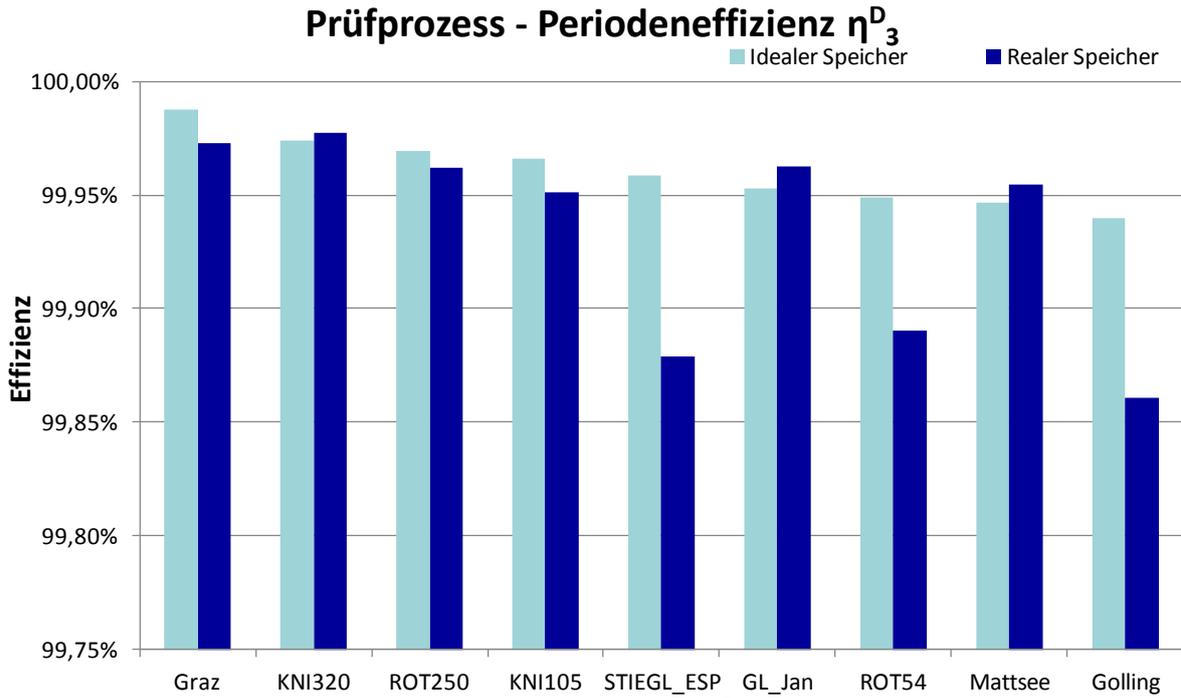


Abbildung A- 3: Periodeneffizienz ; reale – ideale Speicher - η_3^D

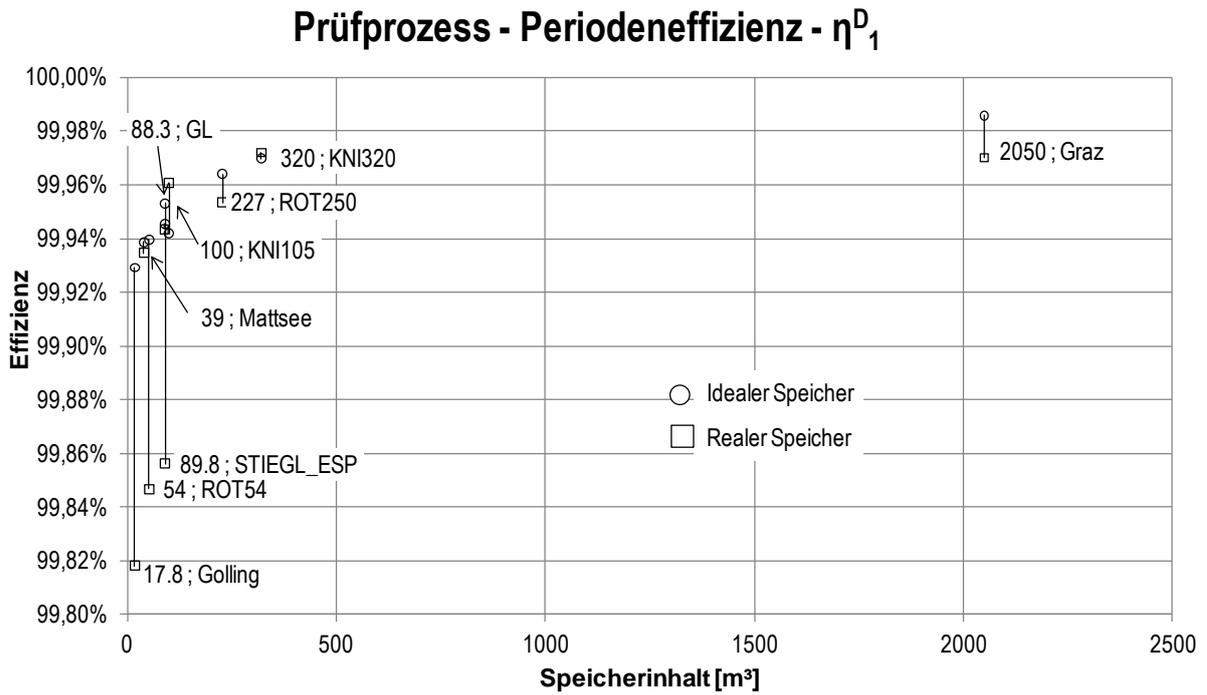


Abbildung A- 4: Periodeneffizienz Prüfprozess ; reale – ideale Speicher vs. Füllvolumen - η_1^D

Prüfprozess - Periodeneffizienz - η^D_2

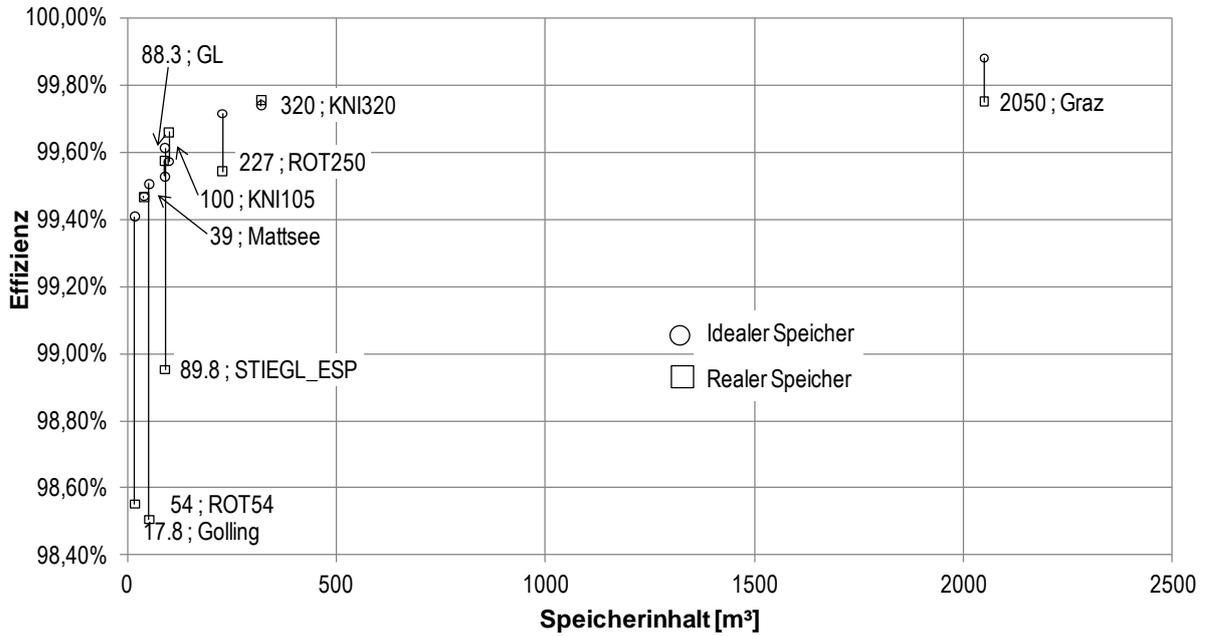


Abbildung A- 5: Periodeneffizienz Prüfprozess ; reale – ideale Speicher vs. Füllvolumen - η^D_2

Prüfprozess - Periodeneffizienz - η^D_3

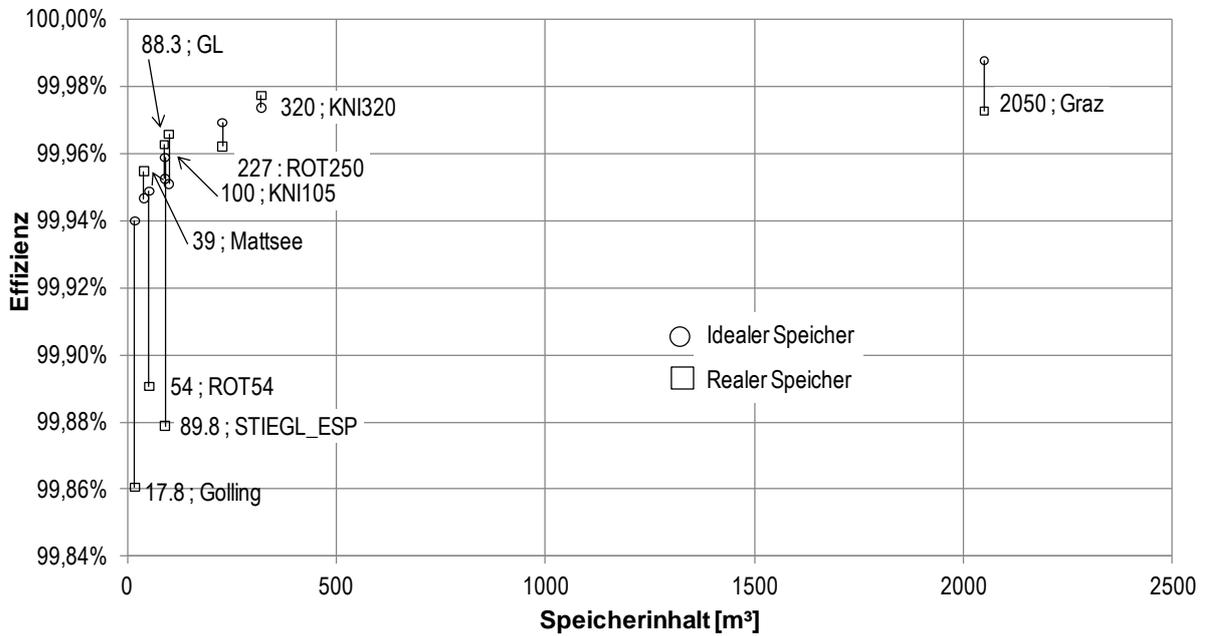


Abbildung A- 6: Periodeneffizienz Prüfprozess ; reale – ideale Speicher vs. Füllvolumen - η^D_3

ANHANG

Anhang - B : Periodeneffizienzen - Reale Prozessdaten

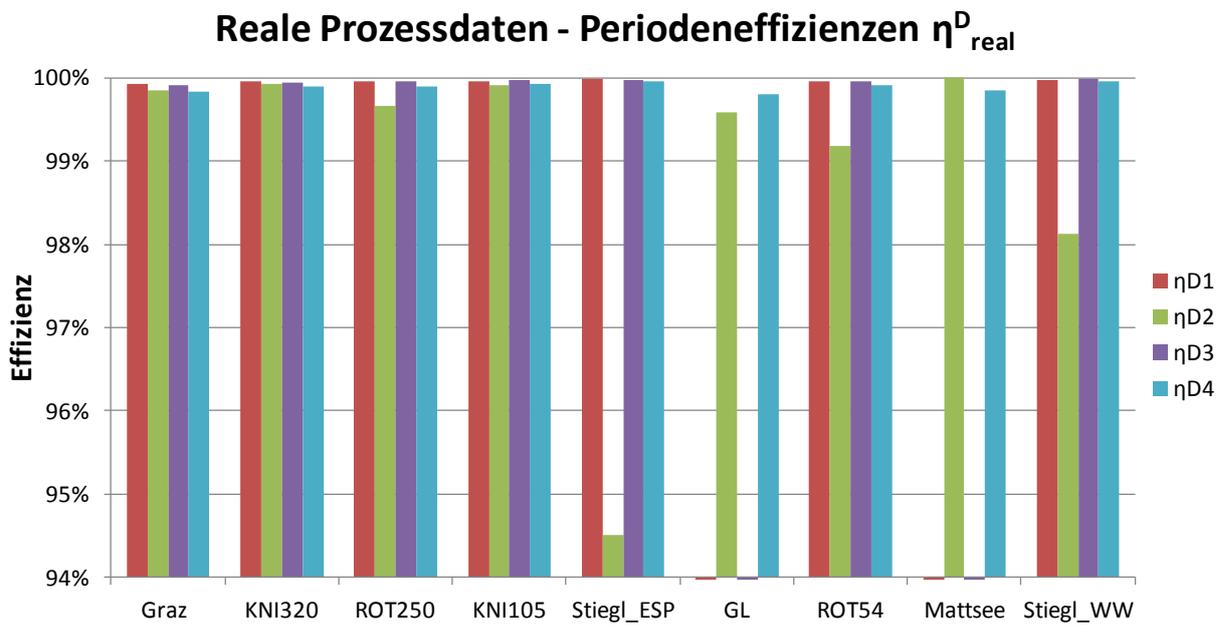


Abbildung A- 7: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_{real}

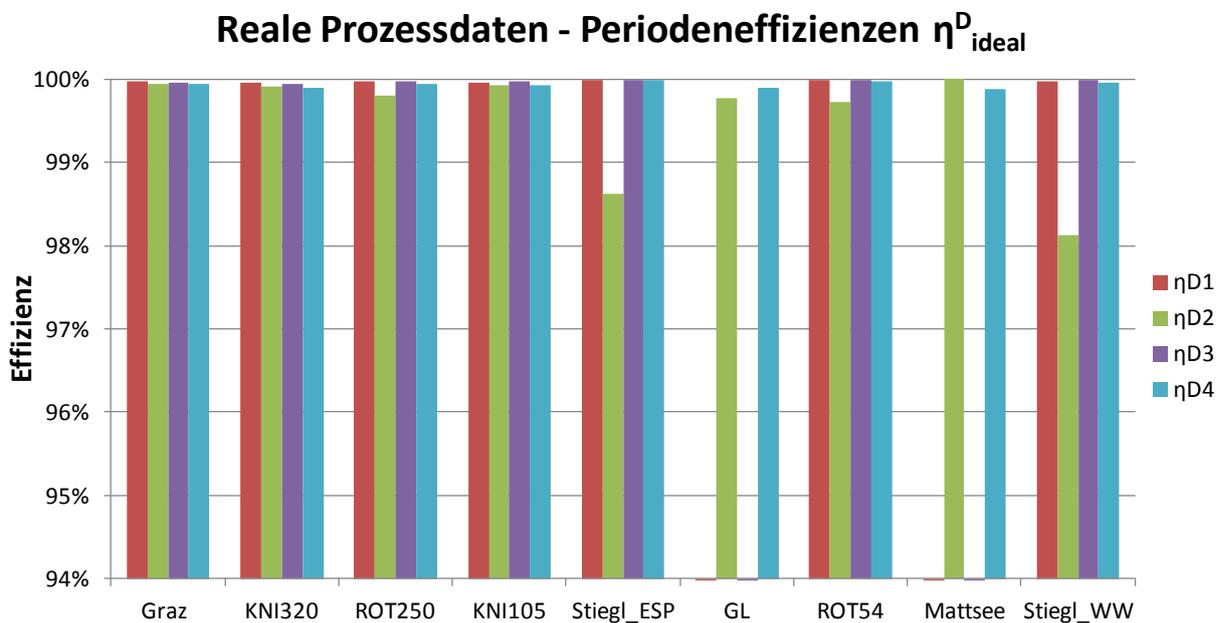


Abbildung A- 8: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_{ideal}

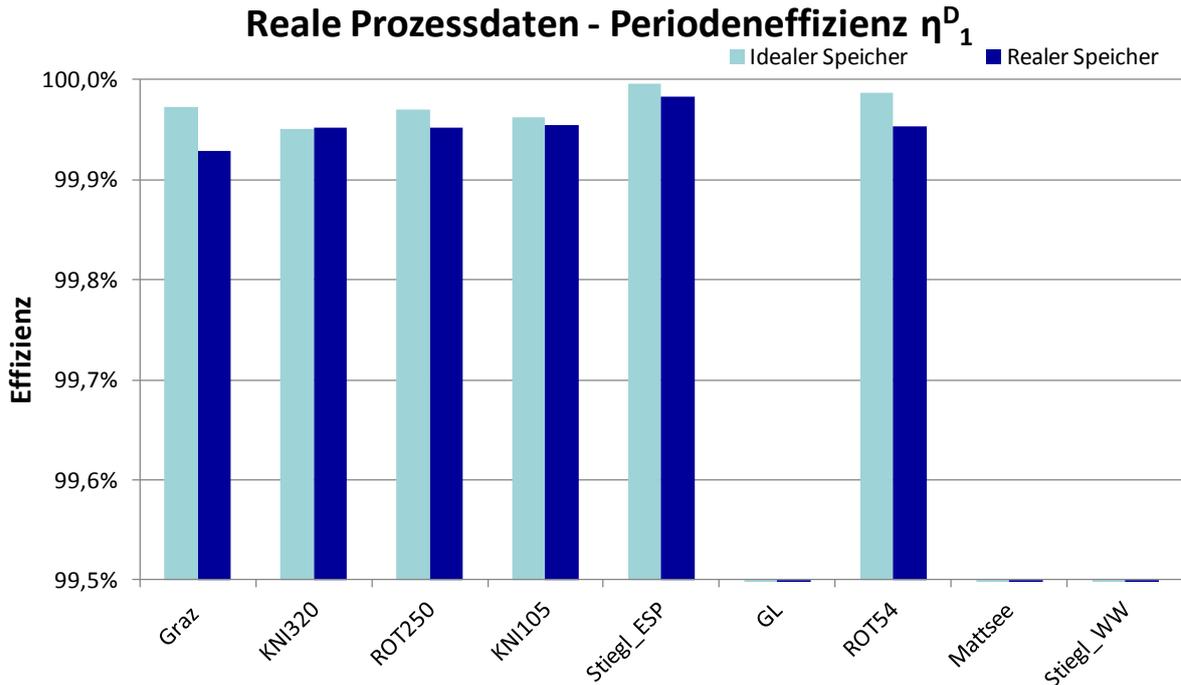


Abbildung A- 9: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_1

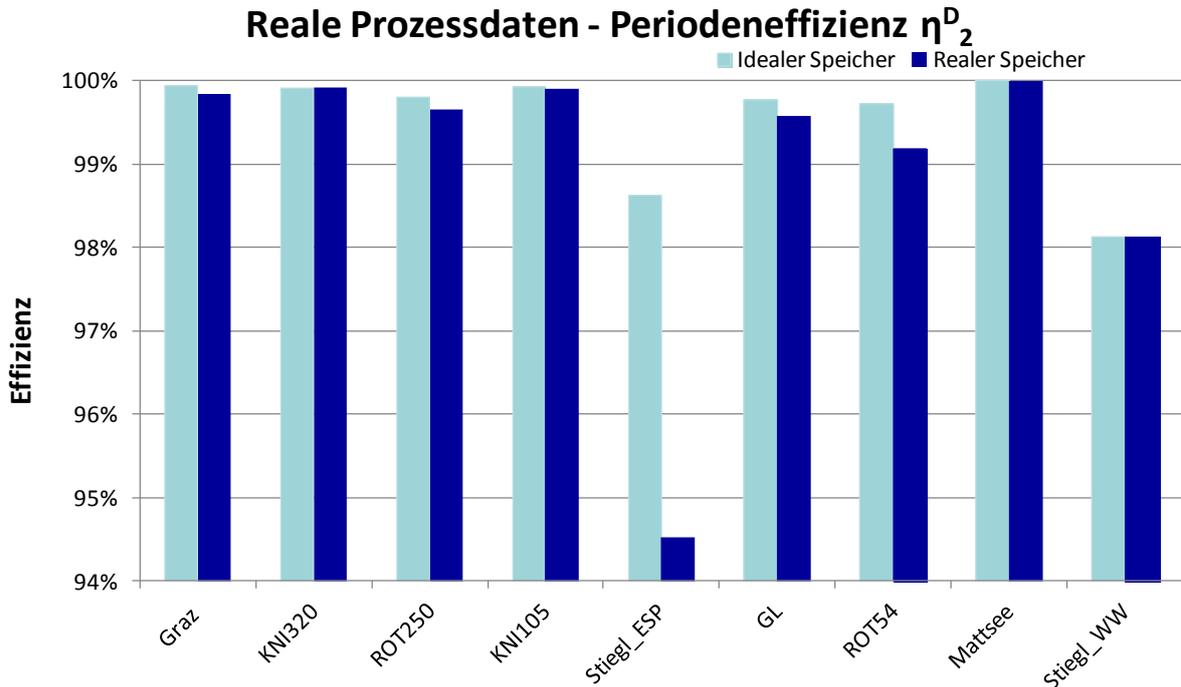


Abbildung A- 10: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_2

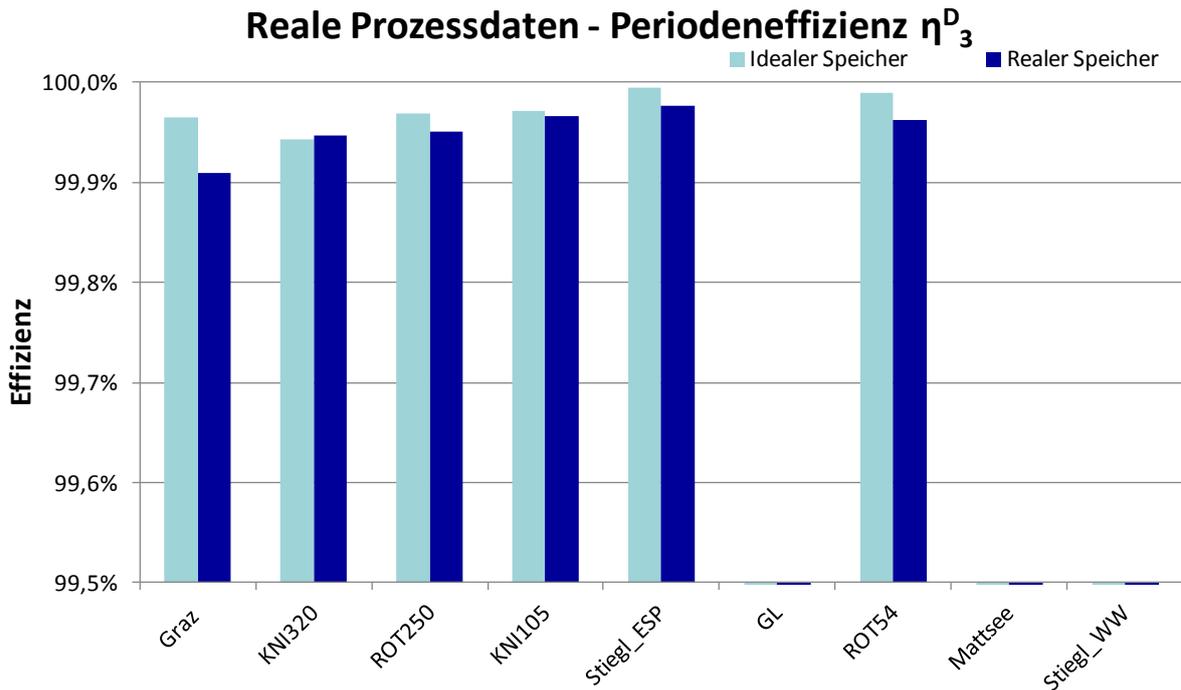


Abbildung A- 11: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_3

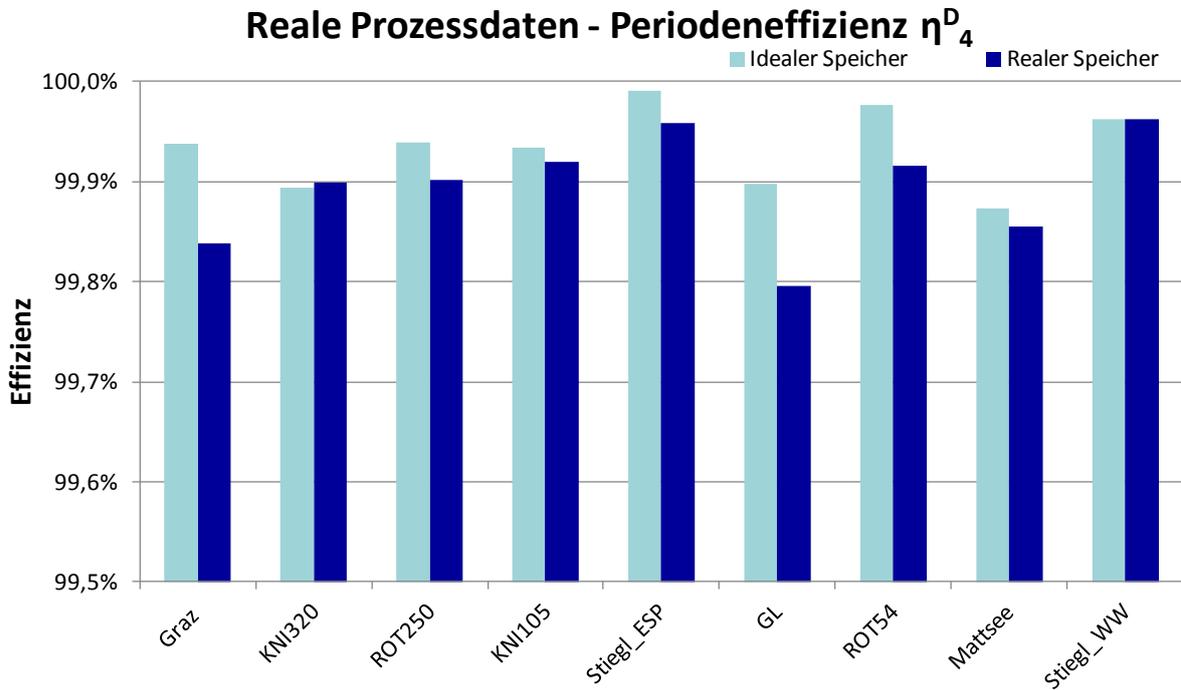


Abbildung A- 12: Gesamteffizienz Realer Prozess ; reale – ideale Speicher vs. Füllvolumen - η^D_4

Anhang - C : Fragebogen

Angaben für die Modellierung thermischer Speicher



Zwischenbericht Projekt SOCO

Graz, am 7.7.2011

Hermann Schranzhofer, Richard Heimrath

Angaben für die Modellierung thermischer Speicher

1 Einleitung

Um einen thermischen Speicher modellieren zu können werden sämtliche relevanten Größen, die den abzubildenden Speicher beschreiben, benötigt. Im Folgenden sollen diese Größen abgefragt werden. Es wird davon ausgegangen, dass im Speicher keine Einbauten und keine Heizstäbe vorhanden sind. Für die Geometrie wird eine Zylinderform angenommen. Bei allen Angaben ist es dringend notwendig die entsprechende Einheit anzuführen. Abbildung 1 zeigt die wesentlichen Größen und Bezeichnungen im Warmwasserspeicher.

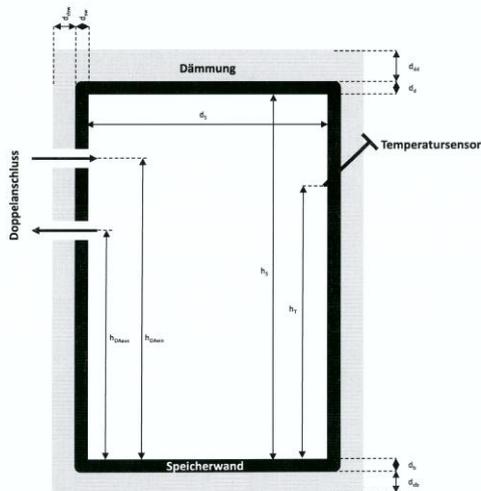


Abbildung 1 Abmessungen, Bezeichnungen und Positionen im Warmwasserspeicher

Zwischenbericht SOCO

2

Angaben für die Modellierung thermischer Speicher

Inhaltsverzeichnis

1 Einleitung 2
 2 Parameter 3
 2.1 Speichervolumen 3
 2.2 Speichermedium 3
 2.3 Speicherverluste 3
 2.4 Anschlüsse 5
 2.5 Temperatursensoren 6
 2.6 Weitere Angaben zu Parametern 7
 3 Inputs 8
 4 Outputs 9
 5 Zusätzliche Angaben für andere Speichertypen 10
 5.1 Speicher mit variablem Speichermedium Volumen 10
 5.2 Kommunizierende Speicher (Speicherbatterie) 11

Zwischenbericht SOCO

1

Angaben für die Modellierung thermischer Speicher

2 Parameter

Hier handelt es sich um Größen, die zeitlich konstant sind. Hauptsächlich sind hier geometrische Parameter und Materialkonstanten gemeint.

2.1 Speichervolumen

Anzugeben sind hier die Innenmaße des Speichers und es sind zumindest zwei der folgenden drei Parameter anzugeben:

Bezeichnung	Wert	Einheit
Speichervolumen V_i	2150	m^3
Speicherhöhe h_i	12,0	m
Speicherdurchmesser d_i	15,1	m

Sind nur Außenmaße bekannt, können auch diese angegeben werden. Wichtig ist jedoch, dass dies vermerkt wird! Gerade bei den geometrischen Angaben sind Skizzen, Pläne und Fotos sehr hilfreich.

2.2 Speichermedium

Für die Berechnungen werden die thermischen Kennwerte des Speichermediums benötigt. Sollten die Parameter nicht bekannt sein, ist zumindest die Art des Speichermediums (z.B. Wasser) anzugeben. Bei speziellen Flüssigkeiten sind eventuell Datenblätter vorhanden und bei Mischungen sollte zumindest das Mischungsverhältnis angegeben werden.

Art des Speichermediums:	Heisswasser
--------------------------	-------------

Bezeichnung	Wert	Einheit
spez. Wärmekapazität	4187	$J / (kg \cdot K)$
Dichte	958,35	kg / m^3
Siedepunkt	100	$^{\circ}C$
Wärmeleitkoeffizient	0,5562	$W / (m \cdot K)$

2.3 Speicherverluste

Im Idealfall wäre hier entweder der Speicherverlustkoeffizient (U-Wert in z.B. $W / (m^2 \cdot K)$) oder die Wärmeverlustrate (in z.B. W / K) anzugeben.

Bezeichnung	Wert	Einheit
Speicherverlustkoeffizient		
Wärmeverlustrate	38,28 *	W

* bei Vernachlässigung der Verluste im Boden

Zwischenbericht SOCO

3

ANHANG

Angaben für die Modellierung thermischer Speicher

Ist keiner dieser beiden Werte bekannt, lässt sich der Speicherverlust über die Angaben der Wand- und Dämmmaterialien ermitteln. Ist hierbei der Wärmeleitkoeffizient (in z.B. W/(mK)) des Materials nicht bekannt, sind zumindest die Dicke und die Art des Materials (z.B. Stahlmantel oder XPS Dämmung) anzugeben. Es besteht auch die Möglichkeit für die Decke und den Boden andere Werte anzuführen.

Speicherwand: **Stahlmantel St37T 6-9mm**

Material der Dämmung	Heralan - DP-7
Material der Speicherwand	Alu-Trapezblech 29/124 0,8mm

Bezeichnung	Wert	Einheit
Dicke der Dämmung $d_{\text{Dämm}}$	100	mm
Dicke der Wand d_{Wand}	6-9	mm
Wärmeleitkoeffizient der Dämmung	0,047	W/mK
Wärmeleitkoeffizient der Wand		

Speicherdecke: **Dachblech 5mm**

Material der Dämmung	Heralan - DP-7
Material der Speicherdecke	Sikaplan 18 G

Bezeichnung	Wert	Einheit
Dicke der Dämmung $d_{\text{Dämm}}$	100	mm
Dicke der Decke d_{Decke}	5	mm
Wärmeleitkoeffizient der Dämmung	0,047	W/mK
Wärmeleitkoeffizient der Decke		

Speicherboden: **Bodenblech 6mm**

Material der Dämmung	
Material des Speicherbodens	

Bezeichnung	Wert	Einheit
Dicke der Dämmung $d_{\text{Dämm}}$		

Zwischenbericht SOCO

4

Angaben für die Modellierung thermischer Speicher

Dicke des Bodens d_{B}	6	mm
Wärmeleitkoeffizient der Dämmung		
Wärmeleitkoeffizient des Bodens		

2.4 Anschlüsse

Für die Position der Ein- und Ausgänge im Speicher ist nur die Angabe der Höhe im Speicher erforderlich (idealerweise bezogen auf den Speicherboden). Um zusätzliche Verluste berücksichtigen zu können ist der Anschlussstyp (a,b,c,d; siehe Abbildung 2) anzugeben. Auch eine Dämmung kann falls vorhanden mit einbezogen werden. Auch hier sind Skizzen, Pläne und Fotos hilfreich.

Abstand Speicher zur Krümmung abfragen und wieder auf Skizzen oder Zeichnungen hinweisen. Doppelschluss erklären.



Abbildung 2 Anschlussarten; a) nach oben, b) nach unten, c) gerade, d) Siphon

Doppelschluss 1 (DA1): **LADEN DN 300**

	Höhe (h_{DAM})	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Abstand Speicher-Krümmung	Einheit
Eingang	11,0	m	b					
Ausgang	0,475	m	b					

Doppelschluss 2 (DA2): **ENTLADEN DN 300**

	Höhe (h_{DAM})	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Abstand Speicher-Krümmung	Einheit
Eingang	0,475	m	b					
Ausgang	11,35	m	b					

Zwischenbericht SOCO

5

Angaben für die Modellierung thermischer Speicher

Doppelschluss 3 (DA3): **BE-ENTLÜFTUNGSVENTIL DN 150**

	Höhe (h_{DAM})	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Abstand Speicher-Krümmung	Einheit
Eingang	1,40	m	c					
Ausgang								

Doppelschluss 4 (DA4): **ÜBERLAUF DN 300**

	Höhe (h_{DAM})	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Abstand Speicher-Krümmung	Einheit
Eingang								
Ausgang	11,8	m	d					

Doppelschluss 5 (DA5):

	Höhe (h_{DAM})	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Abstand Speicher-Krümmung	Einheit
Eingang								
Ausgang								

2.5 Temperatursensoren

Werden im Speicher Temperaturen gemessen wird die Position dieser Sensoren benötigt. Hierbei ist wiederum nur die Höhe im Speicher (idealerweise bezogen auf den Speicherboden, also Innenmaß (siehe Skizze)) anzugeben.

Temperatursensor	Höhe	Einheit
T1 4x	0,5	m
T2	1,5	m
T3	3,0	m
T4	4,5	m
T5	6,0	m
T6	7,5	m
T7	9,0	m
T8	10,5	m
T9 4x	11	m

Zwischenbericht SOCO

6

Angaben für die Modellierung thermischer Speicher

2.6 Weitere Angaben zu Parametern

Sollten noch weitere Angaben zu zeitlich konstanten Werten notwendig sein, können diese im Folgenden angegeben werden.

Raum für weitere Angaben:

Zwischenbericht SOCO

7

Angaben für die Modellierung thermischer Speicher

3 Inputs

Dies sind zeitlich variierende Werte, die dem Modell in einer gewissen Zeitschrittweite zur Verfügung stehen müssen. Ideal wären gemessene Werte über eine repräsentative Zeitperiode, die im Prozess immer wieder kehren (z.B. Tagesverlauf, Wochenverlauf). Die digitalen Datensätze sollten im Text- oder auch Excelformat vorliegen und eine Zeitschrittweite von 10 min nicht überschreiten.

Prinzipiell ist eine Anordnung der Daten in Spalten von Vorteil. Jede Spalte sollte in der ersten Zeile eine eindeutige Beschriftung (z.B. Zeit, Temperatur ...) und in der zweiten Zeile einen Eintrag für die Einheit (z.B. h, °C) haben. Das Trennzeichen zwischen den Spalten sollte eindeutig und einheitlich sein (z.B. TAB, Leerzeichen). Eine Spalte mit Zeitwerten ist dringend erforderlich und kann auch im Format dd.mm.yyy hh:mm:ss vorliegen. Für den Eingang der Doppelschlüsse sind jeweils Temperaturen und Massenströme notwendig (Massenstrom am Ausgang ist der gleiche und die Temperatur am Ausgang ergibt sich aus dem Modell) und für jede Temperaturmessung im Speicher die gemessenen Werte. Zur Berechnung der Verluste muss im Zeitbereich die gemessene Umgebungstemperatur vorhanden sein.

ACHTUNG: Wenn die Messung nicht direkt am Eingang erfolgt ist die Rohrleitungslänge bis zur Messstelle anzugeben. Besonders wichtig ist für diese Rohrleitung die Angabe der Dämmung (Material und Dicke).

Vorschlag einer Anordnung:

Spalte	Bezeichnung	Einheit
1	Zeitstempel	h
2	Temperatur DA1 Eingang	°C
3	Massenfluss DA1 Eingang	kg/h
4	Temperatur DA2 Eingang	°C
5	Massenfluss DA2 Eingang	kg/h
6	Temperatur DA3 Eingang	°C
7	Massenfluss DA3 Eingang	kg/h
8	Temperatur DA4 Eingang	°C
9	Massenfluss DA4 Eingang	kg/h
10	Temperatur DA5 Eingang	°C
11	Massenfluss DA5 Eingang	kg/h
17	Umgebungstemperatur	°C

Sollten weitere Messdaten vorliegen können diese für weitere Analysen nützlich sein. Mit einer kurzen Beschreibung und Angabe der Spaltennummer können diese Daten Verwendung finden.

Angaben für die Modellierung thermischer Speicher

4 Outputs

Das Modell liefert im Wesentlichen folgende Resultate:

Bezeichnung	Einheit
Zeitstempel	h
Temperatur DA1 Ausgang	°C
Massenfluss DA1 Ausgang	kg/h
Temperatur DA2 Ausgang	°C
Massenfluss DA2 Ausgang	kg/h
Temperatur DA3 Ausgang	°C
Massenfluss DA3 Ausgang	kg/h
Temperatur DA4 Ausgang	°C
Massenfluss DA4 Ausgang	kg/h
Temperatur DA5 Ausgang	°C
Massenfluss DA5 Ausgang	kg/h
Abgegebene Leistung an den Verbraucher	kJ/h
Leistungseintrag von der Wärmequelle	kJ/h
Änderung der inneren Energie	kJ/h
Temperaturen in den angesetzten Knoten (verschiedene Höhen im Speicher)	°C

Um die Ergebnisse des Modells validieren zu können sind zu den oben angegebenen Resultaten entsprechende Messdaten erforderlich. Auch hier gelten die schon im vorigen Kapitel angegebenen Hinweise (Kopfzeile, Spalten, Trennzeichen ...). Zumindest die Temperaturen und Massenflüsse an den Ausgängen und die im Speicher gemessenen Temperaturen sollten vorliegen. Weiter Messungen können für detaillierte Analysen hilfreich sein und sollten mit einer kurzen Beschreibung angefügt werden.

Vorschlag einer Anordnung:

Spalte	Bezeichnung	Einheit
1	Zeitstempel	h
2	Temperatur DA1 Ausgang	°C
3	Massenfluss DA1 Ausgang	kg/h
4	Temperatur DA2 Ausgang	°C
5	Massenfluss DA2 Ausgang	kg/h
6	Temperatur DA3 Ausgang	°C
7	Massenfluss DA3 Ausgang	kg/h
8	Temperatur DA4 Ausgang	°C
9	Massenfluss DA4 Ausgang	kg/h
10	Temperatur DA5 Ausgang	°C
11	Massenfluss DA5 Ausgang	kg/h
12	Temperatur Sensor T1	°C
13	Temperatur Sensor T2	°C
14	Temperatur Sensor T3	°C
15	Temperatur Sensor T4	°C
16	Temperatur Sensor T5	°C

Angaben für die Modellierung thermischer Speicher

5 Zusätzliche Angaben für andere Speichertypen

5.1 Speicher mit variablem Speichermedium Volumen

Zusätzlich zu den schon angegebenen Werten ist hier das maximale und minimale Füllvolumen anzugeben.

Maximales Füllvolumen	m ³
Minimales Füllvolumen	m ³

Zusätzlich sind noch gemessene Daten für Füllhöhe notwendig.

Des Weiteren ist anzugeben, wie die Regelung für den Fall des Erreichens des maximalen Füllvolumens angelegt ist. Das Modell bietet hier zwei Möglichkeiten:

Im Modus 1, wird der Überlauf mit dem Inhalt des Speichers gemischt, um eine Zirkulationsströmung zu simulieren (in Abbildung 3 (b) dargestellt). In diesem Fall ist die Temperatur des Überlaufs gleich der Temperatur im Speicher. Im Modus 2 wird der überschüssige Massenstrom vor dem Speichereintritt umgeleitet, wie in Abbildung 3 (c) dargestellt. Die Temperatur des umgeleiteten Stromes ist hier gleich der Temperatur der ankommenden Strömung.

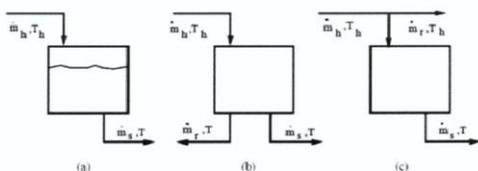


Abbildung 3 Konfiguration des Speichers mit variablem Füllvolumen

Angabe zur Überlaufregelung:

Angaben für die Modellierung thermischer Speicher

5.2 Kommunizierende Speicher (Speicherbatterie)

Die oben angegebenen Werte beziehen sich hier auf einen Einzelspeicher. Sind die Einzelspeicher unterschiedlich aufgebaut, muss jeder vorkommende Einzelspeicher beschrieben werden. Ansonsten sind die Anzahl der Einzelspeicher und die Daten für die Verbindungen anzugeben. Es wird ein geschichteter Speicher mit dem Gesamtvolumen gerechnet und die Verbindungen zwischen den Speichern als zusätzliche Verluste in den betreffenden thermischen Knoten berücksichtigt.

Anzahl der Einzelspeicher	
---------------------------	--

Verbindungen und Dämmmaterial an den Verbindungen:

	Höhe	Einheit	Art (a,b,c,d)	Dämmmaterial	Dicke	Einheit	Länge	Einheit
Verbindung 1 (Speicher 1 mit 2)								
Verbindung 2 (Speicher 1 mit 2)								
Verbindung 3 (Speicher 1 mit 2)								
Verbindung 4 (Speicher 1 mit 2)								
Verbindung 1 (Speicher 2 mit 3)								
Verbindung 2 (Speicher 2 mit 3)								
Verbindung 3 (Speicher 2 mit 3)								
Verbindung 4 (Speicher 2 mit 3)								
Verbindung 1 (Speicher 3 mit 4)								
Verbindung 2 (Speicher 3 mit 4)								
Verbindung 3 (Speicher 3 mit 4)								
Verbindung 4 (Speicher 3 mit 4)								
Verbindung 1 (Speicher 4 mit 5)								
Verbindung 2 (Speicher 4 mit 5)								
Verbindung 3 (Speicher 4 mit 5)								
Verbindung 4 (Speicher 4 mit 5)								

Angaben für die Modellierung thermischer Speicher

5.3 Latentwärmespeicher (PCM)

In jedem Fall sind hier die Materialdaten des verwendeten Phasenwechselmaterials und die Art der Einbringung (Form, Position im Speicher, Menge) in den Speicher notwendig. Da derzeit davon auszugehen ist, dass ein derartiger Speicher bei Industrieprozess noch nicht zum Einsatz kommt, wird dieses Kapitel erst bei Bedarf ausformuliert.

6 Weitere Angaben

Grundsätzlich sind jede Art von weiteren Angaben wertvoll und können zur genauen Analyse beitragen. Vor allem Skizzen, Detailzeichnungen, Diagramme oder auch Fotos sind oft sehr hilfreich Parameter für die Simulationen abzuschätzen. Daher an dieser Stelle noch einmal die Bitte derartige Informationen beizulegen.

Anhang - D: SourceCode – Speichermodell

Option Explicit

```
Sub Core_Var(Time As Double, n As Integer, dt As Double, Inlet As Integer, Outlet As Integer, _
    Para_mdot_In() As Double, Para_mdot_Out() As Double, T_old() As Double, _
    Para_T_in() As Double, T_amb As Double, inl() As Integer, out() As Integer, _
    h_out() As Double, V_Store As Double, h_store As Double, lambda_eff As Double, _
    UA_IO() As Double, d_ins() As Double, lambda_ins() As Double, V_Fluid As Double, _
    curved As Integer, strcha As Integer, Q_Store_old As Double, Result() As Double)
' Speicherroutine welche sowohl fixe als auch Variabel Füllhöhen unterstützt

' #####
' ##### Variablen Deklaration #####
' #####
Dim cp As Double ' [J/kgK] spezifische Wärmekapazität
Dim rho As Double ' [kg/m3] Dichte Wasser
Dim pi As Double ' [-] MagicNumber Pi
Dim n_fluid As Integer ' [-] Knoten mit Fluid gefüllt
Dim mdot_In() As Double ' [kg/h] Zugeführter Massenstrom
Dim mdot_Out() As Double ' [kg/h] Abgeführter Massenstrom
Dim mdot() As Double ' [kg/h] Resultierender Massenstrom
Dim dt_internal As Double ' [s] Interner Zeitschritt
Dim A_Cross_Section As Double ' [m2] Querschnittsfläche
Dim SimTime As Double ' [s] laufende Simulationszeit
Dim T_node() As Double ' [°C] Temperatur der Speicherknoten
Dim T_new() As Double ' [°C] aktualisierte Knotentemperatur
Dim h_Nodes() As Double ' [m] Höhe des Knotens (mit Fluid gefüllt)
Dim m_Nodes() As Double ' [kg] Füllmenge des Knotens mit Fluid
Dim T_out() As Double ' [°C] Array in dem die Aus
Dim Q_Store As Double ' [J] Innere Energie im Speicher
Dim dQ_Store As Double ' [J] Änderung der Inneren Energie
Dim Qdot_bal As Double ' [kJ/h] Abweichung der Leistungsbilanz im Zeitschritt
Dim Qdot_Inlet() As Double ' [J/s] Leistung durch mdot_In und T_in im int. Zeitschritt
Dim Qdot_Vert() As Double ' [J/s] Vertikale Wärmeleitung im int. Zeitschritt
Dim Qdot_Flow() As Double ' [J/s] Wärmetransport im Speicher im int. Zeitschritt
Dim Qdot_Loss() As Double ' [J/s] Wärmeverlust an Umgebung im int. Zeitschritt
Dim Qdot_In() As Double ' [J/s] Eingetretene Leistung im Zeitschritt
Dim Qdot_In_sum As Double ' [J/s] Summe der eingetretenen Leistung im Zeitschritt
Dim Q_out() As Double ' [J] Ausgetretene Energie aus dem Speicher
Dim Qdot_Out() As Double ' [J/s] Ausgetretene Leistung im Zeitschritt
Dim Qdot_out_sum As Double ' [J/s] Summe der ausgetretenen Leistung im Zeitschritt
Dim Q_loss_sum As Double ' [J] Energieverlust durch Verluste an Umgebung
Dim Qdot_loss_sum As Double ' [J/s] Wärmeverlust an Umgebung im Zeitschritt
Dim UA_Store() As Double ' [W/K] Array mit den UA-Werten

Dim mdot_UnderRun As Double ' [kg/s] Massenstrom Fluid das nicht abgeführt werden kann
Dim mdot_OverRun As Double ' [kg/s] Massenstrom Fluid durch Überlauf
Dim V_OverRun As Double ' [m3] Volumen durch Überlauf
Dim Qdot_OverRun As Double ' [J/s] Leistung durch Überlauf
Dim Q_OverRun As Double ' [J] Enrgie durch Überlauf

Dim i As Long ' [-] Temporäre Speichervariable (Temp)
Dim j As Long ' [-] Temp
Dim k As Long ' [-] Temp
Dim l As Long ' [-] Temp

' Konstanten
cp = 4185.5
pi = 3.14159265358979
rho = 1000

' Definieren der Arrays für diesen Zeitschritt
ReDim mdot(n - 1)
ReDim mdot_In(n - 1)
ReDim mdot_Out(n - 1)
```

```
ReDim T_in(n - 1)
ReDim Qdot_Inlet(n - 1)
ReDim Qdot_Vert(n - 1)
ReDim Qdot_Flow(n - 1)
ReDim Qdot_Loss(n - 1)
ReDim T_node(n - 1)
ReDim Q_out(Outlet - 1)
ReDim T_out(Outlet - 1)
ReDim Qdot_In(Inlet - 1)
ReDim Qdot_Out(Outlet - 1)
ReDim T_new(n - 1)
ReDim h_Nodes(n - 1)
ReDim m_Nodes(n - 1)
ReDim UA_Store(n - 1)

' #####
' ##### Initialisierung Simulationsstart #####
' #####
' Laden der Startwerte
' #####
A_Cross_Section = 0
n_fluid = 0
mdot_OverRun = 0
n_fluid = 0
' #####

' Berechnet die Anzahl Fluid- Knoten, weiters die Masse und Höhe der Knoten.
' Prüft auf einen Überlauf im Zeitschritt, Berechnet die Querschnittsfläche
' Da die Input- Arrays nur temporär befüllt werden, werden sie zum Initialisieren bewusst
' später befüllt. Somit wird mit Get_Nodes() der Letztwert der Massen aus dem Füllvolumen
' berechnet. Im Simulationszeitschritt werden die Input-Arrays zuerst befüllt, damit die Knoten
' davor angepasst werden können.

' Test
If Time >= 110.533 Then
    Stop
End If

' Bei einem Überlauf/Unterlauf wird der auch der austretende Massenstrom berechnet.
Call Get_Nodes(n, n_fluid, V_Store, h_store, dt, dt_internal, V_Fluid, rho, _
    A_Cross_Section, m_Nodes(), h_Nodes(), Para_mdot_In(), _
    Para_mdot_Out(), mdot_OverRun, mdot_UnderRun, h_out(), inl(), out(), T_node())

' Kontrolle ob Speicher mehr als leerläuft
If mdot_UnderRun > 0.0001 Then
    MsgBox "Fehlendes Speichervolumen" & vbLf & "im Zeitschritt " _
        & Format(Time, "####.0") & " h", vbCritical, "ERROR"
    Stop
    Exit Sub
End If

' Beladen des Arrays mit den Knotentemperaturen des vorherigen Zeitschritts
For i = 0 To n - 1
    T_node(i) = T_old(i)
Next i

' Schichtladung
If strcha = 1 Then
    ' Verändert nur das inl() - Array
    Call Stratified_Charge(n, n_fluid, T_node(), Para_T_in(), inl())
End If

' Laden der zeitvariablen Eingabedaten
Call Get_Input_Data(n, Para_mdot_Out(), Para_mdot_In(), Para_T_in(), mdot_Out(), _
    mdot_In(), inl(), out(), mdot_OverRun)

' Berechnet die UA-Werte des Speichers inklusive der Anschlussverluste
Call UA(UA_IO(), d_ins(), lambda_ins(), V_Store, h_store, V_Fluid, n, _
```

ANHANG

```

n_fluid, h_Nodes(), A_Cross_Section, curved, UA_Store())

' Berechnen der Energie im Speicher nur einmal zur Initialisierung
' Dies könnte man auch in der Hauptroutine erledigen, da der Letztwert an diese Funktion
' übergeben wird.
' Da hier aber schon alle Größen berechnet wurden, ist es hier einfacher.
If Time = 0 Then
  For i = 0 To n_fluid - 1
    Q_Store_old = Q_Store_old + m_Nodes(i) * cp * T_old(i)
  Next i
End If

' Berechnen des resultierenden Massenfluss für die Berechnung des Wärmeflusses
Call mdot_bal(n, n_fluid, mdot_In(), mdot_Out(), mdot())

' Berechnet den internen Zeitschritt
Call Sim_dt(n, dt, n_fluid, cp, lambda_eff, A_Cross_Section, h_Nodes(), m_Nodes(), _
  mdot_In(), mdot(), UA_Store(), dt_internal)

' Runden des Teilschritts auf ein Vielfaches von dt
i = dt / dt_internal ' Integerrundung

If (i * dt_internal) < dt Then
  dt_internal = dt / (i + 1)
Else
  dt_internal = dt / i
End If

' Zum Einstellen einer Schichtung ist es nötig mehrere kleinere Zeitschritte zum Beginn
' zu berechnen. Ansonsten dauert es bei kleinen Massenströmen sehr lange.
If Time = 0 And dt_internal = dt Then
  dt_internal = dt_internal / 10
End If

' #####
' ##### Hauptschleife- interner Zeitschritt #####
' #####
Do ' main loop
  ' Leeren der temporären Arrays, sie werden nur zur Berechnung von T_new()
  ' im Simulationszeitschritt benötigt.
  For i = 0 To n - 1
    Qdot_Inlet(i) = 0
    Qdot_Flow(i) = 0
    Qdot_Vert(i) = 0
    Qdot_Loss(i) = 0
    mdot(i) = 0
  Next i

  ' Berechnet die Masse und die Höhe jedes Knotens, sowie den Füllstand, Überlauf,...
  Call Get_Nodes(n, n_fluid, V_Store, h_store, dt, dt_internal, V_Fluid, rho, _
    A_Cross_Section, m_Nodes(), h_Nodes(), Para_mdot_In(), Para_mdot_Out(), _
    mdot_OverRun, mdot_UnderRun, h_out(), inl(), out(), T_node())

  ' Anpassen der Eintrittshöhen bei Schichtladung
  If strcha = 1 Then

  ' Geschichtetes Beladen => Modifikation von inl()
  ' Bei variabler Füllhöhe ist es notwendig dies zu jedem Zeitschritt auszuführen
  Call Stratified_Charge(n, n_fluid, T_node(), Para_T_in(), inl())

  For i = 0 To n - 1
    mdot_In(i) = 0
    mdot_Out(i) = 0
    mdot(i) = 0
  Next i

  ' Erneutes Laden der zeitvariablen Eingabedaten mit neuem inl()
  Call Get_Input_Data(n, Para_mdot_Out(), Para_mdot_In(), Para_T_in(), mdot_Out(), _
    mdot_In(), inl(), out(), mdot_OverRun)

  ' Berechnen der UA-Werte im Speicher
  Call UA(UA_IO(), d_ins(), lambda_ins(), V_Store, h_store, _
    V_Fluid, n, n_fluid, h_Nodes(), A_Cross_Section, curved, UA_Store())

  ' Berechnen des resultierenden Massenfluss für die Berechnung des Wärmeflusses
  Call mdot_bal(n, n_fluid, mdot_In(), mdot_Out(), mdot())

  ' Berechnen Qdot_inlet
  Call Qdot_Inlets(n, n_fluid, cp, inl(), Para_mdot_In(), Para_T_in(), T_node(), Qdot_Inlet())

  ' Berechnen Qdot_flow
  Call Qdot_flows(n, n_fluid, cp, mdot(), T_node(), Qdot_Flow())

  ' Berechnen Qdot_vert
  Call Qdot_vertical(n, n_fluid, A_Cross_Section, lambda_eff, T_node(), h_Nodes(), _
    Qdot_Vert())

  ' Berechnen Qdot_loss
  Call Qdot_Losses(n, n_fluid, T_amb, T_node(), UA_Store(), Qdot_Loss())

  ' Aktualisieren der Knotentemperaturen
  Call T_Node_Update(n, n_fluid, cp, dt_internal, T_node(), m_Nodes(), Qdot_Inlet(), _
    Qdot_Flow(), Qdot_Vert(), Qdot_Loss(), T_new())

  ' Ausgleich der Inversen Temperaturschichten
  Call Temp_Invers(T_new(), n, n_fluid, m_Nodes())

  ' Berechnen Qdot_out, Ausgabe T_out
  ' Es wird die gesamte ausgetretene Energie während aller Zeitschritte aufsummiert
  For i = 0 To UBound(out())
    T_out(i) = T_node(out(i))
    Q_out(i) = Q_out(i) + Para_mdot_Out(i) * cp * T_out(i) * dt_internal
  Next i

  ' Volumen und Leistung durch Überlauf
  V_OverRun = V_OverRun - mdot_OverRun / rho * dt_internal
  Q_OverRun = Q_OverRun + mdot_OverRun * cp * T_node(n - 1) * dt_internal

  ' Berechnen der Verluste
  For i = 0 To n - 1
    Q_loss_sum = Q_loss_sum + Qdot_Loss(i) * dt_internal
  Next i

  ' Übergabe der neuen Temperaturen
  For i = 0 To n - 1
    T_node(i) = T_new(i)
  Next i

  SimTime = SimTime + dt_internal

  Loop Until Abs(SimTime - dt) < 0.0001
' #####
' ##### interner Zeitschritt vollständig #####
' #####
' Berechnen der Energien des Zeitschrittes
For i = 0 To UBound(inl())
  Qdot_In(i) = Para_mdot_In(i) * cp * Para_T_in(i)
  Qdot_In_sum = Qdot_In_sum + Qdot_In(i)
Next i

For i = 0 To UBound(out())
  Qdot_Out(i) = Q_out(i) / dt
  Qdot_Out_sum = Qdot_Out_sum + Qdot_Out(i)

```


ANHANG

```
n_fluid = (V_Fluid / A_Cross_Section) / h_Node
h_fluid = V_Fluid / A_Cross_Section
```

' Korrektur bei nur mehr einem Knoten

```
If (V_Fluid < V_Store / n) And (V_Fluid > 0.0000001) Then
  n_fluid = 1
End If
```

' Prüfung ob ein Unterlauf stattfindet

```
For i = 0 To UBound(h_out())
  If h_fluid < h_out(i) Then
    If Para_mdot_Out(i) < 0 Then ' Wird Auslass benutzt?
      MsgBox "Outlet above the fluid level!", vbCritical, "ERROR"
      Stop
    End If
  End If
End For
Next i
```

' Vor der Berechnung des internen Zeitschritts werden die Knoten des letzten Zeitschritts berechnet

```
If dt_internal = 0 Then
```

' Gesamte Masse des Fluids
m_Sum = V_Fluid * rho

' Bei einer Änderung werden die Knoten, das Volumen und die Temperatur modifiziert

' Es wird zuerst eine Bilanz über die Massenströme gebildet

' dm = 0 Makro wird beendet

' dm < 0 Anzahl Knoten <= Anzahl Knoten davor

' dm > 0 Anzahl Knoten >= Anzahl Knoten davor

' Grenzwert ist ein halber Knoten

' darüber wird ein neuer Knoten gebildet, darunter ein vorhandener vergrößert

```
For i = 0 To n - 1
  If (m_Sum - m_Node) >= 0 Then
    m_Nodes(i) = m_Node
    m_Sum = m_Sum - m_Node
  ElseIf m_Sum = 0 Then
    m_Nodes(i) = 0
  Else ' Ist der Fall wenn nur mehr ein Restanteil übrig
    If m_Sum > (0.5 * m_Node) Then
      ' Zur Vermeidung von Rundungsfehlern
      If Abs(m_Sum - m_Node) < (1 * 10 ^ -6) Then
        m_Sum = m_Node
      End If
      m_Nodes(i) = m_Sum
    Else ' Knoten wird verkleinert n_fluid-1
      If i = 0 Then
        m_Nodes(0) = m_Sum
      Else
        m_Nodes(i - 1) = m_Nodes(i - 1) + m_Sum
        m_Nodes(i) = 0
      End If
    End If
  End If
  m_Sum = 0
End For
Next i
```

' Berechnen der Knotenhöhen zum Simulationsstart

```
For i = 0 To n - 1
  h_Nodes(i) = m_Nodes(i) / (A_Cross_Section * rho)
Next i
```

' Überprüfen auf Unterlauf/Überlauf + Ausgabe des Massenstroms des Überlaufs

' Änderung der Masse

```
dm = 0
dV = 0
```

' Massenbilanz über alle In- und Outlets

```
For i = 0 To UBound(inl())
  dm = dm + Para_mdot_In(i)
Next i
```

```
For i = 0 To UBound(out())
  dm = dm + Para_mdot_Out(i)
Next i
```

If dm <> 0 Then ' Differenz der Massenströme

```
dm = dm * dt ' Änderung der Masse im gesamten Zeitschritt _
Umrechnung auf Massenänderung
```

```
dV = V_Fluid + (dm / rho) ' Resultierendes Volumen im Speicher
If (V_Store - dV) < 0.0000001 Then ' Wenn kleiner als 0=> Überlauf
  ' Ausgabe des mdot_Overrun
  dV = dV - V_Store ' Differenz Überzählige Volumen
  mdot_OverRun = (-dV * rho) / dt
```

```
Elseif dV < 0.0000001 Then
  mdot_UnderRun = (-dV * rho) / dt ' Massenstrom welcher in den Speicher fehlt
End If
```

```
dm = 0
dV = 0
End If
```

Else ' Hauptschleife, Zeitschritt wurde berechnet

' Berechnung der neuen Knotenmasse/ Höhe im Zeitschritt

' Änderung der Masse

```
dm = 0
```

' Massenbilanz über alle In- und Outlets

```
For i = 0 To UBound(inl())
  dm = dm + Para_mdot_In(i)
Next i
```

```
For i = 0 To UBound(out())
  dm = dm + Para_mdot_Out(i)
Next i
```

```
dm = dm + mdot_OverRun
```

```
dm = dm * dt_internal ' Änderung der Masse im Zeitschritt
```

' Berechnung der neuen Füllmenge und Höhe des letzten Knoten

If dm <> 0 Then

' Letztwerte sichern

```
m_Node_Last = m_Nodes(n_fluid - 1)
```

```
If n_fluid < 2 Then ' Nur 1 Fluid Knoten
```

```
  m_Nodes(0) = m_Node_Last + dm
```

```
Elseif (m_Node_Last + dm) > (1.5 * m_Node) Then ' Neuen Knoten erzeugen
```

```
  m_Nodes(n_fluid - 1) = m_Node
```

```
  m_Nodes(n_fluid) = m_Node_Last + dm - m_Node
```

```
  h_Nodes(n_fluid - 1) = h_Node
```

```
  h_Nodes(n_fluid) = m_Nodes(n_fluid) / (A_Cross_Section * rho)
```

```
  T_node(n_fluid) = T_node(n_fluid - 1)
```

```
Elseif dm > 0 Then ' Keiner neuer Knoten aber Füllmenge anpassen
```

```
  m_Nodes(n_fluid - 1) = m_Node_Last + dm
```

```
  h_Nodes(n_fluid - 1) = m_Nodes(n_fluid - 1) / (A_Cross_Section * rho)
```

```
Elseif (m_Node_Last + dm) < (0.5 * m_Node) Then ' Knoten löschen
```

```
  m_Nodes(n_fluid - 1) = 0
```

```
  m_Nodes(n_fluid - 2) = m_Node_Last + dm + m_Node
```

```
  h_Nodes(n_fluid - 1) = 0
```

```
  h_Nodes(n_fluid - 2) = m_Nodes(n_fluid - 2) / (A_Cross_Section * rho)
```

```
  T_Node_New = (T_node(n_fluid - 2) * m_Node + T_node(n_fluid - 1) * m_Node_Last) _
  / (m_Node + m_Node_Last)
```

ANHANG

```

For i = n_fluid - 2 To n - 1
    T_node(i) = T_Node_New
Next i
Else ' dm < 0 Füllmenge des Knoten anpassen
    m_Nodes(n_fluid - 1) = m_Node_Last + dm
    h_Nodes(n_fluid - 1) = m_Nodes(n_fluid - 1) / (A_Cross_Section * rho)
End If

' Anpassen der Parameter
V_Fluid = V_Fluid + dm / rho ' Neues Volumen, Überlauf wird in Get_Nodes abgefangen!
n_fluid = (V_Fluid / A_Cross_Section) / h_Node ' Neue Anzahl der Fluid Knoten

If (V_Fluid < V_Store / n) And (V_Fluid > 0.0001) Then
    n_fluid = 1
End If
End If
End Sub

Sub mdot_bal(n As Integer, n_fluid As Integer, mdot_In() As Double, mdot_Out() As Double, _
    mdot() As Double)
' Berechnen des resultierenden Massenfluss für die Berechnung des Wärmeflusses
Dim i As Long ' temp

i = 0
mdot(i) = mdot_In(i) + mdot_Out(i)

For i = 1 To n_fluid - 2
    mdot(i) = mdot(i - 1) + mdot_In(i) + mdot_Out(i)
Next i

If n_fluid = 1 Then ' wenn ein Knoten keine Konvektion
    mdot(0) = 0
End If

End Sub

Sub Sim_dt(n As Integer, dt As Double, n_fluid As Integer, cp As Double, lambda_eff As Double, _
    A_Cross_Section As Double, h_Nodes() As Double, m_Nodes() As Double, _
    mdot_In() As Double, mdot() As Double, UA_Store() As Double, dt_internal As Double)
' Routine berechnet den internen Zeitschritt für die Simulation
' Basierend auf der Energiebilanz an einem Knoten wird der maximale Zeitschritt berechnet
' m*cp*dT/dt=Qdot_in+Qdot_flow+Qdot_vert+Qdot_loss (NUM27)
' dt=(mdot_in*cp*dT+mdot*cp*dT+2*lambda*A/h_node*dT+UA*dT)/(m*cp*dT)
' dt=(mdot_in*cp+mdot*cp+2*lambda*A/h_node+UA)/(m*cp)
' Dem letzten Fluid-Knoten werden alle Qdot_In(i) oberhalb zugeordnet.
' Die vertikale Wärmeleitung könnte für die Knoten 1, n_fluid-1 und n_fluid genauer
' beschrieben werden. Da sich aber dadurch der Zeitschritt nur vergrößern kann wird
' er vereinfacht berechnet und dadurch ein kleiner Zeitschritt gewählt.
Dim dt_Temp As Double ' [s] momentaner interner Zeitschritt
Dim UA_Temp As Double ' [W/K] Summe Wärmeverlust Fluidknoten bis letzten Knoten
Dim mdot_In_Temp As Double ' [kg/s] Eintretender Massenstrom in letzten Fluidknoten
Dim i As Integer ' [-] tmp
Dim j As Integer ' [-] tmp

dt_internal = dt ' Zuweisung des Ausgabezeitschritts

' 1. Knoten
i = 0

' Berücksichtigen des Wärmeflusses da dieser ein Input ist
If dt_Temp < 0 Then
    dt_Temp = m_Nodes(i) * cp / (mdot_In(i) * cp + Abs(mdot(i)) * cp _
        + 2 * A_Cross_Section * lambda_eff / h_Nodes(i) + UA_Store(i))
Else
    dt_Temp = m_Nodes(i) * cp / (mdot_In(i) * cp _
        + 2 * A_Cross_Section * lambda_eff / h_Nodes(i) + UA_Store(i))
End If

If dt_Temp < dt_internal Then
    dt_internal = dt_Temp
End If

' Knoten ober dem ersten, ohne letzten Fluid-Knoten
For i = 1 To n_fluid - 2
    dt_Temp = m_Nodes(i) * cp / (mdot_In(i) * cp + Abs(mdot(i - 1)) * cp _
        + 2 * A_Cross_Section * lambda_eff / h_Nodes(i) + UA_Store(i))

    If dt_Temp < dt_internal Then
        dt_internal = dt_Temp
    End If

Next i

' Letzer Knoten
' Energie in die Luftknoten wird letzten Fluid- Knoten zugewiesen
For i = n_fluid - 1 To n - 1
    UA_Temp = UA_Temp + UA_Store(i)
    mdot_In_Temp = mdot_In_Temp + mdot_In(i)
Next i

i = n_fluid - 1
If n_fluid > 1 Then
    dt_Temp = m_Nodes(i) * cp / (mdot_In_Temp * cp + Abs(mdot(i - 1)) * cp _
        + 2 * A_Cross_Section * lambda_eff / h_Nodes(i) + UA_Store(i))
Else
    dt_Temp = m_Nodes(i) * cp / (mdot_In_Temp * cp + UA_Store(i))
End If

If dt_Temp < dt_internal Then
    dt_internal = dt_Temp
End If

End Sub

Sub Temp_Invers(T_new() As Double, n As Integer, n_fluid As Integer, m_Nodes() As Double)
' Temp_Invers dient zum Ausgleich inverser Temperaturschichten =>
' Es darf sich keine wärmere Schicht unter einer kälteren befinden
Dim T_M As Double ' [°C] Schichttemperatur*Knotenmasse
Dim T_Mittel As Double ' [°C] Mittlere Schichttemperatur
Dim m_Sum As Double ' [kg] Summe der Massen der Schichten
Dim CounT As Integer ' [-] Zähler der inversen Temperaturschichten
Dim Check As Integer ' [-] Zeiger ob Schichtung im jetzigen Zeitschritt stimmt
Dim i As Integer ' [-] temp
Dim k As Integer ' [-] temp

For i = 1 To n_fluid - 1 ' Beginnt mit 1 wegen Vergleich
    If T_new(i) < T_new(i - 1) Then ' Inverse Schicht
        CounT = CounT + 1
        Check = 0
    Else ' Temperaturschichtung passt
        Check = 1
    End If

    If i = n_fluid - 1 And CounT <> 0 Then
        i = i + 1
        Check = 1
    End If

    If Check = 1 And CounT <> 0 Then ' Ausgleich der Temperaturen
        ' Checke die Schichten bis zur letzten durch
        k = i - 1 ' Zeiger auf die Schicht mit der falschen Schichtung
        CounT = 1 ' Schleifenzeiger für Ausgabe
        m_Sum = 0 ' Gesamte Masse der Knoten
        T_M = 0 ' Temperatur*Massee zu Berechnung der mittleren Temperatur
    End If
End Sub

```

ANHANG

```

For k = i - 1 To 1 Step -1 ' Suche geht von der inversen Schicht bis zum 1. Knoten
' Laufende Mittelwertbildung + Vergleich mit dem darunterliegenden Knoten
  m_Sum = m_Sum + m_Nodes(k)
  T_M = T_M + T_new(k) * m_Nodes(k)

' Vergleiche die aktuelle Mischtemperatur der Knoten mit der Temperatur _
des darunterliegenden Knoten
If (T_M / m_Sum) > T_new(k - 1) Then
' Knotentemperatur kleiner als die Mischtemperatur => Durchmischung vollständig
  Exit For
End If
If (k - 1) = 0 Then
  m_Sum = m_Sum + m_Nodes(k - 1)
  T_M = T_M + T_new(k - 1) * m_Nodes(k - 1)
End If

  CounT = CounT + 1
Next k
' Ausgabe der mittleren Temperatur
T_Mittel = T_M / m_Sum

For k = i - 1 To i - CounT Step -1
  T_new(k) = T_Mittel
Next k

CounT = 0 ' Leeren des Schleifenzählers
End If
Next i

' Knoten über Wasser haben dieselbe Temperatur wie der letzte Wasserknoten
For i = n_fluid To n - 1
  T_new(i) = T_new(n_fluid - 1)
Next i

End Sub

Sub T_Node_Update(n As Integer, n_fluid As Integer, cp As Double, dt_internal As Double, _
  T_node() As Double, m_Nodes() As Double, Qdot_Inlet() As Double, _
  Qdot_Flow() As Double, Qdot_Vert() As Double, Qdot_Loss() As Double, _
  T_new() As Double)
' Es wird die Energiebilanz in jedem Speicherknoten aufgestellt
' Die zeitliche Änderung der inneren Energie im Speicherknoten entspricht
' der Summe der ein- und austretenden Energien.
'  $m \cdot cp \cdot dt/dt = Qdot_{in} + Qdot_{flow} + Qdot_{vert} + Qdot_{loss}$  (NUM27)
Dim i As Long ' temp

For i = 0 To n_fluid - 1
  T_new(i) = T_node(i) + (Qdot_Inlet(i) + Qdot_Flow(i) + Qdot_Vert(i) _
    + Qdot_Loss(i)) * dt_internal / (m_Nodes(i) * cp)
Next i

For i = n_fluid To n - 1
  T_new(i) = T_new(n_fluid - 1)
Next i

End Sub

Sub Qdot_flows(n As Integer, n_fluid As Integer, cp As Double, mdot() As Double, _
  T_node() As Double, Qdot_Flow() As Double)
' Berechnet den Wärmefluss durch die angrenzenden Schichten
Dim i As Long ' temp

' Negativer Wärmefluss (nach oben)
For i = 0 To n_fluid - 1
  If i > 0 Then
    If mdot(i - 1) > 0 Then
      Qdot_Flow(i) = Abs(mdot(i - 1)) * cp * (T_node(i - 1) - T_node(i))
    End If
  End If
End For

' Positiver Wärmefluss (nach unten)
If i < n_fluid - 1 Then
  If mdot(i) < 0 Then
    Qdot_Flow(i) = Qdot_Flow(i) + Abs(mdot(i)) * cp * (T_node(i + 1) - T_node(i))
  End If
End If
Next i

End Sub

Sub Qdot_vertical(n As Integer, n_fluid As Integer, A_Sect As Double, lambda_eff As Double, _
  T_node() As Double, h_Nodes() As Double, Qdot_Vert() As Double)
' Beschreibt den Effekt der vertikalen Wärmeleitfähigkeit im Speicher
Dim delta_z() As Double ' [m] Abstand der Knotentemperaturen zueinander
Dim i As Long ' [1] temp

ReDim delta_z(n - 1)

' Abstände der Knotentemperaturen
For i = 0 To n_fluid - 2
  delta_z(i) = (h_Nodes(i) + h_Nodes(i + 1)) / 2
Next i

' 1. Knoten nur Eintritt
If delta_z(0) > 0 Then
  Qdot_Vert(0) = lambda_eff * A_Sect / delta_z(0) * (T_node(1) - T_node(0))
End If

' Fluid Knoten Ein - und Austritt
For i = 1 To n_fluid - 2
  Qdot_Vert(i) = lambda_eff * A_Sect / delta_z(i) * (T_node(i + 1) - T_node(i)) _
    + lambda_eff * A_Sect / delta_z(i - 1) * (T_node(i - 1) - T_node(i))
Next i

' Letzter Fluid-Knoten nur Austritt
i = n_fluid - 1
If n_fluid > 1 Then
  Qdot_Vert(i) = lambda_eff * A_Sect / delta_z(n_fluid - 2) * (T_node(i - 1) - T_node(i))
End If

End Sub

Sub Qdot_Losses(n As Integer, n_fluid As Integer, T_amb As Double, T_node() As Double, _
  UA_Store() As Double, Qdot_Loss() As Double)
' Hier werden die Verluste des Speichers an die umgebung berechnet
Dim i As Long ' temp

For i = 0 To n_fluid - 1
  Qdot_Loss(i) = UA_Store(i) * (T_amb - T_node(i))
Next i

For i = n_fluid To n - 1
  Qdot_Loss(n_fluid - 1) = Qdot_Loss(n_fluid - 1) + UA_Store(i) * (T_amb - T_node(i))
Next i

End Sub

Sub Qdot_Inlets(n As Integer, n_fluid As Integer, cp As Double, inl() As Integer, _
  Para_mdot_In() As Double, Para_T_in() As Double, T_node() As Double, _
  Qdot_Inlet() As Double)
' Makro berechnet den Wärmeeintrag in den Speicherknoten
' Statt einer gemeinsamen Eintrittstemperatur und eines resultierenden Massenstroms _
werden die Einlässe getrennt berechnet + addiert, im letzten Fluidknoten werden _
alle darüberliegenden Ströme zusammengefasst
Dim i As Long ' temp

For i = 0 To UBound(inl())
  If inl(i) <= n_fluid - 1 Then
    Qdot_Inlet(inl(i)) = Qdot_Inlet(inl(i)) + Para_mdot_In(i) * cp * (Para_T_in(i) - T_node(inl(i)))
  End If
End For

```

ANHANG

```

Else ' above fluid level
  Qdot_Inlet(n_fluid - 1) = Qdot_Inlet(n_fluid - 1) + Para_mdot_In(i) * cp * _
    (Para_T_in(i) - T_node(in(i)))
End If
Next i

```

End Sub

```

Sub UA(UA_IO() As Double, d_ins() As Double, _
  lambda_ins() As Double, V_Store As Double, h_store As Double, V_Fluid As Double, _
  n As Integer, n_fluid As Integer, h_Nodes() As Double, A_Cross_Section As Double, _
  curved As Integer, UA_Store() As Double)

```

' Berechnen des neuen UA-Wertes für den Zwischschritt

@ Hier wird der UA-Wert der Dämmung in allen Knoten berechnet. _

Es ist möglich verschiedene Dämmstärke/ Wärmeleitfähigkeiten anzugeben _

Weiters sind folgende Eingaben möglich: _

' 1. Boden

' 2. Decke

' 3. Mantel

' Mit der Option curved ist die Angabe eines gewölbten Speichers möglich, der im Vergleich _
zu einem ebenen Boden/Decke ein größere Oberfläche aufweist

Dim pi As Double

Dim alpha_Inside_air As Double ' [W/m²K] Wärmeübergangszahl innen Luft/Wand

Dim alpha_Inside_Water As Double ' [W/m²K] Wärmeübergangszahl innen Wasser/Wand

Dim alpha_Outside As Double ' [W/m²K] Wärmeübergangszahl aussen Wand/Luft

Dim D_Innen As Double ' [m] Innerer Durchmesser des Speichers

Dim U_Bottom_Air As Double ' [W/m²K] Wärmedurchgangszahl Luft/Wand/Luft

Dim U_Bottom_Water As Double ' [W/m²K] Wärmedurchgangszahl Wasser/Wand/Luft

Dim U_Top_Air As Double ' [W/m²K] Wärmedurchgangszahl Luft/Wand/Luft

Dim U_Top_Water As Double ' [W/m²K] Wärmedurchgangszahl Wasser/Wand/Luft

Dim U_Mantle_Air As Double ' [W/m²K] Wärmedurchgangszahl Luft/Wand/Luft

Dim U_Mantle_Water As Double ' [W/m²K] Wärmedurchgangszahl Wasser/Wand/Luft

Dim h_Node As Double ' [m] Standardhöhe eines Knotens

Dim curv As Double ' [-] Faktor für Oberflächenvergrößerung

Dim i As Long ' [-] temp

' Allgemeine Konstanten

pi = 3.14159265358979

alpha_Inside_Water = 500

alpha_Inside_air = 10

alpha_Outside = 5

h_Node = h_store / n

D_Innen = (4 * (A_Cross_Section) / pi) ^ (1 / 2)

' Zusätzliche Verluste durch gewölbten Boden/Decke

If curved = 1 Then

curv = 0.3829 ' Mittelwert der eingesetzten Speicher

Else

curv = 0

End If

' Berechnen des Wärmeübergangs

U_Bottom_Water = 1 / (1 / alpha_Inside_Water + d_ins(0) * (1 + curv) / lambda_ins(0) _
+ 1 / alpha_Outside)

U_Bottom_Air = 1 / (1 / alpha_Inside_air + d_ins(0) * (1 + curv) / lambda_ins(0) _
+ 1 / alpha_Outside)

U_Top_Air = 1 / (1 / alpha_Inside_air + d_ins(1) / lambda_ins(1) + 1 / alpha_Outside)

U_Top_Water = 1 / (1 / alpha_Inside_Water + d_ins(1) / lambda_ins(1) + 1 / alpha_Outside)

U_Mantle_Air = 1 / (1 / alpha_Inside_air + d_ins(2) * (1 + curv) / lambda_ins(2) _
+ 1 / alpha_Outside)

U_Mantle_Water = 1 / (1 / alpha_Inside_Water + d_ins(2) * (1 + curv) / lambda_ins(2) _
+ 1 / alpha_Outside)

' Nur Mantelverluste + Anschlussverluste

' Wasserkonten, außer der letzte

For i = 0 To n_fluid - 2

UA_Store(i) = UA_IO(i) + D_Innen * pi * h_Node * U_Mantle_Water

Next i

' letzter Wasserkonten (variable Höhe) + erster Luftknoten

If (h_Nodes(n_fluid - 1) - h_Node) > 0.0000001 Then ' Konten größer

UA_Store(n_fluid - 1) = UA_IO(n_fluid - 1) + D_Innen * pi * h_Nodes(n_fluid - 1) * U_Mantle_Water

If n_fluid = n Then

Else

UA_Store(n_fluid) = UA_IO(n_fluid) + D_Innen * pi * (2 * h_Node - h_Nodes(n_fluid - 1)) * U_Mantle_Air

End If

Else ' Knoten kleiner

UA_Store(n_fluid - 1) = UA_IO(n_fluid - 1) + D_Innen * pi * h_Nodes(n_fluid - 1) * U_Mantle_Water _

+ (h_Node - h_Nodes(i)) * U_Mantle_Air

If n_fluid = n Then

Else

UA_Store(n_fluid) = UA_IO(n_fluid) + D_Innen * pi * h_Node * U_Mantle_Air

End If

End If

' restliche Luftknoten

For i = n_fluid + 1 To n - 1

UA_Store(i) = UA_IO(i) + D_Innen * pi * h_Node * U_Mantle_Air

Next i

' + Boden

If V_Fluid < 0.0000001 Then ' Speicher leer

UA_Store(0) = UA_Store(0) + A_Cross_Section * U_Bottom_Air

Else

UA_Store(0) = UA_Store(0) + A_Cross_Section * U_Bottom_Water

End If

' + Decke

If Abs(V_Store - V_Fluid) < 0.0000001 Then ' Speicher voll

UA_Store(n - 1) = UA_Store(n - 1) + A_Cross_Section * U_Top_Water

Else

UA_Store(n - 1) = UA_Store(n - 1) + A_Cross_Section * U_Top_Air

End If

End Sub

Sub Stratified_Charge(n As Integer, n_fluid As Integer, T_node() As Double, _
Para_T_in() As Double, in() As Integer)

' Schichtladung => Eingeschichtet wird mit minimalen Exergieverlust=> _

Wenn ein Knoten gefunden wird, dessen Temperatur größer ist als die des Fluids, _

dann wird in den darunterliegenden Knoten eingeschichtet. _

Sollte der ganze Speicher eine niedrigere Temperatur als das Fluid aufweisen, _

so wird in den letzten Fluid- Knoten eingespeist.

Dim i As Long ' temp

Dim j As Long ' temp

For j = 0 To UBound(in()) ' Alle Inlets

For i = 1 To n - 1 ' Alle Knoten

If T_node(i) > Para_T_in(j) Then

' Schichtet in dem Knoten ein, wo die Temperatur noch unter dem oberen ist

in(j) = i - 1

Exit For

Elsel (n_fluid - 1) = i Then

in(j) = i + 1 ' Einschichten in den letzten Knoten

Exit For

End If

Next i

Next j

End Sub

ANHANG

```

    l = l + 1
    If l = UBound(Para, 2) + 1 Then
        l = 0
    End If
End If
Next i
k = k + 1
Loop
Close #1

#####
##### Input #####
|-----|-----|-----|
| Time   | Input_1 | Input_n |
|-----|-----|-----|
| Einheit| Einheit | Einheit |
|-----|-----|-----|

' Bestimmung der Größe des Datenarrays
strFile = "J:\Input.txt"
strFile = ActiveWorkbook.Path & "\Input.txt"
Open strFile For Input As #1
k = 0 ' Zeilenzeiger
l = 0 ' temporärer Spaltenzeiger
j = 0 ' Spaltenzeiger

Do While Not EOF(1)
Line Input #1, strTemp
'Trennen der Daten
AstrTemp = Split(strTemp, " ")
For i = 0 To UBound(AstrTemp, 1)
    If Not AstrTemp(i) = "" Then ' Wenn Wert
        l = l + 1
        If j < l Then
            j = l
        End If
    End If
End For
Next i
l = 0 ' Rücksetzen des Spaltenzählers
k = k + 1
Loop
Close #1

' Dimensionieren des Daten-Arrays
ReDim Data(k - 1, j - 1)

' Kopieren der Daten
Open strFile For Input As #1
k = 0
l = 0
Do While Not EOF(1) ' End of File
Line Input #1, strTemp
'Trennen der Daten
AstrTemp = Split(strTemp, " ")
For i = 0 To UBound(AstrTemp, 1)
    If Not AstrTemp(i) = "" Then ' Wenn Wert
        Data(k, l) = AstrTemp(i)
        l = l + 1
        If l = UBound(Data(), 2) + 1 Then
            l = 0
        End If
    End If
End For
Next i
k = k + 1
Loop
Close #1

#####

```

```

##### T_init #####
|-----|-----|
| T_Knoten0 |
|-----|-----|
| T_Knoten1 |
|-----|-----|
| T_Knoten2 |
|-----|-----|

' Bestimmung der Größe des Datenarrays
strFile = "J:\T_init.txt"
strFile = ActiveWorkbook.Path & "\T_init.txt"
Open strFile For Input As #1
k = 0 ' Zeilenzeiger
l = 0 ' temporärer Spaltenzeiger
j = 0 ' Spaltenzeiger

Do While Not EOF(1)
Line Input #1, strTemp
'Trennen der Daten
AstrTemp = Split(strTemp, " ")
For i = 0 To UBound(AstrTemp, 1)
    If Not AstrTemp(i) = "" Then ' Wenn Wert
        l = l + 1
        If j < l Then
            j = l
        End If
    End If
End For
Next i
l = 0 ' Rücksetzen des Spaltenzählers
k = k + 1
Loop
Close #1

' Dimensionieren des Daten-Arrays
ReDim T_init(k - 1, j - 1)

' Kopieren der Daten
Open strFile For Input As #1
k = 0
l = 0
Do While Not EOF(1)
Line Input #1, strTemp
'Trennen der Daten
AstrTemp = Split(strTemp, " ")
For i = 0 To UBound(AstrTemp, 1)
    If Not AstrTemp(i) = "" Then ' Wenn Wert
        T_init(k, l) = AstrTemp(i)
        l = l + 1
        If l = UBound(T_init(), 2) + 1 Then
            l = 0
        End If
    End If
End For
Next i
k = k + 1
Loop
Close #1

#####

##### Initialisierung der Startwerte #####

' Konstanten
cp = 4185.5 ' spezifische Wärmekapazität J/kg.K
pi = 3.14159265358979

' Übertragen der zeitvarianten Parameter (Input aus txt oder EXCEL, beides möglich)
Time = Para(0, 1) ' min Simulationszeit
dt = Para(1, 1) ' Ausgabezeitschritt

```

ANHANG

```

n = Para(2, 1)      ' Anzahl der Speicherknoten
Inlet = Para(3, 1)  ' Anzahl der Inlets
Outlet = Para(4, 1) ' Anzahl der Outlets
curved = Para(5, 1) ' Gewölbter Boden? (0=NEIN; 1=JA)
strcha = Para(6, 1) ' Geschichtetes Beladen? (0=NEIN; 1=JA)

V_Store = Para(8, 1) ' m³ Volumen
h_store = Para(9, 1) ' m Höhe des Speichers
s_store = Para(16, 1) ' mm Wandstärke des Speichers
lambda_s = Para(17, 1) ' W/mK Wärmeleitfähigkeit St37
T_sens = Para(4 * Inlet + 3 * Outlet + 20, 1) ' Anzahl der Temperatursensoren im Tank

' Berechnen der Simulationsdauer
' Dynamische Berechnung in Abhängigkeit von der Einheit
If Para(0, 2) = "h" Then
    Time = Para(0, 1)
Elseif Para(0, 2) = "min" Then
    Time = Para(0, 1) / 60
End If

time_out = Time * 3600 / dt ' Anzahl der Ausgabewerte

' ##### Berechnung lambda_eff #####

' Mittlere Temperatur
If Para(18, 1) = -100 Then ' Variable Starttemperatur
    For i = 0 To UBound(T_init())
        T_M = T_M + T_init(i, 0)
    Next i

    T_M = T_M / (UBound(T_init()) + 1) ' °C Mittlere Temperatur des Speichers
Else ' Konstante Starttemperatur
    T_M = Para(18, 1)
End If

' Setzen der Standardwerte, falls leer
If s_store = 0 Then
    s_store = 9.6 ' mm Wandstärke des Speichermantels
End If
If lambda_s = 0 Then
    lambda_s = 50 ' W/mK Wärmeleitfähigkeit Stahl (St37)
End If

A_store_q = V_Store / h_store
d_store = (A_store_q * 4 / pi) ^ 0.5
A_store_s = pi / 4 * ((d_store + s_store * 2 / 1000) ^ 2 - d_store ^ 2) ' Querschnittsfläche Speicherwand m²
' aus Wärmeübertragung Grundlagen und Praxis; Anhang S235
lambda_w = 0.56112 * (T_M / 100) ^ 0 + 0.18825 * (T_M / 100) ^ 1 + 0.03255 * (T_M / 100) ^ 2 _
    - 0.23117 * (T_M / 100) ^ 3 + 0.15512 * (T_M / 100) ^ 4 - 0.00988 * (T_M / 100) ^ 5 _
    - 0.01697 * (T_M / 100) ^ 6

' aus VO Solarthermie - effektive Wärmeleitfähigkeit (überschlagsmäßig)
lambda_eff = (A_store_q * lambda_w + A_store_s * lambda_s) / (A_store_s + A_store_q)

' ##### Dimensionieren der Arrays nach Knoten/ANSchlüssen #####
' ### Eingabewert - Arrays ###
ReDim in(Inlet - 1)
ReDim out(Outlet - 1)
ReDim Para_mdot_In(Inlet - 1)
ReDim Para_mdot_Out(Outlet - 1)
ReDim Para_T_in(Inlet - 1)
ReDim h_in(Inlet - 1)
ReDim h_out(Outlet - 1)
ReDim h_sens(T_sens - 1)
ReDim T_old(n - 1)

```

```

ReDim UA_IO(n - 1)
ReDim d_ins(2)
ReDim lambda_ins(2)

' ### Ausgabewert - Arrays ###
' Global für Simulationsdauer
ReDim Save_Data(time_out + 2, n + Inlet + Outlet * 2 + 6 + T_sens) ' Array besitzt genau eine Zeile mehr für die
Startwerte
' Local für Simulationszeitschritt
ReDim Result(n + Outlet * 2 + Inlet + 6)

' #####

' Belegung des Ein- und Austrittsknoten-Arrays

For i = 0 To Inlet - 1
    h_in(i) = Para(2 * Inlet + Outlet + 20 + i, 1) ' Höhe Inlet1
Next i

For i = 0 To Outlet - 1
    h_out(i) = Para(3 * Inlet + 1 * Outlet + 20 + i, 1) ' Höhe Outlet1
Next i

' Umrechnen der Ein-Austrittshöhen auf Knotenposition in den Arrays 0 Basiert
dz = h_store / n ' Knotenhöhe

zi = 0
For j = 0 To Inlet - 1
    zi = 0
    For i = 0 To n - 1
        zi = zi + dz
        If h_in(j) <= zi Or i = n - 1 Then ' Inlet im Knoten ...
            in(i) = i
            Exit For
        End If
    Next i
Next j

For j = 0 To Outlet - 1
    zi = 0
    For i = 0 To n - 1
        zi = zi + dz
        If h_out(j) < zi Or i = n - 1 Then ' Outlet im Knoten ...
            out(i) = i
            Exit For
        End If
    Next i
Next j

' Umrechnen der Sensorhöhen auf Knotenposition in den Arrays - Base0
For j = 0 To UBound(h_sens())
    zi = 0
    For i = 0 To n - 1
        zi = zi + dz
        If Para(4 * Inlet + 3 * Outlet + 21 + j, 1) <= zi Or i = n - 1 Then ' Inlet im Knoten ...
            h_sens(j) = i
            Exit For
        End If
    Next i
Next j

' Zusätzliche Wärmeverluste aus Parameter
For i = 0 To Inlet - 1
    UA_IO(in(i)) = UA_IO(in(i)) + Para(3 * Inlet + 2 * Outlet + 20 + i, 1)
Next i
For i = 0 To Outlet - 1
    UA_IO(out(i)) = UA_IO(out(i)) + Para(4 * Inlet + 2 * Outlet + 20 + i, 1)
Next i

```


ANHANG

```

    Para_T_in(i) = Data(time_tmp + 2, Inlet + Outlet + 2 + i) ' °C
    ' Para_T_in(i) = Data(time_tmp + 2, 22 + i) ' °C
Else ' Fixwert
    Para_T_in(i) = Para(20 + i, 1) ' °C
End If
Next i

' Umgebungstemperatur aus Input-File oder Parameter
'-100 => variabel aus txt, sonst aus Parameter
If Para(19, 1) = -100 Then ' Variabel
    T_amb = Data(time_tmp + 2, Inlet + Outlet + 1) ' °C
    ' T_amb = Data(time_tmp + 2, 21) ' °C
Else ' Fixwert
    T_amb = Para(19, 1) ' °C
End If

' Leeren des Ausgabe-Arrays Result() für den diesen Zeitschritt
For i = 0 To UBound(Result())
    Result(i) = 0
Next i

' #####
Call Core_Var(Timestep, n, dt, Inlet, Outlet, Para_mdot_In(), Para_mdot_Out(), T_old(), _
    Para_T_in(), T_amb, in(), out(), h_out(), V_Store, h_store, lambda_eff, _
    UA_IO(), d_ins(), lambda_ins(), V_Fluid, curved, strocha, Q_Store_old, Result())
' #####

' Wenn ein leeres Array zurückgegeben wird, war der austretende Volumen, aus dem Speicher, größer
' als das Volumen des Speichers und der eintretende Volumenstrom!

time_tmp = time_tmp + 1 ' nächster Wert in der Spalte
Timestep = Timestep + (dt / 3600)

' Speichern des Zeitschritts
Save_Data(2 + time_tmp, 0) = Timestep

' Speichern des Zwischenergebnis
For i = 1 To (UBound(Result()) - 2)
    Save_Data(2 + time_tmp, i) = Result(i)
Next i

For i = UBound(Save_Data(), 2) - 1 To UBound(Save_Data(), 2)
    Save_Data(2 + time_tmp, i) = Result(i - T_sens)
Next i

If time_tmp = 1 Then ' Im ersten Zeitschritt wird Q_store-old berechnet
    Save_Data(2, 2 + n + Outlet * 2 + Inlet) = Q_Store_old '###
End If
Loop Until time_tmp >= time_out

' #####
' ##### Ende Simulationsschleife #####
' #####

' ##### Qdot_bal #####
Application.StatusBar = "Bereite Ausgabe vor..."

j = 0
k = Outlet + Inlet + 1

For j = n + Outlet + 1 To n + Outlet + k
    sum = 0
    For i = 3 To UBound(Save_Data(), 1)
        sum = sum + Save_Data(i, j)
    Next i
    Save_Data(2, j) = sum
Next j

```

```

'Überlauf
k = T_sens + 5 + n + Outlet * 2 + Inlet
For j = k To k + 1
    sum = 0
    For i = 3 To UBound(Save_Data(), 1)
        sum = sum + Save_Data(i, j)
    Next i
    Save_Data(2, j) = sum
Next j

sum = 0

k = Outlet + Inlet + 1

For j = n + Outlet + 1 To n + Outlet + k
    sum = sum + Save_Data(2, j)
Next j

sum = sum + Save_Data(2, n + Inlet + Outlet * 2 + 6 + T_sens)

k = 2 + n + Outlet * 2 + Inlet

Save_Data(2, k + 1) = sum - (Save_Data(UBound(Save_Data(), 1), k) - Save_Data(2, k)) / dt

'
' ##### Ausgabe T_Sensor #####
Application.StatusBar = "Ausgabe SensorTemperaturen"
' Speichert die betreffenden Temperaturen der Knoten

If T_sens > 0 Then
    k = 5 + n + Outlet * 2 + Inlet ' Spalte
    For j = 2 To UBound(Save_Data(), 1) ' alle Zeilen( Zeitschritte)
        l = 10
        For i = 0 To UBound(h_sens()) ' alle SensorTemp
            Save_Data(j, k + i) = Save_Data(j, h_sens(i) + 1)
        Next i
    Next j
End If

' ##### OUTPUT #####
KnotenTemp = False
' Öffnen der Datei
' strFile = "J:\output.txt"
Application.StatusBar = "Ausgabe Output.txt"
strFile = ActiveWorkbook.Path & "\output.txt"

Open strFile For Output As #1

For k = 0 To UBound(Save_Data(), 1)
    strTemp = ""
    For l = 0 To UBound(Save_Data(), 2)
        If KnotenTemp = True Then ' Knotentemp abspeichern
            strTemp = strTemp + CStr(Save_Data(k, l)) + " "
        Else ' Keine Knotentemp abspeichern
            If l = 0 Or l > n Then
                strTemp = strTemp + CStr(Save_Data(k, l)) + " "
            End If
        End If
    Next l
    strTemp = Left(strTemp, (Len(strTemp) - 1)) ' Löscht dass nun überzählige Leerzeichen
    Print #1, strTemp ' |Wert1|space|Wert2|space|Wert3|
Next k

Close #1

```

ANHANG

```
' #####  
Application.StatusBar = False  
  
End Sub  
  
Sub Import_Store()  
' Makro Importiert selbst erstellte TRNSYS.out- files und fügt diese in das aktive Sheet ein  
' Weiters wird der erste Zeitschritt überprüft und falls ChangeTime = True geändert  
' Einfügebereich kann im neuen Sheet angegeben werden  
' Verknüpfungen werden alle gelöscht  
Dim rngZelle As Range  
Dim Ws As Worksheet  
Dim Dateiname As Variant  
Dim ChangeTime As Boolean  
Dim UserInput As Boolean  
Dim Search_Replace As Boolean  
Dim SkipLine As Integer  
Dim Source As Boolean  
Dim Timestep As Double  
Dim intzeile As Long  
Dim intSpalte As Long  
Dim i As Long 'Temporäre Speichervariable  
Dim temp As String 'Temporäre Stringvariable  
Dim Target As String  
Dim T_Zeile As Integer  
Dim T_Spalte As String  
Dim T_Spalte_N As Integer  
Dim myCom As Object  
Dim kommentar As Comment  
  
Application.ScreenUpdating = False  
Application.DisplayAlerts = False  
  
' #####  
ChangeTime = False ' Ändert die Werte auf Sekunden/Minutenwerte  
UserInput = False ' Vorgabe des Eingabebereichs, False = A1  
Source = True ' Einfügen eines Kommentars mit Quelle und Datum  
SkipLine = 0 ' Zeilen werden beim Einfügen übersprungen  
' #####  
  
' Aktive Zelle als ersten Bezug setzen  
T_Zeile = ActiveCell.Row  
T_Spalte_N = ActiveCell.Column  
T_Spalte = ItoA(T_Spalte_N)  
  
' Abfrage des Einfügebereichs  
If UserInput = True Then  
Target = InputBox("Angabe des Einfügebereichs..." & vbLf & vbLf & "Wenn kein Wert eingegeben wird, " _  
& vbLf & "wird als Ziel die Zelle " & T_Spalte & T_Zeile & " gewählt, "Einfügebereich")  
End If  
  
' Wenn nichts ausgewählt oder InputBox= False dann immer A1  
If Target = "" Then  
Target = T_Spalte & T_Zeile  
Target = "A7"  
End If  
  
' Trennung Spalte/Zeile  
For i = 1 To Len(Target)  
If Not IsNumeric(Right(Target, i)) Then  
T_Spalte = Left(Target, Len(Target) - (i - 1))  
T_Spalte_N = Atol(T_Spalte)  
T_Zeile = Right(Target, i - 1)  
Exit For  
End If  
Next i  
  
Set Ws = ActiveSheet
```

```
' Löscht den alten Bereich  
Rows("7:13000").Delete Shift:=xlUp  
  
Dateiname = Application.GetOpenFilename("Textdateien, *.out; *.plt; *.txt; *.xtxt")  
  
If Dateiname <> False Then  
' Ab hier Unterscheidung txt-File, oder out-File  
temp = FileNameOnly(Dateiname, ".") ' Endung  
If temp = "txt" Then 'txt-File Leerzeichen trennt  
Workbooks.OpenText Filename:=Dateiname, DataType:=xlDelimited, _  
Space:=True, DecimalSeparator=".", ThousandsSeparator=""  
  
ElseIf temp = "out" Then ' TRNSYS- out-File  
Workbooks.OpenText Filename:=Dateiname, DataType:=xlDelimited, _  
Tab:=True, Space:=False, DecimalSeparator=".", ThousandsSeparator=""  
  
ElseIf temp = "plt" Then ' TRNSYS- plt-File  
Workbooks.OpenText Filename:=Dateiname, DataType:=xlDelimited, _  
Tab:=True, Space:=False, DecimalSeparator=".", ThousandsSeparator=""  
  
Else 'txt-Files mit Semikolon als Trennzeichen  
Workbooks.OpenText Filename:=Dateiname, DataType:=xlDelimited, _  
Semicolon:=True, Space:=False, DecimalSeparator=".", ThousandsSeparator=""  
End If  
Cells.NumberFormat = "0.00000"  
  
'Löschen der 1.Zeile (Überschrift)  
If SkipLine <> 0 Then  
ActiveSheet.Rows(SkipLine).Delete xlShiftUp  
End If  
  
ActiveSheet.UsedRange.Copy Ws.Cells(T_Zeile, T_Spalte_N) ' Kopiert die Daten ins Sheet an Position "A1"  
ActiveWorkbook.Close ' Schließt das temporäre Sheet  
Else  
Exit Sub  
End If  
  
' Wieder im alten Tabellenblatt weiter...  
With ActiveSheet  
If Source = True Then  
' Zuerst alle alten Kommentare im Sheet löschen  
  
For Each kommentar In .Comments  
.Comments(1).Delete  
Next  
  
' Neues Kommentar einfügen  
Set myCom = .Cells(T_Zeile, T_Spalte_N).AddComment  
With myCom  
.Visible = False  
.Text Text:=Dateiname & vbLf & "Eingelesen am: " & Now  
.Shape.TextFrame.Characters.Font.Name = "Arial"  
.Shape.TextFrame.Characters.Font.FontStyle = "standard"  
.Shape.TextFrame.Characters.Font.Size = 8  
.Shape.Height = 23  
.Shape.Width = 450  
End With  
End If  
  
If ChangeTime = True Then  
  
intzeile = ActiveSheet.UsedRange.Rows.Count  
  
Timestep = 3600 *.Cells(4, 1).Value  
Timestep = Round(Timestep, 3)  
Select Case Timestep  
Case Is < 60
```

ANHANG

```

.Cells(2, 1).Value = "SECOND"
Case Is < 3600
.Cells(2, 1).Value = "MINUTE"
Timestep = Timestep / 60
Case Else
Timestep = Timestep / 3600
End Select

For i = 4 To intzeile
.Cells(i, 1).Value = .Cells(i - 1, 1).Value + Timestep
Next i
End If

'Formatieren der Werte
Cells.NumberFormat = "0.00000"
Columns("A:A").NumberFormat = "General"
.Columns.AutoFit

' Löscht alle Verbindungen zu Data-Files im Excel-File, falls vorhanden
If ActiveWorkbook.Connections.Count > 0 Then
For i = 1 To ActiveWorkbook.Connections.Count
ActiveWorkbook.Connections.Item(i).Delete
Next i
End If
End With

Set Ws = Nothing
Application.DisplayAlerts = True
Application.ScreenUpdating = True
End Sub
Sub Export_Store() 'v1.1
'Makro exportiert das ganze Tabellenblatt im csv-Format
'Zusätzlich können die Werte Formatiert werden, Standard "0000.0000"
'Weiters kann das Trennzeichen verändert werden
'Das Datum(SpalteA) wird automatisch ausgeschlossen, da sonst Nummernsalat ;-)
'Die 1. Zeile wird auf Leerzeichen überprüft
Dim rngZelle As Range
Dim Zeile As Range
Dim Zelle As Range
Dim s As String
Dim Ws As Worksheet
Dim txt_Dateiname As String
Dim intzeile As Long
Dim intSpalte As Integer
Dim i As Long
Dim imax As Long
Dim Trennzeichen As String
Dim Datum As Boolean
Dim Formatierung As String
Dim k As Long 'Temporäre Variable
Dim Start As Long 'Startpunkt Datum
Dim FilterVar As String
Dim temp As String 'Temporärer String
Dim temp1 'Temporärer String
Dim vx() As Double 'String für Platzhalter (0)

Application.ScreenUpdating = True

'Workbooks.Open FileName:=Dateiname
'Sheets(SheetName).Activate
Set Ws = ActiveSheet

'### Eingabewerte ###
Trennzeichen = ""
'Formatierung = "0000.0000000"
'#####

intzeile = Ws.UsedRange.Rows.Count
intSpalte = Ws.UsedRange.Columns.Count

With Application
.UseSystemSeparators = False
.DecimalSeparator = "."
.ThousandsSeparator = ""
End With

' Ist Datum in Spalte A vorhanden?
For k = 1 To intzeile
If IsDate(Ws.Cells(k, 1).Value) Then
Exit For
End If
Next k

If Trennzeichen = "" Then ' Falls Trennzeichen kein Leerzeichen dann weiter
For k = 1 To intSpalte 'Suche nach Leerzeichen und Vergleich der StringLänge
temp = Cells(1, k).Value
temp1 = FileNameOnly(temp, Trennzeichen)
If Len(temp1) < Len(temp) Then
MsgBox "Trennzeichen in der Zelle " & Itoa(k) & "1 vorhanden" & _
vbLf & "Werte können falsch eingelesen werden", vbCritical, "TRNSYS-DataReaderExport")
Exit Sub
End If
End If
Next k

If k = intzeile + 1 Then 'Kein Datum vorhanden
.Cells.NumberFormat = Formatierung
Else 'Mit Datumsspalte
Range(Cells(1, 2), Cells(intzeile, intSpalte)).NumberFormat = Formatierung
End If

Set rngZelle = Ws.UsedRange

If Trennzeichen = "" Then
FilterVar = "Textdateien (*.txt;*.dat), *.txt"
Elseif Trennzeichen = ";" Then
FilterVar = "Textdateien (*.txt;*.xxt), *.xxt"
Else
MsgBox "Kein zulässiges Trennzeichen" & vbLf & "Leerzeichen für txt" & vbLf & "Semikolon für xxt", vbCritical,
"Trennzeichen"
Application.UseSystemSeparators = True 'Zurücksetzen der Einstellung
Exit Sub ' Falls anderer Separator eingegeben
End If

If txt_Dateiname = "" Then
ChDir ActiveWorkbook.Path 'Vorgabe des Standard- Ordners
txt_Dateiname = Application.GetOpenFilename(FilterVar)
'Definiert die Einlesedatei
End If

'#####

If txt_Dateiname = "Falsch" Then
MsgBox "Keine Datei ausgewählt" & vbLf & "Batch wird beendet!", vbCritical, "Export txt-File"
Application.UseSystemSeparators = True 'Zurücksetzen der Einstellung
Exit Sub
End If

If Trennzeichen = ";" And FileNameOnly(txt_Dateiname, ".") = "txt" Then
Name txt_Dateiname As (Left(txt_Dateiname, Len(txt_Dateiname) - 3) & "txt")
' Falls bei der Endung noch txt-Steh
txt_Dateiname = (Left(txt_Dateiname, Len(txt_Dateiname) - 3) & "xxt")
End If

'##### Bereich füllen - Platzhalter #####
ReDim vx(intzeile) 'Dimensionierung des Arrays
For i = 0 To UBound(vx, 1)

```

ANHANG

```

vx(i) = "0"
Next i
'Wenn Zeile3 leer ist, fügt es 0 als Platzhalter ein
For i = 1 To intSpalte
    If IsEmpty(Cells(3, i).Value) Then
        Ws.Range(Cells(3, i), Cells(intzeile, i)).Value = vx
    End If
Next i
'
#####

Open txt_Dateiname For Output As #1

For Each Zeile In rngZelle.Rows

For Each Zelle In Zeile.Cells
s = s & Zelle.Text & Trennzeichen
Next Zelle

s = Left(s, Len(s) - 1)
Print #1, s
s = ""

Next Zeile

Close #1

Application.UseSystemSeparators = True

'ActiveWorkbook.Close SaveChanges:=False

Application.ScreenUpdating = True
End Sub
Public Function Atol(strBuchstaben As String)
' Funktion kann eine Buchstabenkombination mir 3 Buchstaben
' in eine Zahl umwandeln, zB.Eingabefeld übergibt wert an range
' Public Function, damit sie auch in Tabellenblatt verfügbar ist
Dim intNummer As Integer
Dim intNummer_2 As Integer
If Len(strBuchstaben) = 1 Then ' Ein Buchstabe
    intNummer = Asc(strBuchstaben)
    If intNummer >= 97 Then ' Kleinbuchstabe
        intNummer = intNummer - 96
    Else ' Großbuchstabe
        intNummer = intNummer - 64
    End If
Elseif Len(strBuchstaben) = 2 Then ' Zwei Buchstaben
    intNummer = Asc(Left(strBuchstaben, 1)) ' 1. Buchstabe
    If intNummer >= 97 Then
        intNummer = (intNummer - 96) * 26
    Else
        intNummer = (intNummer - 64) * 26
    End If
    intNummer_2 = Asc(Right(strBuchstaben, 1)) ' 2. Buchstabe
    If intNummer_2 >= 97 Then
        intNummer = intNummer + intNummer_2 - 96
    Else
        intNummer = intNummer + intNummer_2 - 64
    End If
Else ' Drei Buchstaben
    intNummer = Asc(Left(strBuchstaben, 1)) ' 1. Buchstabe
    If intNummer >= 97 Then
        intNummer = (intNummer - 96) * 676
    Else
        intNummer = (intNummer - 64) * 676
    End If
    intNummer_2 = Asc(Right(Left(strBuchstaben, 2), Len(Left(strBuchstaben, 2)) - 1))
' Befehl MID(variable, startwert von links, anzahl der zeichen)

```

```

If intNummer_2 >= 97 Then
    intNummer = intNummer + (intNummer_2 - 96) * 26
Else
    intNummer = intNummer + (intNummer_2 - 64) * 26
End If
intNummer_2 = Asc(Right(strBuchstaben, 1))
If intNummer_2 >= 97 Then
    intNummer = intNummer + (intNummer_2 - 96)
Else
    intNummer = intNummer + (intNummer_2 - 64)
End If
End If
If intNummer > 16384 Then
    MsgBox "Spaltenlimit Excel 16384 = XFD!", vbCritical
End If
Atol = intNummer
End Function

Public Function ltoA(intColumn As Variant) As String ' Änderung des intcolumns von integer auf variant
' Berechnet aus einer Zahl den entsprechenden Buchstabencode für die Excel-Spaltendarstellung
' Quelle:http://www.office-loesung.de/ftopic343198_0_0_asc.php

Dim strLetter As String
Dim strLetter2 As String
Dim strLetter3 As String

' erste Stelle
strLetter = Chr((64 + intColumn) - Int(intColumn / 26) * 26)
If strLetter = "@" Then strLetter = "Z"

' zweite Stelle
' ab 703 fängt "AAA" an
If intColumn < 703 Then
    strLetter2 = Chr(64 + (((Int((intColumn - 1) / 26) * 26) / 26) - 26 * (Int(((intColumn - 1) / 702))))))
Else
    strLetter2 = Chr(64 + (((Int((intColumn - 1) / 26) * 26) / 26) - 26 * (Int(((intColumn - 1) - 702) / 676) + 1)))
End If
If strLetter2 = "@" Or intColumn < 27 Then strLetter2 = ""

' dritte Stelle
If intColumn >= 703 Then strLetter3 = Chr(64 + Int(((intColumn - 1) - 702) / 26) / 26) + 1)

ltoA = strLetter3 & strLetter2 & strLetter
End Function

Function FileNameOnly(SuchString, Zeichen As String) As String ' vs2.4
' 1.Parameter: der zuzurchsuchende String
' 2.Parameter: das zu suchende Zeichen
' Funktion sucht den String von rechts nach links durch
' Wenn das Zeichen gefunden wurde gibt die Funktion den SuchString bis zum Zeichen zurück
' Wenn das Zeichen nicht gefunden wurde gibt die Funktion den gesamten SuchString zurück
Dim i As Integer
Dim Length As Integer
Dim sTr As String

Length = Len(SuchString)
sTr = ""

For i = Length To 1 Step -1
    If Mid(SuchString, i, 1) = Zeichen Then
        FileNameOnly = sTr
        Exit Function
    End If
    sTr = Mid(SuchString, i, 1) & sTr
Next i
FileNameOnly = SuchString
End Function

```

ANHANG

Anhang - F : EXCEL - Makros

```

Option Explicit
Private Function DateToSec(EndTime As Variant, StartTime As Variant) As Long
' Funktion errechnet aus 2 Datum+Zeitschritten den Sekundenwert zurück
Dim Timestep As Date
Dim AnzahlTage As Long
Dim AnzahlDaten_SOLL As Long

    If IsDate(EndTime) Then ' And IsDate(StartTime) Then
        Timestep = CDate(Format(EndTime, "TT.MM.YYYY hh:mm:ss")) - CDate(Format(StartTime, "TT.MM.YYYY
hh:mm:ss"))
        Timestep = EndTime - StartTime ' berechneter Zeitschritt
    Else
        Timestep = CDate(Format(EndTime, "TT.MM.YYYY hh:mm:ss")) - CDate(Format(StartTime, "TT.MM.YYYY
hh:mm:ss"))
    End If

    AnzahlTage = DateDiff("d", StartTime, EndTime)
' Reine Datumsdifferenz- ohne Berücksichtigung der Uhrzeit !!!

    If Format(EndTime, "hh:mm:ss") < Format(StartTime, "hh:mm:ss") Then
        AnzahlTage = AnzahlTage - 1
    End If

    DateToSec = AnzahlTage * 24 * 3600 + TimeToSek(CStr(Format(Timestep, "hh:mm:ss"))) ' Zeitschritt in sec
End Function

Private Function TimeToSek(time As String) As Long
' Funktion errechnet aus einem String mit dem Format "hh:mm:ss" den Sekundenwert
TimeToSek = (Left(time, 2) * 3600 + Mid(time, 4, 2) * 60 + Right(time, 2) * 1) * 60(sec)*5(min) = Sekunden
End Function

Sub Repair_3()
' Makro fügt fehlende Zeitschritte in einen Datenbereich ein, welcher zuvor in das Array kopiert wurde.
Dim i As Long ' temporäre Zählvariable
Dim j As Long ' Zeilenzeiger für fehlende Zeitschritte
Dim k As Long ' Zeilenzeiger Array
Dim l As Long ' Spaltenzeiger Array
Dim m As Long ' Zeilenzeiger für idente Zeitschritte (00:00)
Dim n As Long ' Zeilenzeiger für Zwischenzeitschritte
Dim o As Long ' temporäre Speichervariable

Dim ZZ As Long ' Zeiger auf fehlerhaften Zeitstempel
Dim LZ As Long ' Zeiger auf Leere Zeilen
Dim a As Integer ' Rückgabewert MsgBox
Dim Modulo As Long ' Ergebnis des Modulo-Operators
Dim SpalteN As Long
Dim Spalte As String
Dim ZeileN As Long
Dim AnzahlTage As String
Dim anzahlTage_i As Long
Dim Timestep As Date
Dim LastDate As Long
Dim Timestep_s As String
Dim AnzahlDaten As Long ' Anzahl der einzufügenden Daten
Dim AnzahlDaten_SOLL As Long ' Benötigte Daten - in Sekunden
Dim AnzahlDaten_IST As Long ' Vorhandene Daten - in Sekunden
Dim AnzahlDaten_Loss As Long ' Fehlende Daten - in Sekunden
Dim Fehler As Long ' Anzahl von fehlenden Daten
Dim vx As Variant ' Array
Dim vx_Data As Variant ' Array mit Nullen
Dim wSq As Worksheet ' Tabellenblatt mit Ausgangsdaten
Dim Zeitschritt As String ' Zeitschritt
Dim Zeitschritt_i As Long ' Zeitschritt in Sekunden

```

```

Dim datName As String
Dim wsZ As Worksheet

Application.ScreenUpdating = False
Application.DisplayAlerts = False
#####
Zeitschritt = "00:05:00" ' Zeitschritt der Daten
Spalte = "d" ' Angabe der Spalte mit Datum + Zeit
#####

Set wSq = ActiveSheet

' Befüllen des Ausgangswert-Array
ZeileN = wSq.UsedRange.Rows.Count ' Zeilenzeiger, beginnt mit 1
SpalteN = wSq.UsedRange.Columns.Count ' Spaltenzeiger, beginnt mit 1

vx = wSq.Range(Cells(1, 1), Cells(ZeileN, SpalteN)) ' Zeilen/Spalten beginnt mit 1, da aus EXCEL-Tab - Base 1

' Speichern der Datumsspalte in SpalteN
SpalteN = Atol(Spalte)

' Ermittelt ersten/letzten Datum/Zeitschritt
For i = 1 To UBound(vx, 1)
    If IsDate(vx(i, SpalteN)) Then
        ' Übergibt i als Zeiger für erste Spalte mit Datum
        Exit For
    End If
Next i

' Wenn die falsche Spalte angegeben ist...
If i > UBound(vx, 1) Then
    MsgBox "Es wurde in der Spalte " & Spalte & " kein Datum/Zeitschritt" & vbLf & "im Format: TT.MMYYYY
hh:mm:ss gefunden" & vbLf & "Makro wird abgebrochen", vbCritical, "Fehler"
Exit Sub
End If

For j = UBound(vx, 1) To i Step -1
    If IsDate(vx(j, SpalteN)) Then
        ' Übergibt j als Zeiger auf die letzte Spalte
        LastDate = j
        Exit For
    End If
Next j

' Kontrolle des Zeitschritts mittels des ersten Datensatzes
If Not CDate(Zeitschritt) = Format(vx(i + 1, SpalteN) - vx(i, SpalteN), "hh:mm:ss") Then

    a = MsgBox("Es wurde eine Abweichung im Zeitschritt festgestellt" & vbLf & vbLf & "Voreingestellte Zeitschritt : "
& Zeitschritt & vbLf & "Zeitschritt der ersten beiden Datenreihen : " & CStr(Format(vx(i + 1, SpalteN) - vx(i, SpalteN),
"hh:mm:ss")) & vbLf & "Ja => Voreingestellter Zeitschritt" & vbLf & "Nein => Zeitschritt aus Daten", vbYesNoCancel,
"Zeitschritt")

    If a = vbYes Then

        ElseIf a = vbNo Then ' Zeitschritt wird geändert
            Zeitschritt = CStr(Format(vx(i + 1, SpalteN) - vx(i, SpalteN), "hh:mm:ss"))
        Else
            Exit Sub
        End If
    End If

    Zeitschritt_i = TimeToSek(Zeitschritt)

' Ermittle maximale Anzahl der Zeitschritte des Arrays

AnzahlDaten_SOLL = DateToSec(vx(j, SpalteN), vx(i, SpalteN))
AnzahlDaten = AnzahlDaten_SOLL / Zeitschritt_i ' Anzahl der Datenreihen

```

ANHANG

```

' Vergleich der vorhanden Daten
AnzahlDaten_IST = (UBound(vx, 1) - i) * Zeitschritt_i

AnzahlDaten_Loss = (AnzahlDaten_SOLL - AnzahlDaten_IST)
Fehler = AnzahlDaten_Loss / Zeitschritt_i

' Dimensionieren des neuen Arrays - Base 0
ReDim vx_Data(UBound(vx) + Fehler + 3, UBound(vx, 2) - 1)

' Befüllen des Arrays mit vollständigen Zeitschritt
' Kopieren der Überschriften + 1. Zeitschritt
For l = 0 To UBound(vx, 2) - 1
    For k = 0 To i - 1
        vx_Data(k, l) = vx(k + 1, l + 1)
    Next k
Next l

' main
j = 0 ' Offset Zeilenzähler positiv - fehlende Datenreihen
m = 0 ' Offset Zeilenzähler negativ - Datenreihen mit gleichem Zeitstempel
n = 0 ' Offset Zeilenzähler negativ - Datenreihen ausserhalb des Zeitstempels
LZ = 0 ' Zeiger auf leere Zeilen
ZZ = 0 ' Zeiger auf fehlerhaften Zeitstempel

For k = i To LastDate - 1
    o = 0 ' Temporärer Offset Zeilenzähler Ausgangsarray
    o = k
    If o = 136061 Then
        o = 0
    End If
    o = 0
    If IsDate(vx(k, SpalteN)) And IsDate(vx(k + 1, SpalteN)) Then ' Datenreihen folgen aufeinander

        AnzahlDaten_SOLL = DateToSec(vx(k + 1, SpalteN), vx(k, SpalteN))

        If AnzahlDaten_SOLL = Zeitschritt_i Then ' Zeitschritt korrekt

            For l = 0 To UBound(vx_Data, 2)
                vx_Data(k + j - m - n, l) = vx(k + 1, l + 1)
            Next l

        Else ' Zeitschritt stimmt nicht

            If AnzahlDaten_SOLL = 0 Then ' Gleicher Zeitstempel
                ' Einfügen der Datenreihen auf den Platz der vorherigen (-1)
                For l = 3 To UBound(vx_Data, 2)
                    vx_Data(k - 1 + j - m - n, l) = vx(k + 1, l + 1)
                Next l
                m = m + 1

            Else ' Datenreihen mit fehlerhaftem Zeitstempel
                ' Finden des nächsten richtigen Zeitstempels
                For i = 1 To UBound(vx, 1)

                    o = o + 1 ' Anzahl der Übersprungenen Zeitschritte im vx-array
                    AnzahlDaten_SOLL = DateToSec(vx(k + i, SpalteN), vx(k, SpalteN))
                    Modulo = AnzahlDaten_SOLL Mod Zeitschritt_i
                    If Modulo = 0 Then
                        Exit For
                    End If

                Next i

                ZZ = ZZ + o - 1

            AnzahlDaten = AnzahlDaten_SOLL / Zeitschritt_i - 1 ' Anzahl der fehlenden Zeitschritte

```

```

' Einfügen des fehlenden Zeitschrittes
For i = 1 To AnzahlDaten
    ' Spalte D - Datum+Zeitspalte
    vx_Data(k + j - m - n, SpalteN - 1) = vx(k, SpalteN) + CDate(Zeitschritt) * i
    ' Spalte C - Zeitspalte
    vx_Data(k + j - m - n, SpalteN - 2) = Format((vx(k, SpalteN) + (CDate(Zeitschritt) * i)), "hh:mm")
Uhrzeit

' Spalte B - Datumsspalte
vx_Data(k + j - m - n, SpalteN - 3) = CDate(Format(vx(k, SpalteN) + CDate(Zeitschritt) * i,
"DD.MM.YYYY")) ' Datum
' Fehlende Daten durch 0 ersetzen
For l = SpalteN To UBound(vx_Data, 2)
    vx_Data(k + j - m - n, l) = 0
Next l
j = j + 1 ' Damit erst eine Zeile später im Ziel-Array wieder begonnen wird zu schreiben
Next i

' Und jetzt des ersten Fehlerfreien Zeitschrittes
For l = 0 To UBound(vx_Data, 2)
    vx_Data(k + j - m - n, l) = vx(k + o, l + 1)
Next l

If AnzahlDaten = 0 Then ' Fehler Innerhalb ersten Zeitschritt

Else
    k = k + o - 1
    n = n + o - 1
End If
End If
End If

ElseIf IsEmpty(vx(k, SpalteN)) And IsDate(vx(k + 1, SpalteN)) Then ' Auf Leerzeile folgt Datenreihe
' Nur Einfügen der Datenreihe

For l = 0 To UBound(vx_Data, 2)
    vx_Data(k + j - m - n, l) = vx(k + 1, l + 1)
Next l

Else ' zB. Leerzeile folgt Leerzeile
LZ = LZ + 1
End If

Next k

' Ausgabe der modifizierten Daten
If m = 0 And ZZ = 0 And j = 0 Then
    MsgBox " Zeitschritt fehlerfrei!", vbInformation, "CheckTimeStep"
Else
    MsgBox j & " fehlende Zeitschritte" & vbLf & m & " doppelte Zeitschritte" & vbLf & ZZ & " Zeitschritte <>
Zeitstempel", vbCritical, "Abweichungen"

    If Not LZ = 0 Then
        MsgBox " Es wurden " & LZ & " leere Zellen gefunden" & vbLf & vbLf & "Eventuell dient sie zum Trennen eines
Zeitbereichs???" & vbLf & vbLf & "Ansonsten die leeren Zellen löschen" & vbLf & "und erneuten Durchlauf starten",
vbQuestion, "Leere Zellen"
    End If

    a = MsgBox("Ausgabe der korrigierten Daten?", vbYesNo, "Output")

End If

If a = vbYes Then
    ' Erstellen eines neuen Tabellenblattes
    datName = wSq.Name & "_mod"

    For Each wsZ In ActiveWorkbook.Worksheets
        If wsZ.Name = datName Then

```

ANHANG

```

a = MsgBox("Tabellenblatt wird gelöscht!", vbYesNo, "Warnung")
If a = vbYes Then
    Sheets(datName).Delete ' Löscht bereits vorhandene Datei
End If
End If
Next

Sheets.Add.Name = (datName) ' Fügt neues Tabellenblatt ein

' Ausgabe des Array in das neue Tabellenblatt
Sheets(datName).Range("A1").Resize(UBound(vx_Data, 1) - LBound(vx_Data, 1) + 1, UBound(vx_Data, 2) -
LBound(vx_Data, 2) + 1) = vx_Data
End If

Application.ScreenUpdating = True
Application.DisplayAlerts = True
End Sub

Sub timesteps_gleinst()
' Makro überprüft ob bei jedem Zeitschritt eine Minute Differenz
' Der erste Zeitschritt wird als Muster herangezogen!!
' Fehlt eine Zeile wird diese am Ende per msgbox ausgegeben
' und auch durch einfärben des Bereichs markiert.
' Implementierung der Erkennung von großen Bereichen und getrennte Ausgabe

' Requires Atol(); IToA(); - funktion

Application.ScreenUpdating = False
Dim rngZelle As Range
Dim zeit_vorher As Date ' Vorheriger Wert
Dim zeit_jetzt As Date ' Jetziger Wert
Dim intZeile As Long
Dim s As String
Dim i As Long ' Temporäre Laufvariable
Dim wS As Worksheet
Dim rngBereich As Range ' Range der zu überprüfen ist
Dim t As Long ' Temporärer String
Dim Timestep As Date
Dim Drift As Date
Dim min As String
Dim h As String
Dim sec As String
Dim time2 As String
Dim Zeitschritt As String
Dim ZeitschrittSuche As Boolean
Dim Spalte As String
Dim SpalteN As Long
Dim last_TimeStep As Date
Dim Min_Drift As String
Dim Repair As Boolean
Dim value_range As String
Dim ColorIndex As Integer

' Workbooks.Open Filename:=Dateiname
' Sheets(SheetName).Activate
Set wS = ActiveSheet
'### Eingabewerte ###
Spalte = "C"
Zeitschritt = "00:02:00"
ZeitschrittSuche = False
Min_Drift = "00:15:00" ' Grenze ab der Abweichung ausgegeben wird
Repair = True
ColorIndex = 3 ' Indexfarbe fürs Einfärben
'#####

intZeile = wS.UsedRange.Rows.Count
SpalteN = Atol(Spalte) ' Konvertieren der Spalte
' Suche des Zeitschritts

```

```

For i = 1 To intZeile
    If IsDate(Cells(i, SpalteN)) Then
        t = Cells(i, SpalteN).Row ' t = erste Datumszeile
        Exit For
    End If
Next i

' Auswahl des Bereichs
value_range = ItoA(Cells(t, SpalteN).Column) & Cells(t, SpalteN).Row & ":" & _
ItoA(Cells(intZeile, SpalteN).Column) & Cells(intZeile, SpalteN).Row
Set rngBereich = Range(value_range)

If ZeitschrittSuche = True Then
' Überprüfung welcher Zeitschritt vorhanden ist
' Hierzu wird der 1. Zeitschritt betrachtet
Timestep = Cells(t + 1, SpalteN).Value - Cells(t, SpalteN).Value

sec = Right(Timestep, 2) ' Wenn sec als Integer definiert => statt "01" wird 01
min = Mid(Timestep, 4, 2)
h = Left(Timestep, 2)
'time1 = Format(timestep, "hh:mm:ss")
Zeitschritt = h & ":" & min & ":" & sec
End If

' Suche des letzten Zeitschritts einer Stunde

' Auftrennen des Zeitschritts
h = Left(Zeitschritt, 2)
min = Mid(Zeitschritt, 4, 2)
sec = Right(Zeitschritt, 2)
t = sec + min * 60 + h * 3600 ' Zeitschritt in Sekunden
t = 86400 - t ' Letzer Zeitschritt vor Überlauf

' Berechnung der Stellen
h = t \ 3600 ' \ => Ganzzahldivision!!
t = t Mod 3600 ' Bestimmung des Rest
min = t \ 60
t = t Mod 60
sec = t

last_TimeStep = h & ":" & min & ":" & sec

' Jetzt wird das gesamte Blatt überprüft
t = 0 ' Starterkennung
i = 0 ' Zähler für Abweichungen
Drift = 0

For Each rngZelle In rngBereich
    If rngZelle.Value <> "" And t <> 0 And IsDate(rngZelle.Value) Then ' Makro geht alle Zeilen durch

        zeit_jetzt = Format(rngZelle.Value, "hh:mm:ss") ' Minutenwert in der Variable
        Timestep = zeit_jetzt - zeit_vorher ' Vergleich mit Vorgabe
        time2 = Format(Timestep, "hh:mm:ss") ' Mögliche Abweichung nun in time2
' Abfrage des Tagsprungs
'dayJump=
' Achtung der Tagsprung wird nicht berücksichtigt! Abfangen beim Rücksetzen, mehrmalige Unterbrechung...?

        If time2 <> Zeitschritt Then ' Vergleich IST mit SOLL - Zeitschritt
            If zeit_vorher = last_TimeStep Then ' Suche Letzer Zeitschritt
                Else
                    If s = Empty Then
                        s = s & " Abweichungen >= " & Min_Drift & vbLf & vbLf
                        s = s & "Datum - Uhrzeit - Abweichung" & vbLf
                    End If

                    Drift = Drift + Timestep
                    i = i + 1
                    If Timestep >= Min_Drift Then

```

ANHANG

```

s = s & rngZelle.Value & " - " & time2
s = s & vbLf
End If
rngZelle.Interior.ColorIndex = ColorIndex ' Und mit Farbe rot markieren
End If
End If
End If

If t = 0 And IsDate(rngZelle.Value) Then ' Wird nur einmal beim Start durchlaufen
t = 1
zeit_jetzt = Format(rngZelle.Value, "hh:mm:ss")
End If

zeit_vorher = zeit_jetzt
Next rngZelle

If Repair = True Then
'Call Repair_data(value_range, ColorIndex, Zeitschritt) ' Makro das den Zeitschritt repariert
End If

If s = Empty Then
s = s & " Zeitschritt fehlerfrei" & vbLf
MsgBox s, vbInformation, "Abweichung"
Else
s = s & vbLf & "Zeitschritt: " & Zeitschritt & vbLf
s = s & vbLf & i & " Abweichungen vom Zeitschritt"
s = s & vbLf & "Abweichungsdauer: " & Drift
MsgBox s, vbCritical, "Abweichung"
End If

'ActiveWorkbook.Close SaveChanges:=True
Application.ScreenUpdating = True
End Sub

Sub Repair_data()
' Makro fügt fehlende Zeitschritte in einen Datenbereich ein, welcher zuvor in das Array kopiert wurde.
Dim i As Long
Dim k As Long ' Zeilenzeiger
Dim l As Long ' Spaltenzeiger
Dim j As Long ' Zeilenfortschritt für Modifikation
Dim SpalteN As Long
Dim Spalte As String
Dim ZeileN As Long
Dim AnzahlTage As String
Dim anzahlTage_i As Long
Dim Timestep As Date
Dim Timestep_s As String
Dim AnzahlDaten_SOLL As Long ' Benötigte Daten - in Sekunden
Dim AnzahlDaten_IST As Long ' Vorhandene Daten - in Sekunden
Dim AnzahlDaten_Loss As Long ' Fehlende Daten - in Sekunden
Dim Fehler As Long ' Anzahl von fehlenden Daten
Dim vx As Variant ' Array
Dim vx_Data As Variant ' Array mit Nullen
Dim wSq As Worksheet ' Tabellenblatt mit Ausgangsdaten
Dim Zeitschritt As String ' Zeitschritt
Dim datName As String
Dim wSz As Worksheet

'#####
Zeitschritt = "00:02:00" ' Zeitschritt der Daten
Spalte = "c" ' Angabe der Spalte mit Datum + Zeit
'#####

Set wSq = ActiveSheet

' Dimensionierung des Arrays
ZeileN = wSq.UsedRange.Rows.Count ' Zeilenzeiger, beginnt mit 1
SpalteN = wSq.UsedRange.Columns.Count ' Spaltenzeiger, beginnt mit 1

```

```

' Kopiert das gesamte Daten-Bereich in das Array
vx = wSq.Range(Cells(1, 1), Cells(ZeileN, SpalteN)) ' Zeilen/Spalten beginnt mit 1, da aus EXCEL-Tab

SpalteN = Atoi(Spalte)

' Ermittelt ersten Datumsschritt
For i = 1 To UBound(vx, 1)
If IsDate(vx(i, SpalteN)) Then
' Übergibt i als Zeiger für erste Spalte mit Datum
Exit For
End If
Next i

' Ermittle maximale Anzahl der Zeitschritte des Arrays
Timestep = vx(UBound(vx, 1), SpalteN) - vx(i, SpalteN)
Timestep_s = Format(Timestep, "hh:mm:ss")
AnzahlTage = DateDiff("d", vx(i, SpalteN), vx(UBound(vx, 1), SpalteN))
anzahlTage_i = AnzahlTage

AnzahlDaten_SOLL = anzahlTage_i * 24 * 3600 + TimeToSek(Timestep_s) ' Zeitschritt in sec

' Vergleich der vorhanden Daten
AnzahlDaten_IST = (UBound(vx, 1) - i) * TimeToSek(Zeitschritt)

AnzahlDaten_Loss = (AnzahlDaten_SOLL - AnzahlDaten_IST)
Fehler = AnzahlDaten_Loss / TimeToSek(Zeitschritt)

' Dimensionieren des neuen Arrays
ReDim vx_Data(UBound(vx) - 1 + Fehler, UBound(vx, 2) - 1)

' Befüllen des Arrays mit vollständigen Zeitschritt
' Kopieren der Überschriften + 1. Zeitschritt
For l = 0 To UBound(vx, 2) - 1
For k = 0 To i - 1
vx_Data(k, l) = vx(k + 1, l + 1)
Next k
Next l

' Beginn des Vergleichs
' Falls Leerzeilen eingefügt sind - wie bei Gleinstätten Daten
j = 0 ' Temporärer Offset Zeilenzähler

For k = i To UBound(vx, 1) - 1
If IsDate(vx(k, SpalteN)) And IsDate(vx(k + 1, SpalteN)) Then ' 2 Datumsangaben hintereinander
Timestep = vx(k + 1, SpalteN) - vx(k, SpalteN) ' berechneter Zeitschritt

If Not CStr(Format(Timestep, "hh:mm:ss")) = Zeitschritt Then ' Hier der Fall wenn Zeitschritt ungleich
Fehler = (TimeToSek(CStr(Timestep)) / TimeToSek(Zeitschritt)) - 1 ' fehlende Zeitschritte

' Einfügen des fehlenden Zeitschritts
For i = 1 To Fehler
' Spalte D - Datum+Zeitspalte
vx_Data(k + j, SpalteN - 1) = Format((vx(k, SpalteN) + (CDate(Zeitschritt) * i)), "DD.MM.YYYY") & " " &
"hh:mm:ss")
' Spalte C - Zeitspalte
vx_Data(k + j, SpalteN - 2) = Format((vx(k, SpalteN) + (CDate(Zeitschritt) * i)), "hh:mm:ss") ' Uhrzeit
' Spalte B - Datumsspalte
vx_Data(k + j, SpalteN - 3) = Format((vx(k, SpalteN) + (CDate(Zeitschritt) * i)), "DD.MM.YYYY") ' Datum

' Fehlende Daten durch 0 ersetzen
For l = SpalteN To UBound(vx_Data, 2)
vx_Data(k + j, l) = 0
Next l
j = j + 1 ' Damit erst eine Zeile später im Ziel-Array wieder begonnen wird zu schreiben
Next i
End If

```

ANHANG

```
' Einfügen des Originalbereichs
For l = 0 To UBound(vx_Data, 2)
    vx_Data(k + j, l) = vx(k + 1, l + 1)
Next l
Elseif IsDate(vx(k + 1, SpalteN)) Then
    ' Einfügen des Originalbereichs
    For l = 0 To UBound(vx_Data, 2)
        vx_Data(k + j, l) = vx(k + 1, l + 1)
    Next l
End If
Next k
j = 0
' Erstellen eines neuen Tabellenblattes
datName = wSq.Name & "_mod"

For Each wsZ In ActiveWorkbook.Worksheets
    If wsZ.Name = datName Then
        Sheets(datName).Delete ' Löscht bereits vorhandene Datei
    End If
Next

Sheets.Add.Name = (datName) ' Fügt neues Tabellenblatt ein

' Ausgabe des Array in das neue Tabellenblatt
Sheets(datName).Range("A1").Resize(UBound(vx_Data, 1) - LBound(vx_Data, 1) + 1, UBound(vx_Data, 2) - LBound(vx_Data, 2) + 1) = vx_Data

Application.ScreenUpdating = True
End Sub

Sub Import()
' Makro Importiert selbst erstellte TRNSYS.out- files und fügt diese in das aktive Sheet ein
' Weiters wird der erste Zeitschritt überprüft und falls ChangeTime = True geändert
' Einfügebereich kann im neuen Sheet angegeben werden
' Verknüpfungen werden alle gelöscht
Dim rngZelle As Range
Dim wS As Worksheet
Dim dateiname As Variant
Dim ChangeTime As Boolean
Dim UserInput As Boolean
Dim Search_Replace As Boolean
Dim SkipLine As Integer
Dim Source As Boolean
Dim Timestep As Double
Dim intZeile As Long
Dim intSpalte As Long
Dim i As Long ' Temporäre Speichervariable
Dim temp As String ' Temporäre Stringvariable
Dim Target As String
Dim T_Zeile As Integer
Dim T_Spalte As String
Dim T_Spalte_N As Integer
Dim myCom As Object
Dim kommentar As Comment

Application.ScreenUpdating = False
Application.DisplayAlerts = False

#####
ChangeTime = False ' Ändert die Werte auf Sekunden/Minutenwerte
UserInput = True ' Vorgabe des Eingabebereichs, False = A1
Source = True ' Einfügen eines Kommentars mit Quelle und Datum
SkipLine = 0 ' Zeilen werden beim Einfügen übersprungen
#####
```

```
' Aktive Zelle als ersten Bezug setzen
T_Zeile = ActiveCell.Row
T_Spalte_N = ActiveCell.Column
T_Spalte = Itoa(T_Spalte_N)

' Abfrage des Einfügebereichs
If UserInput = True Then
    Target = InputBox("Angabe des Einfügebereichs..." & vbCrLf & vbCrLf & "Wenn kein Wert eingegeben wird, _" & vbCrLf & "wird als Ziel die Zelle " & T_Spalte & T_Zeile & " gewählt", "Einfügebereich")
End If

' Wenn nichts ausgewählt oder InputBox= False dann immer A1
If Target = "" Then
    Target = T_Spalte & T_Zeile
End If

' Trennung Spalte/Zeile
For i = 1 To Len(Target)
    If Not IsNumeric(Right(Target, i)) Then
        T_Spalte = Left(Target, Len(Target) - (i - 1))
        T_Spalte_N = Atoi(T_Spalte)
        T_Zeile = Right(Target, i - 1)
        Exit For
    End If
Next i

Set wS = ActiveSheet

dateiname = Application.GetOpenFilename("Textdateien, *.out; *.plt; *.txt; *.xxt; *.dat")

If dateiname <> False Then
    ' Ab hier Unterscheidung txt-File, oder out-File
    temp = FileNameOnly(dateiname, ".") ' Endung
    If temp = "txt" Then ' txt-File Leerzeichen trennt
        Workbooks.OpenText Filename:=dateiname, DataType:=xlDelimited, _
            Space:=True, DecimalSeparator:=".", ThousandsSeparator:=""
    Elseif temp = "out" Then ' TRNSYS- out-File
        Workbooks.OpenText Filename:=dateiname, DataType:=xlDelimited, _
            Tab:=True, Space:=False, DecimalSeparator:=".", ThousandsSeparator:=""
    Elseif temp = "plt" Then ' TRNSYS- plt-File
        Workbooks.OpenText Filename:=dateiname, DataType:=xlDelimited, _
            Tab:=True, Space:=False, DecimalSeparator:=".", ThousandsSeparator:=""
    Else
        ' txt-Files mit Semikolon als Trennzeichen
        Workbooks.OpenText Filename:=dateiname, DataType:=xlDelimited, _
            Semicolon:=True, Space:=False, DecimalSeparator:=".", ThousandsSeparator:=""
    End If
    Cells.NumberFormat = "0.00000"

    ' Löschen der 1.Zeile (Überschrift)
    If SkipLine <> 0 Then
        ActiveSheet.Rows(SkipLine).Delete xlShiftUp
    End If

    ActiveSheet.UsedRange.Copy wS.Cells(T_Zeile, T_Spalte_N) ' Kopiert die Daten ins Sheet an Position "A1"
    ActiveWorkbook.Close ' Schließt das temporäre Sheet
Else
    Exit Sub
End If

' Wieder im alten Tabellenblatt weiter...
With ActiveSheet
    If Source = True Then
        ' Zuerst alle alten Kommentare im Sheet löschen

        For Each kommentar In .Comments
```

ANHANG

```

.Comments(1).Delete
Next

'Neues Kommentar einfügen
Set myCom = .Cells(T_Zeile, T_Spalte_N).AddComment
With myCom
.Visible = False
.Text Text:=dateiname & vbLf & "Eingelesen am: " & Now
.Shape.TextFrame.Characters.Font.Name = "Arial"
.Shape.TextFrame.Characters.Font.FontStyle = "standard"
.Shape.TextFrame.Characters.Font.Size = 8
.Shape.Height = 23
.Shape.Width = 450
End With
End If

If ChangeTime = True Then

intZeile = ActiveSheet.UsedRange.Rows.Count

Timestep = 3600 * .Cells(4, 1).Value
Timestep = Round(Timestep, 3)
Select Case Timestep
Case Is < 60
.Cells(2, 1).Value = "SECOND"
Case Is < 3600
.Cells(2, 1).Value = "MINUTE"
Timestep = Timestep / 60
Case Else
Timestep = Timestep / 3600
End Select

For i = 4 To intZeile
.Cells(i, 1).Value = .Cells(i - 1, 1).Value + Timestep
Next i
End If

'Formatieren der Werte
Cells.NumberFormat = "0.00000"
Columns("A:A").NumberFormat = "General"
.Columns.AutoFit

'Löscht alle Verbindungen zu Data-Files im Excel-File, falls vorhanden
If ActiveWorkbook.Connections.Count > 0 Then
For i = 1 To ActiveWorkbook.Connections.Count
ActiveWorkbook.Connections.Item(i).Delete
Next i
End If
End With

Set wS = Nothing
Application.DisplayAlerts = True
Application.ScreenUpdating = True
End Sub

Sub Phase_1()
'Das Makro wird aus 2 Datenbereichen, wovon eines als Ausgangspunkt angenommen wird,
einen Min, Avg, Max -Wert ermitteln welcher aus dem Zeitschritt der 1. Spalte t1 und den Zeitschritt -
t1-60min bis t1+60min gebildet wird ermitteln. Diese Werte werden dann in in Array gespeichert,
wovon sie dann ausgewertet werden könne. Die Auswertung wird mit MsgBox, Tabellenblatt und Diagramm erfolgen
'Hiezu wird eine Spalte als Ausgangspunkt definiert eine weitere als Zeitfolger
'Bei der 2-ten Spalte müssen die Werte von zB.:60 min vorher vorhanden sein, andernfalls wird abgebrochen
'Der Datenbereich wird zuerst fix definiert und in weiterer Folge als Abfrage in einer User-Form implementiert,

'Funktion Atol muss vorhanden sein

Dim intZeile As Long '!!! intger nur bis 32 767 werte, long 2 147 483 647 !!!
Dim OutSp As Integer 'Ausgangsspalte

```

```

Dim TSSp As Integer 'Zeitfolgerspalte
Dim TSMIn As Date 'Startzeit Ausgangsspalte
Dim TSMMax As Date 'Endzeit Ausgangsspalte
Dim OutDateSp As Integer 'Spalte mit dem Zeitschritt des Ausgangspunkt
Dim TSDateSp As Integer 'Spalte mit dem Zeitschritt des Zeitfolgers
Dim TSMInZeile As Long 'Zeile in dem der Zeitschritt steht
Dim TSMMaxZeile As Long
Dim OUTMinZeile As Long 'Zeile in dem der Zeitschritt steht
Dim OUTMaxZeile As Long
Dim Values As Long 'Anzahl der Spalten im Array, beginnend mit 0
Dim timesteps As Long 'Anzahl der Zeilen im Array, beginnend mit 0
Dim k As Long '1Dimension in vx bzw temporäre Variable
Dim l As Long '2Dimension in vx bzw temporäre Variable
Dim x As Date
Dim y As Variant
Dim vx As Variant 'Array mit Temperaturen in der 1.Dim, und der Zeit in der 2.Dim
Dim vx_2 As Variant 'Array mit den deltaT-Werten in der 1.Dim und der Zeit i der 2.Dim
Dim vx_Min As Variant 'Array mit den Min-Werten für Diagramm
Dim vx_Max As Variant 'Array mit Max-Werten
Dim vx_Avg As Variant 'Array mit Durchschnittswerten
Dim vx_Time As Variant 'Array mit dem jeweiligen Zeitschritt
Dim wSq As Worksheet 'Tabellenblatt mit Ausgangsdaten
Dim wsZ As Worksheet 'Tabellenblatt mit Ausgabewerten/diagrammen
Dim i As Long 'temporäre Laufvariable
Dim delta As Long 'Anzahl der Zeitschritte vor/ zurück
Dim min As Double 'Temporärer Minimalwert des Arrays
Dim max As Double 'Temporärer Maximalwert des Arrays
Dim avg As Double 'Temporärer Durchschnittswert des Arrays
Dim result As Double 'Speicher für den berechneten Zeitschritt in min
Dim objChart As ChartObject 'Zum Ansprechen des Diagramms
Dim Ausgabe As Boolean 'Wenn True dann werden Werte in das Tabellenblatt"Array" ausgegeben
Dim UserDefined As Boolean 'Wenn True dann wird der vom User definierte Bereich vorgegeben
Dim Dia_Sheet As Boolean 'Diagramm wird als Sheet eingefügt
Dim s As String 'Temporäre Stringvariable z.B.:Überschrift im Diagramm
Dim OutDatName As String 'Adresse der QuellDatei
Dim OutTabName As String 'Name des Tabellenbalttes
Dim TSDatName As String 'Adresse der zu überprüfenden Datei
Dim TSTabName As String 'Name des Tabellenblattes

```

```

Application.ScreenUpdating = False
Application.StatusBar = True
Application.StatusBar = "Beginn des Makros"

```

```

'Bereich fällt bei User-Form dann weg
OutSp = Atol("g") 'Ausgangsdaten in Spalte L
TSSp = Atol("b") 'deltaT Daten in Spalte F
OutDateSp = Atol("e") 'Datum der Ausgangsdaten in Spalte J
TSDateSp = Atol("a") 'Datum der deltaT -Werte in Spalte A
TSMIn = "03.08.2011 17:00" 'Timestep
TSMMax = "10.08.2011 06:00"
delta = 160 'Anzahl der deltaT- Werte
Ausgabe = False
UserDefined = True ****fehlt noch***

```

```

'Übergabewerte aus User Form *****
OutSp = Atol(Diag.txt_quell_value)
OutDateSp = Atol(Diag.txt_quell_date)
OutDatName = Diag.txt_quell_datName
OutTabName = Diag.cbo_quell_value
TSSp = Atol(Diag.txt_check_value)
TSDateSp = Atol(Diag.txt_check_date)
TSDatName = Diag.txt_check_datName
TSTabName = Diag.cbo_check.value
TSMIn = Diag.txt_start_date & " " & Diag.txt_start_time
TSMMax = Diag.txt_end_date & " " & Diag.txt_end_time
delta = Diag.txt_delta 'Anzahl der deltaT- Werte
Ausgabe = Diag.chk_out_value

```

ANHANG

```

*****
Set wSq = Sheets("Verschub_Tank1_mod_p1h")
Set wsZ = Sheets("Array")

' Worksheets.Open Filename:=OutDatName
' Sheets(OutTabName).Activate
' Set wsQ = ActiveSheet

' Überprüfung ob Datei schon geöffnet
' If OutDatName <> TSDatName Then
'     Worksheets.Open Filename:=TSDatName
' End If
' Sheets(TSTabName).Activate
' Set wsZ = ActiveSheet

intZeile = wSq.UsedRange.Rows.Count

' Code sucht die entsprechenden Zeitschritt im Code und gibt die Zeile zurück
If UserDefined = True Then
    For i = 1 To intZeile ' für Ausgangspunkt
        If Format(wSq.Cells(i, OutDateSp).Value, "dd.mm.yyyy hh:mm") = Format(TSMIn, "dd.mm.yyyy hh:mm") Then
            OUTMinZeile = i
        End If
        If Format(wSq.Cells(i, OutDateSp).Value, "dd.mm.yyyy hh:mm") = Format(TSMMax, "dd.mm.yyyy hh:mm") Then
            OUTMaxZeile = i
        End If
    Next i

    For i = 1 To intZeile ' für Zeitfolger
        If Format(wSq.Cells(i, TSDateSp).Value, "dd.mm.yyyy hh:mm") = Format(TSMIn, "dd.mm.yyyy hh:mm") Then
            TSMInZeile = i
        End If
        If Format(wSq.Cells(i, TSDateSp).Value, "dd.mm.yyyy hh:mm") = Format(TSMMax, "dd.mm.yyyy hh:mm") Then
            TSMMaxZeile = i
        End If
    Next i

Else ' Automatische Erkennung des Bereichs
    ' Es wird die erset und die letzte Zeile mit Datum gesucht
    For i = 1 To intZeile
        If IsDate(wSq.Cells(i, OutDateSp).Value) Then
            OUTMinZeile = i
        End If
    Next i

    For i = intZeile To 1 Step -1
        If IsDate(wSq.Cells(i, OutDateSp).Value) Then
            OUTMaxZeile = i
        End If
    Next i

' ****unvollständig****
End If

' Kontrolle ob Zeitschritt gleich
If (OUTMaxZeile - OUTMinZeile) = (TSMMaxZeile - TSMInZeile) Then
    ' If .cells(tsmInzeile-delta,tssp) prüfen ob numerischer wert
Else
    MsgBox "Zeitschritt der Spalten ist ungleich!", vbCritical, "Phase-Shift"
End Sub
End If

' Kontrolle ob ein durch den Delta Zeiger verschobener Wert existiert
k = TSMInZeile - delta

```

```

If k <= 0 Then
    k = 1 ' Damit cells() die 1 Zeile erhält, sonst Laufzeitfehler!
End If
If Not IsDate(wSq.Cells(k, TSDateSp)) Then ' wenn kein Datum in der Zelle vorhanden
    MsgBox "Nicht alle Zeitschritte vorhanden!" & vbLf & "Späteren Startzeitpunkt wählen," _
        & vbLf & "oder Delta-Wert verkleinern"
    Exit Sub ' oder Rücksprung zur UserForm
End If
If Not IsDate(wSq.Cells(TSMMaxZeile + delta, TSDateSp)) Then ' wenn kein Datum in der Zelle vorhanden
    MsgBox "Nicht alle Zeitschritte vorhanden!" & vbLf & "Früheren Startzeitpunkt wählen," _
        & vbLf & "oder Delta-Wert verkleinern"
    Exit Sub ' oder Rücksprung zur UserForm
End If

' Dimensionierung des Arrays
Application.StatusBar = "Erstelle Ausgangswert- Array"
Values = OUTMaxZeile - OUTMinZeile
I = Values
timesteps = delta * 2 + 2
k = timesteps * T0 + T0-n - T0-1 | T0+1 - T0+n
ReDim vx(k, I) ' nachträglich Dimensioniert

' Einlesen der Ausgangswerte
i = 0 ' Laufvariable
For k = 0 To timesteps
    For I = 0 To Values
        If k = 0 Then
            vx(0, I) = Format(wSq.Cells(OUTMinZeile + I, OutDateSp).Value, "dd/mm/yyyy hh:mm")
        Elseif k = 1 Then
            vx(1, I) = wSq.Cells(OUTMinZeile + I, OutSp).Value
        Else
            vx(k, I) = wSq.Cells(TSMInZeile - delta + k + I - 2, TSSp).Value
        End If
    Next I
    i = i + 1
Next k

' Dimensionieren des 2.Array
Application.StatusBar = "Erstelle Abweichungs- Array"
k = timesteps
I = Values
ReDim vx_2(k, I)

' Kopieren des 1.Array in das 2.Array + Abweichung
For I = 0 To Values
    For k = 0 To timesteps
        If k >= 2 Then ' Array2 = Array1 - Ausgangswert
            vx_2(k, I) = vx(k, I) - vx(1, I)
        Else ' Die ersten beiden Zeilen werden 1:1 kopiert
            vx_2(k, I) = vx(k, I)
        End If
    Next k
Next I

' Nachträgliches Dimensionieren der Ausgabe-Arrays
k = timesteps - 2
ReDim vx_Min(k) ' nachträglich Dimensioniert
ReDim vx_Max(k) ' nachträglich Dimensioniert
ReDim vx_Avg(k) ' nachträglich Dimensioniert
ReDim vx_Time(k) ' nachträglich Dimensioniert

' Berechnen der Min - Max -Werte mittels Array
For k = 2 To timesteps ' Minimalwert
    min = vx_2(k, 0) ' Ersten Wert des Arrays
    For I = 0 To Values
        If min > vx_2(k, I) Then
            min = vx_2(k, I)
        End If
    Next I

```

ANHANG

```

    End If
Next l
vx_Min(k - 2) = min ' Werte ins Min-Array ; -2 = Start array bei 0
Next k

For k = 2 To timesteps ' Maximalwert
    max = vx_2(k, 0) ' Ersten Wert des Arrays
    For l = 0 To Values
        If max < vx_2(k, l) Then
            max = vx_2(k, l)
        End If
    Next l
    vx_Max(k - 2) = max ' Werte ins Max-Array ; -2 = Start array bei 0
Next k

' Berechnen der Spannweite zwischen Min-Max-Werten mittels Array und finden eines optimalen Punktes
' Es müssen 3 Fälle unterschieden werden: Max > 0 | Min > 0
'                               Max > 0 | Min < 0
'                               Max < 0 | Min < 0
For k = 0 To timesteps - 2
    If vx_Max(k) > 0 And vx_Min(k) > 0 Then
        avg = vx_Max(k) - vx_Min(k)
    ElseIf vx_Max(k) > 0 And vx_Min(k) < 0 Then
        avg = vx_Max(k) + Abs(vx_Min(k))
    Else ' vx_Max(k) < 0 And vx_Min(k) < 0
        avg = Abs(vx_Min(k)) - Abs(vx_Max(k))
    End If
    vx_Avg(k) = avg
Next k

min = vx_Avg(0) ' Ersten Wert des Arrays
i = -delta
avg = 0
For k = 0 To timesteps - 2 ' Minimalwert
    If min >= vx_Avg(k) Then
        min = vx_Avg(k)
        avg = i
    End If
    i = i + 1
Next k

result = avg ' Speichert das Ergebnis

' Einfügen der dynamischen Beschriftung
i = -delta
For k = 0 To timesteps - 2
    If i > 0 Then
        vx_Time(k) = "t+" & i & "min"
    ElseIf i < 0 Then
        vx_Time(k) = "t" & i & "min"
    Else
        vx_Time(k) = "t " & i & "min"
    End If
    i = i + 1
Next k

wsZ.Activate ' Damit Range und das Diagramm in sheet array gespeichert wird
If Ausgabe = True Then ' Ausgabe der erstellten Arrays
    Application.StatusBar = "Kopiere Werte ins Tabellenblatt"
    wsZ.Cells.ClearContents

    For k = 0 To timesteps - 2 ' Min-array
        Sheets("Array").Cells(k + 3, Values + 2).Value = vx_Min(k)
    Next k
    For k = 0 To timesteps - 2 ' Max-Array
        Sheets("Array").Cells(k + 3, Values + 3).Value = vx_Max(k)
    Next k
    For k = 0 To timesteps - 2 ' Avg-Array
        Sheets("Array").Cells(k + 3, Values + 4).Value = vx_Avg(k)
    Next k
    Sheets("Array").Cells(1, Values + 4).Value = result
    Range(Cells(3, Values + 2), Cells(timesteps + 1, Values + 4)).NumberFormat = "0.000"
    For k = 0 To timesteps - 2 ' Time-Array
        Sheets("Array").Cells(k + 3, Values + 5).Value = vx_Time(k)
    Next k
    For l = 0 To Values ' Delta_array
        For k = 0 To timesteps
            Sheets("Array").Cells(k + 1, l + 1).Value = vx_2(k, l)
        Next k
    Next l
    For l = 0 To Values ' Timestep_array
        For k = 0 To timesteps
            Sheets("Array").Cells(k + timesteps + 4, l + 1).Value = vx(k, l)
        Next k
    Next l
    For k = 0 To timesteps - 2 ' Time-Array
        Sheets("Array").Cells(k + timesteps + 6, Values + 2).Value = vx_Time(k)
    Next k
    Range(Cells(3, Values + 2), Cells(Timesteps + 1, Values + 2)).value = vx_Min
    Range(Cells(3, Values + 3), Cells(Timesteps + 1, Values + 3)).value = vx_Max
    Range(Cells(3, Values + 4), Cells(Timesteps + 1, Values + 4)).value = vx_Avg
    Range(Cells(3, Values + 5), Cells(Timesteps + 1, Values + 5)).value = vx_Time

    Range(Cells(1, 1), Cells(Timesteps + 1, Values + 1)).value = vx_2
    Range(Cells(30, 1), Cells(Timesteps + 30, Values + 1)).value = vx
    Cells.NumberFormat = "0.000"
    ActiveSheet.Columns.AutoFit ' Spaltenbreite an Datum anpassen
End If

' Berechnen des Min/Max -Wertes
k = 0
min = vx_Min(k) ' Ersten Wert des Arrays
For k = 0 To timesteps - 2 ' Minimalwert
    If min > vx_Min(k) Then
        min = vx_Min(k)
    End If
Next k
k = 0
max = vx_Max(k) ' Ersten Wert des Arrays
For k = 0 To timesteps - 2 ' Maximalwert
    If max < vx_Max(k) Then
        max = vx_Max(k)
    End If
Next k

' Bereichabhängiges runden
avg = max - min
If avg >= 10 Then
    i = -1 ' Zenerstelle runden
Elseif avg >= 1 Then
    i = 0
Else
    i = 1
End If

' Kontrolle ob Fehlerkurve innerhalb des Min-Max-Bereichs
k = 0
For k = 0 To timesteps - 2 ' Minimalwert
    If min > vx_Avg(k) Then
        min = vx_Avg(k)
    End If
Next k
k = 0
For k = 0 To timesteps - 2 ' Maximalwert
    If max < vx_Avg(k) Then
        max = vx_Avg(k)
    End If
Next k

```

ANHANG

```

End If
Next k

' Runden in Abhängigkeit vom Vorzeichen und relativer Größe
If min <= 0 Then
    min = Application.WorksheetFunction.RoundUp(min, i) ' -werte müssen aufrunden werden
Else
    min = Application.WorksheetFunction.RoundDown(min, i)
End If
If max >= 0 Then
    max = Application.WorksheetFunction.RoundUp(max, i)
Else
    max = Application.WorksheetFunction.RoundDown(max, i)
End If

'Ausgabe des Diagramms
Application.StatusBar = "Erstelle Diagramm"
s = FormatDateTime(TSMIn, vbShortDate) & " - " & FormatDateTime(TSMIn, vbShortTime) _
    & " <-> " & FormatDateTime(TSMMax, vbShortDate) & " - " & FormatDateTime(TSMMax, vbShortTime) _
    & " ; Timestep: " & delta & vbLf & "Werte um " & Format(result, "#0") & "min verschoben" ' Diagramm -
Überschrift
Set objChart = ActiveSheet.ChartObjects.Add(200, 50, 1000, 500)
With objChart
    .Name = "Abweichung" ' Diagramm - Name
    With .Chart
        .ChartType = xlLineMarkers
        .SeriesCollection.NewSeries
        .SeriesCollection(1).XValues = vx_Time
        .SeriesCollection(1).Values = vx_Avg
        .SeriesCollection(1).Name = "Avg"
        .SeriesCollection.NewSeries
        .SeriesCollection(2).XValues = vx_Time
        .SeriesCollection(2).Values = vx_Min
        .SeriesCollection(2).Name = "Min"
        .SeriesCollection.NewSeries
        .SeriesCollection(3).XValues = vx_Time
        .SeriesCollection(3).Values = vx_Max
        .SeriesCollection(3).Name = "Max"
        .SetElement (msoElementPrimaryCategoryGridLinesMajor)
        .SetElement (msoElementChartTitleAboveChart)
        .ChartTitle.Text = s ' Diagramm - Überschrift
    With .Axes(xlCategory)
        .TickMarkSpacing = 1
        .HasTitle = True
        .AxisTitle.Text = "Zeitschritt" ' Horizontale Primärachse
        .AxisTitle.Orientation = xlHorizontal
        .AxisBetweenCategories = False
        .TickMarkSpacing = 1 ' Linie bei jedem 2.Wert
        .TickLabelSpacing = 1 ' Beschriftung bei jedem 2.Wert
        .TickLabels.Orientation = xlUpward
    End With
    With .Axes(xlValue)
        .HasTitle = True
        .AxisTitle.Text = "Abweichung" ' Vertikale Primärachse
        .CrossesAt = min
        .MinimumScale = min
        .MaximumScale = max
    End With
    End With
End With

Set wsQ = Nothing
Set wsZ = Nothing
Set objChart = Nothing

Application.ScreenUpdating = True
Application.StatusBar = False

End Sub

Sub Change_Timestep()
' Makro verschiebt einen Zeitschritt um einen angegebenen Wert in min
' Eingabeparameter ist nur der Wert in "hh:mm:ss"
' Zeitschritt ist in der angegeben Spalte
' Es wird nur der Zeitschritt geändert und neu abgespeichert

' Requires Atol() - funktion

Dim Spalte As String
Dim SpalteN As Long
Dim change_TS As String
Dim change_TS_date As Date
Dim intZeile As Long
Dim wsZ As Worksheet
Dim i As Long
Dim k As Long ' Feldkoordinate Zeile
Dim vx_c As Variant ' Ausgabefeld
Dim Start As Long ' Erste Datumzelle
Dim ende As Long ' Letzte Datumzelle
Dim Zeitschritt As String
Dim Antwort As Variant
Set wsZ = ActiveSheet
Application.ScreenUpdating = False
Application.StatusBar = True
Application.StatusBar = "Change_Timestep"

'### Eingabewerte ###
change_TS = "02:03:00"
Spalte = "a"
Zeitschritt = "+"
'#####

change_TS_date = change_TS
SpalteN = Atol(Spalte) ' Buchstabe -> Atol -> Zahl
intZeile = wsZ.UsedRange.Rows.Count
' Ermittlung des ersten Datumswert i = erster Datumswert
' und des letzten Datumswert = intZeile
For i = 1 To intZeile
    If IsDate(Cells(i, SpalteN).Value) Then
        Exit For
    End If
Next i
Start = i
For i = Start To intZeile
    If Not IsDate(Cells(i, SpalteN).Value) Then
        Exit For
    End If
Next i
ende = i - 1

' Dimensionieren des Arrays
k = ende - Start ' Eigentlich +1 aber Array beginnt bei 0
ReDim vx_c(k, 0) ' (k,0) = Bei 1-dim Array müsste sonst transponiert werden...

' Kopiert Werte ins Array + Bearbeitung Zeitschritt
For k = 0 To ende - Start
    If Zeitschritt = "+" Then
        vx_c(k, 0) = wsZ.Cells(Start + k, SpalteN).Value + change_TS_date '### Vorzeichen ###
    Else
        vx_c(k, 0) = wsZ.Cells(Start + k, SpalteN).Value - change_TS_date '### Vorzeichen ###
    End If
    'Debug.Print k ' Ausgabe des k-Wertes im Direktfenster
Next k

Antwort = MsgBox(" Warnung! " & vbLf & "Es werden " & ende - Start + 1 & " Datumswerte überschrieben!", _

```

ANHANG

```

vbCritical + vbOKCancel, "Änderung des Zeitschritts")

' Ausgabe des Arrays
If Antwort = vbOK Then
    For k = 0 To ende - Start ' intzeile-2
        wsZ.Cells(Start + k, SpalteN).Value = vx_c(k, 0)
        Application.StatusBar = k + 1 & " von " & ende - Start
    Next k

    Sheets("Tabelle1").Range(Cells(Start, SpalteN), Cells(ende, SpalteN)).Value = vx_c

End If

Set wsZ = Nothing
Application.ScreenUpdating = True
Application.StatusBar = False
End Sub

Sub Export() v1.1
    ' Makro exportiert das ganze Tabellenblatt im csv-Format
    ' Zusätzlich können die Werte Formatiert werden, Standart "0000.0000"
    ' Weiters kann das Trennzeichen verändert werden
    ' Das Datum(SpalteA) wird automatisch ausgeschlossen, da sonst Nummernsalat ;-)
    ' Die 1. Zeile wird auf Leerzeichen überprüft
    Dim rngZelle As Range
    Dim Zeile As Range
    Dim Zelle As Range
    Dim s As String
    Dim wS As Worksheet
    Dim txt_Dateiname As String
    Dim intZeile As Long
    Dim intSpalte As Integer
    Dim i As Long
    Dim iMax As Long
    Dim Trennzeichen As String
    Dim Datum As Boolean
    Dim Formatierung As String
    Dim k As Long ' Temporäre Variable
    Dim Start As Long ' Startpunkt Datum
    Dim FilterVar As String
    Dim temp As String ' Temporärer String
    Dim temp1 ' Temporärer String

    Application.ScreenUpdating = True

    'Workbooks.Open FileName:=Dateiname
    'Sheets(SheetName).Activate
    Set wS = ActiveSheet

    '### Eingabewerte ###
    Trennzeichen = ";"
    Formatierung = "0000.00000000"
    '#####

    intZeile = wS.UsedRange.Rows.Count
    intSpalte = wS.UsedRange.Columns.Count

    With Application
        .UseSystemSeparators = False
        .DecimalSeparator = "."
        .ThousandsSeparator = ""
    End With

    ' Ist Datum in Spalte A vorhanden?
    For k = 1 To intZeile
        If IsDate(wS.Cells(k, 1).Value) Then
            Exit For
        End If

        Next k

        If Trennzeichen = "" Then ' Falls Trennzeichen kein Leerzeichen dann weiter
            For k = 1 To intSpalte 'Suche nach Leerzeichen und Vergleich der StringLänge
                temp = Cells(1, k).Value
                temp1 = FileNameOnly(temp, Trennzeichen)
                If Len(temp1) < Len(temp) Then
                    MsgBox "Trennzeichen in der Zelle " & Itoa(k) & "1 vorhanden" & _
                        vbLf & "Werte können falsch eingelesen werden", vbCritical, "TRNSYS-DataReaderExport")
                    Exit Sub
                End If
            Next k
        End If

        If k = intZeile + 1 Then ' Kein Datum vorhanden
            Cells.NumberFormat = Formatierung
        Else ' Mit Datumsspalte
            Range(Cells(1, 2), Cells(intZeile, intSpalte)).NumberFormat = Formatierung
        End If

        Set rngZelle = wS.UsedRange

        If Trennzeichen = "" Then
            FilterVar = "Textdateien (*.txt;*.dat), *.txt"
        Elseif Trennzeichen = ";" Then
            FilterVar = "Textdateien (*.txt;*.txt), *.txt"
        Else
            MsgBox "Kein zulässiges Trennzeichen" & vbLf & "Leerzeichen für txt" & vbLf & "Semikolon für txt", vbCritical,
                "Trennzeichen"
            Application.UseSystemSeparators = True ' Zurücksetzen der Einstellung
            Exit Sub ' Falls anderer Separator eingegeben
        End If

        If txt_Dateiname = "" Then
            ChDir ActiveWorkbook.Path ' Vorgabe des Standard- Ordners
            txt_Dateiname = Application.GetOpenFilename(FilterVar)
            ' Definiert die Einlesedatei
        End If
        '*****
        If txt_Dateiname = "Falsch" Then
            MsgBox "Keine Datei ausgewählt" & vbLf & "Batch wird beendet!", vbCritical, "Export txt-File"
            Application.UseSystemSeparators = True ' Zurücksetzen der Einstellung
            Exit Sub
        End If

        If Trennzeichen = ";" And FileNameOnly(txt_Dateiname, ".") = "txt" Then
            Name txt_Dateiname As (Left(txt_Dateiname, Len(txt_Dateiname) - 3) & ".txt")
            ' Falls bei der Endung noch txt-Steht
            txt_Dateiname = (Left(txt_Dateiname, Len(txt_Dateiname) - 3) & ".txt")
        End If

        Open txt_Dateiname For Output As #1

        For Each Zeile In rngZelle.Rows

            For Each Zelle In Zeile.Cells
                s = s & Zelle.Text & Trennzeichen
            Next Zelle

            s = Left(s, Len(s) - 1)
            Print #1, s
            s = ""

        Next Zeile

        Close #1

        Application.UseSystemSeparators = True

```

ANHANG

```
'ActiveWorkbook.Close SaveChanges:=False

Application.ScreenUpdating = True
End Sub

Sub delta_var()
' Makro addiert xxx Werte und bildet den Durchschnitt
' Die Durchschnittswerte werden dann in ein anderes Tabellenblatt kopiert
' Der Zeitschritt wird nur beim Durchlaufen der ersten Zeile mitkopiert
' Status Bar zur Visualisierung des Vorgangs

' Requires Atol() - function

Dim wsZ As Worksheet
Dim wSq As Worksheet
Dim intZeile As Long
Dim intSpalte As Integer
Dim Zeile As Long
Dim Sum As Double
Dim Counter As Long ' Laufvariable für Ausgabe
Dim SpalteZ As Long
Dim i As Double ' StatusBar
Dim iMax As Double
Dim delta As Double
Dim sec As String
Dim min As String
Dim h As String
Dim sec_M As String
Dim min_M As String
Dim h_M As String
Dim m As Long ' tempräre Laufvariable
Dim k As Long ' Laufvariable Datenfeld -> Zeilenrichtung
Dim l As Long ' Laufvariable Datenfeld -> Spaltenrichtung
Dim Ausgabe As Variant ' Ausgabefeld, wird später deklariert
Dim Eingabe As Variant ' Eingabefeld, wird später deklariert
Dim Zeitschritt_Auto As Boolean
Dim Spalte As String
Dim SpalteN As Integer ' Buchstabe -> Atol -> Zahl
Dim Mittelwert As String
Dim Zeitschritt As String
Dim Timestep As Date
Dim Start As Integer ' startdatum/wert
Dim test As Boolean
Application.ScreenUpdating = False
Application.StatusBar = True
Application.StatusBar = " Starte Mittelwertbildung"

'### Eingabewerte ###
Mittelwert = "00:01:00" ' 1 min
Zeitschritt = "00:00:02" ' 3 sec
Zeitschritt_Auto = True ' True -> es wird der Zeitschritt aus dem Quell-Tabellenblatt genommen
Spalte = "A"
Set wSq = ActiveWorkbook.Sheets(" csv_Import")
Set wsZ = ActiveWorkbook.Sheets(" csv_1min")
test = False ' False -> Erstellung mittels Array / True -> Erstellung mittels Zellenzugriff (langsamer)
'#####

'Workbooks.Open Filename:=DateiName
'Sheets(SheetName).Activate
'Set wsQ = ActiveSheet
'Sheets.Add After:=Sheets(Sheets.count)
'Sheets(Sheets.count).Name = "1min"
'Set wsZ = ActiveWorkbook.Sheets("1min")

SpalteN = Atol(Spalte)
intZeile = wSq.UsedRange.Rows.Count
```

```
intSpalte = wSq.UsedRange.Columns.Count

' Einlesen der gesamten Tabelle ins Array
Application.StatusBar = " Erstellung Eingangswert-Array"
k = intZeile - 1
l = intSpalte - 1
ReDim Eingabe(k, l)

For l = 0 To intSpalte - 1 ' Zur Überprüfen des gesamten Arrays
For k = 0 To intZeile - 1
i = i + 1
Eingabe(k, l) = wSq.Cells(k + 1, l + 1).Value
Next k
Next l

' Ausgabe des Array zum Testen
' For l = 0 To intSpalte - 1 ' Zur Überprüfen des gesamten Arrays
' For k = 0 To intZeile - 1
' wsQ.Cells(k + 1, intspalte+2 + l).value = Eingabe(k, l)
' Next k
' Next l
Application.StatusBar = " Berechnung der Konstanten"
' Zu Erstellung eines definierten Zeitschritts (zB 1 min)
If Zeitschritt_Auto = True Then
For k = 1 To intZeile
If test = False Then
If IsDate(Eingabe(k - 1, 0)) Then
Exit For
End If
Else ' true
If IsDate(wSq.Cells(k, SpalteN)) Then
Exit For
End If
End If
Next k
Start = k
If test = False Then
Timestep = Eingabe(k - 1, 0) - Eingabe(k, 0)
Else ' True
Timestep = wSq.Cells(k + 1, SpalteN).Value - wSq.Cells(k, SpalteN).Value
End If

Zeitschritt = Timestep ' Zwischenschritt Variablendeklaration Date/String
End If

sec = Right(Zeitschritt, 2) ' Wenn sec als Integer definiert => statt "01" wird 01
min = Mid(Zeitschritt, 4, 2)
h = Left(Zeitschritt, 2)
sec = sec + min * 60 + h * 3600 ' Zeitschritt in Sekunden

' Mittelwert
sec_M = Right(Mittelwert, 2) ' Wenn sec als Integer definiert => statt "01" wird 01
min_M = Mid(Mittelwert, 4, 2)
h_M = Left(Mittelwert, 2)
sec_M = sec_M + min_M * 60 + h_M * 3600

delta = sec_M / sec ' Zeilen mit denen ein Mittelwert gebildet werden muss
' Anzahl der Messwert= Zeitschritt/ Einzelschritte

' Überprüfen ob Überschriften vorhanden - Dynamische Anpassung
Application.StatusBar = " Kopiere Überschriften"
If Start = 2 Then
k = (intZeile - 1) / delta + 1 ' Zeilenrichtung beginnt bei Null Nachkommastelle wird durch start bei 0 berücksichtigt
l = intSpalte - 1 ' Spaltenrichtung beginnt bei 0
ReDim Ausgabe(k, l)
For l = 0 To intSpalte - 1
If test = False Then
```

ANHANG

```

        Ausgabe(0, I) = wSq.Cells(1, I + 1).Value      ' Kopiert die erste Zeile (Überschriften) ins Array
    Else
        wsZ.Cells(1, I + 1).Value = wSq.Cells(1, I + 1).Value  ' Kopiert die erste Zeile (Überschriften) ins
Tabellenblatt
    End If
    Next I
Else
    k = intZeile / delta  ' Zeilenrichtung beginnt bei Null Nachkommastelle wird durch start bei 0 berücksichtigt
    I = intSpalte - 1    ' Spaltenrichtung beginnt bei 0
    ReDim Ausgabe(k, I)
    ' keine Überschriften - keine Ausgabe
End If

Counter = Start - 1
' Kopieren des Zeitschritts
Application.StatusBar = " Kopiere Zeitschritt"
For k = Start To intZeile Step delta
    Counter = Counter + 1
    If test = False Then
        Ausgabe(Counter - 1, 0) = Eingabe(k - 1, 0)
    Else
        wsZ.Cells(Counter, SpalteN).Value = wSq.Cells(k + delta / 2 - 1, SpalteN).Value
    End If
End If
Next k

' Jetzt die Daten, aber mit Mittelwert
i = 0
For I = 1 To intSpalte - 1
    i = i + 1
    Application.StatusBar = " Kopiere " & i & " Spalte"
    Counter = Start - 1  ' Überschrift wenn vorhanden berücksichtigen
    For k = Start To intZeile
        m = m + 1
        If test = False Then
            Sum = Sum + Eingabe(k - 1, I)
        Else
            Sum = Sum + wSq.Cells(k, I + 1).Value
        End If

        If m = delta Then  ' Anzahl der mittelwerte erreicht
            If test = False Then
                Ausgabe(Counter, I) = Sum / delta
            Else
                wsZ.Cells(Counter + 1, I + 1).Value = Sum / delta
            End If

            Sum = 0
            m = 0
            Counter = Counter + 1
        End If
    Next k
Next I
If test = False Then
    Application.StatusBar = " Ausgabe Delta- Array"
    For I = 0 To intSpalte - 1  ' Zur Überprüfen des gesamten Arrays
        For k = 0 To intZeile / delta
            wsZ.Cells(k + 1, I + 1).Value = Ausgabe(k, I)
        Next k
    Next I
End If

Set wsZ = Nothing
Set wSq = Nothing

ActiveWorkbook.Close SaveChanges:=True
Application.ScreenUpdating = True
Application.StatusBar = False

```

```

End Sub

Sub delta20()
' Makro addiert 20 Werte und bildet den Durchschnitt
' Die Durchschnittswerte werden dann in ein anderes Tabellenblatt kopiert
' Der Zeitschritt wird nur beim Durchlaufen der ersten Zeile mitkopiert
' Status Bar zur Visualisierung des Vorgangs
Dim wsZ As Worksheet
Dim wSq As Worksheet
Dim intZeile As Long
Dim intSpalte As Integer
Dim Zeile As Long
Dim Zeiger As Long
Dim Sum As Double
Dim Counter As Long  ' Laufvariable für Ausgabe
Dim Spalte As Integer
Dim i As Double  ' StatusBar
Dim iMax As Double

Application.ScreenUpdating = False
Application.StatusBar = True
Application.StatusBar = " Starte Mittelwertbildung"

Set wSq = ActiveWorkbook.Sheets("csv_Import")
Set wsZ = ActiveWorkbook.Sheets("csv_1min")

intZeile = wSq.UsedRange.Rows.Count
intSpalte = wSq.UsedRange.Columns.Count

i = 0
iMax = intZeile * intSpalte
For Spalte = 2 To intSpalte
    Zeiger = 2  ' 1 Spalte  Laufvariable für Quelle
    Counter = 2  ' 1 Spalte

    Do While wSq.Cells(Zeiger, intSpalte).Value <> ""
        Sum = 0  ' Summe der 20 Messwerte

        If Spalte = 2 Then  ' Kopiert Datumswert
            wsZ.Cells(Counter, 1).Value = wSq.Cells(Zeiger + 10, 1).Value
        End If

        For Zeile = 0 To 19
            Sum = Sum + wSq.Cells(Zeiger, Spalte).Value  ' Summiert Werte
            Zeiger = Zeiger + 1
            i = i + 1
            Application.StatusBar = i & " von " & iMax & "Datensätzen"
        Next Zeile

        Sum = Sum / 20  ' Mittelwert aus 20 Messungen
        wsZ.Cells(Counter, Spalte).Value = Sum
        Counter = Counter + 1
    Loop

Next Spalte

Set wsZ = Nothing
Set wSq = Nothing
Application.ScreenUpdating = True
Application.StatusBar = False
End Sub

Sub timesteps()
' Makro überprüft ob bei jedem Zeitschritt eine Minute Differenz

```

ANHANG

```
' Der erste Zeitschritt wird als Muster herangezogen!!!
' Fehlt eine Zeile wird diese am Ende per msgbox ausgegeben
' und auch durch einfärben des Bereichs markiert.
' Implementierung der Erkennung von großen Bereichen und getrennte Ausgabe
```

```
' Requires Atol() - function
```

```
Application.ScreenUpdating = False
Dim rngZelle As Range
Dim zeit_vorher As Date ' Vorheriger Wert
Dim zeit_jetzt As Date ' Jetziger Wert
Dim intZeile As Long
Dim s As String
Dim i As Long
Dim wS As Worksheet
Dim rngBereich As Range ' Range der zu überprüfen ist
Dim t As Long ' Temporäre Laufvariable
Dim Timestep As Date
Dim Drift As Date
Dim min As String
Dim h As String
Dim sec As String
Dim time2 As String
Dim Zeitschritt As String
Dim ZeitschrittSuche As Boolean
Dim Spalte As String
Dim last_TimeStep As Date
Dim Min_Drift As String
Dim Repair As Boolean
' Workbooks.Open Filename:=Dateiname
' Sheets(SheetName).Activate
Set wS = ActiveSheet

'### Eingabewerte ###
Spalte = "d"
Zeitschritt = "00:05:00"
ZeitschrittSuche = False
Min_Drift = "00:15:00" ' Grenze ab der Abweichung ausgegeben wird
Repair = True
#####

intZeile = wS.UsedRange.Rows.Count
Set rngBereich = Range(Cells(1, Atol(Spalte)), Cells(intZeile, Atol(Spalte)))

If ZeitschrittSuche = True Then
' Überprüfung welcher Zeitschritt vorhanden ist
' Hierzu wird der 1. Zeitschritt betrachtet
For t = 1 To intZeile
If Cells(t, 1).Value <> "" Then
Timestep = Cells(t + 1, 1).Value - Cells(t, 1).Value
End If
If Timestep <> 0 Then ' Wenn erster Wert erhalten, dann Abbruch
Exit For
End If
Next t
sec = Right(Timestep, 2) ' Wenn sec als Integer definiert => statt "01" wird 01
min = Mid(Timestep, 4, 2)
h = Left(Timestep, 2)
'time1 = Format(timestep, "hh:mm:ss")
Zeitschritt = h & ":" & min & ":" & sec
End If

' Suche des letzten Zeitschritts einer Stunde

' Auftrennen des Zeitschritts
h = Left(Zeitschritt, 2)
min = Mid(Zeitschritt, 4, 2)
sec = Right(Zeitschritt, 2)
```

```
t = sec + min * 60 + h * 3600 ' Zeitschritt in Sekunden
t = 86400 - t ' Letzter Zeitschritt vor Überlauf
```

```
' Berechnung der Stellen
h = t \ 3600 '\ => Ganzzahldivision!!!
t = t Mod 3600 ' Bestimmung des Rest
min = t \ 60
t = t Mod 60
sec = t
```

```
last_TimeStep = h & ":" & min & ":" & sec
```

```
' Jetzt wird das gesamte Blatt überprüft
t = 0 ' Starterkennung
i = 0 ' Zähler für Abweichungen
Drift = 0
```

```
For Each rngZelle In rngBereich
If rngZelle.Value <> "" And t <> 0 And IsDate(rngZelle.Value) Then ' Makro geht alle Zeilen durch

zeit_jetzt = Format(rngZelle.Value, "hh:mm:ss") ' Minutenwert in der Variable
Timestep = zeit_jetzt - zeit_vorher ' Vergleich mit Vorgabe
time2 = Format(Timestep, "hh:mm:ss") ' Mögliche Abweichung nun in time2

If time2 <> Zeitschritt Then ' Vergleich IST mit SOLL - Zeitschritt
If zeit_vorher = last_TimeStep Then ' Suche Letzter Zeitschritt
Else
If s = Empty Then
s = s & " Abweichungen >= " & Min_Drift & vbLf & vbLf
s = s & "Datum - Uhrzeit - Abweichung" & vbLf
End If

Drift = Drift + Timestep
i = i + 1
If Timestep >= Min_Drift Then
s = s & rngZelle.Value & " - " & time2
s = s & vbLf
End If
rngZelle.Interior.ColorIndex = "3" ' Und mit Farbe rot markieren
End If
End If
End If

If t = 0 And IsDate(rngZelle.Value) Then ' Wird nur einmal beim Start durchlaufen
t = 1
zeit_jetzt = Format(rngZelle.Value, "hh:mm:ss")
End If

zeit_vorher = zeit_jetzt
Next rngZelle

If s = Empty Then
s = s & " Zeitschritt fehlerfrei" & vbLf
MsgBox s, vbInformation, "Abweichung"
Else
s = s & vbLf & "Zeitschritt: " & Zeitschritt & vbLf
s = s & vbLf & i & " Abweichungen vom Zeitschritt"
s = s & vbLf & "Abweichungsdauer: " & Drift
MsgBox s, vbCritical, "Abweichung"
End If
```

```
' ActiveWorkbook.Close SaveChanges:=True
Application.ScreenUpdating = True
```

ANHANG

```

End Sub

Sub timesteps_old()
' Makro überprüft ob bei jedem Zeitschritt eine Minute Differenz
' Es muss nur die oberste Zelle markiert werden
' Fehlt eine Zeile wird diese am Ende per msgbox ausgegeben
' und auch durch einfärben des Bereichs markiert.
' Ändern auf Bereichvorgabe durch auskommentieren von Z18 - Z19

Application.ScreenUpdating = False
Application.StatusBar = True
Dim rngZelle As Range
Dim minute_vorher As Integer ' Vorheriger Wert
Dim minute_jetzt As Integer ' Jetziger Wert
Dim intZeile As Double
Dim s As String
Dim i As Integer ' Fehlerfall
Dim Count As Double
' Ändern zum selektieren per User *****
intZeile = ActiveSheet.UsedRange.Rows.Count
Range(Cells(ActiveCell.Row, ActiveCell.Column), Cells(intZeile, ActiveCell.Column)).Select
'*****
minute_vorher = 99 ' Startwert zum Auswerten
i = 0
Count = 0
For Each rngZelle In Selection
If rngZelle.Value <> "" Then ' Makro geht alle Zeilen durch
minute_jetzt = Minute(rngZelle.Value) ' Minutenwert in der Variable
Application.StatusBar = Count & " von " & intZeile
Count = Count + 1
' Fehler beim letzten Schritt?
If i = 1 Then
minute_vorher = 99
i = 0
End If

If minute_vorher = 99 Then
minute_vorher = minute_jetzt
Elseif minute_vorher = 59 Then
minute_vorher = "00"
Else
minute_vorher = minute_vorher + 1
End If
If minute_jetzt = minute_vorher Then
Else
i = 1
s = s & rngZelle.Value
s = s & vbCrLf
rngZelle.Interior.ColorIndex = "3" ' Und mit Farbe rot markieren
End If
End If
Next rngZelle
ActiveCell.Select

If s = Empty Then
s = s & " Zeitschritt fehlerfrei"
End If

MsgBox s ' Ausgabe des Errorcodes

Application.StatusBar = False
Application.ScreenUpdating = True
End Sub

Public Function Atol(strBuchstaben As String)
' Funktion kann eine Buchstabenkombination mir 3 Buchstaben
' in eine Zahl umwandeln, zB.Eingabefeld übergibt wert an range
' Public Function, damit sie auch in Tabellenblatt verfügbar ist

```

```

Dim intNummer As Integer
Dim intNummer_2 As Integer
If Len(strBuchstaben) = 1 Then ' Ein Buchstabe
intNummer = Asc(strBuchstaben)
If intNummer >= 97 Then ' Kleinbuchstabe
intNummer = intNummer - 96
Else ' Großbuchstabe
intNummer = intNummer - 64
End If
Elseif Len(strBuchstaben) = 2 Then ' Zwei Buchstaben
intNummer = Asc(Left(strBuchstaben, 1)) ' 1. Buchstabe
If intNummer >= 97 Then
intNummer = (intNummer - 96) * 26
Else
intNummer = (intNummer - 64) * 26
End If
intNummer_2 = Asc(Right(strBuchstaben, 1)) ' 2. Buchstabe
If intNummer_2 >= 97 Then
intNummer = intNummer + intNummer_2 - 96
Else
intNummer = intNummer + intNummer_2 - 64
End If
Else ' Drei Buchstaben
intNummer = Asc(Left(strBuchstaben, 1)) ' 1. Buchstabe
If intNummer >= 97 Then
intNummer = (intNummer - 96) * 676
Else
intNummer = (intNummer - 64) * 676
End If
intNummer_2 = Asc(Right(Left(strBuchstaben, 2), Len(Left(strBuchstaben, 2)) - 1))
' Befehl MID(variable, startwert von links, anzahl der zeichen)
If intNummer_2 >= 97 Then
intNummer = intNummer + (intNummer_2 - 96) * 26
Else
intNummer = intNummer + (intNummer_2 - 64) * 26
End If
intNummer_2 = Asc(Right(strBuchstaben, 1))
If intNummer_2 >= 97 Then
intNummer = intNummer + (intNummer_2 - 96)
Else
intNummer = intNummer + (intNummer_2 - 64)
End If
End If
If intNummer > 16384 Then
MsgBox "Spaltenlimit Excel 16384 = XFD!", vbCritical
End If
Atol = intNummer
End Function

Function FileNameOnly(pname, Zeichen As String) As String ' vs2.4
' 1.Parameter: der zudurchsuchende String
' 2.Parameter: das zu suchende Zeichen
' Funktion sucht den String von rechts nach links durch
' Wenn das Zeichen gefunden wurde gibt die Funktion den SuchString bis zum Zeichen zurück
' Wenn das Zeichen nicht gefunden wurde gibt die Funktion den gesamten SuchString zurück
Dim i As Integer
Dim length As Integer
Dim temp As String
length = Len(pname)
temp = ""
For i = length To 1 Step -1
If Mid(pname, i, 1) = Application.PathSeparator Then
If Mid(pname, i, 1) = Zeichen Then
FileNameOnly = temp
Exit Function
End If
temp = Mid(pname, i, 1) & temp
Next i

```

ANHANG

```
FileNameOnly = pname
End Function

Sub Delete_All_Diag()
    Dim Dia As ChartObject
    Dim wS As Worksheet
    Set wS = ActiveSheet
    For Each Dia In ActiveSheet.ChartObjects
        Dia.Delete
    Next Dia
End Sub

Sub value_change()
    ' Makro führt auf allen darunterliegenden Zellen die selbe Operation durch
    ' Es muss nur die erste Zeile selektiert sein
    Dim intZeile As Double
    Dim rngZelle As Range
    Dim Zelle As Range
    Dim i As Double

    Application.ScreenUpdating = False
    Application.StatusBar = True
    Application.StatusBar = " Batch beginnt"

    intZeile = ActiveSheet.UsedRange.Rows.Count

    ' Bereichzuweisung
    Set rngZelle = Range(Cells(ActiveCell.Row, ActiveCell.Column), Cells(intZeile, ActiveCell.Column))
    i = 0

    For Each Zelle In rngZelle
        With Zelle
            ' Ab hier ändern was mit der Zelle geschehen soll
            .Value = .Value / 10
        End With
        i = i + 1
    Application.StatusBar = i & " von " & intZeile
    Next Zelle

    Application.StatusBar = False
    Application.ScreenUpdating = True
End Sub

Public Function ItoA(intColumn As Variant) As String ' Änderung des intcolumns von integer auf long
    ' Berechnet aus einer Zahl den entsprechenden Buchstabencode für die Excel-Spaltendarstellung
    ' Quelle: http://www.office-loesung.de/ftopic343198\_0\_0\_asc.php
    Dim strLetter As String
    Dim strLetter2 As String
    Dim strLetter3 As String

    ' erste Stelle
    strLetter = Chr((64 + intColumn) - Int(intColumn / 26) * 26)
    If strLetter = "@" Then strLetter = "Z"

    ' zweite Stelle
    ' ab 703 fängt "AAA" an
    If intColumn < 703 Then
        strLetter2 = Chr(64 + (((Int((intColumn - 1) / 26) * 26) / 26) - 26 * (Int((intColumn - 1) / 702))))
    Else
        strLetter2 = Chr(64 + (((Int((intColumn - 1) / 26) * 26) / 26) - 26 * (Int(((intColumn - 1) - 702) / 676) + 1)))
    End If
    If strLetter2 = "@" Or intColumn < 27 Then strLetter2 = ""

    ' dritte Stelle
    If intColumn >= 703 Then strLetter3 = Chr(64 + Int(((intColumn - 1) - 702) / 26) + 1)

    ItoA = strLetter3 & strLetter2 & strLetter
End Function
```