Übertragung und Validierung des TRNSYS - MULTIPORT Store Modells

Type 340 für MATLAB R2014b

Im Rahmen eines Fachpraktikums im Forschungsprojekt PVstore

Marc Jakobi

FB1/HTW Berlin – UT/RE (B)

Inhaltsverzeichnis

1. Einleitung

Im Forschungsprojekt „Energiemanagement und Optimierung von Photovoltaiksystemen mit Batterie- und Wärmespeichern (PVstore)“ im Verbundvorhaben „Langlebige Qualitätsmodule für PV-Systeme mit Speicheroption und intelligentem Energiemanagement (LAURA)“ werden intelligente Regelstrategien für photovoltaische Eigenverbrauchssysteme entwickelt [1]. Im Rahmen des Projektes werden u. a. PV-Speichersysteme mit Wärmespeichern in MATLAB simuliert. Da das TRNSYS-Modell Type 340 (MULTIPORT Store Model) in MATLAB nicht zur Verfügung steht, musste dafür eine Funktion programmiert werden. Diese wird im Folgenden dokumentiert und validiert.

2. Beschreibung des Modells und Veränderungen gegenüber der TRNSYS-Version

Die Funktionsweise des Modells wurde aus [2] und [3] entnommen und teilweise entsprechend der Ansprüche der im Rahmen von PVstore durchgeführten Simulationen an das Modell vereinfacht. Der Speicher wird in N_{max} Volumensegmente (Nodes) eingeteilt, die jeweils die gleiche Temperatur besitzen. An jeden Node kann mit sogenannten Doppelports direkt be- und entladen werden. Ein Doppelport ist ein Anschluss-Paar, das zum selben Kreislauf gehört, mit einem Eingang und einem Ausgang. Der Massenstrom kann in dem zugehörigen Kreislauf für jeden Zeitschritt als konstant betrachtet werden. In dieser Variante des Modells ist die Anzahl Nodes und somit die Anzahl Doppelports frei wählbar. Desweiteren kann der Speicher mit bis zu vier internen Wärmetauschern, sowie einem Heizstab (von der Seite in einem der Nodes positioniert) beladen werden. In Bild 2.1 ist die MATLAB-Version des Speichers schematisch dargestellt.

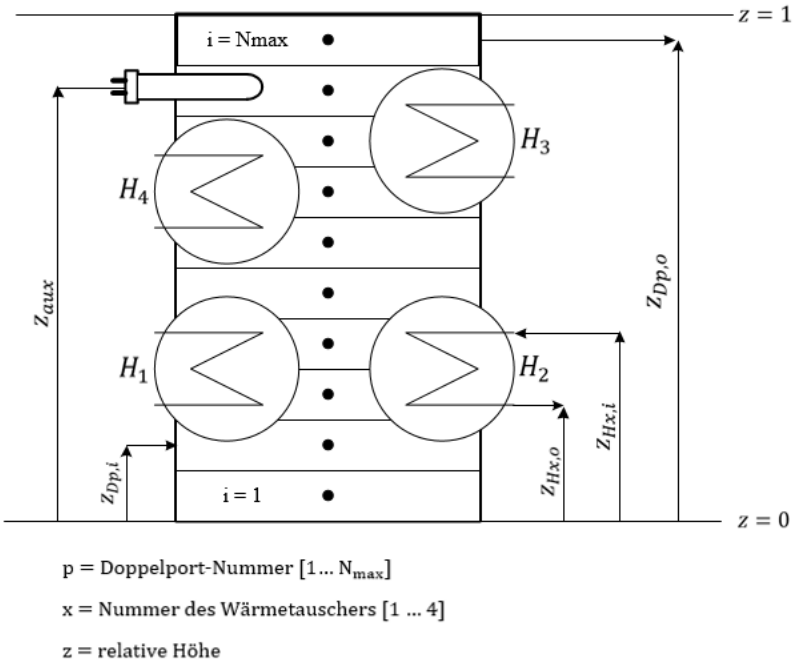

Bild 2.1: Schematische Darstellung des MULTIPORT Store-Modells nach der Implementierung in MATLAB

2.1 Höhenangaben

Wie in der TRNSYS-Variante, erfolgen zur Erleichterung der Anwendung durch den Nutzer alle Höhenangaben relativ zur Höhe des Speichers [3]. Die relative Höhe des Heizstabes z_{aux} wird z. B. nach Gl. 2.1 berechnet.

$$z_{aux} = \frac{H_{aux}}{H_S} \qquad (2.1)$$

$H_{aux} =$ Höhe des Speichers in m
$z_{aux} =$ Höhe des Heizstabes in m

2.2 Wärmeverluste an die Umgebung

Im Gegensatz zu der TRNSYS-Version des Modells, wird der Speicher nicht in vier Zonen konstanter Wärmeverlustrate (*WVR*) eingeteilt. Da in vielen Fällen nur eine effektive *WVR* für einen Speicher angegeben wird [4], führt die Einteilung in eine Zone aufgrund der ersparten Eingabeparameter zu einer einfacheren Anwendung durch den Nutzer. Für den Speicherdeckel und –Boden können getrennte *WVR* eingegeben werden.

2.3 Doppelports

Bezüglich dem Be- und Entladen mit Doppelports wurden gegenüber der TRNSYS-Version keine Veränderungen vorgenommen, außer dass die Maximale Anzahl der Doppelports auf N_{max} begrenzt wird, statt auf 10, wobei N_{max} unbegrenzt ist.

Bei der Unterscheidung zwischen Be- und Entladen durch Doppelports ist das Vorzeichen des Massenstromes entscheidend. Befindet sich der Einlass z_{Di} unter dem Auslass z_{Do}, fließt also das Wasser von unten nach oben, bekommt der Massenstrom ein positives Vorzeichen [3]. Ist $z_{Di} > z_{Do}$, ist der Massenstrom positiv. Bei der Anwendung des Modells in MATLAB kann der Betrag angegeben werden. Hinzu kommt die Unterscheidung zwischen geschichtetem und ungeschichtetem Be- und Entladen durch Doppelports. Bei geschichtetem Laden wird die Position des Einlasses so neu berechnet, dass z_{Di} sich dort befindet, wo die Speichertemperatur gleich oder geringfügig über der Einlasstemperatur ist. Bei durchgehend konstanter Speichertemperatur unter der Einlasstemperatur steigt das Wasser ganz nach oben. Dieses Phänomen lässt sich durch das Aufsteigen des warmem Wassers aufgrund des Dichteunterschiedes erklären [3]. Beim geschichteten Entladen wird der Einlass analog neu berechnet. Durch den Operator-Input *scdp* kann der Funktion für jeden Doppelport geschichtetes $(scdp_D = true)$ oder ungeschichtetes $(scdp_D = false)$ Be- und Entladen vorgegeben werden. Bei ungeschichtetem Be- und Entladen wird das Wasser im Speicher an der Einlassposition des Doppelports direkt aufgewärmt/abgekühlt und durch Durchmischung des Speichers verteilt.

2.4 Wärmetauscher

In der TRNSYS-Variante des Speichermodells gibt es für jeden Wärmetauscher vier Möglichkeiten: (virtuell) geschichtetes und ungeschichtetes Laden mit internen Wärmetauschern und mit Mantelwärmetauschern. Für die Anforderungen im Forschungsprojekt PVstore sind die Fälle geschichtetes Laden und Mantelwärmetauscher nicht von Bedeutung. Somit wird in der MATLAB-Ausführung nur der Fall ungeschichtetes Laden mit internen Wärmetauschern betrachtet. Somit entfällt auch die Betrachtung der Wärmeverluste eines Wärmetauschers an die Umgebung. Es können bis zu vier Wärmetauscher eingesetzt werden (s. Bild 1), es ist jedoch zu beachten, dass die Wärmetauscher H_1 und H_4, bzw. H_2 und H_3 nicht dieselben Nodes im Speicher teilen können. H_1 und H_4 können aber jeweils die Speichernodes mit H_2 und H_3 teilen.

2.4.1 Zeit-, Temperatur- und Massenstromabhängige Wärmeverlustrate $(UA)^*_{Hx}$ der Wärmetauscher

Wie in der TRNSYS-Ausführung des Type 340 kann für jeden Wärmetauscher die Nennwärmeverlustrate $(\mathrm{UA})_{\mathrm{Hx}}$ separat angegeben werden. Diese wird nach Gl. 2.2 und Gl. 2.3 angewendet, um für jeden Zeitschritt in Abhängigkeit der Zeit, des Massenstroms und der Temperaturdifferenz zwischen Wärmetauscher-Eintritt und Speicher, bzw. der mittleren Temperaturdifferenz zwischen Wärmetauscher-Eintritt und Speicher, die aktuelle *WVR* des Wärmetauschers für jedes besetzte Node i zu berechnen [3].

$$(UA)^*_{Hx,i} = (UA)_{Hx} \cdot F_{Hx} \cdot \dot{m}_{Hx}^{b_{Hx,1}} \cdot \left(\vartheta_{Hx,in} - \vartheta_{S,i}\right)^{b_{Hx,2}} \cdot \left(\frac{\vartheta_{Hx,in} - \vartheta_{S,i}}{2}\right)^{b_{Hx,3}} \qquad (2.2)$$

Mit:
$(UA)_{Hx}$ = Nennwärmeverlustrate zwischen WT und Speicher in W/K
F_{Hx} = Faktor für die Zeitabhängigkeit der WVR nach Gl. 2.3
$\dot{m}_{Hx}$ = Massenstrom durch den WT in kg/s
$\vartheta_{Hx,in}$ = Einlasstemperatur des WT in °C
$\vartheta_{S,i}$ = Speichertemperatur im entsprechenden Speichernode in °C
$b_{Hx,1}$ = Parameter für die Abhängigkeit vom Massenstrom
$b_{Hx,2}$ = Parameter für die Abhängigkeit von der Temperaturdifferenz zwischen WT-Eintritt und Speicher
$b_{Hx,3}$ = Parameter für die Abhängigkeit von der mittleren Temperaturdifferenz zwischen WT-Eintritt und Speicher
x = Nummer des Wärmetauschers [1…4]

Der Faktor $(0 \leq F_{Hx} \geq 1)$ für die Zeitabhängigkeit aufgrund der Massenträgheit beim Anfahren des Wärmetauschers wird nach Gl. 2.3 berechnet.

$$F_{Hx} = \frac{1}{S_{Hx}} \cdot \int_{t=0}^{t=t-1} \dot{m}_{Hx} \cdot (1 - F_{Hx}) \cdot dt_{Hx} \qquad (2.3)$$

Mit S_{Hx} = Faktor für das Startverhalten des Wärmetauschers.

In TRNSYS lassen sich die Parameter *b* für die Abhängigkeiten, sowie der Faktor S_{Hx} für das Startverhalten des WT im Deck festlegen. Dafür lassen sich jedoch schwer Herstellerangaben finden, sodass diese experimentell ermittelt werden müssen. In [3] wird erwähnt, dass für S_{Hx} typische Werte zwischen 0,01 und 0,05 liegen. Deswegen wird zwecks der einfacheren Anwendung der Funktion für den Nutzer in MATLAB ein Mittelwert für alle Wärmetauscher festgelegt. Für die Parameter $b_{Hx,1}$ und $b_{Hx,3}$ werden Mittelwerte aus den in [5] messtechnisch ermittelten Werten festgelegt, wobei $b_{Hx,2}$ i. d. R. vernachlässigt werden kann [5] und in diesem Fall wird.

2.5 Heizstab

Da die Nachheizung mittels Heizstab von oben heute selten Anwendung findet, wird in der MATLAB-Anwendung des Type 340 nur die Möglichkeit des Nachheizens von der Seite gegeben. Wird ein Heizstab verwendet, gibt es zwei Betriebsarten, die zur Verfügung stehen: eine konstante Leistung, die intern anhand eines Thermostates am Heizstab anhand der im entsprechenden Node herrschenden Temperatur ein- oder ausgeschaltet wird, oder eine variable Leistung, für die eine externe Regelung notwendig ist. Bei konstanter Leistung kann wie im TRNSYS-Type 340 für den Node, in dem der Heizstab sich befindet eine Mindesttemperatur T_{set} festgelegt werden. Bei der Nachheizung mit konstanter Leistung erfolgt die Rate des Ein- und Ausschaltens (Stagnation) mit der internen (variablen) Zeitschrittweite der Funktion. So wird ein übermäßiges Erhitzen innerhalb der externen Zeitschrittweite der Simulation vermieden. Die Deadband-Temperatur ΔT_{DB} aus dem TRNSYS-Modell wird in der MATLAB-Ausführung ausgelassen.

2.6 Temperatur-Sensoren

Um die Temperatur einer ausgewählten Höhe des Speichers ausgeben zu lassen, können bis zu N_{max} Temperatur-Sensoren angebracht werden. Dafür wird jeweils die relative Höhe des Sensors angegeben.

3. Sonstige Abweichungen der MATLAB-Funktion von dem TRNSYS Type 340

Zusätzlich zu den o. g. Änderungen wurden die folgenden Parameter für die MATLAB-Funktion festgelegt. Bei Bedarf können sie in der Syntax der Funktion („MultiStoreModel.m“) geändert werden.

Tabelle 3.1: Festgelegte Annahmen im MATLAB-Modell, die im TRNSYS-Modell als Parameter eingegeben werden.

Parameter		
Dichte	$\rho_{S,}\ \rho_{Hx}$	Für die Dichte des Fluids im Speicher und des Fluids in den Wärmetauschern werden konstant 992,21 kg/m³ angenommen. [Dichte von Wasser bei 40 °C]
Effektive Wärmeleitfähigkeit	λ_{con}	Die effektive Wärmeleitfähigkeit beinhaltet die Wärmeleitfähigkeit über Einbauten, Wände und Konvektion und kann nur experimentell ermittelt werden [2]. Festgelegt wird ein Mittelwert aus Literaturwerten [6] [7] mit 1,52 W/mK
Spezifische Wärmekapazität	$c_{p,S,}\ c_{p,Hx}$	Für Speicher und Wärmetauscher wird eine konstante spez. Wärmekapazität von 4179,5 Ws/kgK angenommen. [spez. Wärmekapazität von Wasser bei 40 °C]
Faktor für das Startverhalten der Wärmetauscher	S_{Hx}	0,025 (siehe Abschnitt 2.4.1)
Faktoren für die Abhängigkeiten von $(UA)^{*}_{Hx,i}$	$b_{Hx,1}$ $b_{Hx,2}$ b_{Hx3}	0,237 0 0,509 (siehe Abschnitt 2.4.1)

4. Mathematische Beschreibung

Die Energiebilanz aus [3] wird im Folgenden um die Energiezufuhr durch den Heizstab und durch die Doppelport-Eintritte ergänzt. Alle Temperaturen abgesehen von der Umgebungstemperatur ϑ_{amb} werden jeweils mit i, j indiziert. Dabei steht i für das jeweilige Node im Speicher, $j = 1$ für die Wärmetauscher 1 und 4, $j = 2$ für den Speicher und $j = 3$ für die Wärmetauscher 2 und 3. Die erste und dritte Zeile von Gl. 4.1 beschreiben die Energieflüsse aufgrund der bis zu N_{max} Doppelports durch die Nodes. Da in jedem Node bis zu N_{max} Massenströme fließen können, werden diese aufsummiert. Die zweite Zeile beschreibt die Energiezufuhr/abfuhr durch die Doppelport Ein- und Ausgänge. Wird an einem Node Energie zu- oder abgeführt, muss der entsprechende Massenstrom separat behandelt werden.

In der vierten Zeile werden die Wärmeströme zwischen Speicher und Wärmetauscher berechnet, in der fünften die Durchmischung des Speichers und in der letzten Zeile die Energiezufuhr durch den Heizstab. Für unterschiedliche Betriebszustände können Teile der Gleichung 4.1 durch Nullsetzen der logischen Operatoren ξ entfallen.

Wichtig bei der Anwendung ist die Richtung der Energieflüsse. Diese hängt von dem Vorzeichen der entsprechenden Massenströme ab. Ein Massenstrom von unten nach oben wird mit positivem Vorzeichen dargestellt und ein Massenstrom von oben nach unten mit negativem Vorzeichen.

$$
\begin{aligned}
\frac{V_S \cdot \rho_S \cdot c_{p,S}}{N_{max}} \cdot \frac{\partial \vartheta_{i,2}}{\partial t} = & \sum_{p=1}^{i-1} \dot{m}_{Dp} \cdot c_{p,S} \cdot \left[\xi_{1,p} \cdot \left(\vartheta_{i-1,2} - \vartheta_{i,2}\right) + \xi_{2,p} \cdot \left(\vartheta_{i,2} - \vartheta_{i+1,2}\right)\right] \\
& + \xi_{D,i} \cdot \dot{m}_{Di} \cdot c_{p,S} \cdot \left[\xi_{1,i} \cdot \left(\vartheta_{Di,in} - \vartheta_{i,2}\right) + \xi_{2,i} \cdot \left(\vartheta_{i,2} - \vartheta_{Di,in}\right)\right] \\
& + \sum_{p=i+1}^{n_D} \dot{m}_{Dp} \cdot c_{p,S} \cdot \left[\xi_{1,p} \cdot \left(\vartheta_{i-1,2} - \vartheta_{i,2}\right) + \xi_{2,p} \cdot \left(\vartheta_{i,2} - \vartheta_{i+1,2}\right)\right] \\
& + \xi_3 \cdot \frac{(UA)^*_{h1/4,S}}{n_{h1/4}} \cdot \left(\vartheta_{i,1} - \vartheta_{i,2}\right) + \xi_4 \cdot \frac{(UA)^*_{h2/3,S}}{n_{h2/3}} \cdot \left(\vartheta_{i,3} - \vartheta_{i,2}\right) \\
& + \lambda_{con} \cdot \frac{A_q}{H_S} \cdot N_{max} \cdot \left[\left(\vartheta_{i+1,2} - \vartheta_{i,2}\right) + \left(\vartheta_{i-1,2} - \vartheta_{i,2}\right)\right] \\
& - \frac{(UA)_S}{N_{max}} \cdot \left(\vartheta_{i,2} - \vartheta_{amb}\right) \\
& + \xi_{av} \cdot P_{aux} + \xi_{ak} \cdot P_{aux}
\end{aligned}
\qquad (4.1)
$$

$(UA)^*_{h1/4,S}$ = WVR nach Gl. 2.2 für die Wärmetauscher 1 und 4 $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$

$n_{h1/4}$ = Anzahl Nodes im Speicher, die von den WT 1 und 4 besetzt sind

$(UA)^*_{h2/3,S}$ = WVR nach Gl. 2.2 für die Wärmetauscher 2 und 3 $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$

$n_{h2/3}$ = Anzahl Nodes im Speicher, die von den WT 2 und 3 besetzt sind

p = Index für die Doppelports $p \in [0, n_D]$

i = Index für die Nodes im Speicher $i \in [1, N_{max}]$

n_D = Anzahl genutzter Doppelports $n_D \in [0, N_{max}]$

λ_{con} = Effektive Wärmeleitfähigkeit im Speicher $\left[\frac{\mathrm{W}}{\mathrm{mK}}\right]$

$(UA)_S$ = Nennwärmeverlustrate des Speichers. Für den Node 1 wird die WVR des Speicherbodens $(UA)_{S,bot}$ / für den Node N_{max} wird die WVR des Deckels $(UA)_{S,top}$ hinzu addiert. $\left[\frac{\mathrm{W}}{\mathrm{K}}\right]$

P_{aux} = Leistung mit dem Heizstab, wobei $\eta_{aux} = 1$ [W]

$\xi_{1,p} = 1$, wenn $\dot{m}_{Dp} > 0$ und wenn $\dot{m}_{Dp}$ durch den Node i fließt, sonst $\xi_{1,p} = 0$

$\xi_{2,p} = 1$, wenn $\dot{m}_{Dp} < 0$ und wenn $\dot{m}_{Dp}$ durch den Node i fließt, sonst $\xi_{2,p} = 0$

$\xi_3 = 1$, wenn der Node i des Speichers im Kontakt mit dem Node i eines der Wärmetauscher 1 oder 4 ist, sonst $\xi_3 = 0$

$\xi_4 = 1$, wenn der Node i des Speichers im Kontakt mit dem Node i eines der Wärmetauscher 2 oder 3 ist, sonst $\xi_4 = 0$

$\xi_{av} = 1$, wenn der Heizstab zum internen Zeitschritt t_{int} eingeschaltet ist und wenn die Heizstableistung variabel ist, sonst $\xi_{av} = 0$

$\xi_{ak} = 1$, wenn die Heizstableistung konstant ist, sonst $\xi_{ak} = 0$

Zur gleichen Zeit der Durchmischung im Speicher erfolgt die Wärmeverteilung in den Wärmetauschern. Die Energiebilanz für die Wärmetauscher-Nodes wird anhand von Gl. 4.2 beschrieben. Da in der Matlab-Ausführung des Modells keine Mantelwärmetauscher betrachtet werden, entfallen die Verluste der WT an die Umgebung aus [3].

$$\frac{V_{hx} \cdot \rho_{hx} \cdot c_{p,hx}}{n_{hx}} = \xi_5 \cdot \dot{m}_{hx} \cdot c_{p,hx} \cdot \left(\vartheta_{i-1,j} - \vartheta_{i,j}\right) + \xi_6 \cdot \dot{m}_{hx} \cdot c_{p,hx} \cdot \left(\vartheta_{i,j} - \vartheta_{i+1,j}\right) + \frac{(UA)^*_{hx,S}}{n_{hx}} \cdot \left(\vartheta_{i,2} - \vartheta_{i,j}\right) \quad (4.2)$$

$\xi_5 = 1$, wenn $\dot{m}_{hx} > 0$, sonst $\xi_5 = 0$

$\xi_6 = 1$, wenn $\dot{m}_{hx} < 0$, sonst $\xi_6 = 0$

Für die Temperaturen $\vartheta_{i-1,j}$ und $\vartheta_{i+1,j}$ [$j = 1$ oder 3] wird die Eintrittstemperatur des Wärmetauschers eingesetzt, sofern sie am entsprechenden Node vorhanden ist.

5. Anwendung der MATLAB-Funktion *MultiportStoreModel.m*

Für die Ausführung der MATLAB-Funktion *MultiportStoreModel.m* gibt es zwei Fälle:

1. Die Initialisierung, bei der bestimmte Anfangszustände angenommen werden müssen – diese Ausführung muss in jeder Simulation nur einmal angewendet werden und hat eine etwas längere Rechenzeit.
   Die Initialisierung erfolgt mit der Syntax:

```
[Tho, Qls, Qlbot, Qltop, Qlsx, Qd, Ts, Taux, Qh, Qhs, Thm, Qaux, Tz hxdat, storedat]...
           = MultiportStoreModel(@solver, Hs, Vs, dz, UA_a, UA_u, UA_o, UA_h, Nmax, zdi,...
             zdo, scdp, aux, zhi, zho, Vh, zs, Tdi, mdotd, Thi, mdoth, Paux, delt, Tamb);
```

2. Alle weiteren Ausführungen unter Berücksichtigung des vorherigen Zeitschrittes: Zu jedem Zeitschritt werden für den darauf folgenden Zeitschritt wichtige Größen gespeichert. Die „Rückführparameter“ eines vorherigen Zeitschrittes werden für den aktuellen Zeitschritt als Eingabeparameter eingegeben.
   Jede nicht-initialisierende Ausführung erfolgt mit der Syntax:

```
[Tho, Qls, Qlbot, Qltop, Qlsx, Qd, Ts, Taux, Qh, Qhs, Thm, Qaux, Tz hxdat, storedat] ...
           = MultiportStoreModel(@solver, Hs, Vs, dz, UA_a, UA_u, UA_o, UA_h, Nmax, zdi,...
             zdo, scdp, aux, zhi, zho, Vh, zs, Tdi, mdotd, Thi, mdoth, Paux, delt, Tamb,...
             Tz, hxdat, storedat);
```

5.1 Eingabeparameter

@solver	Funktionshandle [z. B. @ode23] zur Festlegung des zu verwendenden Solver

5.1.1 Speicherparameter

Hs	Höhe des Speichers	[m]
Vs	Speichervolumen	[m^3]
UA_a	Nennwärmeverlustrate des Speichers	$\left[\frac{W}{K}\right]$
UA_u	Wärmeverlustrate des Speicherdeckels	$\left[\frac{W}{K}\right]$
UA_o	Wärmeverlustrate des Speicherbodens	$\left[\frac{W}{K}\right]$
UA_h	Wärmeverlustrate zwischen Wärmetauscher und Speicher (4x1-Vektor)	$\left[\frac{W}{K}\right]$
Nmax	Anzahl Volumensegmente im Speicher	-

5.1.2 Doppelports

zdi	Relative Höhen der Doppelport-Eingänge (Nx1-Vektor), wobei N <= Nmax	-
zdo	Relative Höhen der Doppelport-Ausgänge (Nx1-Vektor), wobei N <= Nmax	-
scdp	Nx1-Vektor entsprechend zdi und zdo: scdp = 1 für geschichtetes Laden durch einen Doppelport und scdp = 0 für ungeschichtetes Laden.	-

5.1.3 Heizstab

Für die Parameter des Heizstabes wird ein Struct „aux“ verwendet, welches folgende Felder enthält:

var	var = 1 für variable Leistung var = 0 für konstante Leistung var = 2, wenn kein Heizstab verwendet wird	-
pos	Relative Höhe des Heizstabes	-
T	Gesetzte Temperatur für den Regler (falls var = 0)	[°C]

5.1.4 Wärmetauscher

Alle Wärmetauscher werden durch 4x1-Vektoren parametriert. Dabei ist für einen nicht eingesetzten Wärmetauscher an der entsprechenden Stelle in den Vektoren NaN zu setzen (z. B. `Vh = [0,01; nan; 0,01; 0,01];` wenn der zweite Wärmetauscher nicht angeschlossen ist.

zhi	Relative Höhen der WT-Eingänge (4x1-Vektor)	-
zho	Relative Höhen der WT-Ausgänge (4x1-Vektor)	-
Vh	Volumen der Wärmetauscher (4x1-Vektor)	[m³]

5.1.5 Temperatur-Sensoren

Für die Ausgabe von Temperaturen an ausgewählten Positionen im Speicher kann ein entsprechender Vektor mit den relativen Höhen der Sensoren zs eingegeben werden. Ist dies nicht erwünscht, kann als Input auch nan eingegeben werden.

5.1.6 Physikalische Inputs

Tdi	Eingangstemperaturen der N genutzten Doppelports (Nx1-Vektor)	[°C]
mdotd	Massenströme der Doppelports (Nx1-Vektor)	$\left[\frac{kg}{m^3}\right]$
Thi	Eingangstemperaturen der Wärmetauscher 1…4 (4x1-Vektor; NaN für nicht genutzte WT)	[°C]
mdoth	Massenströme der Wärmetauscher 1…4 (4x1-Vektor; NaN für nicht genutzte WT)	$\left[\frac{kg}{m^3}\right]$
Paux	Heizstab-Leistung (NaN, wenn kein Heizstab verwendet wird)	[W]
delt	Externe Zeitschrittweite der Simulation	[s]
Tamb	Umgebungstemperatur	[°C]
Tz	Bei Initialisierung: Startwerte für die Temperaturverteilung im Speicher Bei folgenden Zeitschritten: Temperaturverteilung im Speicher des vorherigen Zeitschrittes [Nmax x 1-Vektor	[°C]

5.2 Ausgabeparameter

Kann ein Parameter aufgrund von fehlenden Eingangsparametern nicht berechnet werden (z. B. wenn ein Parameter, wie der Heizstab nicht genutzt wird), wird NaN ausgegeben.

Tho	Ausgangstemperaturen der Wärmetauscher (NaN bei nicht genutzten WT)	[°C]
Qls	Gesamter Wärmeverlust des Speichers an die Umgebung	[W]
Qlbot	Wärmeverlust des Speicherbodens an die Umgebung	[W]
Qltop	Wärmeverlust des Speicherdeckels an die Umgebung	[W]
Qlsx	Wärmeverluste der einzelnen Nodes an die Umgebung (Nmax x 1-Vektor)	[W]
Tdo	Ausgangstemperaturen der Doppelports	[°C]
Qd	Wärmeströme durch die Doppelports	[W]
Ts	Temperaturen bei den Temperatursensoren	[°C]
Taux	Temperatur an der Position des Heizstabes	[°C]
Qh	Wärmeleistung durch die Wärmetauscher (4x1-Vektor)	[W]
Qhs	Nutz-Wärmeleistung zwischen WT und Speicher (4x1-Vektor)	[W]
Thm	Mittlere Temperaturen der WT (4x1-Vektor)	[°C]
Qaux	Tatsächlich genutzte Heizstab-Leistung (bei aux.var = 0 kann diese aufgrund von mehrfachem Ein- und Ausschalten innerhalb eines Zeitschrittes geringer sein, als die eingestellte Leistung)	[W]

5.3 Rückführparameter

Die folgenden Parameter werden von der Funktion ausgegeben, um im darauf folgenden Zeitschritt als Eingabeparameter genutzt zu werden.

Tz	siehe Tz (Eingabeparameter)	[°C]
hxdat	Struct mit Informationen zu den Wärmetauschern (Anzahl besetzter Nodes, Temperaturen der WT-Nodes, Indizes, Massenströme der vergangenen Zeitschritte, logische Indizes ξ_3 und ξ_4)	-
storedat	Struct mit Informationen zu dem Speicher (Temperaturen der genutzten WT an den Speichernodes, Wärmeströme zwischen Speichernodes und WT-Nodes, Verlustleistungen oben/unten)	-

5.4 Fehlermeldungen

Um eine möglichst geringe Rechenzeit zu erzielen wird in der Funktion auf die Überprüfung von Fehleingaben verzichtet. Im Folgenden werden Hinweise aufgelistet, um Fehler in der Anwendung zu vermeiden.

Wie in Kapitel 4 erwähnt, ist das Vorzeichen der Massen- und Wärmeströme von großer Bedeutung für die Funktionsweise des Speichers. Massenströme von unten nach oben werden mit positivem Vorzeichen angegeben und Massenströme von oben nach unten mit negativem. Wärmeströme werden immer von dem jeweiligen Bauteil aus betrachtet. Ein Wärmestrom, der vom Speicher aus gesehen den Speicher verlässt wird mit negativem Vorzeichen angegeben und ein Wärmestrom, der in den Speicher fließt mit positivem. Dies ist vor allem für die Unterscheidung zwischen Be- und Entladen des Speichers von Bedeutung. Die Wärmeströme Qh und Qhs werden analog aus Sicht der Wärmetauscher betrachtet. Hat Qhs z. B. ein positives Vorzeichen, bedeutet dies, dass Wärme vom Speicher in den Wärmetauscher fließt.

Es kann vorkommen, dass eine Situation simuliert wird, in der der Speicher mit konstanter Leistung be- oder entladen wird. In diesem Fall müssen die benötigten Massenströme extern zu jedem Zeitschritt anhand der Austrittstemperaturen der vorangehenden Zeitschritte neu ermittelt werden. Dabei kann es vorkommen, dass MATLAB folgende (oder eine ähnliche) Warnung anzeigt:

```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND =  2.144711e-16.
> In ode23s at 364
  In MultiportStoreModel at 590
```

Dies ist ein Hinweis, dass ein Massenstrom-Input entweder als extrem kleine Zahl (wie im obigen Beispiel), als extrem große Zahl, als $-\infty, \infty$ oder als NaN eingegeben wurde. Zum Beispiel lässt sich der Fehler bei $\dot{m}_{Dp} \rightarrow \infty$ damit erklären, dass der Ladezustand des Speichers nicht mit der Enladeleistung mithalten kann. Somit sinkt die Temperaturdifferenz zwischen Doppelport Ein- und Austritt auf nahezu Null und der Massenstrom muss extrem hoch gewählt werden, damit die Entladeleistung konstant bleibt. In der Realität ist dies nicht möglich. Deswegen müssen bei konstanter Be- und Entladeleistung die Massenstrom-Inputs begrenzt werden.

6. Geschwindigkeitsoptimierung

MATLAB hat den Vorteil gegenüber vielen anderen Simulationsprogrammen, dass Iterationen durch die Rechnung mit Matrizen unter logischer Indizierung von Schnittpunkten vermieden werden können. So kann z. B. eine Temperaturermittlung in VBA durch Iteration, die ca. 60 Rechenschritte benötigt in MATLAB durch logische Indizierung von Matrizen in vier Rechenschritten zu Selben Lösung führen. Dennoch führt bei der Lösung der gleichzeitig geschehenden Differentialgleichungen 4.1 und 4.2, sowie der Gleichung 2.2 kein Weg an der Iteration vorbei, da in diesen Gleichungen bereits zweidimensionale Matrizen verwendet werden und eine Verwendung von dreidimensionalen Matrizen viel zu viel Arbeitsspeicher in Anspruch nehmen würde, obwohl dies in MATLAB theoretisch möglich wäre. Für die Lösung der voneinander abhängigen Gleichungen 2.2, 4.1 und 4.2 wird ein sogenannter ODE-Solver angewendet. ODE steht für „ordinary differential equations“, was definiert ist als eine Differentialgleichung, welche ein oder mehrere Ableitungen einer abhängigen Variablen y enthält, bezogen auf eine einzelne unabhängige Variable t, z. B. der Zeit [8]. Dies trifft auf die o. g. Gleichungen zu. Ein großer Vorteil der ODE-Solver in MATLAB ist, dass sie fähig sind, intern logische Operationen zu berücksichtigen, was vor Allem bei der internen Regelung des Heizstabes notwendig ist (bei aux.var = 0). Jedoch wird durch jede logische Operation innerhalb des Solvers die Rechenzeit signifikant verlängert, sodass diese möglichst vermieden werden sollten.

Tabelle 6.1: Liste der ODE-Solver [8], die in MATLAB zur Verfügung stehen.

Solver	**Algorithmus**	**Verfahren**	**Anwendbar auf das Modell?**
ode45	Runge-Kutta 4,5 (explizit)	Einzelschritt	Ja
ode23	Runge-Kutta 2,3 (explizit)	Einzelschritt	Ja
ode113	Adams-Bashforth-Moulton PECE	Mehrschritt	Ja
ode15s	Numerische Differentiations-Formeln (NDFs)	Mehrschritt	Beschränkt[1]
ode23s	Modifizierte Rosenbrock-Funktion (2. Ordnung)	Einzelschritt	Beschränkt[1,2]
ode23t	Trapezregel	k. A.	Beschränkt[1]
ode23tb	TR-BDF2 – Runge-Kutta (implizit)	k. A.	Beschränkt[1]
ode15i	BDFs – Runge-Kutta (implizit)	k. A.	Nein

[1] Bei aux.var = 0 steigt die Anzahl Rechenschritte in unermessliche Höhen, sodass der Rechner bei der Analyse hängen blieb.
[2] Führt bei aux.var = 0 und bei bestimmten Δt_{ext} zu Fehlermeldungen.

Aufgrund der hohen Anzahl an Rechenschritten bei einer Iteration entsteht in der Funktion MultiportStoreModel.m die größte Zeiteinbuße durch den ODE-Solver. Aus diesem Grund werden im Folgenden die Einflüsse unterschiedlicher ODE-Solver auf die Rechenzeit des Modells untersucht.

6.1 Auswahl der Solver

In Tabelle 6.1 sind die zur Verfügung stehenden Solver aufgelistet. Fast alle führen zu nahezu der gleichen Lösung und können somit auf das Multi-Port-Speichermodell angewendet werden. Eine Ausnahme bildet der ode15i. Dafür wird zusätzlich zu der Startbedingung für die Funktion y auch eine Startbedingung für die Funktion $\frac{\partial y}{\partial t} \coloneqq y'$ benötigt, welche für das Multi-Port-Speichermodell nicht von Bedeutung ist. Außerdem muss dafür $f(t, y, y') = 0$ sein, was in diesem Modell nicht der Fall ist. Da jeder Solver ein eigenes Verfahren, bzw. einen eigenen Algorithmus anwendet, ist auch die Geschwindigkeit eines Solver von dem Anwendungsfall und von der externen Zeitschrittweite abhängig.

In Bild 6.1.1 wurden die Geschwindigkeiten der anwendbaren Solver für unterschiedliche Betriebsarten des Speichers in Abhängigkeit der externen Zeitschrittweite Δt_{ext} der Simulation untersucht. Bei größer werdendem Δt_{ext} nimmt die benötigte Anzahl an Anwendungen der Funktion, um ein bestimmtes Zeitfenster zu simulieren, ab. Jedoch muss der Solver bei größerem Δt_{ext} mehrere Iterationen und somit mehrere interne Zeitschritte Δt_{int} berechnen, um eine ausreichende Genauigkeit zu erzielen. Somit ist der Zusammenhang zwischen Rechenzeit und externer Zeitschrittweite nicht linear. Der Teil der Differentialgleichung, der die Durchmischung und die Doppelports enthält, kommt in jeder Anwendung vor. Aus diesem Grund werden die folgenden Betriebsarten untersucht: Betrieb nur mit Doppelports, mit Doppelports und Wärmetauscher, mit Doppelports und Heizstab (variable Leistung) und mit Doppelports, Wärmetauscher und Heizstab (variable Leistung). Bei der Betriebsart mit Heizstab konstanter Leistung wird lediglich die Heizstableistung zur Energiebilanz addiert, sodass es bezüglich der Solver-Geschwindigkeiten keine Änderung gegenüber der Betriebsart ausschließlich mit Doppelports gibt.

Bild 6.1.2 verdeutlicht, dass es sinnvoll ist, den verwendeten Solver des Modells dynamisch anhand der Betriebsweise und anhand von Δt_{ext} zu anzupassen. Bei einer Anwendung, in der ausschließlich mit Doppelports be- und entladen wird, kommt ode23 in Frage. Kommt ein Wärmetauscher hinzu, wird bei Zeitschrittweiten ab ca. 15 min auf ode23tb umgeschaltet. Bei Heizstab konstanter Heizleistung anhand eines internen Reglers wird unabhängig von Δt_{ext} auf ode45 umgestellt und im Betrieb mit Verwendung von allen Möglichkeiten wird je nach Δt_{ext} entweder ode45, ode23 oder ode113 verwendet. Die Solver für steife Differentialgleichungen, ode15s, ode23s und ode23t sind in jedem Fall zu langsam.

> ***Hinweis:*** *Die Verwendung eines Heizstabes mit aux.var = 0 ermöglicht zwar schnelle Reaktionen auf Unterschreitung der Solltemperatur im entsprechenden Node, jedoch wird dadurch die Rechenzeit drastisch erhöht (s. Bild 6.1.1 und Bild 6.1.2). Ist die Simulation zu langsam, kann alternative aux.var = 1 gewählt werden und ein Temperatursensor an entsprechender Position angebracht werden, damit die Regelung extern erfolgen kann. Dies ist vor allem bei Simulationen hoher Auflösung zu empfehlen, da die Reaktionsgeschwindigkeit mit geringerer Auflösung steigt.*

Einfluss der ODE-Solver auf die Geschwindigkeit

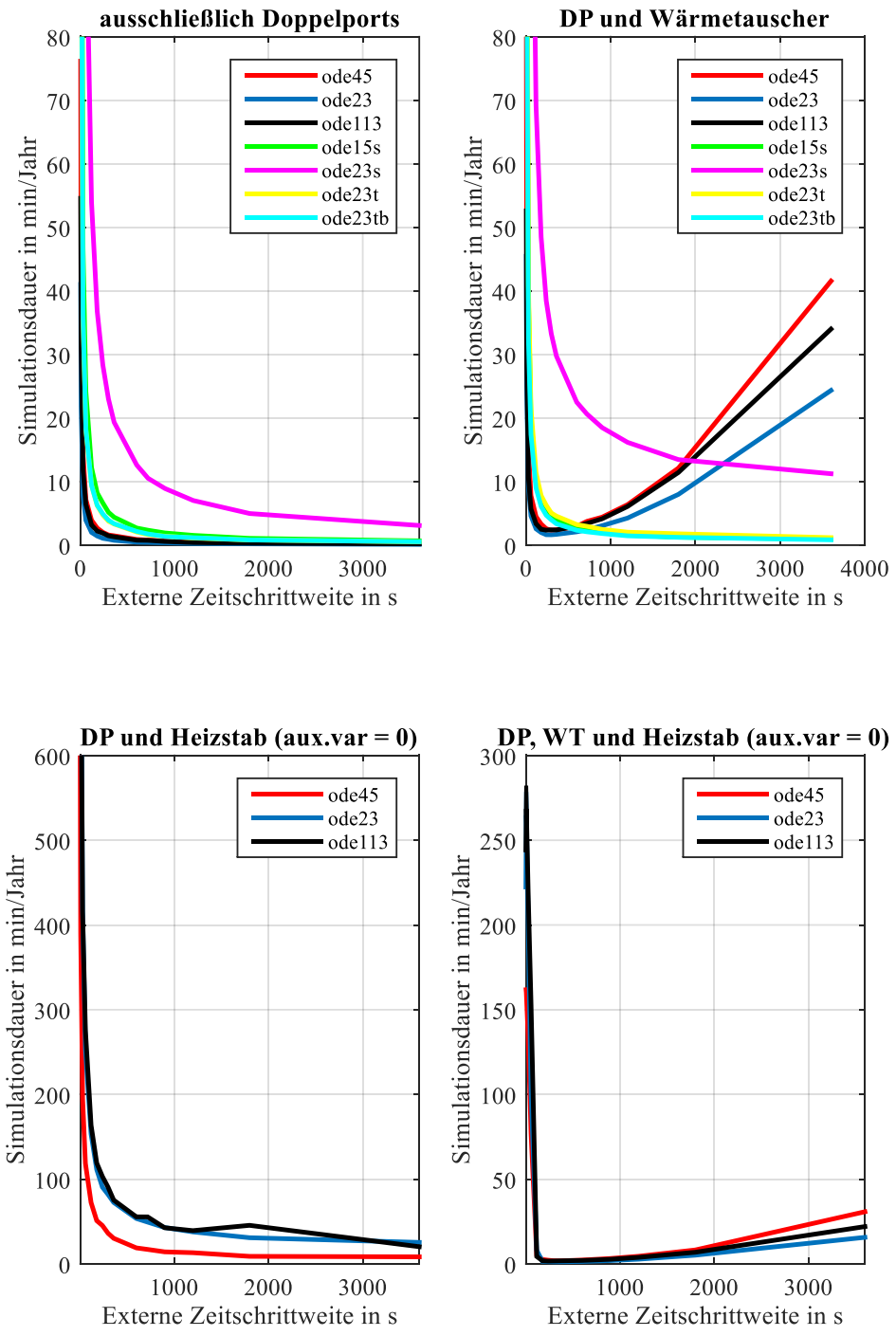

Bild 6.1.1: Einfluss der ODE-Solver auf die Simulationsgeschwindigkeit bei unterschiedlichen Betriebsweisen. Um die Simulationsdauer zu schätzen, wurde jeweils unter Zeitmessung 2 Stunden Betrieb simuliert und die Dauer der Simulation auf ein Jahr hochkalkuliert. Die Angabe in min/Jahr dient dem Vergleich der Solver untereinander. Je nach Rechenleistung und -Belastung kann die tatsächliche Dauer variieren. Verwendete Rechenleistung: CPU: 3,0 GHz (dual-core); RAM: 16 GB

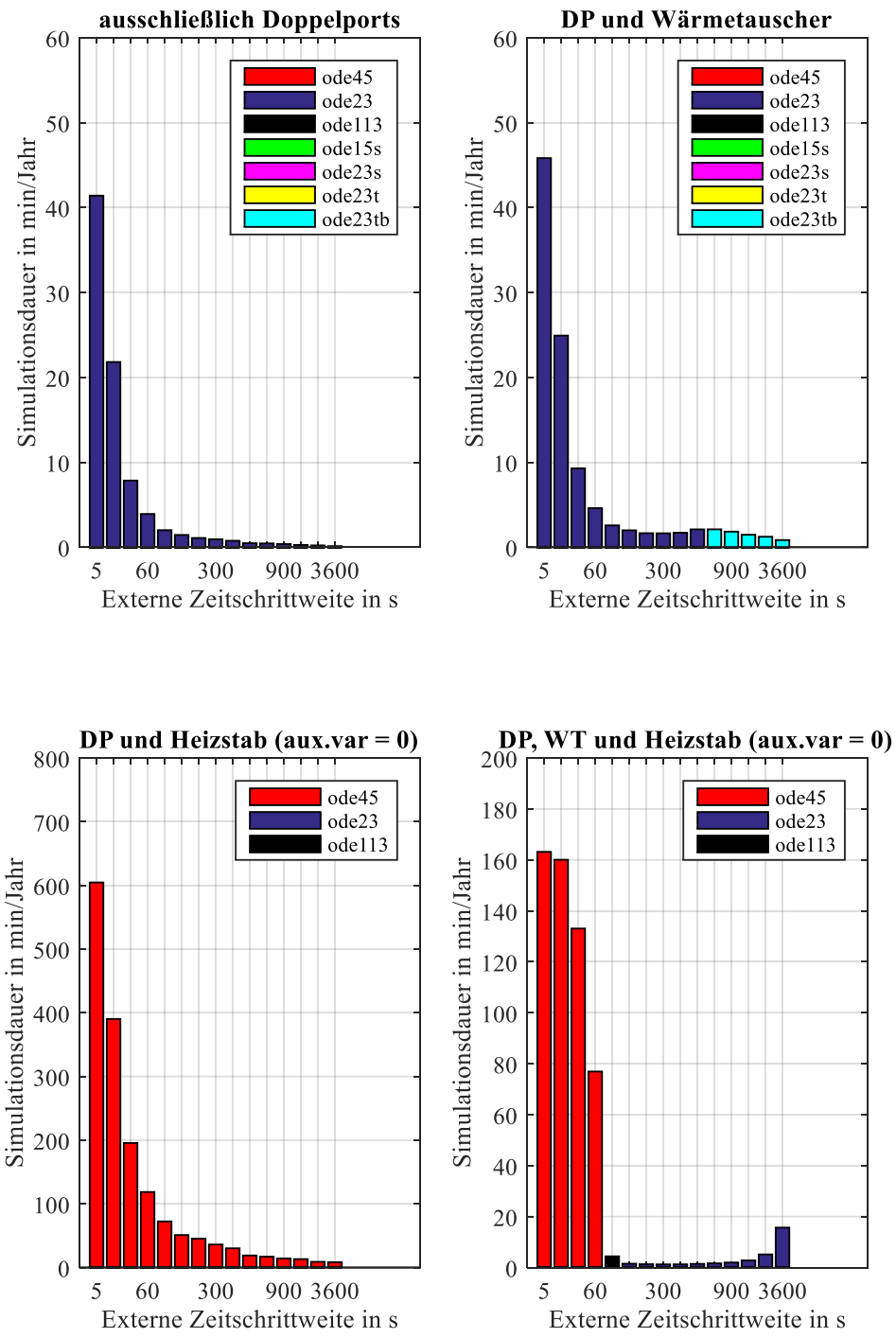

Bild 6.1.2:Darstellung der Geschwindigkeit des schnellsten Solver für jede externe Zeitschrittweite Δt_{ext} bei vier Betriebsarten des Speichermodells nach der Untersuchung. Verwendete Rechenleistung: CPU: 3,0 GHz (dual-core); RAM: 16 GB

Bezüglich des Komforts für den Anwender erscheint es attraktiv, den idealen Solver innerhalb der Funktion dynamisch auszuwählen. So kann bei oft wechselnder Betriebsweise viel Zeit eingespart werden. Jedoch erhöht die IF-Abfrage die Anzahl Rechenschritte, vor Allem bei hoch aufgelösten Simulationen. Im schlimmsten Fall kann die Dauer der Simulation fast verdoppelt werden. In Bild 6.1.3 sind die Geschwindigkeiten, sowie die Geschwindigkeitseinbußen einer dynamischen Solverauswahl gegenüber der festen Implementierung eines Solver dargestellt.

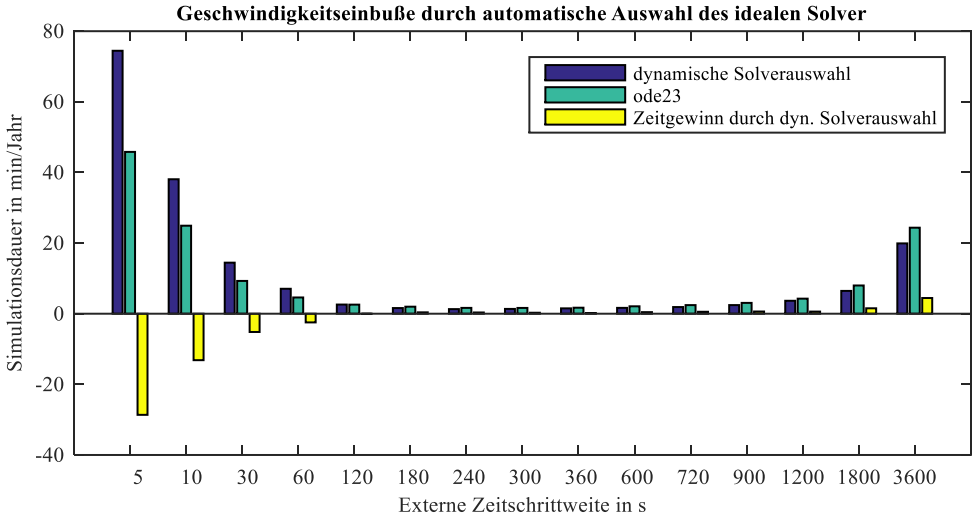

Bild 6.1.3: Geschwindigkeiten der festen Einbindung des ode23 in das Modell, der dynamischen Solverauswahl und der Zeitgewinn bei dynamischer Solverauswahl (hier ode23/ode23tb) gegenüber der festen Einbindung des ode23 für den Betriebsfall: Doppelports und Wärmetauscher. Ein negativer Zeitgewinn beschreibt eine Geschwindigkeitseinbuße.

Um die Geschwindigkeitseinbußen bei geringem Δt_{ext} zu vermeiden, werden die Inputs der MATLAB-Funktion um den Solver ergänzt. Dieser ist als „Handle“ mit @solver [z. B. @ode23 oder @ode45] einzugeben. Für die korrekte Auswahl wird in der Funktionshilfe zusätzlich auf dieses Dokument hingewiesen. Die IF-Abfrage zur automatischen Solverauswahl verbleibt auskommentiert in der Funktion für den Fall, dass dem Nutzer die Zeiteinbuße für den Komfort in der Anwendung wert ist. In diesem Fall muss der Input ebenso entfernt werden. Auf die entsprechenden Stellen im Quellcode der Funktion sind, die de/aktiviert werden müssen wird mit Kommentaren hingewiesen.

6.2 Erhöhen der Fehlertoleranz

In 6.1 wurden bisher nur Solver mit den MATLAB-Standard-Einstellungen verwendet. Über die Funktion „*odeset“* lassen sich diese verändern. Einen großen Einfluss auf die Geschwindigkeit der ODE-Solver hat die zugelassene (absolute) Fehlertoleranz. Diese ist standardmäßig auf $\pm 10^{-6}$ gesetzt [9], d. h. es wird so lange iteriert, bis die Lösung der Differentialgleichung auf die sechste Nachkommastelle genau berechnet ist. Für die Zwecke dies Speichermodells ist eine solch geringe Toleranz viel zu niedrig, da der Fehler, bedingt durch Vernachlässigungen im Modell weitaus höher ist. In Bild 6.2 wird die Simulationsdauer

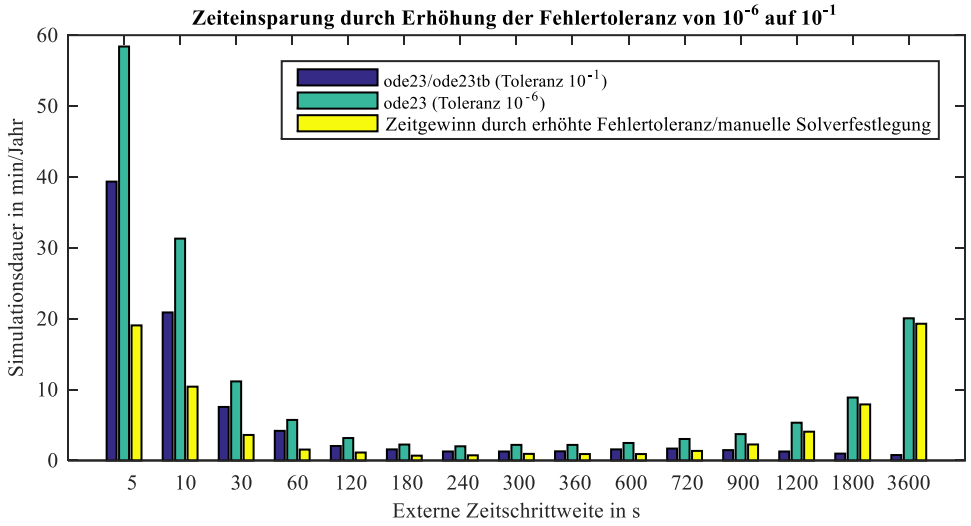

Bild 6.2: Gegenüberstellung der Simulationsdauer bei manueller Solver-Eingabe und Fehlertoleranz von $\pm 10^{-1}$ und der Simulationsdauer bei Anwendung mit festem Solver und Fehlertoleranz von $\pm 10^{-6}$ [Betriebsfall: Doppelports und Wärmetauscher]

bei manueller Solver-Eingabe und erhöhter Fehlertoleranz der festen ode23-Simulationsdauer ohne Toleranzerhöhung gegenübergestellt. Vor allem sind bei hohen Zeitschrittweiten, bei denen die Genauigkeit des Solver die geringste Bedeutung hat, wird in diesem Beispiel ca. 30 % der Rechenzeit eingespart. Bezieht man zusätzlich alle Zeitschrittweiten und die optimale Solver-Auswahl mittels Eingabeparameter ein, beträgt die Zeiteinsparung gegenüber der ursprünglichen Standardausführung bis zu 96 %.

6.3 Verwendung von Rückführparametern

Eine dritte Maßnahme zur Minimierung der Rechenzeit ist die Unterscheidung zwischen Initialisierung und aller restlichen Schritte. Alle Berechnungen zur logischen Indizierung (siehe Gl. 4.1 und 4.2) nehmen Rechenzeit in Anspruch. Wenn sich an den Anschlüssen nichts ändert, ändert sich nichts an der Indizierung. Um unnötige Wiederholungen der Berechnungen zu umgehen, werden u. a. die logischen Switches aus Gl. 4.1 und 4.2 nur bei Initialisierung, bzw. bei Änderung der Anschlüsse innerhalb einer Simulation berechnet und als „Rückführparameter“ in Struct-Feldern abgespeichert. Nach der Initialisierung müssen diese nur noch aus den zurückgeführten Structs geladen werden.

7. Umsetzung in MATLAB

Im Folgenden wird versucht, die Vorgehensweise zur Berechnung des Modells in MATLAB detailliert zu erläutern, sodass alle Berechnungen für den Nutzer nachvollziehbar sind. Im Anhang A befindet sich der Quellcode für die Funktion.

Zunächst wird anhand der Anzahl an Inputs festgestellt, ob es sich um den Initialisierungsschritt handelt, oder nicht. Bei der Initialisierung sind 24 Inputs vorhanden, bei allen weiteren Schritten 26. Zwecks Komfort für den Nutzer sind alle Anschlusshöhen relativ einzugeben. Diese müssen den entsprechenden Nodes zugeteilt werden. Alle Vektoren der Dimension $(N_{max}, 1)$ werden so indiziert, dass $\vec{v}(1)$ dem untersten Node i entspricht und $\vec{v}(N_{max})$ dem obersten. Eine Ausgabe der Temperaturen in MATLAB ist also folgendermaßen auszuwerten:

$$\vec{v} = \begin{pmatrix} i = 1 \\ \dots \\ i = N_{max} \end{pmatrix}$$

Also sozusagen „auf den Kopf gestellt“.

7.1 AUX-Switches ξ_{ak} und ξ_{av}

ξ_{ak} und ξ_{av} werden als Nullvektoren mit den Dimensionen $(N_{max}, 1)$ vorinitialisiert. Je nach Betriebsart [aux.var], werden die Positionen von ξ_{ak} *oder* von ξ_{av}, an denen sich der Heizstab befindet Eins gesetzt. Im Fall, dass kein Heizstab verwendet wird, verbleiben die Switches als Nullvektoren und der Index für die Node-Position des Heizstabes wird Eins gesetzt, um Indizierungsfehler im Solver zu vermeiden.

7.2 Umrechnung der Eintrittspositionen für den Fall geschichtetes Be-/Entladen mit Doppelports

Die Inputs [zdi], [zdo] und [scdp] sind der Dimension $(n, 1)$, wobei n_D die Anzahl angeschlossener Doppelports ist. Für jedes $\overrightarrow{scdp}(k) = 1$ wird geschichtet be- oder entladen. Für die entsprechenden Indizes k wird, wenn der jew. Doppelport-Massenstrom nicht Null ist, geprüft, an welchem Node i im Speicher die Temperatur gleich oder größer der Doppelport-Eingangstemperatur ist. Der nächste Node zum Eintritt, an dem dies zutrifft, wird als virtuelle Eintrittsposition festgehalten. Das Vorzeichen der Massenströme wird dann entsprechend der Ein- und Austrittspositionen angepasst (s. Kapitel 2.3). Zusätzlich wird eine Matrix $\dot{M}_{ik}$ mit den Dimensionen (N_{max}, n_D) erstellt, welcher anhand der Austritte und der virtuellen Eintritte dem Solver mitteilt, welche Massenströme durch welche Nodes fließen. Fließt ein Massenstrom $\dot{m}_k$ durch ein Node i ist $\dot{M}_{ik}(i, k) = 1$, ansonsten Null.

Weiterhin wichtig für den Solver ist die Feststellung, ob sich in einem Node ein Doppelport-Eingang befindet. Hierfür wird ein logischer Switch $\overrightarrow{tf}$ erstellt, sowie ein weiterer $(N_{max}, 1)$ – Vektor mit den Indizes k der den Doppelporteingängen zugehörigen Massenströme. Mithilfe dieser Variablen wird es dem Solver ermöglicht, die Speichernodes ohne Doppelport-Eingang getrennt zu berechnen, um die Eingangstemperatur mit einzubeziehen.

7.3 Wärmetauscher-Switches und Initialisierung der Temperaturen

Anhand der Position-Inputs der Wärmetauscher wird festgestellt, wie viele Wärmetauscher verwendet werden. Deswegen ist es wichtig, dass die Vektoren entsprechenden Ein- und Ausgänge die Dimension $(4{,}1)$ besitzen, auch wenn nicht alle vier Wärmetauscher verwendet werden. Dabei entspricht $\overrightarrow{zho}(1)$ der relativen Ausgangshöhe des Wärmetauschers 1 und $\overrightarrow{zho}(4)$ der relativen Ausgangshöhe des Wärmetauschers 4. Da WT1 und WT4, bzw. WT2 und WT3 sich nicht überschneiden dürfen, werden diese jeweils in einem Vektor der Dimension $(N_{max}, 1)$ zusammengefasst.

Die Temperaturverläufe in den Wärmetauscher-Nodes werden bei Initialisierung zunächst linear angenommen. Bei Entladen nimmt die Temperatur vom Eintritt bis zu der Temperatur des Speichers im Node an der Austrittsposition + 5 K ab, und bei Beladen nimmt sie bis zur Speichertemperatur im WT-Austritts-Node – 5 K zu (Die zum Anfang anzunehmenden Speichertemperaturen sind mittels Input anzugeben). Es wird also angenommen, dass der WT-Eintritt beim Entladen unter dem WT-Austritt liegt und beim Entladen analog; eine sinnvolle Betriebsweise mit minimalen Verlusten. Weil für o. g. Berechnungen mehrfach verkettete IF-Abfragen notwendig sind, und sie nicht in jedem Zeitschritt notwendig sind, werden die ermittelten Daten in einem Struct als Rückführparameter abgespeichert. Ändert sich die Position eines Wärmetauschers, werden die Berechnungen neu durchgeführt.

Für Gleichung 2.1 ist das Startverhalten eines Wärmetauschers von Bedeutung. Zu jedem Simulationszeitschritt wird der aktuelle Massenstrom jedes Wärmetauschers gespeichert und mit dem des letzten Zeitschrittes verglichen. Ist ein Massenstrom des letzten Zeitschrittes Null und der des aktuellen Zeitschrittes größer, als Null, wird dem Solver mittels logischem Switch mitgeteilt, dass es sich um ein Anfahren handelt. Die logischen Switches der Wärmetauscher für die Gleichungen 4.1 und 4.2 werden anhand des Vorhandenseins der Wärmetauscher in den Speichernodes (ξ_3 und ξ_4) und anhand der Fließrichtung (ξ_5 und ξ_6) ermittelt. ξ_3 und ξ_4 haben die Dimensionen $(N_{max}, 1)$, ξ_5 und ξ_6 haben die Dimensionen $(4{,}1)$.

7.4 Berechnung der Energiebilanzen mit dem Solver

Aufgrund der Übergabe der Solver-Auswahl an den Nutzer, ist es möglich, jeden ODE-Solver (außer ode15i) anzuwenden. Empfohlen wird eines der Solver nach Kapitel 6.1. Verwendet werden also die Algorithmen Runge-Kutta-4,5 (explizit), Runge-Kutta-2,3 (explizit), Adams-Bashforth-Moulton PECE und TR-BDF2 – Runge-Kutta (implizit). Auf die Funktionsweise der Algorithmen wird hier nicht näher eingegangen. Die Anwendung in MATLAB ist für jeden Solver dieselbe.

Es wird zunächst eine ode-Funktion erstellt; im Fall des Speichermodells als „nested function“, also als eingebettete Funktion, in der der Einsatz von globalen Variablen möglich ist, die außerhalb der „nested function“ definiert und übergeben werden können. In dieser Funktion wird eine Differentialgleichung der Form $\frac{\partial y}{\partial t} = f(t, y)$ definiert. Dabei können mehrere voneinander abhängige Differentialgleichungen zusammengefasst werden, sodass in dem Solver die Temperaturen $\frac{\partial T}{\partial t} = f(t, T)$ und die Energien $\frac{\partial \dot{Q}}{\partial t} = f\left(q, \dot{Q}\right)$ mit demselben Solver

```
%Energiebilanz für den Wärmetauscher 4
Tdot(Nmax+mm4(1)) = length(mm4)./(Vh(4).*rho_h.*cp_h(4))...
    .*(ksi5(4).*mdoth(4).*cp_h(4).*(Thi(4)-T(Nmax+mm4(1)))+ksi6(4).*mdoth(4).*cp_h(4).*(T(Nmax+mm4(1))-T(Nmax+mm4(2)))...
    + UA_h14(mm4(1))./length(mm4).*(T(mm4(1))-T(Nmax+mm4(1))));
%T(end-2) = Thx4_aus
for k = 2:length(mm4)-1
    Tdot(Nmax+mm4(k)) = length(mm4)./(Vh(4).*rho_h.*cp_h(4))...
        .*(ksi5(4).*mdoth(4).*cp_h(4).*(T(Nmax+mm4(k-1))-T(Nmax+mm4(k)))+ksi6(4).*mdoth(4).*cp_h(4).*(T(Nmax+mm4(k))-T(Nmax+mm4(k+1)))...
        + UA_h14(mm4(k))./length(mm4).*(T(mm4(k))-T(Nmax+mm4(k))));
end
Tdot(Nmax+mm4(end)) = length(mm4)./(Vh(4).*rho_h.*cp_h(4))...
    .*(ksi5(4).*mdoth(4).*cp_h(4).*(T(Nmax+mm4(end-1))-T(Nmax+mm4(end)))+ksi6(4).*mdoth(4).*cp_h(4).*(T(Nmax+mm4(end))-Thi(4))...
    + UA_h14(mm4(end))./length(mm4).*(T(mm4(end))-T(Nmax+mm4(end))));
```

Bild 7.4: Ausschnitt aus der ode-Funktion: Erstellung DGL für die Energiebilanz des Wärmetauschers 4 anhand der Anzahl Nodes, die durch WT4 besetzt sind

gelöst werden. Die logischen Switches und physikalischen Größen, die bereits vor dem Solver definiert werden, dienen als globale Variablen. Die Anzahl der Gleichungen hängt von N_{max} und von der Anzahl Nodes, die von den Wärmetauschern besetzt werden, ab. Sie werden anhand dieser Größen in einer Schleife definiert. Bild 7.4 zeigt einen Ausschnitt aus dem Quellcode der ode-Funktion, in dem die Energiebilanz des WT4 anhand der Anzahl Nodes, die durch WT4 besetzt sind [*length(mm1)*], definiert wird.

In der ode-Funktion „*tempsolver*“ werden die Differentialgleichungen nach Gl. 2.2, Gl. 2.3, Gl. 4.1 und Gl. 4.2 zusammengefasst. Zunächst werden die Temperaturverläufe in den Speichernodes, sowie in den Wärmetauschernodes ermittelt, dann die Energieverluste des Speichers an die Umgebung, die Energieströme zwischen Wärmetauscher und Speicher, die Energie durch die Wärmetauscher und der Energieverbrauch des Heizstabes. Dafür werden in derselben Funktion die entsprechenden Zeilen aus den o. g. Gleichungen als eigene Gleichungen definiert. Für die Temperaturen wird jeweils mit $\frac{N_{max}}{V_S \cdot \rho_S \cdot c_{p,S}}$ bzw. mit $\frac{n_{hx}}{V_{hx} \cdot \rho_{hx} \cdot c_{p,hx}}$ multipliziert. Für den Heizstab mit aux.var = 0 muss intern logisch indiziert werden, d. h. die Energie des Heizstabes wird berechnet mit

$$\frac{\partial \dot{Q}_{aux,var=1}}{\partial t} = P_{aux}(\xi_{T_{SET}}) \qquad (7.4)$$

P_{aux} = Heizstableistung elektrisch ($\eta_{aux} = 1$) [W]

$$\xi_{T_{SET}} = \begin{cases} true, & \text{Temperatur an Position des Heizstabes} \geq T_{SET} \\ false, & \text{Temperatur an Position des Heizstabe} < T_{SET} \end{cases}$$

T_{SET} = vom Nutzer festgelegte Temperatur, die an Position des Heizstabes nicht unterschritten werden darf

Die Funktion wird mittels Funktionshandle „@“ an den Solver übergeben. Zusätzlich wird ein Zeitfenster (zwischen $t = t_0$ und $t = t_0 + \Delta t_{ext}$) übergeben, sowie die Startwerte für $y_n(t)$. Bei Initialisierung werden die Startwerte für die Temperaturverläufe nach Kapitel 7.3 definiert und die Energien Null, bzw. die Energie durch den Heizstab gleich der elektrischen Heizstab-Leistung gesetzt. Zu jedem Weiteren schritt werden die Rückführparameter aus dem letzten Zeitschritt als Startparameter an den Solver übergeben. Herausgegeben wird eine Matrix mit den aktuellen Temperaturen für jeden internen Zeitschritt Δt_{int} innerhalb des

Bereiches $(t_0, t_0 + \Delta t_{ext})$, sowie die Energien für jedes $\Delta t_{int.}$ Die Temperaturen können also direkt aus der Matrix an den Positionen $t = t_0 + \Delta t_{ext}$ entnommen werden. Für die Entnahme der Leistungen werden die Energien an den Positionen $t = t_0 + \Delta t_{ext}$ durch Δt_{ext} geteilt.

Die Wärme ströme durch die Doppelports werden nach der Ausführung des Solver mit $\dot{Q}_{Dp} = \dot{m}_{Dp} \cdot c_{p,S} \cdot (T_{Dp,in} - T_{Dp,out})$, wobei $T_{Dp,out}$ gleich der Speichertemperatur ist, die im Node des jew. Doppelport-Ausganges herrscht. Mit $T_{hx,out} = T_{hx,in} - \frac{\dot{Q}_{hx,S}}{c_{p,hx} \cdot \dot{m}_{hx}}$ wird für jeden Wärmetauscher die Ausgangstemperatur ermittelt. $\dot{Q}_{hx,S}$ ist die durch den Solver ermittelte Wärmeleistung, die von den jew. Wärmetauschern an den Speicher übergeben wird.

8. Validierung des MATLAB-Modells

Um die Korrektheit der Ausgaben der Funktion zu überprüfen, wird eine Simulation aus [10] (s. Bild 8.1) mit unterschiedlichen Betriebsweisen nachgebildet. In der Simulation wird ein Schichtenspeicher in INSEL zunächst eine Stunde lang mit konstanter Leistung (30 kW) beladen. Nach einem Stillstand von einer Stunde wird vier Stunden lang mit 20 kW beladen, bis ein SOC von 100 % erreicht ist. Dann erfolgt ein weiterer Stillstand von einer Stunde, worauf eine 3,5 stündige Entladung mit 20 kW folgt. Zuletzt wird unter Sicherung der Bereitschaftstemperatur weitere 3,5 Stunden lang mit 15 kW entladen. Mit welchem Mittel und in welcher Speicherschicht die Bereitschaftstemperatur gesichert wird, ist nicht angegeben, also werden in der Validierung drei Varianten durchgeführt: Nachheizung mit Heizstab, mit Wärmetauschern und mit Doppelports (geschichtetes Laden).

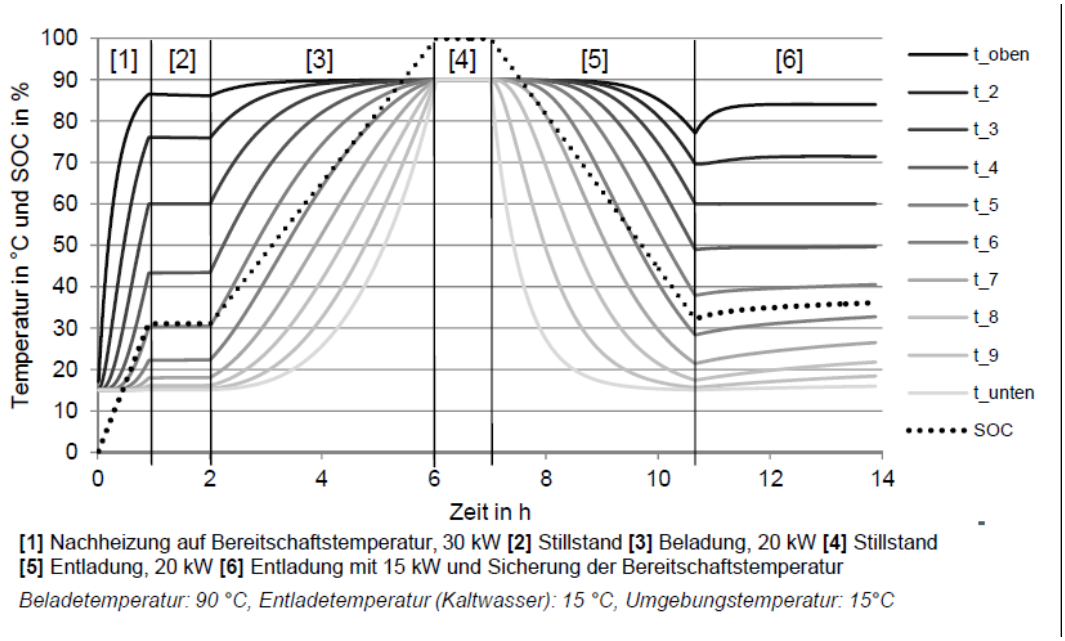

Bild 8.1: Simulation eines Schichtenspeichers in INSEL [10, s. 27]

Bei der Nachbildung der Simulation sind folgende Fehlerursachen, bzw. Ursachen für Abweichungen zu beachten: In dem MATLAB-Speichermodell sind Massenströme als Inputs festzulegen, nicht Eingangsleistungen. Für die Simulation müssen also die notwendigen Massenströme anhand von Temperaturdifferenzen zu Beginn von jedem Zeitschritt berechnet werden. So kann es vorkommen, dass bei niedrigen Temperaturdifferenzen (z. B. bei $SOC \rightarrow 100\,\%$) unrealistisch hohe Massenströme ($\dot{m} \rightarrow \infty$) eingegeben werden. Zudem sind nicht alle Speicherparameter (z. B. die Wärmeverlustraten) aus der nachzubildenden Simulation bekannt und müssen geschätzt werden. Statt eines Speichervolumens von 1000 l werden 1250 l angenommen. Für die Speicherhöhe werden 1,7 m angenommen, für die Wärmeverlustrate des Speichers 4,27 $\frac{\mathrm{W}}{\mathrm{K}}$, bzw. 0,33 $\frac{\mathrm{W}}{\mathrm{K}}$ am Boden und am Deckel. Bei der Nachheizung mittels Wärmetauscher wird eine Wärmeverlustrate (zwischen Speicher und WT) von 542 $\frac{\mathrm{W}}{\mathrm{K}}$ angenommen. Als Bereitschaftstemperatur werden in

der 8. Speicherschicht 60 °C festgelegt (bzw. 50 °C in der 7. Speicherschicht bei Nachheizung mit Heizstab). Ansonsten sind alle Parameter gleich den Parametern der INSEL-Simulation in [10]. Simuliert wird mit einer Zeitschrittweite Δt_{ext} von 60 s. In Bild 8.2 sind die Simulationsresultate dargestellt.

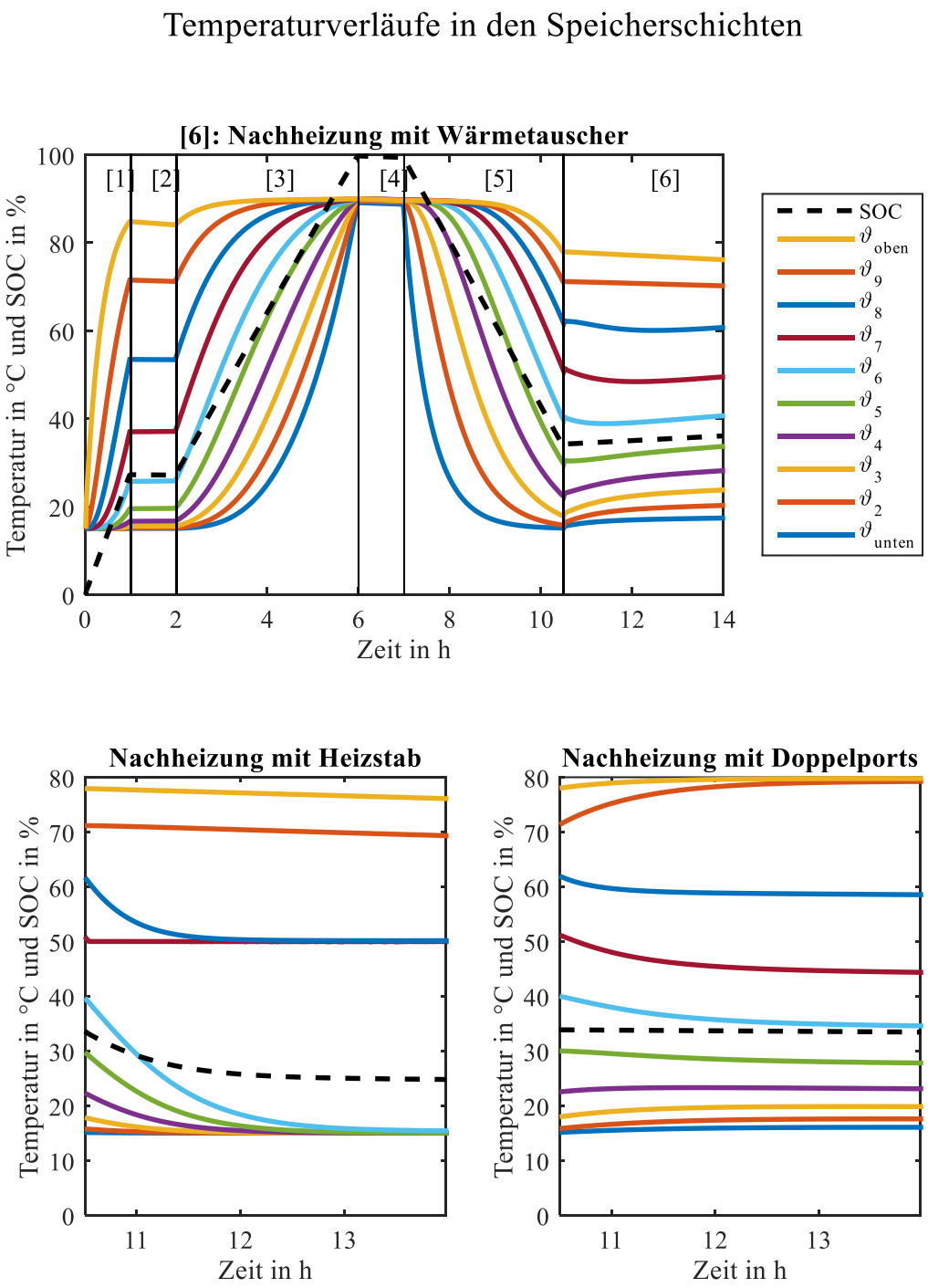

Bild 8.2: Temperaturverläufe in den Speicherschichten bei der Simulation nach [10, s. 27] mit dem MATLAB Speichermodell bei unterschiedlichen Betriebsweisen der Nachheizung

Die Verläufe bis zur Entladung mit Nachheizung ähnlich den Verläufen in der Simulation aus [10]. Es sind etwas höhere Wärmeverluste in den Stillstands-Abschnitten zu erkennen und die Nachheizung mit Wärmetauschern kommt der INSEL-Nachheizung am nächsten. Wird ausschließlich mit Heizstab nachgeheizt, kann die Bereitschaftstemperatur zwar (bei ausreichend hoher Heizstab-Leistung) aufrechterhalten werden, jedoch sinken die Temperaturen darunter weiterhin auf 15 °C. Der Massenstrom durch die Entladung mit Doppelport (in den Schichten 1…8) wirkt einer Durchmischung entgegen. Durch eine geschichtete Nachheizung mit Doppelports wird die Durchmischung gefördert und der SOC bleibt nahezu konstant. Die ermittelten Leistungen sind in einer realistischen Größenordnung und verhalten sich bei der Simulation in Koordination mit den Temperaturverläufen.

9. Quellenverzeichnis

[1] „Forschungsprojekte | pvspeicher.htw-berlin.de“. [Online]. Verfügbar unter: http://pvspeicher.htw-berlin.de/forschungsprojekte/. [Zugegriffen: 06-Mai-2015].

[2] H. Drück, „Weiterentwicklung und Validierung des Modells für solare Warmwasserspeicher ‚4Port‘ für das Simulationsprogramm TRNSYS“, Diplomarbeit ITW, Universität Stuttgart, Stuttgart, 1994.

[3] H. Drück und T. Pauschinger, *MULTIPORT Store-Model*. Type, 2006.

[4] Deutsches Institut für Normung e. V., Hrsg., „DIN EN 12977-3 Thermische Solaranlagen und ihre Bauteile - Kundenspezifisch gefertigte Anlagen - Teil 3. Leistungsprüfung von Warmwasserspeichern für Solaranlagen“, Berlin: Beuth Verlag, 2012.

[5] H. Drück, S. Bachmann, und H. Müller-Steinhagen, „Testing of solar hot water stores by means of up-and down-scaling algorithms“, *Proc. ISES EuroSun*, Bd. 6, S. 27–30, 2006.

[6] H. Drück und E. Hahne, „Kombispeicher auf dem Prüfstand“, in *8. Symposium Thermische Solarenergie*, Bad Staffelstein, 1998.

[7] G. Stryi-Hipp, „Abschlussbericht des Projektes Mitarbeit der deutschen Solarindustrie bei der Überarbeitung der europäischen Normen für thermische Solaranlagen (Eurosol)“, Bundesverband Solarwirtschaft e. V., Berlin, AZ 21589, Sep. 2006.

[8] „Ordinary Differential Equations - MATLAB & Simulink - MathWorks Deutschland“. [Online]. Verfügbar unter: http://de.mathworks.com/help/matlab/math/ordinary-differential-equations.html?refresh=true. [Zugegriffen: 19-Mai-2015].

[9] „Create or alter options structure for ordinary differential equation solvers - MATLAB odeset - MathWorks Deutschland“. [Online]. Verfügbar unter: http://de.mathworks.com/help/matlab/ref/odeset.html. [Zugegriffen: 22-Mai-2015].

[10] T. Tjaden, „Techno-ökonomischer Vergleich von Solarthermieanlagen mit Photovoltaik-Wärmepumpen-Systemen“, Masterthesis, Hochschule für Technik und Wirtschaft Berlin, Berlin, 2013.