

Masterarbeit

Yuan Li

State-of-Charge-Bestimmung für eine
Starterbatterie mit einem Kalman-Filter-Modell

Yuan Li

State-of-Charge-Bestimmung für eine
Starterbatterie mit einem Kalman-Filter-Modell

Masterarbeit eingereicht im Rahmen der Masterprüfung
im gemeinsamen Studiengang Mikroelektronische Systeme
am Fachbereich Technik
der Fachhochschule Westküste
und
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. -Ing. Karl-Ragmar Riemschneider
Zweitgutachter : Prof. Dr. -Ing. Alfred Ebberg

Abgegeben am 24. Juli 2012

Yuan Li

Thema der Masterarbeit

State-of-Charge-Bestimmung für eine Starterbatterie mit einem Kalman-Filter-Modell

Stichworte

Starterbatterie, Modellierung, Kalman-Filter, Methode der kleinsten Quadrate, Ladezustand, Ruhespannung

Kurzzusammenfassung

Die Ladezustandsbestimmung spielt eine sehr wichtige Rolle bei der Überwachung der Batterie. Dabei wird die noch in der Batterie verbleibende Kapazität beurteilt. In dieser Arbeit wird mit elektrischen Elementen die Blei-Starterbatterie modelliert. Damit wird eine zuverlässige Aussage zum Ladezustand getroffen.

Yuan Li

Title of the paper

Determination of Automotive SLI Battery State of Charge with Kalman-Filter-Model

Keywords

SLI, modeling, Kalman-Filter, Least Squares, State of charge, Open Circuit Voltage

Abstract

The determination of state-of-charge plays an important role in monitoring the automobile SLI battery. This paper discusses the modeling of a lead-acid-battery. With the help of the new developed model the state-of-charge will be determined.

Danksagung

An dieser Stelle möchte ich mich bei Herrn Prof.Dr.-Ing. Karl-Ragmar Riemschneider, betreuender Prüfer und Projektleiter des Forschungsprojekts „BATSEN“ für das Angebot dieser Masterarbeit und den sehr engagierten Einsatz bedanken. Weiterhin bedanke ich mich bei Herrn Prof.Dr.-Ing. Alfred Ebborg, der meine Arbeit als Zweitgutachter bewertet hat.

Ein weiterer Dank geht an Herrn Prof.Dr.-Ing. Jürgen Vollmer, ebenfalls Projektleiter des Forschungsvorhabens „BATSEN“ für seine fachliche Unterstützung und Diskussionen über die aufgetauchten Probleme. Für die ausführliche Korrektur der Arbeit bedanke ich mich sehr bei Herrn Dipl.-Ing. Günter Müller.

Besonders möchte ich mich ebenso bei Herrn Dip.-Ing Matthias Schneider, den wissenschaftlichen Mitarbeiter im Forschungsprojekt „BATSEN“, für sein fachliches Wissen und stetiges Engagement bedanken. Mit ihm konnte ich viele technische Details diskutieren, was ich als sehr wertvoll erachte. Er half mir bei den Messungen und der Korrektur.

Zuletzt danke ich noch meinen Freunden und meiner Familie, die mich im Laufe der Arbeit in vielfältiger Weise unterstützt haben.

Inhaltsverzeichnis

1. Motivation und Einführung	9
1.1. Starterbatterie	9
1.1.1. Aufbau	9
1.1.2. Kenngrößen	10
1.2. Batterieüberwachung	13
1.2.1. Grundlagen	13
1.2.2. Zellweise Überwachung großer Batterien	14
1.3. Verfahren zur Überwachung des Ladezustandes	14
2. Konzept & Algorithmus	16
2.1. Kalman-Filter	16
2.1.1. Grundlagen	17
2.1.2. Algorithmus	18
2.1.3. Beispiel	20
2.1.4. Problematik der Modellierung und Festlegung der Modellparameter	22
2.2. Modellierung	24
2.2.1. Anforderung an das Modell	24
2.2.2. Vergleich des Modells nach verschiedenen Kriterien	25
2.2.3. Auswahl eines Modells	29
2.3. Verfahren zur Parameteridentifikation	30
2.3.1. Algorithmus	31
2.3.2. Beispiel	32
3. Implementation von Batterienmodellen in Matlab/Simulink und Messung	34
3.1. Messplatz	34
3.2. Modellierung in Simulink	36
3.3. Minimierungsverfahren in Matlab	36
3.4. Allgemeiner Ablauf des Programms	37
3.5. Einfaches Modell - Modell 4	39
3.5.1. Implementation des ausgewählten Modells 4	40
3.5.2. Auswertung und auftretende Probleme	46
3.6. Angepasstes Modell 4 Klasse 1	47
3.6.1. Implementation des angepassten Modells 4 Klasse 1	50

3.6.2. Auswertung und auftretende Probleme	57
3.6.3. Betrachtung des Temperatureinflusses auf das Batterieverhalten	58
3.7. Das angepasste Modell 4 Klasse 2	64
3.7.1. Festlegung der Ruhespannung der Starterbatterie bei drei Temperaturen	64
3.7.2. Festlegung des Innenwiderstandes der Batterie bei drei Temperaturen	72
3.8. Implementation des angepassten Modells Klasse 2	75
3.8.1. Vergleich des simulierte Ergebnisses mit dem Messwert	76
3.9. Auswertung des angepassten Modells 4 Klasse 2	81
3.10. Das angepasste Modell 4 Klasse 3	82
3.11. Auswertung aller implementierten Modelle	83
4. Implementation des Batteriemodells in das Kalman-Filter	86
4.1. Umsetzung des angepassten Modells 4 Klasse 2	86
4.2. Implementation des angepassten Modells 4 Klasse 2 in das Kalman-Filter	90
4.3. Messrauschen des Sensors	93
4.4. Vergleich des Ergebnisses des Kalman-Filters mit dem verrauschten Messwert	95
4.5. Systemrauschen, Messrauschen und Iteration	101
4.6. Vergleich des Ergebnisses des Kalman-Filters mit dem stark verrauschten Messwert	105
4.7. Auswertung der Korrektur mit Hilfe des Kalman-Filters	108
5. Zusammenfassung und Ausblick	109
Literaturverzeichnis	113
Anhang	116
A. Modell	116
B. Säuerdichtemessung	119
C. Quellcode	121
C.1. kontinuierliche Entladung bei 3 Temperatur	121
C.2. partielle Entladung bei 3 Temperatur	125
C.3. Dynamische Entladung bei 3 Temperatur	128
C.4. Beispiel der Simulation des Modells 4	132
C.5. Angerufene Funktion 1 in der Simulation des Modells 4	135
C.6. Angerufene Funktion 2 in der Simulation des Modells 4	137
C.7. Simulation des angepassten Modell 4 Klasse 2	138
C.8. Angerufene Funktion 1 in der Simulation des angepassten Modell 4 Klasse 2	142
C.9. Angerufene Funktion 2 in der Simulation des angepassten Modell 4 Klasse 2	143
C.10. Angerufene Funktion 3 in der Simulation des angepassten Modell 4 Klasse 2	144

C.11. Angerufene Funktion 4 in der Simulation des angepassten Modell 4 Klasse 2	150
C.12. Vergleich der Ruhespannung bzw. des Widerstandes	151
C.13. Kalman-Filter mit verschiedener Varianz	164
C.14. 3d-Plot (Farbe)	175
C.15. Abweichung des Sensors	176
C.16. Kalman-Filter mit dem verrauschten Messwert	178
Tabellenverzeichnis	192
Abbildungsverzeichnis	193

1. Motivation und Einführung

Heutzutage ist die Bleibatterie einer der am häufigsten eingesetzten Energiespeicher für vielfältige Anwendungen, insbesondere für das Starten von Verbrennungsmotoren (Starterbatterie), für den Elektroantrieb (Traktionsbatterien) und die Notstromversorgung (ortsfeste Batterien). Der Energiespeicher ist ein sehr wichtiger Bestandteil im Automobil, der den normalen Betrieb des Autos gewährleistet. Eine unzuverlässige Bleibatterie kann zu unvorhersehbaren Defekten führen. Deswegen ist es notwendig, den Akkumulator während des Betriebs des Motors zu überwachen und seinen Zustand zu erfassen.

An der HAW erfolgen gegenwärtig Forschungsarbeiten zu drahtlosen Sensoren für Batteriezellen (BATSEN). Die Sensoren liefern Daten an ein Steuergerät, das den Zustand der Batterie analysiert. Dort wird eine Aussage zum Ladezustand (State of Charge, SOC) und zu anderen Eigenschaften getroffen. Im Rahmen dieser Arbeit wird die Modellierung der Starterbatterie und verschiedene Verfahren zur Bestimmung des Ladezustands untersucht.

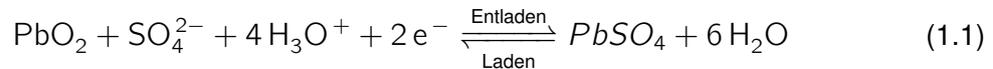
1.1. Starterbatterie

Die Starterbatterie ist ein Akkumulator, häufig ein Bleiakkumulator, der für das Starten des Verbrennungsmotors ausreichend Strom liefert. Während des Startvorganges benötigt der Startermotor kurzfristig einen hohen Strom von mehreren hundert Ampere. Außerdem versorgt die Starterbatterie bei Stillstand des Motors die elektrischen Verbraucher. Die Starterbatterie wird vom Generator aufgeladen, wenn der Motor gestartet ist.

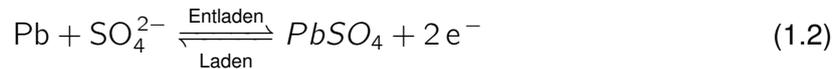
1.1.1. Aufbau

Die Bleibatterie besteht im Allgemeinen aus einer positiven Elektrode aus Bleidioxid, einer negativen Elektrode aus Blei, einem Elektrolyten, im Falle der Starterbatterie Schwefelsäure und einen Ionenleitfähigen Separator. Die Elektroden einer Batteriezelle sind die parallel geschalteten Platten. Die Kapazität der Zelle ist von Anzahl und Fläche dieser Platten abhängig. Je nach der Ablaufrichtung wird der elektrochemische Prozess bei der Entladung und Ladung durch die Gleichungen [1.1](#) - [1.3](#) dargestellt.

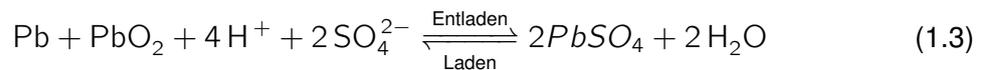
positive Elektrode :



negative Elektrode :



Allgemeine chemische Reaktion:



Bei der Entladung werden 1 Teil Bleidioxid von der positiven Elektrode und 1 Teil Blei der negativen Elektroden mit Hilfe von 2 Teilen Schwefelsäure in je 1 Teil Bleisulfat an beiden Elektroden und 2 Teile Wasser umgewandelt. Dabei wird Schwefelsäure verbraucht und Wasser gebildet, so dass die Schwefelsäure in der Zelle verdünnt wird und die Säuredichte sinkt. Der Ladezustand der Batterie korreliert direkt mit der Säuredichte. Eine Säuredichte von 1.265g/cm^3 wird in der Literatur als Wert für die geladene Batterie angegeben. Bei 1.120g/cm^3 wird die Batterie als 'leer' bezeichnet. Beim Laden verläuft der Vorgang umgekehrt [24].

Eine 12-Volt-Starterbatterie bildet sich aus 6 in Reihe geschalteten Zellen. In der Literatur [32] wird der Aufbau der Starterbatterie ausführlich dargestellt. Der Aufbau der modernen Blei-Starterbatterie wird in Abb. 1.1 dargestellt.

1.1.2. Kenngrößen

Verwendete Spannungsbegriffe :

Die an den Klemmen einer Zelle gemessene Spannung wird als Klemmenspannung bezeichnet. Die Klemmenspannung einer Batterie mit Zellen in Reihenschaltung ist die Summe der Klemmenspannungen der Zellen. Die Klemmenspannung einer Bleibatterie soll beim Aufladen über der Entladeschlussspannung von ca. 1,7 V und beim Entladen unter der Ladeschlussspannung bzw. Gasungsspannung von ca. 2,39 V (25°C) liegen [24]. Falls die Bleibatterie über die Gasungsspannung geladen wird, tritt Knallgasbildung auf, weil Wasser in Wasserstoff und Sauerstoff umgewandelt wird. Falls die Bleibatterie unter die Entladeschlussspannung entladen wird, wird dies als Tiefentladung bezeichnet. Weil bei der Tiefentladung aktive Massen sulfatiert werden, kommt es zu einem Kapazitätsverlust. Die nach einer ausreichenden Ruhephase gemessene Spannung an den Klemmen wird als Ruhespannung bzw. Open Circuit Voltage (OCV) bezeichnet.

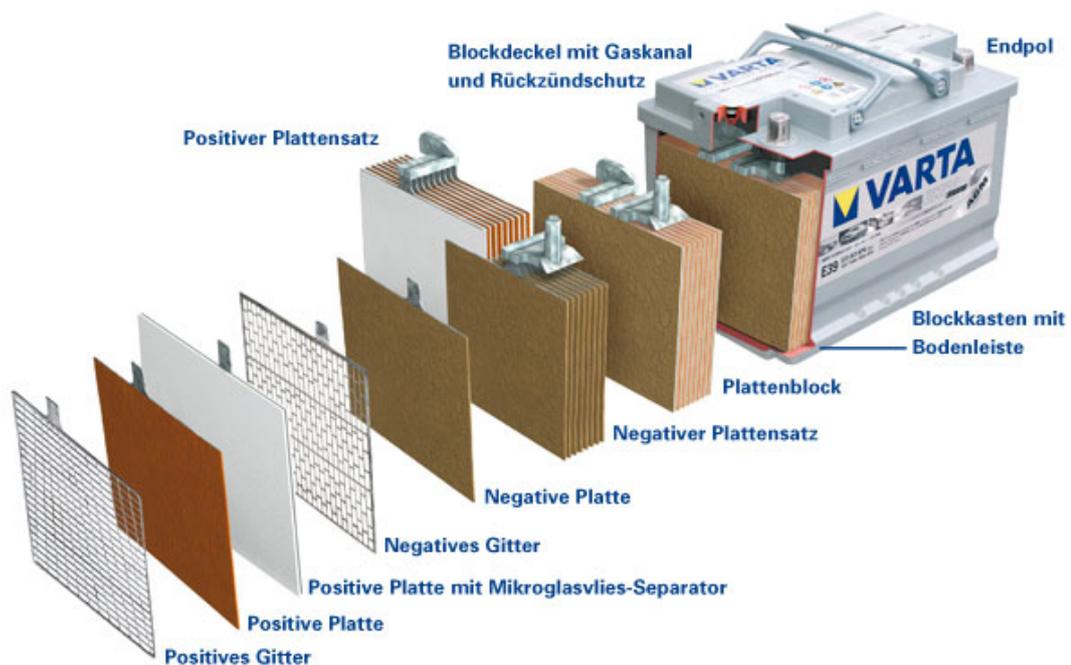


Abbildung 1.1.: Konstruktion einer VARTA - Starterbatterie[9]

Verwendete Kapazitätsbegriffe :

Kapazität bezeichnet die Ladungsmenge, die in einer Batterie gespeichert wird. Häufig wird die Kapazität in Amperestunden (Ah) angegeben. Die Nennkapazität ist die Elektrizitätsmenge, die einer Batterie bei der vollständigen Entladung unter Nennbedingungen zugeordnet werden kann. Diese Nennkapazität wird vom Hersteller angegeben.

Definition des Alterungsprozesses :

Im Laufe des Betriebs kann die Batterie nicht mehr die Nennkapazität bzw. Nennladung abgeben. Verschiedene Ursachen bewirken den irreversiblen Verlust der Kapazität. Eine mögliche Ursache liegt in der Überladung. Dabei wird aufgrund von Korrosion die positive Elektrode langsam zersetzt. Der Innenwiderstand steigt mit zunehmender Korrosion an, wodurch der Spannungsverlust in der Batterie vergrößert wird. Falls die Batterie für einen längeren Zeitraum in einem niedrigen Ladezustand ruht, schließen sich die feinen Bleisulfat-Kristalle zu immer größeren Ablagerungen zusammen. Solche vergrößerten Bleisulfat-Kristalle lassen sich nicht durch Laden wieder abbauen. Damit wird die chemische Umsetzung erschwert. Außerdem spielt die Temperatur eine wichtige Rolle beim Alterungsprozess. Je

höher die Temperatur der Batterie ist, desto schneller laufen die chemische Prozesse ab. Selbstentladung bezeichnet den Kapazitätsverlust von der Batterie im Ruhezustand ohne externen Stromfluss. Die Ursache liegt in den metallischen Verunreinigungen. Pro 10°C Anstieg der Temperatur wird der Selbstentladungsfaktor in etwa verdoppelt [9].

Definition des Ladezustandes:

Als State of Charge (SoC) wird das Verhältnis der noch in der Batterie gespeicherten Ladungsmenge zu der maximalen Ladungsmenge bezeichnet, die die Batterie speichern kann (Gl. 1.4). Die Depth of Discharge (DoD) ist das umgekehrte Verhältnis zum Ladezustand (SoC) (Gl. 1.5). Der Depth of Discharge ist ebenfalls ein Verhältniswert zwischen 0..1 bzw. 0%.....100%.

$$SoC = \frac{Q_{ist}}{Q_{max}} \quad (1.4)$$

$$DoD = 1 - SoC \quad (1.5)$$

Definition des Gesundheitszustandes:

Als State of Health (SoH) wird der Alterungszustand der Batterie bezeichnet. Falls man den Gesundheitszustand einer Batterie bewertet, soll der Spannungsabfall dieser Batterie mit dem Spannungsabfall einer neuen Batterie beim bestimmten Belastungsprofil verglichen werden, d.h. unter gleicher Temperatur, Entladestrom und Entladedauer. U_1 ist Entladeschlussspannung. Der SoH wird durch die Gl. 1.6 definiert [32].

$$SoH = \frac{U_{alt} - U_1}{U_{neu} - U_1} \quad (1.6)$$

Definition des Batteriezustandes:

Als State of Function (SoF) wird die Fähigkeit der Batterie bezeichnet, eine bestimmte Funktion im aktuellen Batteriezustand zu erfüllen. Der SoF ist vom SoH, SoC und Temperatur abhängig.

1.2. Batterieüberwachung

Die Batterieüberwachung spielt immer eine wichtige Rolle im Automobil, z.B. durch eine Überprüfung des Batteriezustandes kann festgestellt werden, ob die Batterie in der nächsten Zeit noch in der Lage sein wird, den benötigten Startstrom zu liefern. Beim Laden sollte man berücksichtigen, ob die Batterie über den vom Hersteller vorgegebenen Spannungpegel geladen wird, weil bei der Überladung das Knallgas gebildet wird. Gegebenenfalls muss anschließend Wasser nachgefüllt werden. Die Bleibatterie ist sehr empfindlich gegenüber Tiefentladung. Tiefentladung führt zur irreversiblen Schädigung der Batterie. Deswegen ist die Überwachung der Batterie zur Verhinderung des Tiefentladens jeder Zelle wichtig.

1.2.1. Grundlagen

Gegenwärtig wird häufig die gesamte Batterie zur Überwachung betrachtet. Die Klemmenspannung an den äußeren Anschlüssen der Batterie, der Strom und die Temperatur werden mittels entsprechender Sensoren gemessen.

Die Vorteile der Überwachung der gesamten Batterien sind bekannt. Im Vergleich mit der Überwachung einzelner Zelle benötigt man weniger Aufwand. Außerdem ist es nicht nötig, die Batterie zu öffnen.

Jedoch hat dieses Verfahren Nachteile, da nur der Mittelwert der Spannung der Batterie gemessen werden kann. Wegen der Toleranz bei der Fertigung in der Fabrik gibt es Unterschiede zwischen einzelnen Zellen im Ladezustand und Alterungszustand. Diese Unterschiede nehmen im Laufe der Betriebszeit immer weiter zu. Solche Unterschiede einzelner Zellen sind bei Betrachtung der Gesamtspannung der Batterie nicht erfassbar, weil 'schwächere' Zellen durch andere Zellen, die einen besseren Ladezustand besitzen, verborgen werden. Ab einem gewissen Ladezustand stören diese 'schwachen' Zellen das Verhalten der gesamten Batterie. Deswegen ist es sinnvoll, jede einzelne Zelle individuell zu betrachten, so dass das obengenannte Problem gelöst werden kann.

1.2.2. Zellweise Überwachung großer Batterien

Gemäß dem Grundkonzept des Forschungsprojektes BATSEN wird die einzelne Batteriezelle durch einen drahtlosen Zellsensor überwacht. Der Sensor misst die Spannung und Temperatur einer Zelle und sendet die Werte an ein zentrales Batteriesteuergerät. Dann werden Ladezustand und Alterungszustand einzelner Zelle ermittelt. In Abb. 1.2 ist das Überwachungssystem dargestellt.

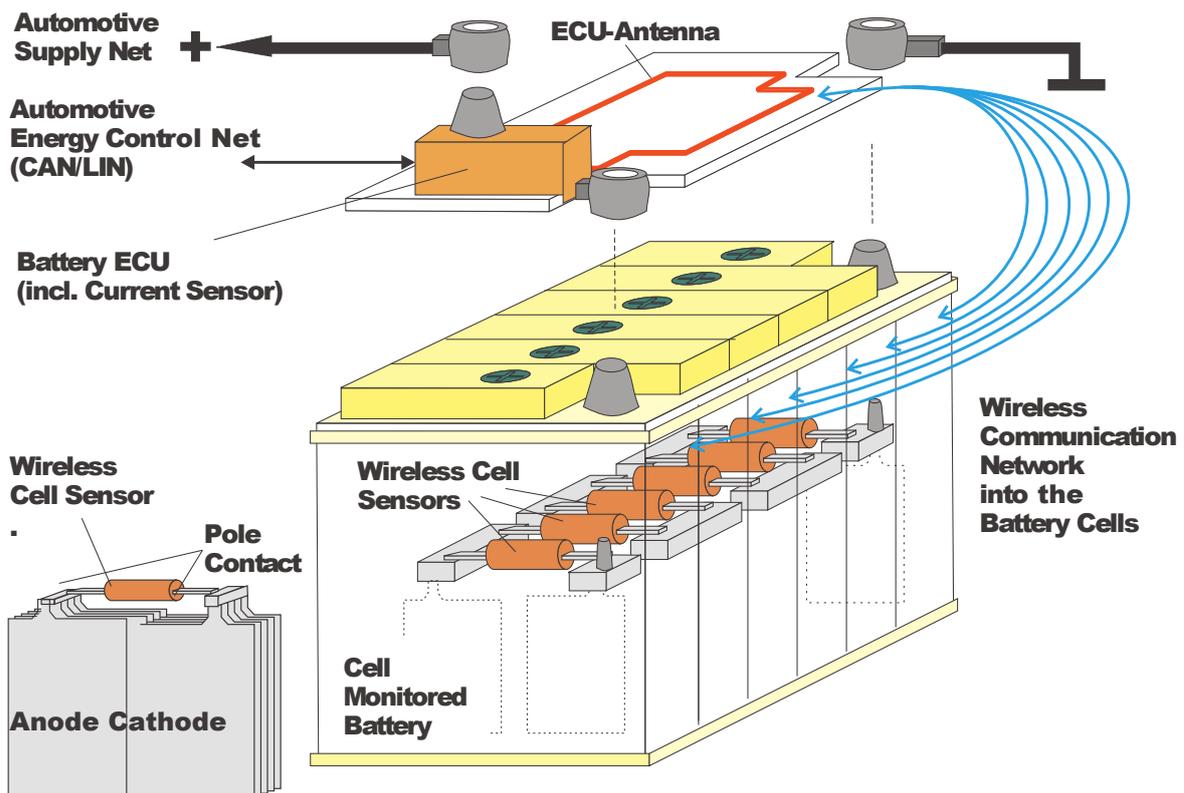


Abbildung 1.2.: Batterieüberwachungssystem [33]

1.3. Verfahren zur Überwachung des Ladezustandes

Die am häufigsten verwendeten Verfahren zur Bestimmung des Ladezustandes werden in Tabelle. 1.1 gegenübergestellt. Außerdem werden Vor- und Nachteile zu jedem Verfahren und dazu benötigte Messwerte aufgelistet.

Verfahren	Algorithmus	Vorteile	Nachteile	Messwert
Coulomb - Counting	Berechnung der Restkapazität durch Integration des Stromes über der Zeit	einfacher Algorithmus, niedriger Rechenaufwand	Kenntnis eines initialen Ladezustandes nötig, Messfehler summieren sich	Strom
Kalman Filter	Der Schätzwert eines in Kalman Filter modellgestützten Systems wird durch den Messwert korrigiert	laufend Schätzung des Ladezustandes über Systemmodell, Korrektur falscher Strommesswerte durch Spannungswert	komplexer Algorithmus, Rechenaufwand	Strom, Spannung
Open Circuit Voltage - Ladezustand	der Ladezustand wird mithilfe annähernd linearer Beziehung zwischen Ladezustandes und Ruhespannung berechnet, die nach einer ausreichenden Ruhephase beim Entladen bzw. Laden gemessen wird.	niedriger Rechenaufwand	lange Wartezeit nur annähernd lineare Beziehung zwischen Ruhespannung und Ladezustand	Spannung
Säuredichtemessung	Öffnung der einzelnen Zelle und Ansaugen einer Probe des Elektrolyts Nachschlagen des entsprechenden Ladezustandes in einer Tabelle	Kompensation des Temperatureinflusses	lange Wartezeit Öffnung der Zelle	Säuredichte
Impedanzspektroskopie	Frequenzabhängigkeit der Diffusionsprozesse in Elektrodengrenzschichten	Überwachung des Innenwiderstandes	Temperatureinfluss Rechenaufwändig	Spannung Strom

Tabelle 1.1.: Verfahren zur Überwachung des Ladezustandes

2. Konzept & Algorithmus

Im vorangegangenen Kapitel wurde bereits die Notwendigkeit der Überwachung der Batterie erwähnt und Verfahren zur Bestimmung des Ladezustandes in der Tabelle 1.1 aufgelistet. In dieser Arbeit soll ein Verfahren untersucht werden, das nicht nur den Ladezustand der Batterie so genau wie möglich erfasst werden kann, sondern es auch eine Fähigkeit zum Vorhersagen des Ladezustandes der Batterie zur Verfügung stellt. Außerdem spielt der Hardwareaufwand eine wichtige Rolle zur Bewertung des Verfahrens. Der Kalman-Filter erfüllt alle oben genannten Anforderungen und verfügt über einen durch Messwerte korrigierbaren Schätzwert des Ladezustandes. In diesem Kapitel sollen einige im Rahmen dieser Arbeit verwendete Konzepte bzw. Algorithmen erklärt werden. Dabei steht ein Beispiel zur Verfügung, um Verständnis des Algorithmus des Kalman Filters zu erleichtern.

2.1. Kalman-Filter

Das von Rudolf E. Kalman im Jahr 1960 entdeckte Filter wird als Kalman-Filter genannt. Das Kalman-Filter wird in dem allgemeinen Rahmen des Beobachterentwurfs gesehen. Ein modellgestütztes Konzept, bei dem das Modell eines gegebenen Grundsystems in einer Regelschleife liegt, korrigiert mit einer Gewichtung, die über ein dynamisches Kriterium zum deterministischen Lünberger-Beobachter und über ein statistisches Kriterium zum Kalman-Beobachter führt. Das modellgestützte Verfahren spielt eine zunehmende Rolle bei der Entwicklung von Sensortechnik, weil es über die Modellierung eine Art „a priori -Wissen“ über den verfahrenstechnischen Prozess einbringt und optimal gefilterte Schätzwerte für nicht messbare (aber beobachtbare) Systemzustände zur Verwendung in Prozessregelungskonzepten liefert. In die gleiche Anwendungskategorie fällt der Einsatz von Kalman-Beobachtern zur Vorausschätzung kritischer Reaktorzustände in Sicherheits- und Überwachungssystemen. In diesem Zusammenhang sind auch die Einsatzbereiche von Kalman-Beobachtern zur Fehlererkennung und -Korrektur bei diskreten Regelungssystemen zu nennen. Kalman-Filter sind im zahlreichen Anwendungsbereichen zu finden [15], wie z.B:

- * Navigation System
- * Radarsystem
- * Selbstlokalisierung von Robotern

- * Wettervorhersage
- * Satellitenverfolgung
- * Gesichtserkennung
- * Kurssteuerungsprobleme

2.1.1. Grundlagen

Im Unterschied zu den klassischen FIR- und IIR-Filtern wird beim Kalman-Filter ein Zustandsraum aufgebaut. Oft werden nicht nur messbare Variablen im Zustandsraum zusammengefasst, sondern auch die nicht messbaren (aber beobachtbaren) Variablen, die im betrachteten System relevant sind, weil solche nicht messbare Variablen zur präzisen Beschreibung des Verhaltens des Systems nötig sind. Diese Variablen werden im Zustandsraum in der Form eines mehrdimensionalen Vektors x dargestellt. Die Zustandsgleichung wird durch Gl. 2.1 angegeben.

$$x_{k+1} = A * x_k + B * u_k + w_k \quad (2.1)$$

x_k ist der Zustandsvektor zum Zeitpunkt k . x_{k+1} ist der Zustandsvektor nach dem bestimmten Zeitintervall ΔT . Die Matrix A beschreibt die Übergänge zwischen zeitlich aufeinanderfolgenden Zuständen und wird als Zustandsübergangsmatrix bezeichnet. Die Einflüsse externen Kräfte werden durch Produkt der Eingangswerte u zum Zeitpunkt k und des Vektors B dargestellt. w_k bildet einen zeitlich unkorrelierten und gaußverteilten Prozessfehler (auch Systemrauschen genannt), der die Abweichung des modellgestützten Systems vom tatsächlichen Systemzustand beschreibt. Die Kovarianzmatrix des Prozessfehlers wird als Q angegeben. Der Systemfehler sollte nicht aufeinander einwirken, deshalb sollte nur die Hauptdiagonale der Kovarianzmatrix Q mit Varianzen belegt werden und die restlichen Werte 0.

Der fehlerbehaftete Messprozess wird durch Gl. 2.2 präsentiert.

$$y_k = H * x_k + v_k \quad (2.2)$$

Mit der Größe v_k wird der von Messgeräten verursachte Messfehler angegeben und als zeitlich unkorreliert und normalverteilt mit dem Erwartungswert 0 angenommen. Die Kovarianzmatrix des Messrauschens wird als R bezeichnet. Damit sich Messfehler nicht gegenseitig beeinflussen, soll nur die Varianz des Messrauschens auf der Hauptdiagonalen der Kovarianzmatrix gelegt werden. H ist die Ausgabematrix. y_k ist der Ausgabewert, den man tatsächlich beobachten will.

2.1.2. Algorithmus

Der Algorithmus des Kalman - Filters wird in der Abb. 2.1 dargestellt. Der Kalman-Filter verfügt 3 Phasen: Initialisierung-, Prädiktion- und Korrekturphase.

Der Ablauf des Kalman - Filters wird in Abb. 2.1 dargestellt.

Initialisierung

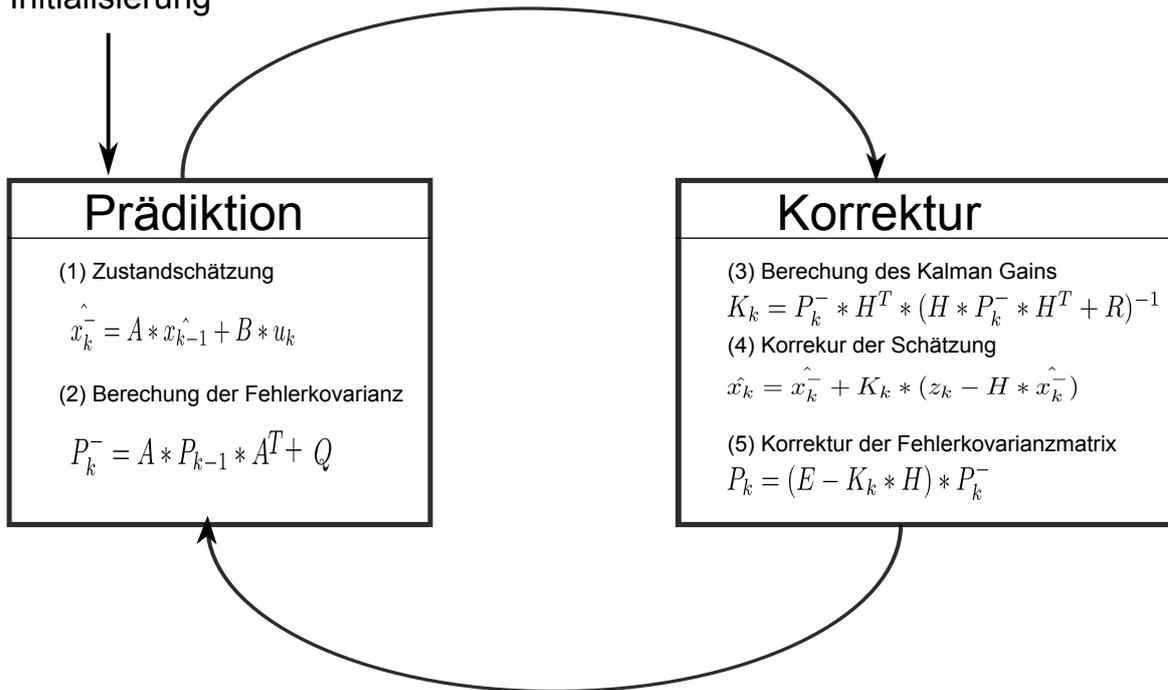


Abbildung 2.1.: Ablauf des Algorithmus

Zur Initialisierung des Filters sollen alle Matrizen mit einem Anfangswert belegt werden.

In der Prädiktionsphase wird aus dem vorherigen geschätzten Systemzustand a priori (lat. "Vom Früheren her") Schätzung \hat{x}_k^- berechnet. Zusätzlich wird die berechnete Schätzung mit dem Einfluss der externen Kräfte $B * u_k$ addiert (Gl. 2.3). Die a priori Fehlerkovarianz P_k^- bildet sich aus der vorherigen a posteriori (lat. "was aus der Erfahrung stammt") Fehlerkovarianz P_{k-1}^- mit Übergangsmatrix A sowie dem Addieren der Kovarianzmatrix des Systemrauschens Q (Gl. 2.4). Diese a posteriori Schätzung wird anhand Kalman Gain in der Korrekturphase aktualisiert.

In der Korrekturphase wird eine fehlerhafte Messung durchgeführt. K ist der Kalman-Gain, der als Korrekturfaktor festlegt, wie stark die Differenz zwischen der priori Schätzung und des aktuellen Messwertes in eine weitere Schätzung eingeht (Gl. 2.5). Mithilfe des Messwertes,

der a priori Schätzung der Prädiktionsphase und des Kalman-Gain wird a posteriori Abschätzung erzeugt und ausgegeben. Diese wird als aktuelles Schätz-Ergebnis x_k zum nächsten Zeitpunkt übergeben (Gl. 2.6). Außerdem wird die a posteriori Fehlerkovarianzmatrix P_k berechnet (Gl. 2.7).

$$\hat{x}_k^- = A * \hat{x}_{k-1} + B * u_k \quad (2.3)$$

$$P_k^- = A * P_{k-1} * A^T + Q \quad (2.4)$$

$$K_k = P_k^- * H^T * (H * P_k^- * H^T + R)^{-1} \quad (2.5)$$

$$\hat{x}_k = \hat{x}_k^- + K_k * (z_k - H * \hat{x}_k^-) \quad (2.6)$$

$$P_k = (E - K_k * H) * P_k^- \quad (2.7)$$

Tabelle. 2.1 liefert einen Überblick über die im Kalman Algorithmus verwendeten Symbole und die entsprechende Bedeutung

Symbol	Bedeutung
\hat{x}_k^-	ein Schätzwert vor der Korrektur zum Zeitpunkt k (a priori Schätzwert)
\hat{x}_k	ein Schätzwert gemäß der Größe des Kalman Gains zum Zeitpunkt k (a posteriori Schätzwert)
u_k	Eingabe
z_k	Messwert
A	Zustandsübergangsmatrix
B	Eingabematrix
H	Ausgabematrix
Q	Kovarianzmatrix des Systemrauschens
R	Kovarianzmatrix des Messrauschens
P_k^-	eine Fehlerkovarianzmatrix vor der Korrektur zum Zeitpunkt k (a priori Fehlerkovarianzmatrix)
P_k	eine Fehlerkovarianzmatrix gemäß der Größe des Kalman Gains zum Zeitpunkt k (a posteriori Fehlerkovarianzmatrix)
K_k	Kalman Gain zum Zeitpunkt k
T	die Transponierte der Matrix
E	Einheitsmatrix

Tabelle 2.1.: Überblick über die in Kalman Filter verwendeten Symbole

2.1.3. Beispiel

In den vorliegenden Abschnitten wurde der Algorithmus des Kalman-Filters erklärt. Der Kalman-Filter nähert anhand eines mathematischen Systems die fehlerhaften Messwerte an die in der Realität vorliegenden Werte des Systems an. Mithilfe eines Beispiels vom Fall eines Balles lässt sich der Algorithmus des Kalman Filters verdeutlichen. Der Ball befindet sich am Anfang in Ruhe und fällt unter dem Einfluss der Gravitation aus einer Höhe h_0 . Der Verlauf des fallenden Balls wird durch eine Videokamera beobachtet und aufgenommen. Der Kalman-Filter dient dazu, dass bei der Messung verursachtes Rauschen entfernt wird und den tatsächlichen Verlauf des fallenden Balls berechnet wird. Laut dem Fallgesetz wird die Bewegung des fallenden Balls ohne Berücksichtigung des Luftwiderstandes durch Gl. 2.8 und Gl. 2.9 beschrieben.

Fallgeschwindigkeit:

$$v_k = v_{k-1} - g * \Delta t \quad (2.8)$$

Fallhöhe:

$$h_k = h_{k-1} + \Delta t * v_{k-1} - \frac{1}{2} * g * (\Delta t)^2 \quad (2.9)$$

Um präzise sich dem tatsächlichen System anzunähern, wird nicht nur die messbare Fallhöhe h berücksichtigt, sondern auch die beobachtete Geschwindigkeit des fallenden Balles v . Die Geschwindigkeit und die Fallhöhe bilden den Zustandsraum. Sie werden als Zustandsvektor angenommen: $x_k = \begin{pmatrix} h \\ v \end{pmatrix}$ (Gl. 2.10)

Die Gravitation g steht im Vektor u und wird als die externe Kraft bzw. Eingabe angenommen. $\begin{pmatrix} h \\ v \end{pmatrix}_{k-1} \begin{pmatrix} -\frac{1}{2} * \Delta t^2 \\ -\Delta t \end{pmatrix}$ ist die Gewichtungsmatrix der externen Eingabe. $\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$ entspricht der Zustandsübergangsmatrix. Nur die Höhe des fallenden Balles sollte als Ausgabe bewertet werden. Aus diesem Grund ist die Ausgabematrix $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. w_k und v_k sind entsprechend das Systemrauschen und das Messrauschen.

$$\begin{pmatrix} h \\ v \end{pmatrix}_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} h \\ v \end{pmatrix}_{k-1} + \begin{pmatrix} -\frac{1}{2} * \Delta t^2 \\ -\Delta t \end{pmatrix} * g + w_k \quad (2.10)$$

$$y_k = \begin{pmatrix} 1 \\ 0 \end{pmatrix} * x_k + v_k \quad (2.11)$$

Die Abb. 2.2 zeigt das Ergebnis dieses Beispiels. Die blaue Kurve beschreibt den tatsäch-

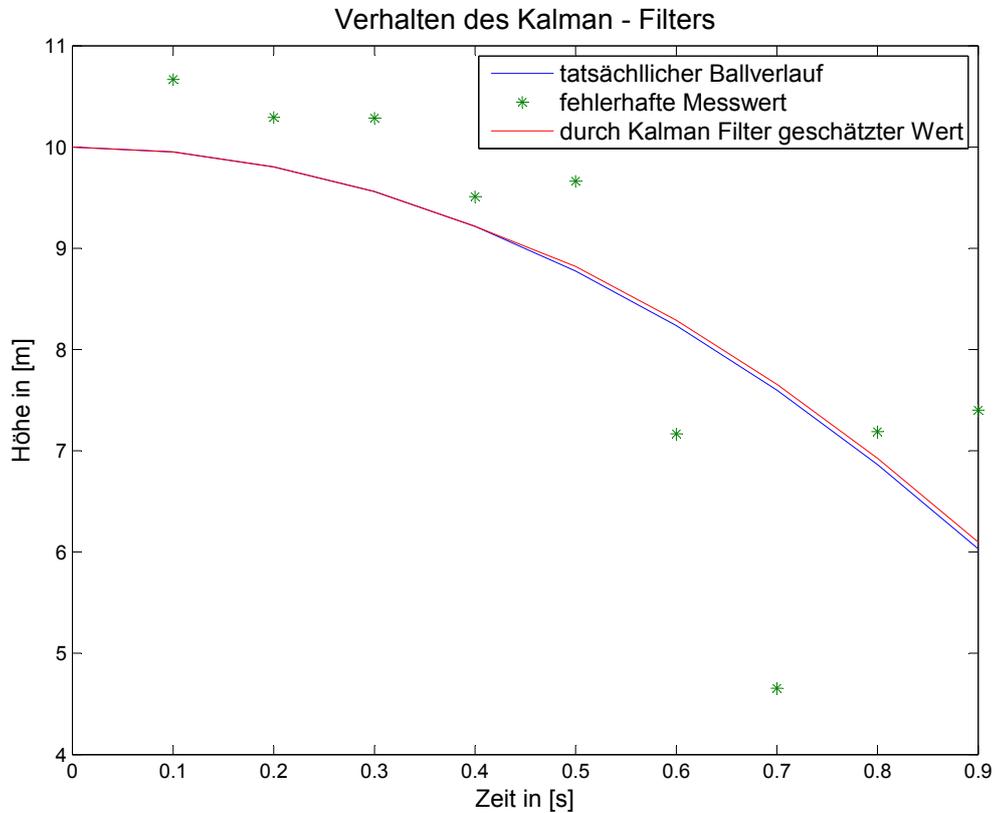


Abbildung 2.2.: Beispiel des Kalman Filters

lichen Verlauf des fallenden Balles. Die grünen Punkte stellen den fehlerhaften Messwert dar. Das Messrauschen verursacht die Abweichung zwischen dem Messwert und dem tatsächlichen Verlauf. Die rote Kurve bezeichnet den vom Kalman-Filter rekonstruierten Verlauf des fallenden Balles. Statt 0.7833 m bei der Messung liegt der durchschnittliche Schätzfehler beim Kalman-Filter nur bei 0.0010 m, wodurch die Wirkung des Kalman-Filters gezeigt wird.

2.1.4. Problematik der Modellierung und Festlegung der Modellparameter

Anhand möglichst genauer mathematischer Modellierung kann das Kalman-Filter die fehlerhaften Messungen glätten und die folgenden Zustände vorhersagen. Falls ein exaktes Modell zur Beschreibung des Verhaltens des Systems im Kalman-Filter zur Verfügung steht, werden zuverlässige Schätzwerte getroffen. Dabei sind nicht häufig zur Korrektur des Schätzwertes dienende Messwerte nötig. D.h., das Zeitintervall zwischen zwei Messpunkten kann sich verlängern und die Häufigkeit zur Korrekturphase reduziert sich. Dadurch lässt sich vermeiden, dass sich der Messfehler beim Ablauf des Kalman Filters zu einem hohen Betrag aufsummiert.

Falls das im Kalman-Filter eingesetzte Modell kein vollständiges Verhalten des tatsächlichen Systems modellieren oder zum Teil des Systems nicht richtig darstellen kann, werden die Schätzwerte beim Ablauf des Kalman-Filters einigermaßen verfälscht. Vorausgesetzt, dass bei der Messung durch Messgeräte verursachte Messfehler noch akzeptabel sind, können solche fehlerhafte Schätzwerte anhand des Messwertes bei der Korrekturphase korrigiert werden.

Häufig hilft der genau ermittelte und analysierbare Messwert, das im Kalman-Filter angewendete Modell zu vereinfachen. Der vom unvollständigen Modell ausgelöste Schätzfehler kann durch häufige Korrektur mit genauem Messwert kompensiert werden. D.h., durch häufige Korrektur kann ein vereinfachtes Modell auch eine zuverlässige Schätzung ermöglichen. Im Vergleich zum komplexen Modell kostet ein vereinfachtes Modell weniger Implementierungsaufwand. Je komplexer das Modell ist, desto mehr Elemente bzw. Parameter werden dort enthalten und müssen berechnet werden.

Bei der Modellierung sollte immer die Komplexität des Modells bzw. des Implementierungsaufwands berücksichtigt werden. Man sollte aber stets daran denken, dass wirkliche Gewinne an Genauigkeit nur in solchen Fällen zu erwarten ist, in denen das Modell flexibel und leicht umzusetzen ist.

In der Praxis sollten vor der Modellierung die Mindestanforderungen an die Genauigkeit der Schätzung des Kalman Filters festgelegt werden. Im bestimmten Fall sollte ein Kompromiss zwischen Rechenaufwand und Genauigkeit der Schätzung geschlossen werden.

Die Parameteridentifikation spielt eine wichtige Rolle bei der Modellierung und der modellbasierten Simulation. Das Modell benötigt die Parameter, womit das Modell das physikalische System vollständig beschreiben kann. Wie gut ein Modell das Verhalten des tatsächlichen Systems darstellen kann, ist stark von ausgewählten Parametern abhängig. Die Parameteridentifikationsaufgabe lässt sich durch folgende Maßnahmen durchführen.

- Offline - Identifikation

Die Parameter sollten vor dem normalen Betrieb festgelegt werden. Solche Parameter werden während des Betriebs nicht mehr geändert.

1. experimentelle Versuche bzw. Messungen

Durch experimentelle Versuche können die Parameter festgelegt werden. Im Beispiel des fallenden Balls kann man vor der Durchführung erst die Gravitation dieses Standorts und die Fallhöhe durch den experimentellen Versuch bzw. die Messungen vorherbestimmen.

2. Approximation der Messkurven

Ein möglicher Weg zur Identifikation der Parameter steht in der Approximation der kritischen Messkurven. Dabei wird eine Fehlerfunktion aufgestellt, die die Differenz zwischen den im parametrisierten Modell berechneten Ausgangsgrößen und in der Tatsache gemessenen Wert ermittelt. Die gesuchten Parameter helfen dabei, die Fehlerfunktion zu minimieren. Es stehen verschiedene Minimierungsverfahren zur Verfügung. z.B. die Gauß - Newton - Methode, die Methode der Kleinsten Quadrate usw.

- Online-Identifikation

Im Vergleich zur Offline-Identifikation sollten die Parameter während des laufenden Prozesses bestimmt werden. Solche veränderliche Parameter sind von der äußeren Umgebung beeinflusst, z.B. Temperatur, Standort, Luftfeuchtigkeit usw. Ein rekursives Modell wird aufgebaut, wenn die aktuellen Parameter durch vorherige bestimmte Parameter und Messgrößen neu berechnet werden sollten. Die neu berechneten Parameter werden zur Berechnung der nachfolgenden Ausgaben und Parametergrößen beitragen.

In der Modellierung des Kalman-Filters können die Parameter im Zustandsvektor belegt werden. Sind die Korrekturphase im Laufe des Kalman Filters aufgerufen, werden die Parameter durch den Vergleich mit den gemessenen Größen korrigiert.

Eine andere Möglichkeit zur Online-Parameteridentifikation steht im Erstellen einer allgemeinen Formel, die den von der äußeren Umgebung beeinflussten Parameter beschreibt. Im Laufe des Prozesses sollten die Parameter anhand des aktuellen Messwertes und der allgemeinen Formel berechnet werden.

Wegen des im Betriebes zusätzlichen Berechnungsaufwands an den Parametern kann sich die Rechenzeit im Vergleich zur Offline-Identifikation beträchtlich erhöhen. Die Rechenzeit sollte je nach der Anforderung der zu lösenden Aufgabe beschränkt werden.

Je nach der Komplexität des Modells, der Abhängigkeit von der Umgebung und die Anforderung an der Genauigkeit der Ausgabe wird über ein passendes Parameteridentifikationsverfahren zur Modellierung entschieden.

2.2. Modellierung

Das Batteriemodell beschreibt das elektrische Verhalten der Bleibatterie. Anhand eines Batteriemodells kann der Verlauf der Klemmenspannung und der Ladezustand der Batterie bei festgelegten Lastprofilen im Voraus berechnet werden. Damit kann beispielsweise die Entladeschlussspannung vorausbestimmt werden. Häufig wird mithilfe einer Gleichung oder eines elektrischen Ersatzschaltbildes das Verhalten der Batterie modelliert. In den folgenden Abschnitten werden die Anforderungen zur Modellierung und einige Modelle erläutert.

2.2.1. Anforderung an das Modell

Ein Batteriemodell sollte abhängig von den gemessenen Werten (Strom, Klemmenspannung, Temperatur) beim Laden und beim Entladen zuverlässige Voraussagen über

- den momentanen Ladezustand
- den Zeitpunkt zum Entladeschluss bzw. Ladeende
- die noch verfügbare Ladung bzw. die entnommene Ladung
- den Temperaturverlauf
- die Alterungseffekte

treffen. Außerdem sollte die Komplexität des Batteriemodells berücksichtigt werden.

2.2.2. Vergleich des Modells nach verschiedenen Kriterien

- Modell 1 – "Modified Thevenin Equivalent Battery Model"

Das Modell in Abb. 2.3 besteht aus einer Spannungsquelle V_0 , welche die Ruhespannung (OCV) der Batterie darstellt, einem Kondensator C_p , womit die aufgrund der Diffusion verursachte Polarisation nachgebildet wird, und drei Widerständen. Das Verhalten beim Laden und Entladen lässt sich durch zwei Widerstände R_d und R_c unterscheiden. Der Widerstand R_t ist der Klemmenwiderstand [8].

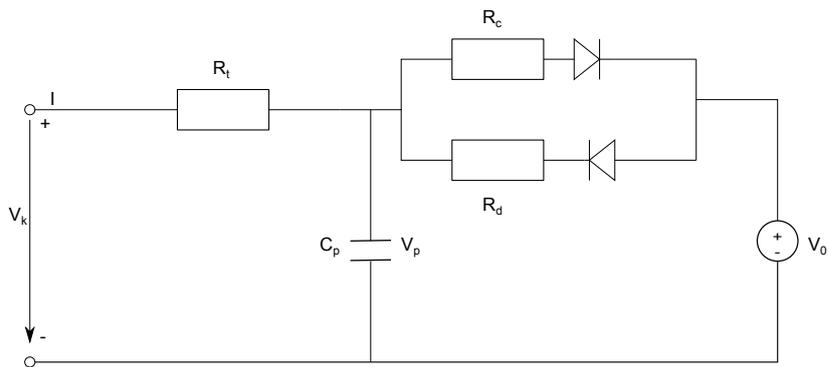


Abbildung 2.3.: Modell 1

- Modell 2 – "lineares elektrisches Batteriemodell"

Das in Abb. 2.4 gezeigte Modell umfasst ein aus drei Widerständen (R_1, R_2, R_3) und drei Kondensatoren (C_1, C_2, C_3) bestehendes Netzwerk. Es beschreibt das Verhalten des Überladens. Eine Reihenschaltung mit Spannungsquellen E_0 und C_b stellt die Ruhespannung der Batterie dar. Den Effekt der Selbstentladung bildet der Widerstand R_p nach. Dieses Modell passt besonders gut zur Beschreibung des Verhaltens des Ladeprozesses [17].

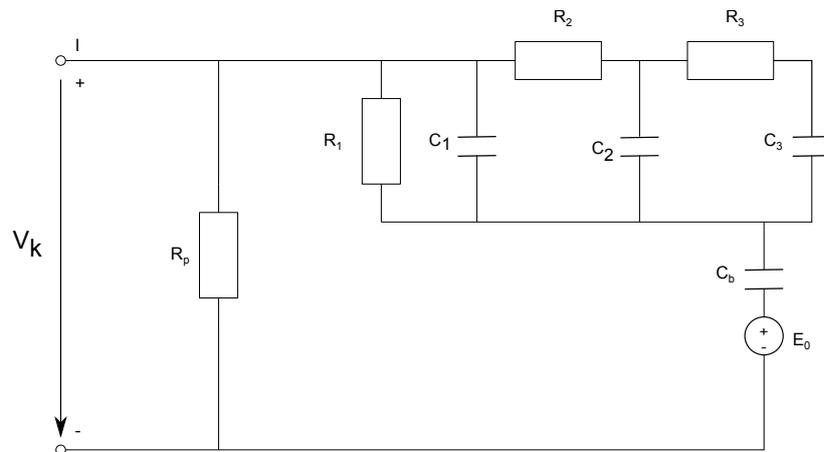


Abbildung 2.4.: Modell 2

- Modell 3 – "Battery Model with a parasitic reaction"
Das Modell in Abb. 2.5 enthält zwei Bereiche. Ein Bereich beschreibt die Hauptreaktion des chemischen Prozesses in der Batterie, der andere bezeichnet das dynamische Verhalten der Batterie. R_t ist der Klemmenwiderstand [21].

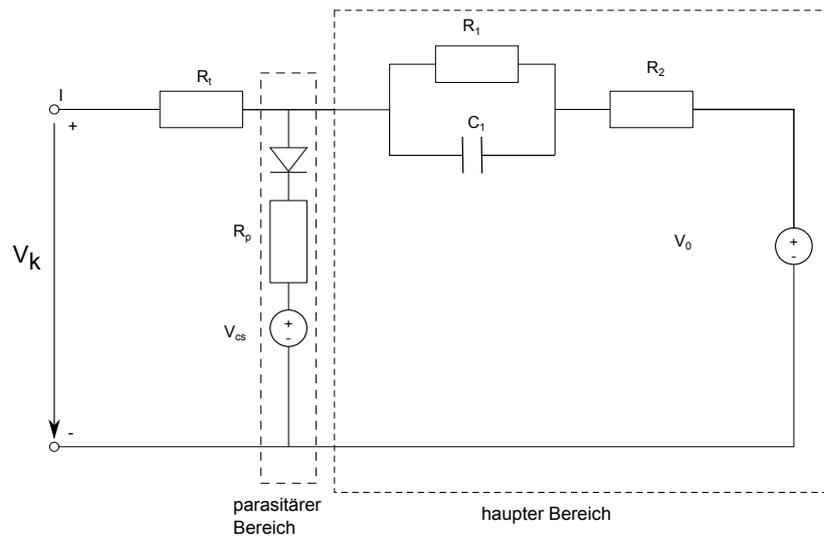


Abbildung 2.5.: Modell 3

- Modell 4 – "RC Battery 1"

Dieses Modell Abb. 2.6 besteht aus einem großen Kondensator C_{bulk} , der die Fähigkeit zur Speicherung der Ladungen besitzt. Es gibt einen Kondensator $C_{surface}$, der dynamische Effekte und die Diffusionseffekte der Batterie darstellt und drei Widerstände. R_t entspricht dem Klemmenwiderstand [7].

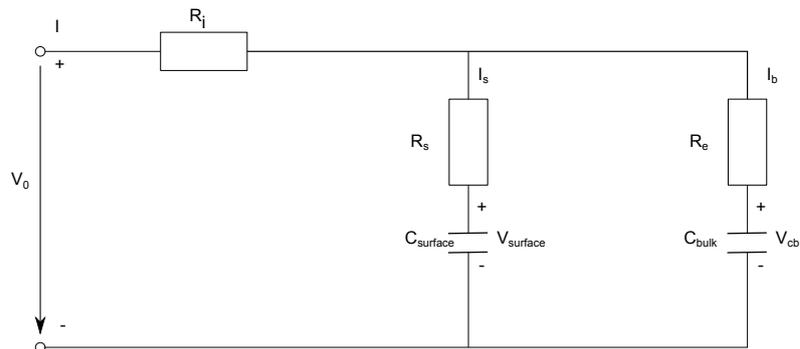


Abbildung 2.6.: Modell 4

- Modell 5 – "RC Battery 2"

Dieses in Abb. 2.7 gezeichnete Modell hat einen Kondensator zur Bildung der Kapazität der Batterie. Eine RC - Parallelschaltung stellt das dynamische Verhalten der Batterie dar. Der Klemmenwiderstand R_t bildet einen Teil des Innenwiderstandes. Der Ladezustand der Batterie wird durch lineare Beziehung zwischen V_{cb} und Ladezustand SOC gebildet [41].

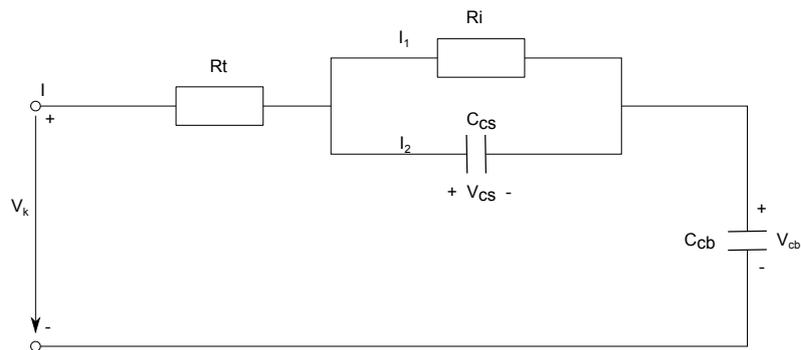


Abbildung 2.7.: Modell 5

Die Unterschiede zwischen den zuvor genannten Modellen werden durch Tabelle 2.2 verdeutlicht.

Kriterien / Modell	Modell 1	Modell 2	Modell 3	Modell 4	Modell 5
Laden	++	++	+	++	++
Entladen	++	+	++	++	++
Polarisation/ Diffusion	+	+	+	+	+
Gasung	-	-	-	-	-
Selbstentladung	-	+	-	-	-
Induktivität	-	-	-	-	-
Wechselstrom- widerstand	-	-	+	-	-
Temperatur- abhängigkeit	-	-	+	-	-
SOC- Abhängigkeit	-	-	++	-	+
Anzahl der Pa- rameter*	2+0+8	2+3+6	6+5+n	1+3+5	1+2+6
verwendetes Verfahren	Minimierungs- verfahren	Minimierungs- verfahren	Versuch	Minimierungs- verfahren / Extended Kalman- Filter	Minimierungs- verfahren / Extended Kalman- Filter
Aufwand	normal	normal	hoch	normal	niedrig
Durchführbarkeit	normal	normal	schwer	normal	normal
Literatur	[8]	[17]	[21]	[7]	[41]

Tabelle 2.2.: Vergleich der Modelle

Bemerkung zur Tabelle:

Das Zeichen „+“ bedeutet, dass das Modell dieses Kriterium erfüllt.

Das Zeichen „++“ bedeutet, dass mit diesem Modell diese Eigenschaft gut dargestellt werden kann.

Das Zeichen „-“ bedeutet, dass das Modell keine Fähigkeit zur Modellierung dieser Eigenschaft der Batterie hat.

* Anzahl der gebrauchten Parameter = Anzahl der durch Messungen bestimmten Parameter + Anzahl der Hersteller-Parameter + Anzahl der durch die Versuche bestimmten Parameter

2.2.3. Auswahl eines Modells

Die Tabelle. 2.2 zeigt verschiedenen Eigenschaften einzelner Modelle.

Im Vergleich zum Modell 1, 2, 3 benötigen die Modelle 4 und 5 weniger Parameter, die im Voraus durch Versuche oder von Datenblättern des Herstellers bestimmt werden sollten.

Wenige Parameter erzeugen einen geringeren Rechenaufwand

Im Vergleich zum Modell 4 und Modell 5 sind das Modell 2 und das Modell 3 eher schwer in einer Differentialgleichung umzusetzen, die später im Kalman Filter angewendet wird.

Im Vergleich zu anderen Modelle berücksichtigt das Modell 3 besonders den Einfluss der Temperatur und des Ladezustandes auf das Verhalten der Batterie.

Im Vergleich zum Modell 2, 3, 4 bezeichnet das Modell 1 und das Modell 5 nur eine einfache lineare Beziehung zwischen Ladezustand und Ruhespannung.

Modell 4 wird als Erstes zur Implementation ausgewählt. Dieses ist wie folgt zu begründen

- Das Modell 4 benötigt zur Beschreibung des Verhaltens der Batterie nur Widerstände und Kondensatoren und kann einfach in Differentialgleichungen umgesetzt werden, die im Kalman Filter angewendet werden.
- Das Modell 4 besitzt nur fünf Parameter. Dabei wird zur Implementation dieses Modells weniger Rechenaufwand benötigt.
- Das Modell 4 passt gut zur Beschreibung des dynamischen Verhaltens der Batterie, weil es zur Beschreibung des dynamischen Verhaltens einen zusätzlichen Kondensator zur Verfügung stellt.
- Das Modell 4 kann einfach durch das Hinzufügen anderer Elemente erweitert werden.

Trotzdem ist ein Nachteil des Modells 4 bemerkbar. Es vernachlässigt den Einfluss der Temperatur und des aktuellen Ladezustandes auf das Verhalten der Batterie, weil alle Parameter konstant gehalten werden.

Während der weiteren Untersuchung im Rahmen dieser Arbeit sollte das Modell 4 stets variiert werden, um das Verhalten der Batterie unter unterschiedlichen Lastprofilen besser darzustellen.

2.3. Verfahren zur Parameteridentifikation

Die Methode der kleinsten Quadrate ist eines der am häufigsten verwendeten Verfahren zur Parameteridentifikationsaufgabe. Dieses Verfahren wird in Rahmen dieser Arbeit zur Bestimmung der Parameter verwendet.

Im Folgenden wird ein Anlehnung an [1] die Methode der Kleinsten Quadrate beschrieben. Die Methode der kleinsten Quadrate (enGl: method of least squares) ist das mathematische Standardverfahren zur Ausgleichsrechnung. Dabei wird zu einer Datenpunktwolke eine Kurve gesucht, die möglichst nahe an den Datenpunkten verläuft. Die Daten können physikalische Messwerte, wirtschaftliche Größen oder Ähnliches repräsentieren, während die Kurve aus einer parameterabhängigen problemangepassten Familie von Funktionen stammt. Die Methode der kleinsten Quadrate besteht dann darin, die Kurvenparameter so zu bestimmen, dass die Summe der quadratischen Abweichungen der Kurve von den beobachteten Punkten minimiert wird. Die Abweichungen werden Residuen genannt. In der Abb. 2.8 sind Datenpunkte eingetragen. In einem ersten Schritt wird eine Funktionenklasse ausgewählt, die zu dem Problem und den Daten passen sollte, hier eine logistische Funktion. Deren Parameter werden nun so bestimmt, dass die Summe der Quadrate der Abweichungen e der Beobachtungen y zu den Werten der Funktion minimiert wird. In der Abb. 2.8 ist die Abweichung e an der Stelle x als senkrechter Abstand der Beobachtung y von der Kurve zu erkennen.

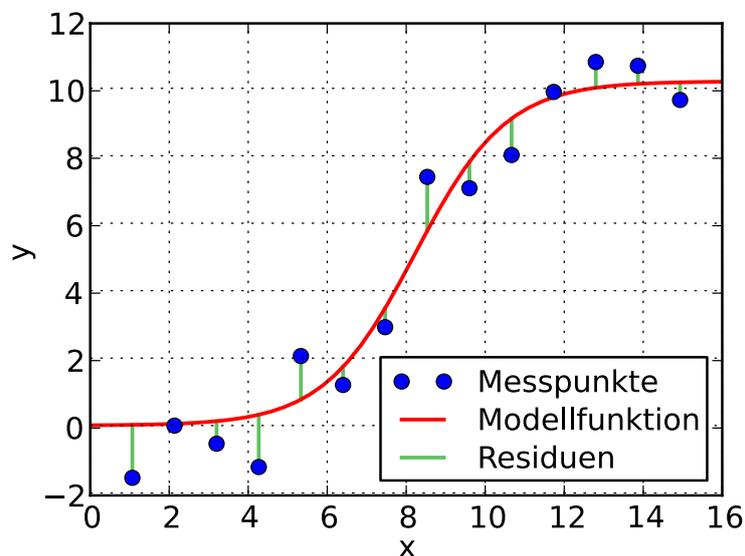


Abbildung 2.8.: Beispiel der Methode kleinsten Quadrate [1]

In der Stochastik wird die Methode der kleinsten Quadrate meistens als Schätzmethode in der Regressionsanalyse benutzt, wo sie auch als Kleinste-Quadrate-Schätzung bezeichnet wird. Angewandt als Systemidentifikation ist die Methode der kleinsten Quadrate in Verbindung mit Modellversuchen für Ingenieure ein Ausweg aus der paradoxen Situation, Modellparameter für unbekannte Gesetzmäßigkeiten zu bestimmen [1].

2.3.1. Algorithmus

Eine Modellfunktion mit m Parametern (a_1, \dots, a_m) stellt das Polynom $m - 1$ te Ordnung dar. (Gl. 2.12) Gesucht werden zu gemessenen Werten $((x_1, y_1), \dots, (x_n, y_n))$ die Koeffizienten (a_1, \dots, a_m) . Die Modellfunktion kann in Matrizenform umformuliert werden wie Gl. 2.13.

$$\begin{aligned} a_1 + a_2 * x_1 + \dots + a_m * x_1^{m-1} &= y_1 \\ &\vdots \\ a_1 + a_2 * x_n + \dots + a_m * x_n^{m-1} &= y_n \end{aligned} \quad (2.12)$$

$$\underbrace{\begin{pmatrix} 1 & x_1 & \dots & x_1^{m-1} \\ \vdots & & & \vdots \\ 1 & x_n & \dots & x_n^{m-1} \end{pmatrix}}_{M \in \mathbb{R}^{n \times m}} \underbrace{\begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}}_{a \in \mathbb{R}^m} = \underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}_{y \in \mathbb{R}^n} \quad (2.13)$$

Die Abweichung γ zwischen der gesuchten Kurve und den Messwerten wird als Anpassungsfehler oder Residuen bezeichnet (Gl. 2.14). Gesucht sind die Koeffizienten (a_1, \dots, a_m) , die den Summe der quadratischen Fehler minimiert (Gl. 2.15 - Gl. 2.16)

$$\begin{aligned} \gamma_1 &= y_1 - a_1 - a_2 * x_1 - \dots - a_m * x_1^{m-1} \\ &\vdots \\ \gamma_n &= y_n - a_1 - a_2 * x_n - \dots - a_m * x_n^{m-1} \end{aligned} \quad (2.14)$$

$$|Ma - y|^2 = \left(\sum_{k=1}^m a_k x_1^{k-1} - y_1 \right)^2 + \dots + \left(\sum_{k=1}^m a_k x_n^{k-1} - y_n \right)^2 \quad (2.15)$$

$$\begin{aligned} f(a_1, \dots, a_m) &= |Ma - y|^2 \\ &= (Ma - y)^T (Ma - y) \\ &= a^T M^T Ma - 2a^T M^T y + y^T y \end{aligned} \quad (2.16)$$

Durch partielle Ableitung der Funktion (GL. 2.16) enthält man GL. 2.17. Die Lösung a^* löst diese Gleichung.

$$0 = \nabla_a f = 2M^T Ma - 2M^T y \quad (2.17)$$

$$a^* = (M^T M)^{-1} M^T y \quad (2.18)$$

Ist die Lösung a^* (Gl. 2.18) tatsächlich ein Minimum, sollte die Hess - Matrix an diesen Punkt positiv semidefinit sein. D.h., $M^T M$ invertierbar soll sein.

2.3.2. Beispiel

Die Regressionsgerade durch die 4 Punkte $(0, 1)$, $(2, 3)$, $(3, 3)$, $(4, 5)$ soll mithilfe der Methode der kleinsten Quadrate festgelegt werden. Laut der Modellfunktion Gl. 2.12 wird die Regressionsgleichung aufgebaut Gl. 2.19.

$$y = a_1 + a_2 * x \quad (2.19)$$

Hierzu werden 4 Punkte in Gl. 2.19 eingesetzt.

$$\begin{aligned} a_1 + 0 * a_2 &= 1 \\ a_1 + 2 * a_2 &= 3 \\ a_1 + 3 * a_2 &= 3 \\ a_1 + 4 * a_2 &= 5 \end{aligned} \quad (2.20)$$

Anhand der Gl. 2.18 werden die am besten angepassten Parameter der Regressionsgerade gefunden. (Gl. 2.21-Gl. 2.25)

$$M = \begin{pmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} \quad (2.21)$$

$$y = \begin{pmatrix} 1 \\ 3 \\ 3 \\ 5 \end{pmatrix} \quad (2.22)$$

$$M^T M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 9 \\ 9 & 29 \end{pmatrix} \quad (2.23)$$

$(M^T M)^{-1}$ existiert, weil $\det(M^T M) = 4 * 29 - 9 * 9 = 35 \neq 0$

$$(M^T M)^{-1} = \frac{1}{35} \begin{pmatrix} 29 & -9 \\ -9 & 4 \end{pmatrix} \quad (2.24)$$

Damit ergibt sich die Parameter a^* :

$$\begin{aligned} a^* &= (M^T M)^{-1} M^T y \\ &= \frac{1}{35} \begin{pmatrix} 29 & -9 \\ -9 & 4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 4 \\ 5 \end{pmatrix} \\ &= \begin{pmatrix} 0.9429 \\ 0.9142 \end{pmatrix} \end{aligned} \quad (2.25)$$

Die Regressionsgerade lautet somit $y = 0.9429 + 0.9142 * x$

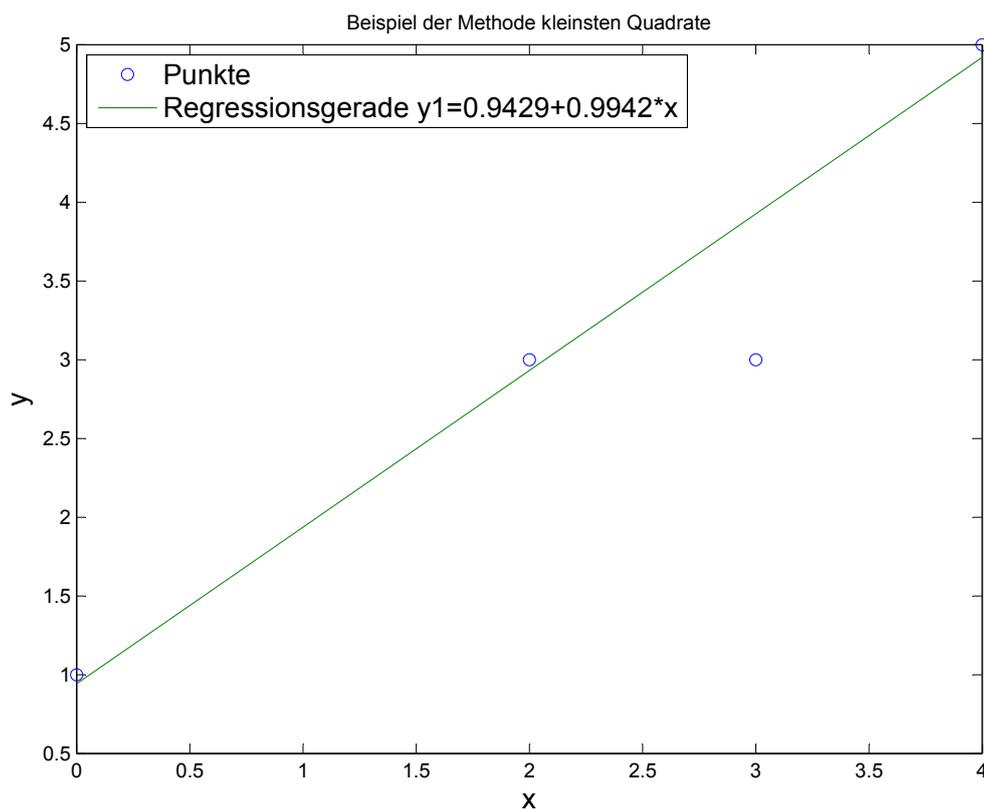


Abbildung 2.9.: Beispiel der Methode kleinster Quadrate

3. Implementation von Batteriemodellen in Matlab/Simulink und Messung

In diesem Kapitel wird ein Modell, das das Verhalten der Batterie so genau wie möglich darstellen kann, untersucht. Wie bereits im Abschnitt 2.2 erwähnt, sollte mit Hilfe dieses Modells eine zuverlässige Aussage zu dem momentanen Ladezustand bzw. der noch verfügbaren Ladung getroffen werden. Ein Messplatz wird aufgebaut. Er sollte in der Lage sein, die Messung der Klemmenspannung, des Stroms und der Temperatur zu ermöglichen und die gemessenen Werte zur weiteren Verarbeitung zur Verfügung zu stellen. Um das Verhalten der Batterie unter unterschiedlichen Lastprofilen zu erfassen und anschließend zu vergleichen, werden verschiedene Messungen durchgeführt. Der Messwert wird mit der simulierten Ausgabe des implementierten Modells verglichen. Damit wird bewertet, ob das eingesetzte Modell alle oder Teile des Verhaltens der Batterie präzise beschreiben kann. Danach wird untersucht, ob es Möglichkeiten gibt, das Modell durch das Hinzufügen zusätzlicher Elemente zu erweitern, um ein anderes Verhalten der Batterie charakterisieren zu können. Das Modell wird in Simulink erstellt und die im Modell angepassten Parameter werden durch die Annäherung der Messkurve in Matlab identifiziert.

3.1. Messplatz

Eine Zelle der Bleibatterie wird beim Entladen bzw. Laden gemessen. Beim Entladen wird die Zelle an einer elektronischen Last angeschlossen. Die elektronische Last gewährleistet, dass während der Messung die Last konstant bleibt. Wird die Zelle mit dem Ladegerät verbunden, startet der Ladevorgang. Anhand der manuell einstellbaren Spannungsgrenze regelt das Ladegerät selbst den Strom. Eine Relaiskarte dient zur Steuerung des Lade- bzw. Entladevorgangs. Die Ein- bzw. Ausschaltzeit des Relais wird durch das Matlab-Skript gesteuert. Der in der Batteriezelle hinein- bzw. herausfließende Strom wird durch einen Hall-Sensor (LEM DHAB S/24) erfasst. Der gemessene Spannungswert an dem Hall-Sensor wird anhand der Formel vom Datenblatt des Herstellers in einen Stromwert umgewandelt. Die Klemmspannung der Zelle wird durch ein Tischmultimeter Fluke 45 gemessen. Ein Heißleiter NTC dient

zur Messung der Temperatur der Batterie. Der Heißleiter wird an einer Seite des Gehäuses der Bleibatterie angeklebt. Der Widerstand wird direkt gemessen. Alle von den drei Tischmultimetern gemessenen Werte werden via USB an den Computer übertragen und in Matlab weiterverarbeitet. In Abb. 3.1 ist der schematische Aufbau des Messplatzes dargestellt und die eingesetzten Geräte sind in der Tabelle 3.1 aufgelistet.

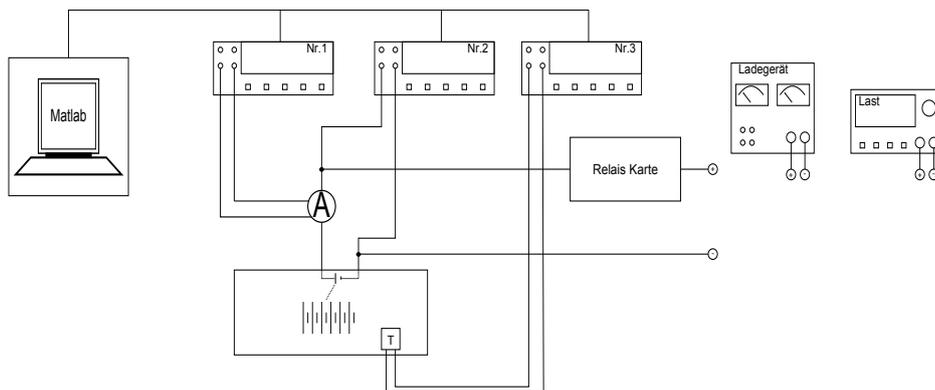


Abbildung 3.1.: Messplatz

Gerät	Funktion
Banner Starterbatterie	Starterbatterie 12V 88Ah 395A
NTC	Temperatur Sensor
Fluke 45 Nr.1	Messung der Spannung am Hall-Sensor
Fluke 45 Nr.2	Messung der Klemmenspannung einer Batteriezelle
Fluke 45 Nr.3	Messung der Spannung am NTC-Sensor
LEM DHAB S/24	Strom Sensor
8 Fach Relais Karte 230V / 16A	Steuerung des Lade- bzw. Entladevorgangs
Stromversorgung	5V Spannungsversorgung für Hall-Sensor
Elektronische Last	konstante Last
Ladegeräte	Ladevorgang

Tabelle 3.1.: Eingesetzte Messgeräte

3.2. Modellierung in Simulink

Das Ersatzschaltbild des Modells kann in Simulink erstellt werden. Die dazu benötigten Funktionsblöcke wie z.B. Kondensator, Spannungsquelle, Widerstand usw. befinden sich in der Simulink - Bibliothek „SimPowerSystems“. Nach der Erstellung des zu untersuchenden Modells in Simulink durch Verknüpfung der benötigten Funktionsblöcke werden die Parameter in die zugehörigen Funktionsblöcke eingetragen. Nachdem die Simulationsbedingungen bestimmt werden, z.B. Dauer der Simulation, Abtastrate usw. wird die Simulation gestartet. Die Simulationsergebnisse lassen sich beispielsweise durch den „Scope - Block“ veranschaulichen. Durch den Block „To File“ oder „To Workspace“ können die Signalverläufe in einer Datei oder im Workspace von Matlab abgelegt werden.

3.3. Minimierungsverfahren in Matlab

Unter den Begriff Methode der kleinsten Quadrate gibt es verschiedene Befehle in Matlab, wodurch eine quadratische Fehlersumme minimiert wird. Der Aufruf erfolgt beispielsweise durch den Befehl:

$$[x, resnorm, res, exit] = lsqin(@funktion, x0, lb, ub, options)$$

In der Funktion wird die Simulation des Modells mit dem Befehl „sim“ aufgerufen und gestartet. Anschließend wird die Ausgabe der Simulation mit dem Messwert verglichen. Die Summe der quadratischen Abweichung zwischen der Ausgabe und dem Messwert wird berechnet.

Ziel des Minimierungsprozesses ist es, die Parameter zu finden, für die sich eine minimale quadratische Abweichung ergibt. Dabei ist „x0“ der Startwert der Parameter. Ein geeigneter Anfangswert hilft dabei, die Dauer der Suche nach angepassten Parametern zu verkürzen. Bei vielen Optimierungsaufgaben interessiert nicht nur ein Minimum der Zielfunktion, vielmehr müssen die Parameter verschiedene Nebenbedingungen erfüllen. Solche Nebenbedingungen können je nach der Aufgabestellung oder physikalischen Begrenzungen bedingt sein, z.B. Ober- und Untergrenze des Widerstandes.

Wenn Messdaten nicht durch ein Polynom sondern durch eine beliebige nichtlineare Funktion approximiert werden, erfolgt die Lösung des Optimierungsproblems beispielsweise in Matlab mit dem Befehl „lsqcurvefit“.

$$[x, resnorm, res, exit] = lsqcurvefit(@funktion, x0, xdata, ydata, lb, ub, options)$$

Zusätzlich sollten die Datenvektoren $xdata$ und $ydata$ eingegeben werden. $xdata$ ist der Zeitpunkt $t = [t_1, \dots, t_n]$ und in $ydata$ steht der zugehörige Messwert $y = [y_1, \dots, y_n]$.

Sobald irgendeine Abbruchbedingung erreicht ist, wird der Optimierungsprozess beendet und neben den berechneten Parametern x werden die Informationen über das Abbruchkriterium und die erreichte minimale Toleranz ausgegeben. Sonst wird die Optimierung weiter

laufen.

Der Abbruch erfolgt, wenn Änderung zwischen 2 Iterationen nur noch geringfügige Verbesserungen bringt, die Änderung der Summe der quadratischen Fehler geringer als die vorgegebene Toleranz ist, die Änderung in der Suchrichtung kleiner als die angegebene Toleranz ist, und die maximale Anzahl von Iterationen erreicht oder andere Fehlermeldung aufgetreten ist.

3.4. Allgemeiner Ablauf des Programms

Zunächst wird ein Modell in Simulink erstellt, das sich für die Beschreibung des Verhaltens der Batterie eignet. Als nächster Schritt ist die Zielfunktion aufzustellen. In der Zielfunktion wird die Simulation des in Simulink erstellten Modells mit Befehl „sim“ aufgerufen und anschließend wird die quadratische Abweichung zwischen der Ausgabe der Simulation und dem Messwert aufsummiert. Im Anschluss daran werden Startwerte der Parameter, Optimierungsstrategie und Nebenbedingungen, womit die Parameter sich beschränken lassen, festgelegt. Schließlich kann die Optimierung gestartet werden.

In der Optimierung wird die Zielfunktion aufgerufen und darin die Summe des quadratischen Fehlers berechnet. Anhand des Simulationsergebnisses wird geprüft, ob ein Abbruchkriterium erreicht ist. Falls nicht, wird die Optimierung mit angepassten Parametern wiederholt. Die Abb. 3.2 zeigt den Ablauf des allgemeinen Programms.

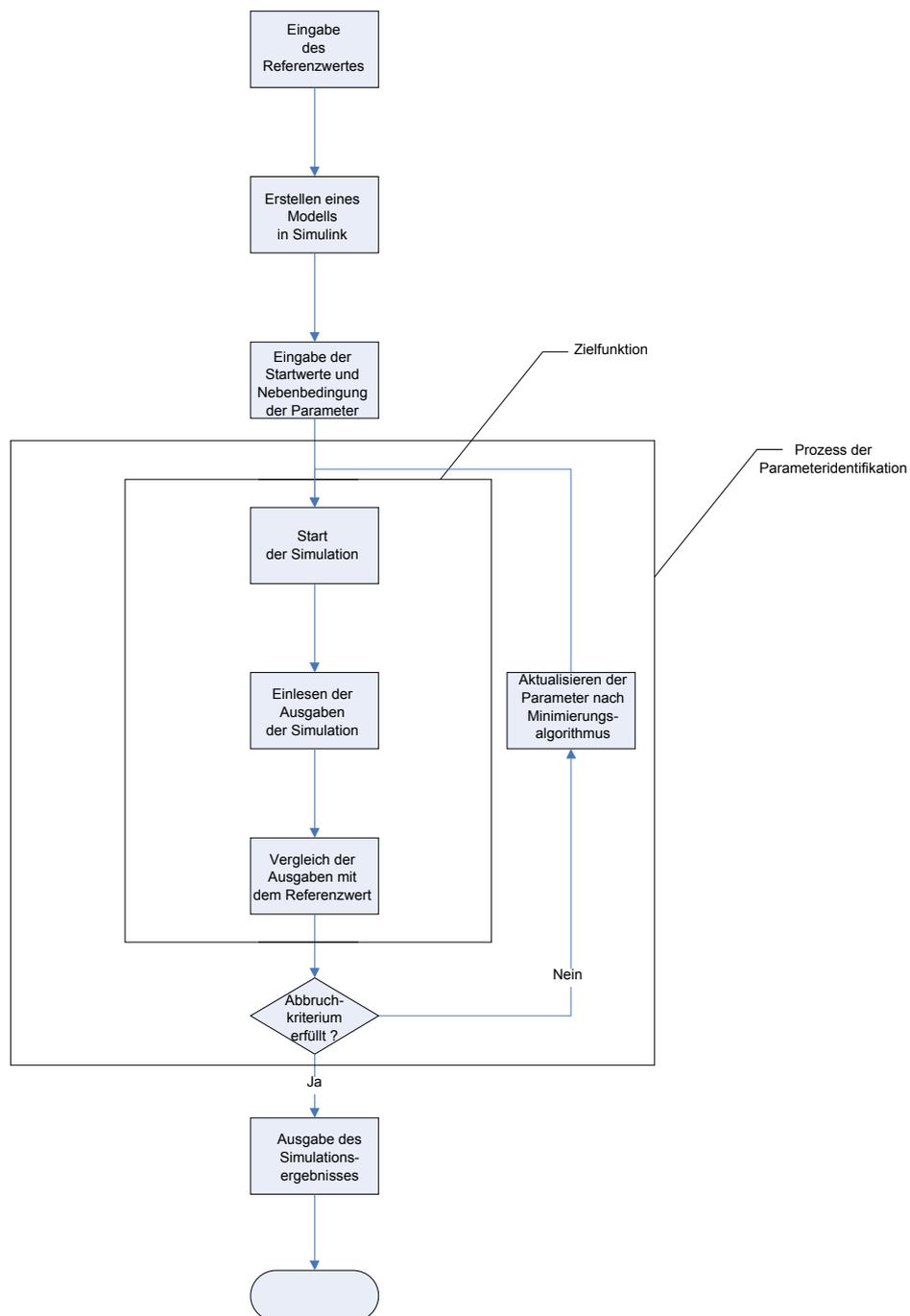


Abbildung 3.2.: Allgemeiner Ablauf des Programms

3.5. Einfaches Modell - Modell 4

Als Erstes wird das Modell 4 (Abb. 3.3) in Matlab implementiert, weil es das dynamische Verhalten der Batterie relativ gut darstellen kann und wenig Rechenaufwand benötigt. Bei der Starterbatterie ist das Verhalten bei kurzzeitiger Belastung wichtiger als das bei Langzeitbelastung, weil beim Startvorgang die Starterbatterie nur einige Sekunden belastet wird. Außerdem wird die Starterbatterie selten vollständig entladen, weil nach dem Startvorgang der Generator die Starterbatterie sofort wieder auflädt. Eine andere Ursache für die Auswahl des einfachen Modells 4 besteht darin, dass es leichter in den Kalman-Filter integriert werden kann. Das Simulationsergebnis des Modells 4 wird mit dem Messwert verglichen. Anhand der Abweichung dazwischen wird das Modell bewertet.

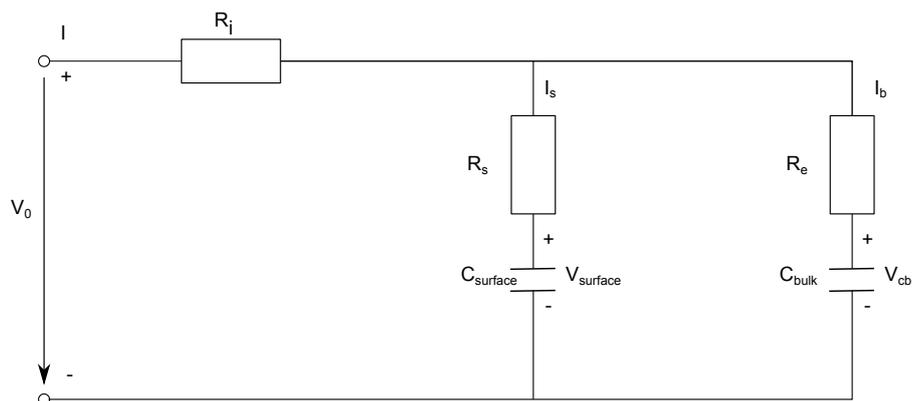


Abbildung 3.3.: Modell 4

3.5.1. Implementation des ausgewählten Modells 4

In Abb. 3.4 wird das in Simulink erstellte Modell dargestellt.

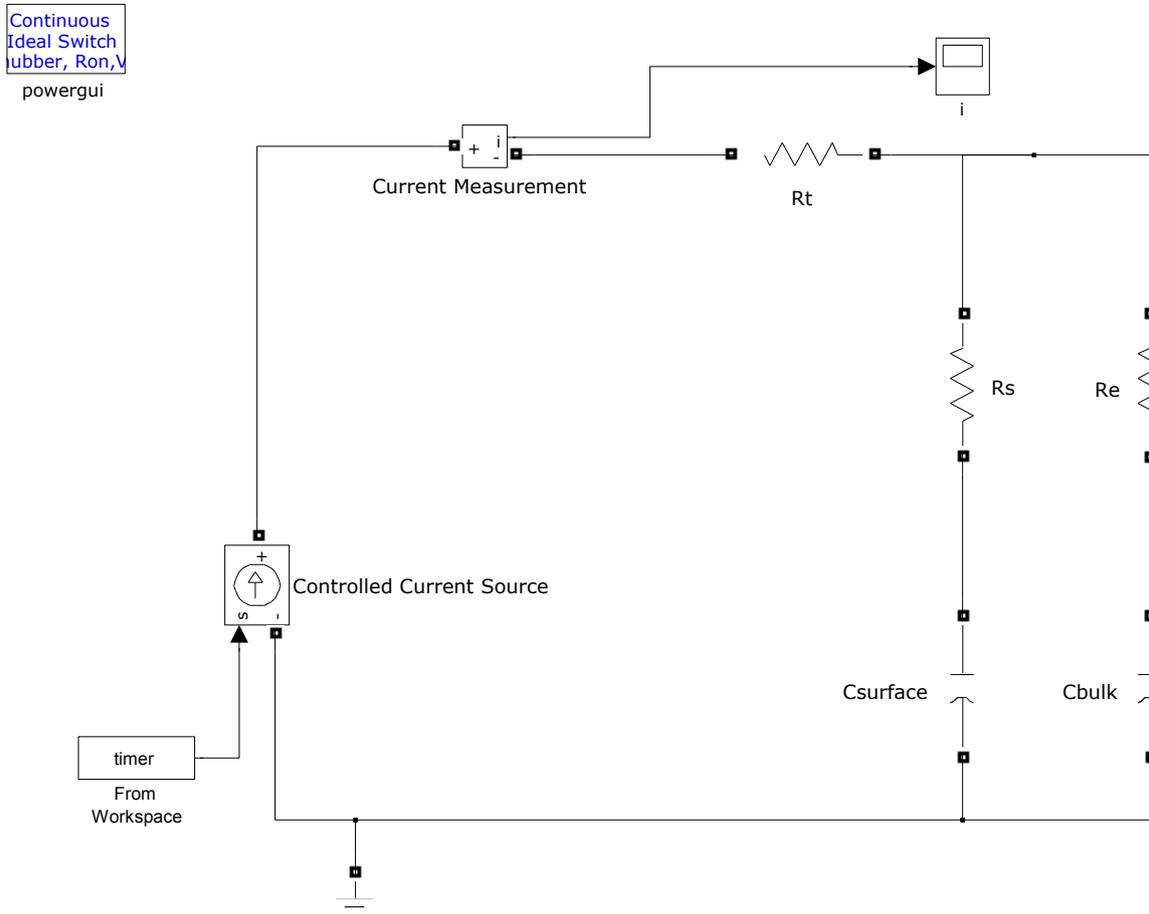


Abbildung 3.4.: Das in Simulink erstellte Modell 4

Dynamisches Verhalten beim Entladen

Eine Zelle der Starterbatterie wird mit einem konstanten Strom (ungefähr zehn Ampere) belastet. Im unteren Teil der Abb 3.5 lässt sich der Lastvorgang veranschaulichen.

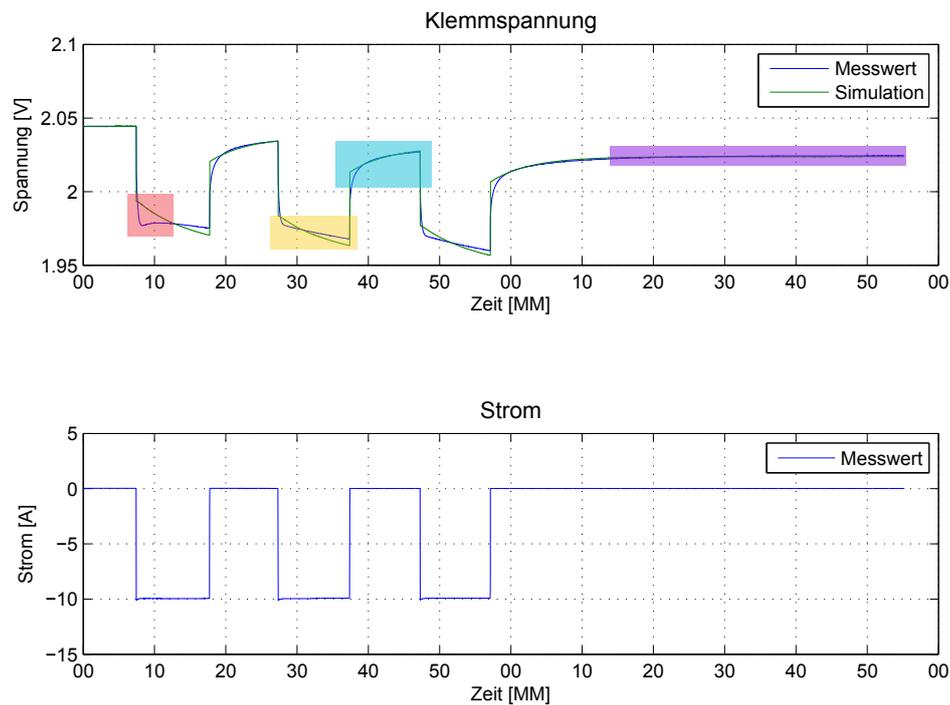


Abbildung 3.5.: Vergleich des dynamischen Verhaltens des Modells 4 mit dem Messwert
Lastzyklus: 30 Minuten

Abb. 3.5 zeigt die Klemmenspannung im Vergleich zwischen Modell 4 und dem Messwert

- Es ist zu beobachten, dass beim ersten Entladevorgang die Zellspannung sinkt und zwar zuerst stark, dann erholt sie sich wieder ein bißchen (roter Bereich). Diese Spannungsabsenkung (Spannungssack) erklärt sich aus dem Konzentrationsgefälle zwischen innerer und äußerer Säure. Die Ruhespannung nimmt erst durch die Bedeckung der Oberfläche mit Bleisulfat ab, bis die Keimbildung abgeschlossen ist. Der Spannungssack tritt nicht auf, wenn noch Kristallkeime vorhanden sind, d.h. Batterie war nicht vollgeladen [40].

Diese Spannungsabsenkung wird durch das einfache Modell 4 nicht dargestellt. Um eine solche Spannungsabsenkung nachzubilden, sollte mindestens ein zusätzlicher Energiespeicher im Modell hinzugefügt werden. In der Langzeit-Messung kann der Einfluss des Spannungssacks vernachlässigt werden.

- In der ersten Hälfte des Entladezyklus (gelber Bereich) nimmt die Spannung unter der Last schneller ab, als sie sich in der zweiten Hälfte (blauer Bereich) wieder erholt. Dieser Unterschied ist auch am Winkel des Verlaufs der Messkurve in beiden Hälften erkennbar.

Dieser Unterschied in der Zeitkonstante beim Absinken und Erholen der Spannung kann durch das Modell 4 nicht realisiert werden, weil im Modell 4 nur eine RC-Reihenschaltung (R_{CS} und $C_{surface}$) zur Beschreibung des dynamischen Verhaltens zur Verfügung gestellt ist. Es müssen weitere Elemente im Modell eingesetzt werden, um diesen Unterschied zu realisieren.

- In der Ruhephase stimmt der Verlauf der simulierten Kurve mit dem Messwert überein (lilaer Bereich). Bei der Aufsummierung der Abweichungen zwischen der simulierten Ausgabe und dem Messwert in der Zielfunktion (siehe Abschnitt 3.4) werden die, sich in der Ruhephase befindlichen Abweichungen höher gewichtet, als die Abweichungen außerhalb der Ruhephase, weil bei der Bestimmung des Ladezustands der Batterie die Ruhespannung eine wichtigere Rolle spielt.

Dynamisches Verhalten bei langer Entladung

Eine Zelle der Starterbatterie wird mit einem konstanten Strom (ungefähr vier Ampere) in verschiedenen Lastzyklen belastet, siehe Abb. 3.6.

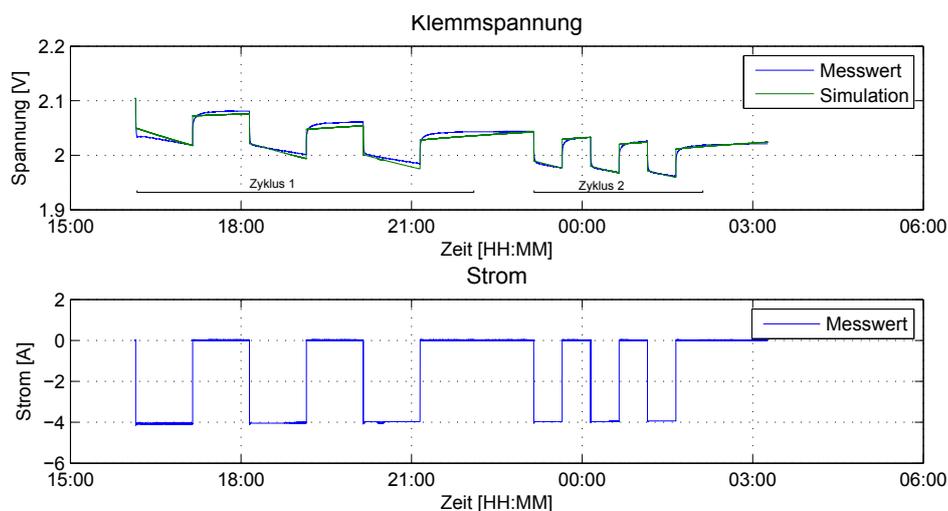


Abbildung 3.6.: Vergleich des langen dynamischen Verhaltens des Modells 4 mit dem Messwert Zyklus 1: 1 Stunde; Zyklus 2: 30 Minuten

Im Vergleich zum Messwert verläuft die simulierte Kurve sowohl in der Lastphase als auch in der Ruhephase zu flach. Es ist offensichtlich erkennbar, dass die Abweichung zwischen der Simulationskurve und der Messkurve im Lastzyklus von 30 Minuten (Zyklus 2) deutlich geringer ist, als die sich im Lastzyklus von einer Stunden (Zyklus 1) befindenden Abweichung. Außerdem stellt das Modell 4 den Spannungsverlauf bei der Abschaltzeit bzw. Einschaltzeit von der Last immer mit großer Abweichung dar.

Kontinuierliche Entladung

Eine Zelle der Starterbatterie wird mit einem Konstantstrom (etwa fünf Ampere) belastet. Der Verlauf der Messkurve zeigt den Spannungsverlauf einer Zelle der Bleibatterie vom vollgeladenen Zustand bis zum entladenen Zustand. Für die Ruhespannung von $2,1\text{ V}$ wird die Batterie als voll angenommen, $1,95\text{ V}$ Ruhespannung wird als Wert für eine leere Batterie genommen. Für die Entladeschlussspannung wurde $1,75\text{ V}$ festgelegt [24].

Die Entladezeit ist abhängig von der Höhe des Entladestroms, dem initialen Ladezustand und der Temperatur der Umgebung. Je niedriger die Stromentnahme, desto länger dauert der Entladeprozess und desto mehr Kapazität kann der Batterie entnommen werden. Der Unterschied zwischen entnehmbarer Kapazität beruht grundsätzlich auf dem Konzentrationsgefälle zwischen innerer und äußerer Säure.

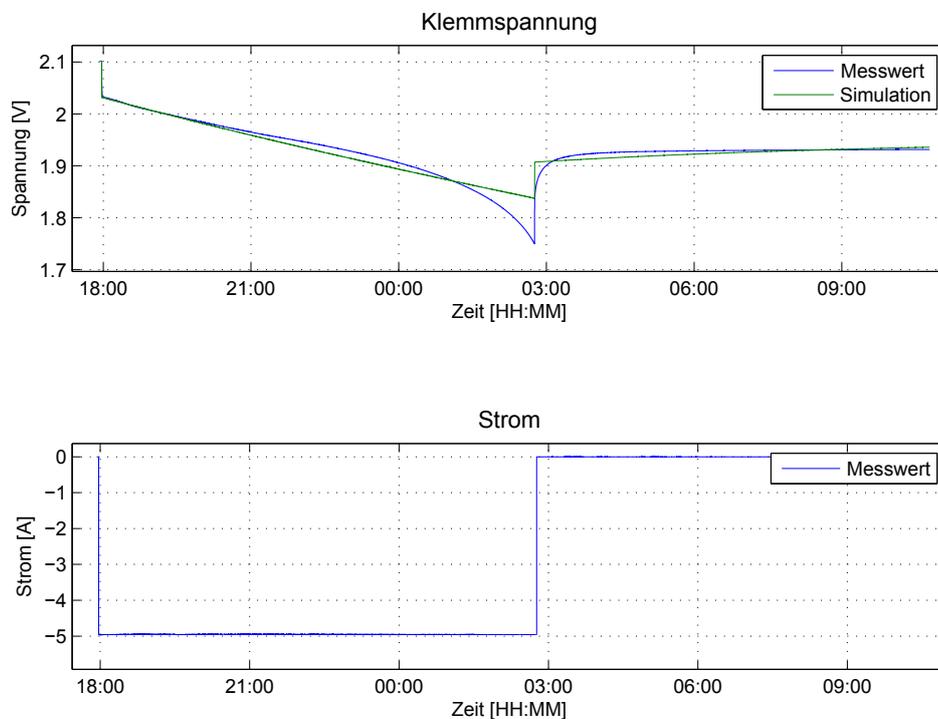


Abbildung 3.7.: Vergleich des Verhaltens der durchgehenden Entladung vom Modell 4 mit dem Messwert

Im oberen Teil der Abb. 3.7 ist zu sehen, dass der Verlauf der simulierten Spannung nur im Anfangs- und Endbereich gut zur gemessenen Spannung passt. Durch zwei konstanten Kondensatoren mit großer Kapazität kann die Messkurve nicht dargestellt werden.

Aufladen

Das Laden wird mit einem durch Ladegeräte geregelten I-U Verfahren durchgeführt. Das I-U Verfahren verbindet das Konstantstrom-Ladeverfahren mit dem Konstantspannung-Ladeverfahren. Erst wird die Batterie mit einem konstanten, durch das Ladegerät begrenzten Strom geladen. Erreicht die Spannung der Batterie die ausgewählte Ladeschluss-spannung, wird von Strom- auf Spannungsregelung umgeschaltet. Anschließend wird die Batterie mit konstanter Spannung geladen, Dabei nimmt der Strom ständig ab. Während der Ladung mit konstanten Strom (I-Ladung) kann innerhalb einer kurzen Zeit ein Großteil der Kapazität geladen werden. Während der Ladung mit konstanter Spannung (U-Ladung) wird mit reduzierter Geschwindigkeit die Batterie vollständig aufgeladen. Die I-U Ladekennlinie wird in die Abb. 3.8 dargestellt.

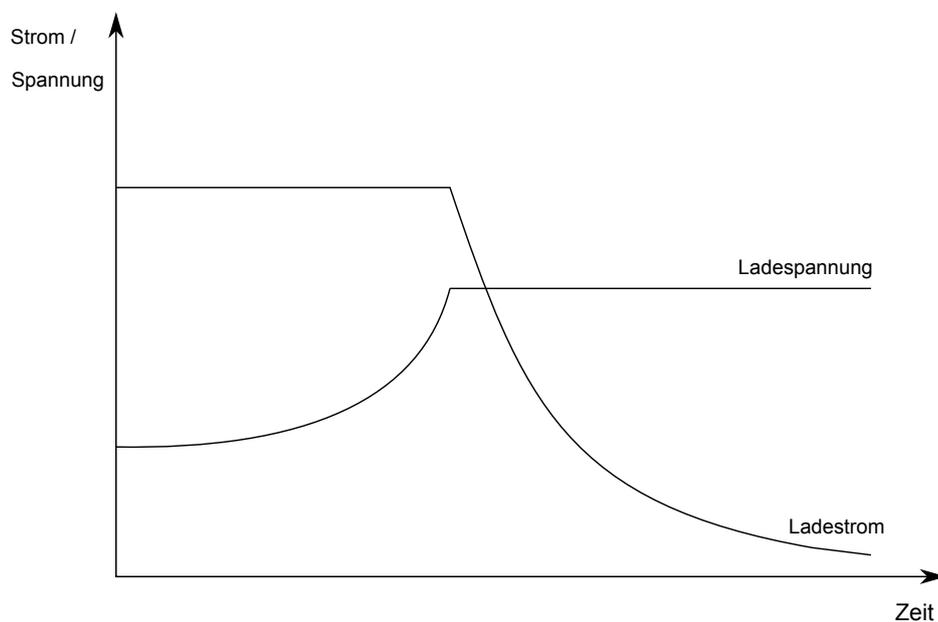


Abbildung 3.8.: I-U Ladeverfahren

Erreicht die Klemmspannung der Zelle 2,5V, trennt die Relais-Karte den Anschluss vom Ladegerät.

Eine Zelle der Batterie wird durch das Ladegerät geladen. Der Ladevorgang wird in der Abb. 3.9 angezeigt.

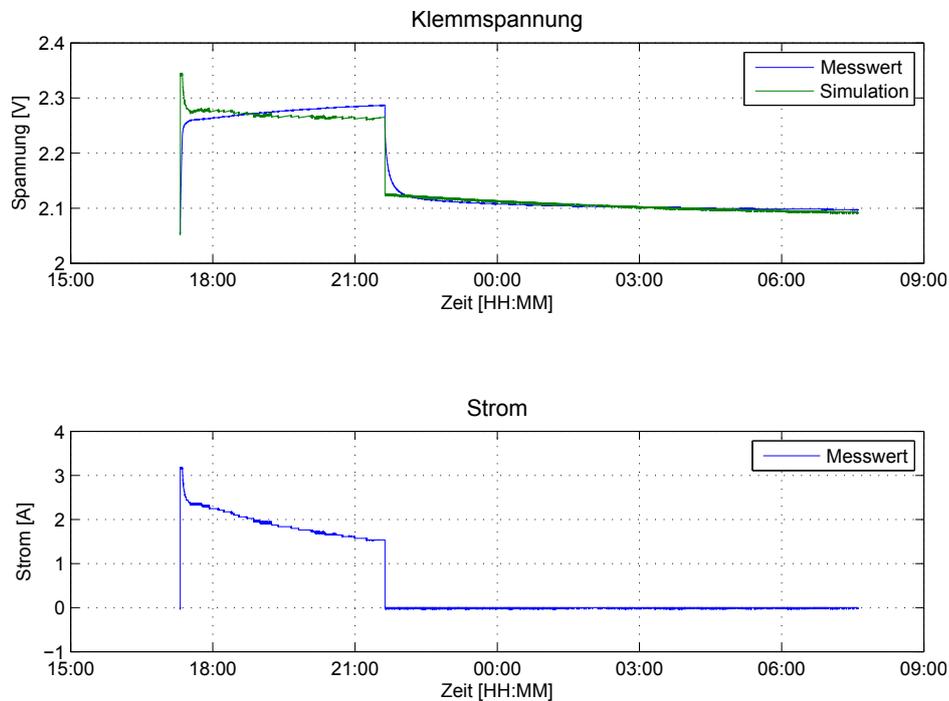


Abbildung 3.9.: Vergleich des Verhaltens beim Aufladen von Modell 4 mit dem Messwert

Der Ladevorgang mit konstantem Strom wird kurz nach dem Anfang des Ladevorgangs beendet, weil die Klemmspannung der Zelle schon 2,25 V erreicht hat. Bei Ladung mit konstanter Spannung nimmt der Strom ständig ab. Beim Aufladen müssen Bleiionen nach der Auflösung der Bleisulfatkristalle durch Diffusion zur Oberfläche der Elektrode transportiert werden. Mit abnehmendem Bleisulfatvorrat verlangsamt sich die Diffusionsgeschwindigkeit merklich, weil die Transportwege zur Oberfläche immer größer werden. Dabei vergrößert sich der Innenwiderstand. Erreicht die Klemmspannung der Zelle die im Programm voreingestellte Ladeschlussspannung, wird der Ladevorgang beendet, indem der Anschluss vom Ladegerät durch die Relais-Karte unterbrochen wird.

Der hohe Strom am Anfang des Ladevorgangs führt zum Spannungssprung. Der starke Spannungseinbruch erscheint beim Abschalten des Ladestroms. Der im Modell eingefügte Kondensator $C_{surface}$ ist hier für das Nachbilden der gemessenen Spannung nicht ausreichend. Der Wechsel der Spannung ist stark von Strom abhängig. Zur Beschreibung des Verlaufs beim Ein- und Ausschalten des Stroms fehlt es zwingend an einigen Elementen. Um den Sprung und den Einbruch zu dämpfen, muss möglicherweise ein zusätzliches RC-Glied in den Zweig, der das dynamische Verhalten beschreibt, hinzugefügt werden.

3.5.2. Auswertung und auftretende Probleme

Im vorliegenden Abschnitt wird das Simulationsergebnis des Modells 4 mit dem Entlade- und Ladeprofil und dem zugehörigen Messwert verglichen. Nun wird das Verhalten des Modells 4 kurz zusammengefasst. Wie erwartet, eignet sich das Modell 4 zur Beschreibung des Verhaltens der Batterie unter kurzzeitiger Belastung, d.h. der Zyklus der Belastung ist kürzer als 1 Stunde. Das Modell 4 kann den Verlauf der durchgehenden Entladung der Batterie nicht nachbilden. Beim Laden kann das Modell 4 lediglich den Hauptteil des Ladevorgangs gut darstellen.

Folgende Unterschiede zwischen dem gemessenen und dem simulierten Verlauf sind besonders auffällig.

- die Spannungsabsenkung beim ersten Entladevorgang nach einer längeren Ruhephase
- der dynamische Verlauf beim Ein- und Abschalten des Lade- und Entladestromes
- der Hauptteil bei durchgehender Entladung

3.6. Angepasstes Modell 4 Klasse 1

In diesem Abschnitt wird das Modell 4 noch erweitert, um obengenannte Fehler durch hinzugefügte Elemente zu korrigieren. Zum Vergleich wird das Modell 4 in der Abb. 3.10 dargestellt.

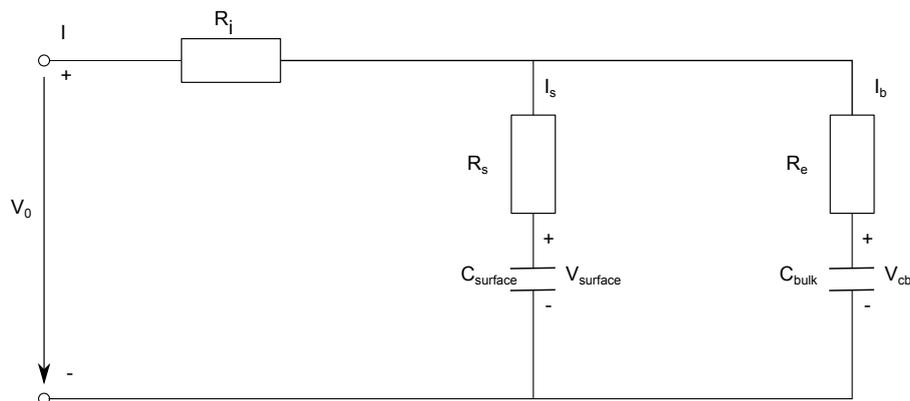


Abbildung 3.10.: Modell 4

Eine Parallelschaltung von einer idealen Diode (*Diode2*) und einer Induktivität (L) wird in Serie mit dem Kondensator C_{bulk} geschaltet. Bei schaltvorgängen bewirkt „L“ einen langsameren Anstieg bzw. Abfall des Stromes.

Wenn die Last eingeschaltet ist, entladen sich der Kondensator $C_{surface}$ und der Kondensator C_{bulk} . Die Spannung an der Spule nimmt zuerst relativ schnell zu und dann immer langsamer ab. Die Geschwindigkeit der Spannungszunahme bzw. Spannungsabnahme ist von der angelegten Spannung und der Induktivität abhängig. Dieser Effekt der Spule kann den Verlauf des Spannungssacks beim Entladen nachbilden. Nach dem Ausschalten der Last fließt weiter Strom aus dem Kondensator C_{bulk} mit großer Kapazität zu dem Kondensator $C_{surface}$ mit kleiner Kapazität, bis die Spannung in beiden Zweigen ausgeglichen ist. Beim nächsten Schaltvorgang hemmt die Spule wieder den Strom aus dem Kondensator C_{bulk} . Wegen der parallelgeschalteten *Diode2* wird die Spule beim Laden wirkungslos. Der im Modell 4 vor dem Kondensator $C_{surface}$ stehende Widerstand R_s wird durch ein Netzwerk ersetzt. Dieses Netzwerk besteht hauptsächlich aus zwei parallel geschalteten Widerständen R_{su} und R_{sd} mit jeweils einer Diode. Die Richtung der beiden Dioden ist entgegengesetzt.

Vor dem Widerstand R_{sd} wird noch eine RC-Parallelschaltung (R_1, C_1) aufgebaut. Diese Erweiterung der Schaltung dient dazu, die Zeitkonstante beim Entladen und beim Laden

zu variieren. Die zusätzliche RC-Parallelschaltung hilft dabei, den langsamen Verlauf beim Abschalten des Stroms noch deutlicher darzustellen. In der Abb. 3.11 wird das angepasste Modell 4 Klasse 1 dargestellt.

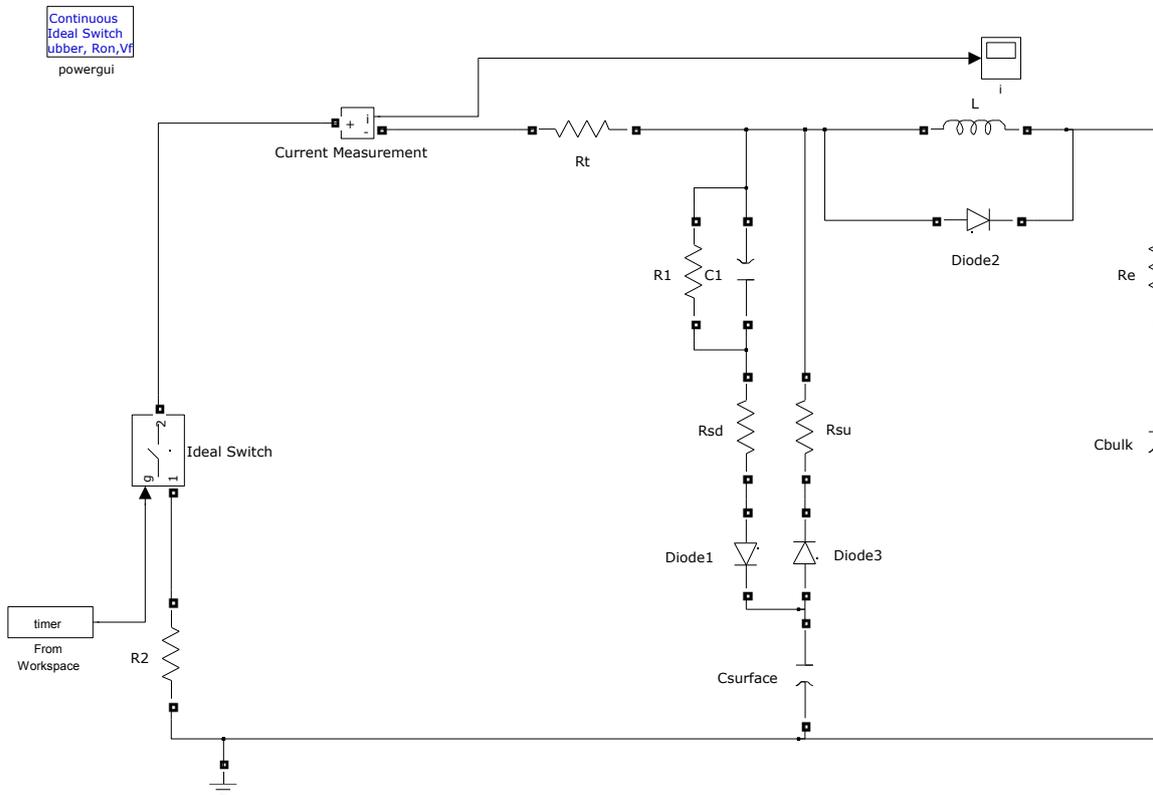


Abbildung 3.11.: Das angepasste Modell 4 Klasse 1

Die Wirkung der einzelnen Elemente lässt sich durch die Abb. 3.12 zeigen. Die Farbe bezeichnet, welches Element für welchen Teil der Messkurve verantwortlich ist.

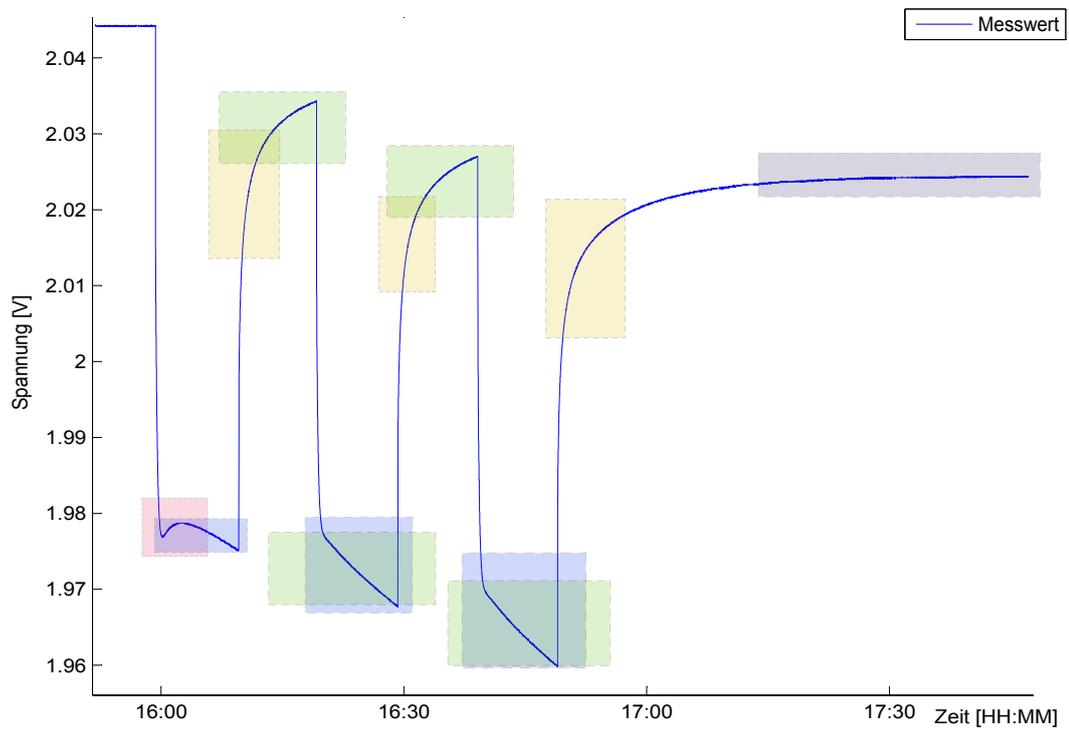
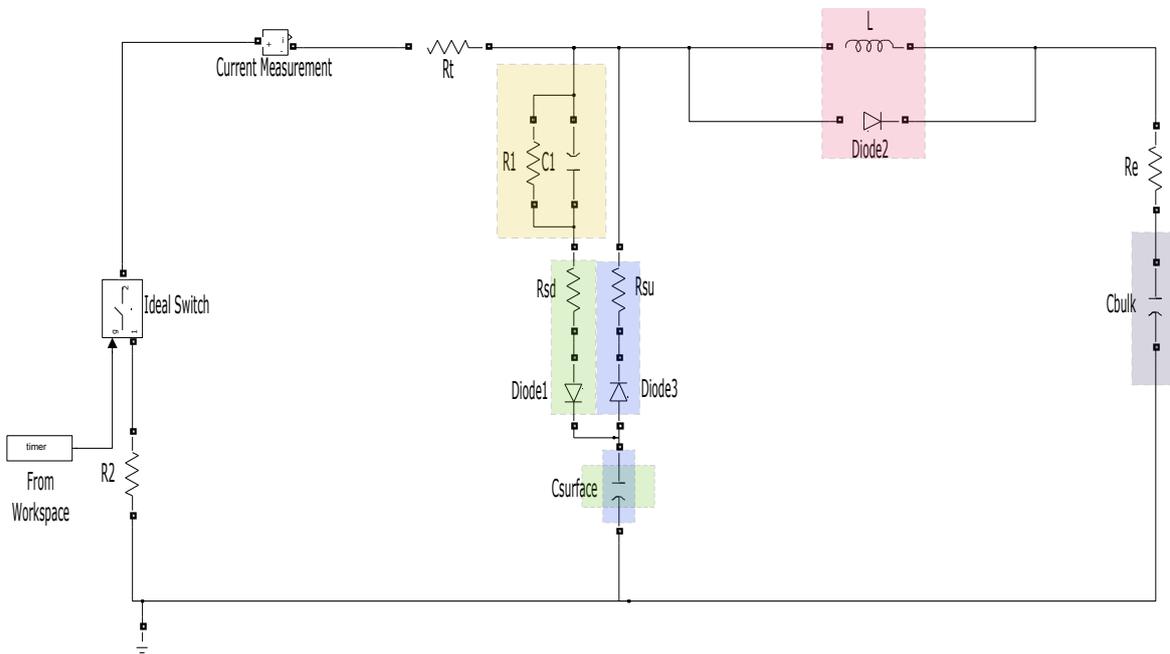


Abbildung 3.12.: Wirkung der einzelnen Elemente

3.6.1. Implementation des angepassten Modells 4 Klasse 1

Dynamisches Verhalten bei der Entladung

In der Abb. 3.13 wird die simulierte Ausgabe des Modells mit bzw. ohne die zusätzlichen RC-Parallelschaltung und dem Messwert verglichen. Der Verlauf des Modells mit zusätzlichen RC-Glied beim Ein- bzw. Ausschalten des Stroms ist wesentlich besser rekonstruiert.

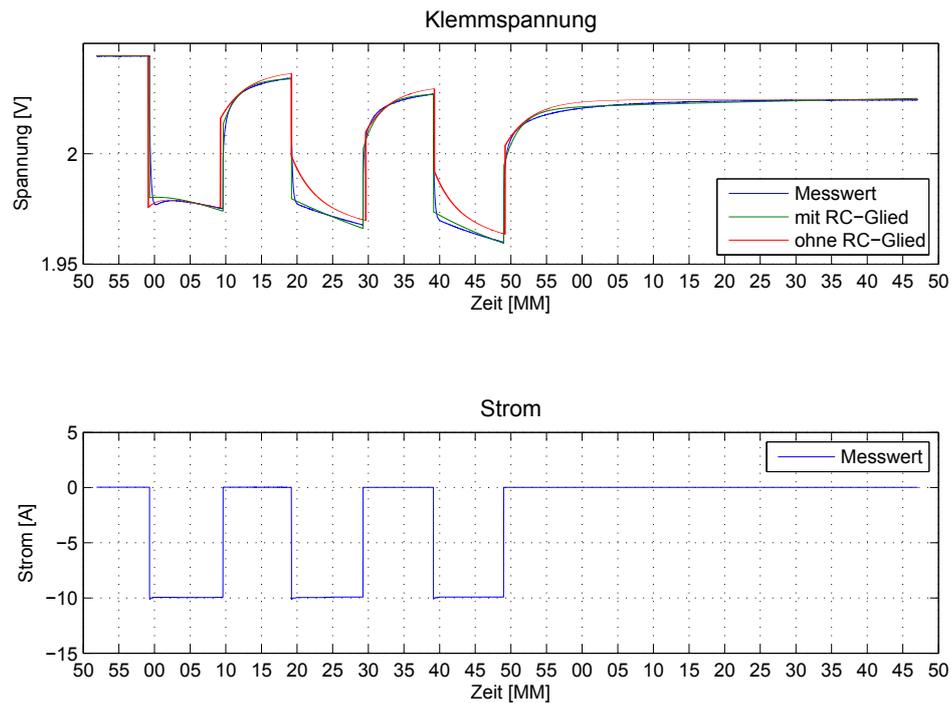


Abbildung 3.13.: Vergleich des dynamischen Verhaltens des Modells mit dem Messwert

Dynamisches Verhalten bei langer Entladung

In der Abb. 3.14 wird die simulierte Kurve des angepassten Modells 4 Klasse 1 beim langen Entladezyklen mit dem Messwert verglichen. Bei der langen Entladung kann die Wirkung der Spule schon vernachlässigt werden, weil im Vergleich zu der gesamten Dauer der Entladung die Dauer des Spannungssacks kurz ist. Die Abweichung liegt innerhalb von drei Millivolt. Die Messkurve kann ausreichend genau von dem angepassten Modell 4 Klasse 1 dargestellt werden.

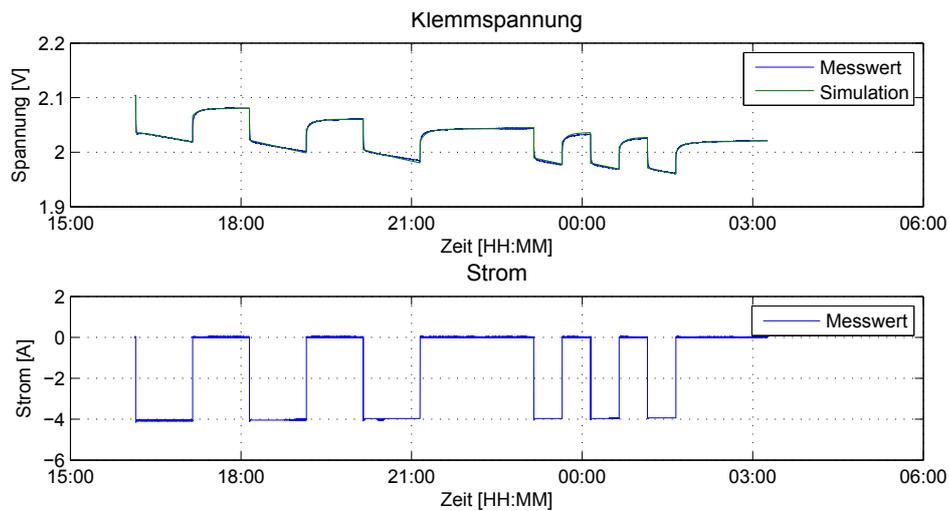


Abbildung 3.14.: Vergleich des langen dynamischen Verhaltens des Modells mit dem Messwert Zyklus1: 1 Stunde; Zyklus 2: 30 Minuten

Kontinuierliche Entladung

In der Abb. 3.15 wird das Verhalten bei der kontinuierlichen Entladung mit der Simulation des angepassten Modells 4 Klasse 1 verglichen. Das angepasste Modell 4 Klasse 1 eignet sich nicht für die Beschreibung des Verlaufs der kontinuierlichen Entladung. Das Modell mit einem konstanten Kondensator kann den nichtlinearen Verlauf der Klemmspannung nicht nachbilden.

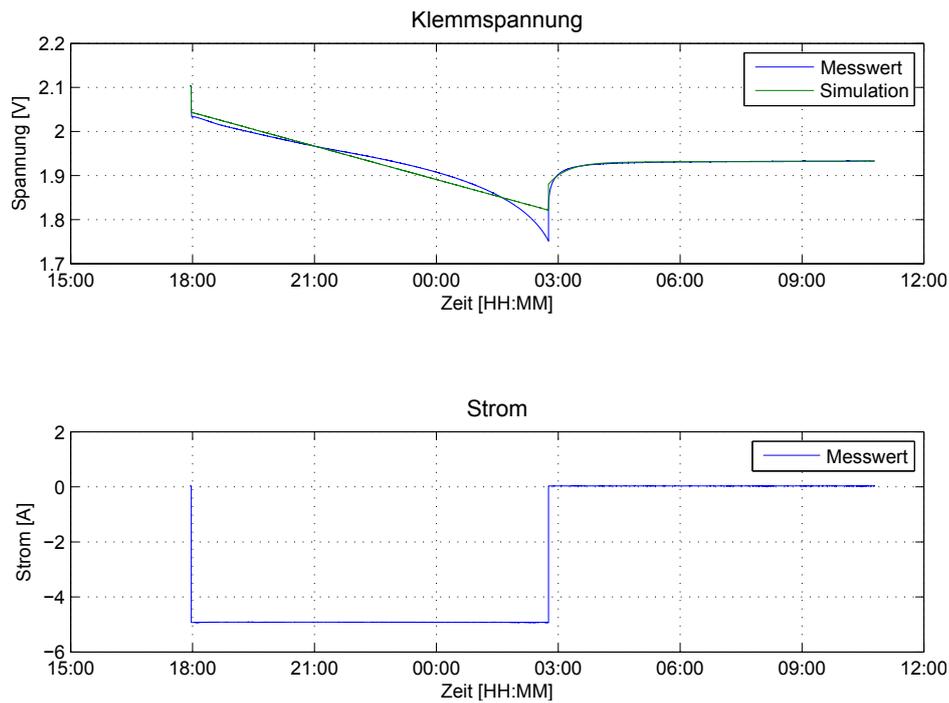


Abbildung 3.15.: Vergleich des Verlaufs der kontinuierlichen Entladung des Modells mit dem Messwert

Aufladung

Im Abschnitt 3.5.1 wurde das Ladeverfahren schon erläutert. Die Last ist beim Aufladen mit dem I-U Ladeverfahren nicht konstant. Eine verstellbare Last ist nicht in der Bibliothek von Simulink vorhanden und kann nur durch eine verstellbare Spannungsquelle erzeugt werden. Es tritt häufig eine Fehlermeldung auf, wenn in der Schaltung in Simulink gleichzeitig mehrere Energiespeicher und mehrere Spannungs- bzw. Stromquellen entstehen. Solche Elemente stören sich gegenseitig. Bei längerem Aufladen bis 2, 3 V kann die Wirkung der Spule vernachlässigt werden, weil im Vergleich mit der gesamten Dauer des Aufladevorgangs das Zeitintervall beim Spannungssack kurz ist. Beim Aufladen wird das Modell der Abb. 3.16 implementiert.

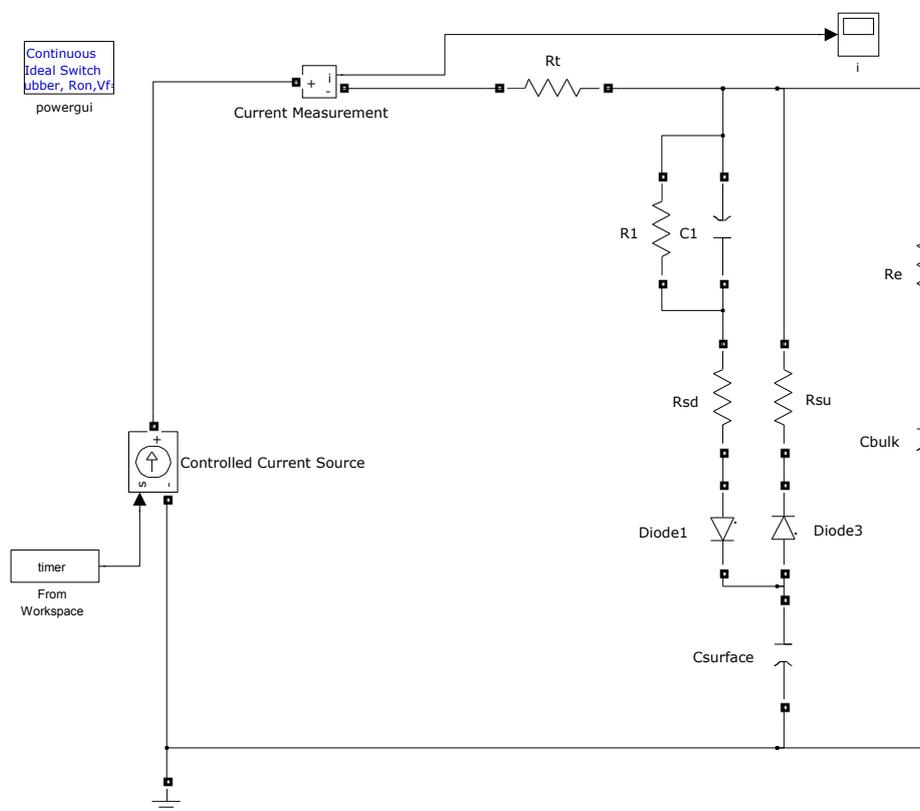


Abbildung 3.16.: Modell beim Aufladen

Das simulierte Ergebnis wird in der Abb. 3.17 dargestellt.

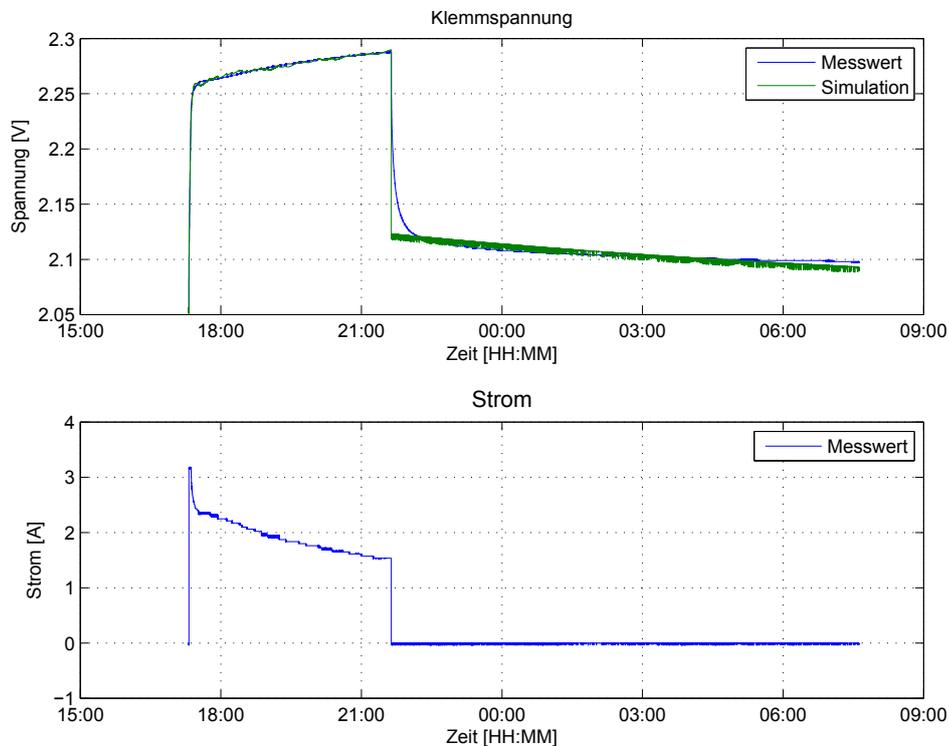


Abbildung 3.17.: Vergleich der Simulation des Modells beim Aufladen mit dem Messwert

Der wesentliche Verlauf der Messwerte lässt sich durch dieses Modell gut darstellen. Nur beim Ruhezustand nach der Aufladung kommt es zu Abweichungen. Die Spannung verändert sich beim Ausschalten des Stroms zu stark. Diese Unterschiede werden durch das Hinzufügen der zusätzlichen RC-Parallelschaltung vor dem R_{sd} vermieden. Das erweiterte Modell wird in der Abb. 3.18 dargestellt und die Simulation lässt sich in der Abb. 3.19 ersehen.

Mittels dieses Modells lässt sich der Verlauf der Ruhespannung gut nachbilden. Aber es tritt noch ein Fehler beim Einschalten des Stroms auf. Es wird versucht, eine RC-Parallelschaltung vor dem Widerstand R_{su} einzusetzen, wie in der Abb. 3.20

Die Simulation in der Abb. 3.21 ist deutlich schlechter als das vorherige Modell. Vermutlich besteht die Ursache darin, dass eine Rückkopplung zwischen den beiden zusätzlichen Kondensatoren gebildet wird.

Vergleichsweise wird das Modell in der Abb. 3.19 zur Implementation bei der Aufladung vorgeschlagen, weil die Ruhespannung eine wichtige Rolle spielt und die Abweichung beim Einschalten des Stroms durch das Kalman-Filter korrigiert werden kann.

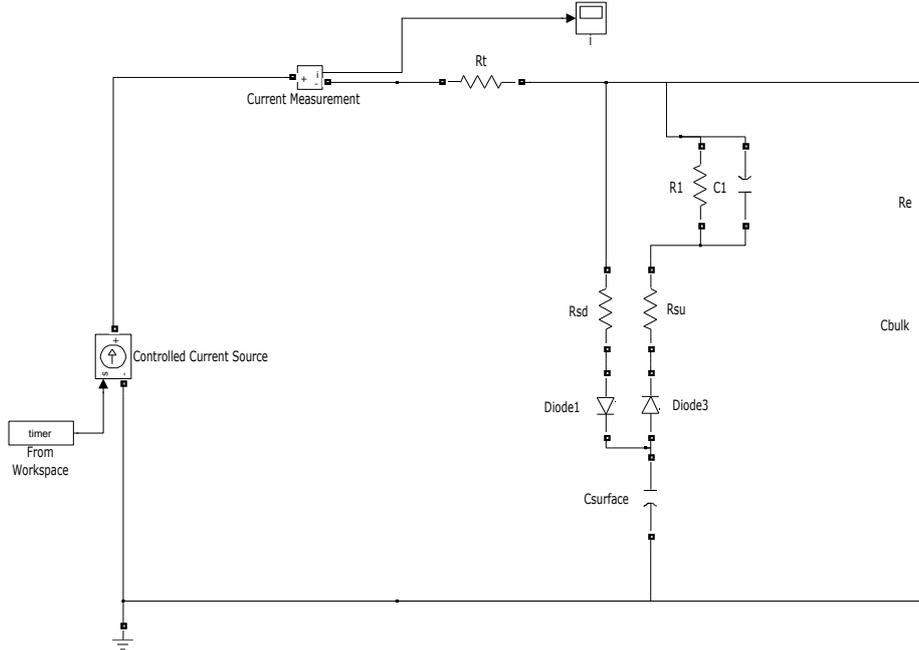


Abbildung 3.18.: erweitertes Modell beim Aufladen

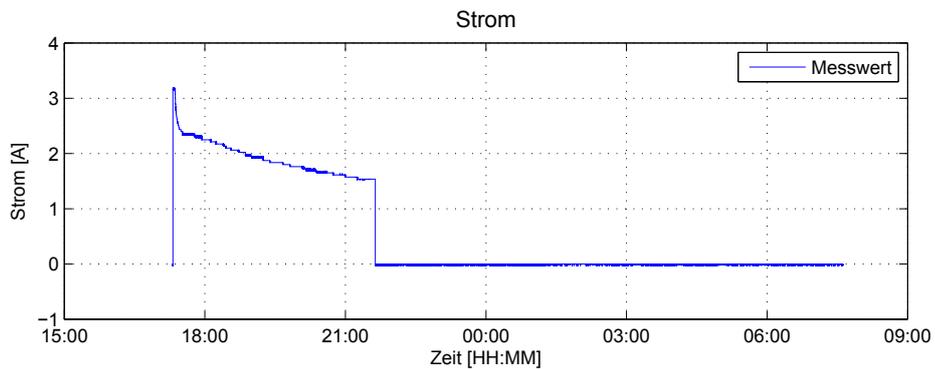
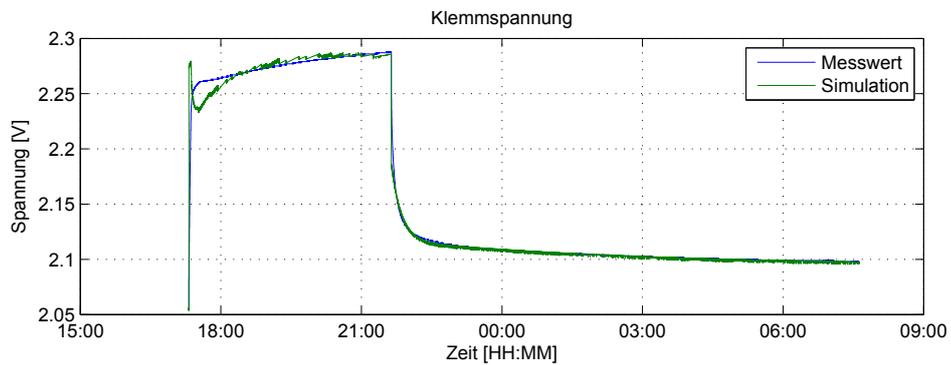


Abbildung 3.19.: Vergleich der Ausgabe des erweiterten Modells beim Aufladen mit dem Messwert

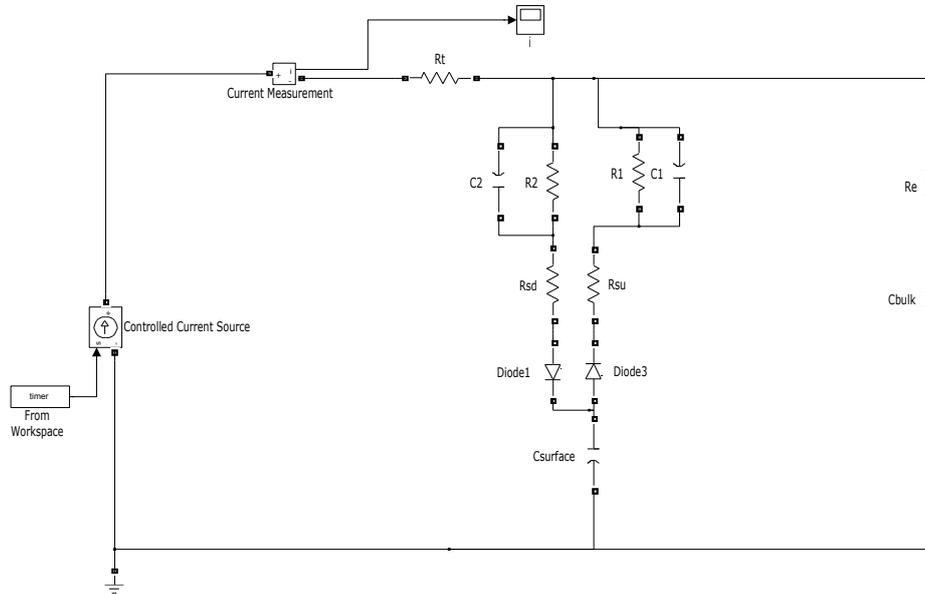


Abbildung 3.20.: Modell mit zwei RC-Parallelschaltungen beim Aufladen

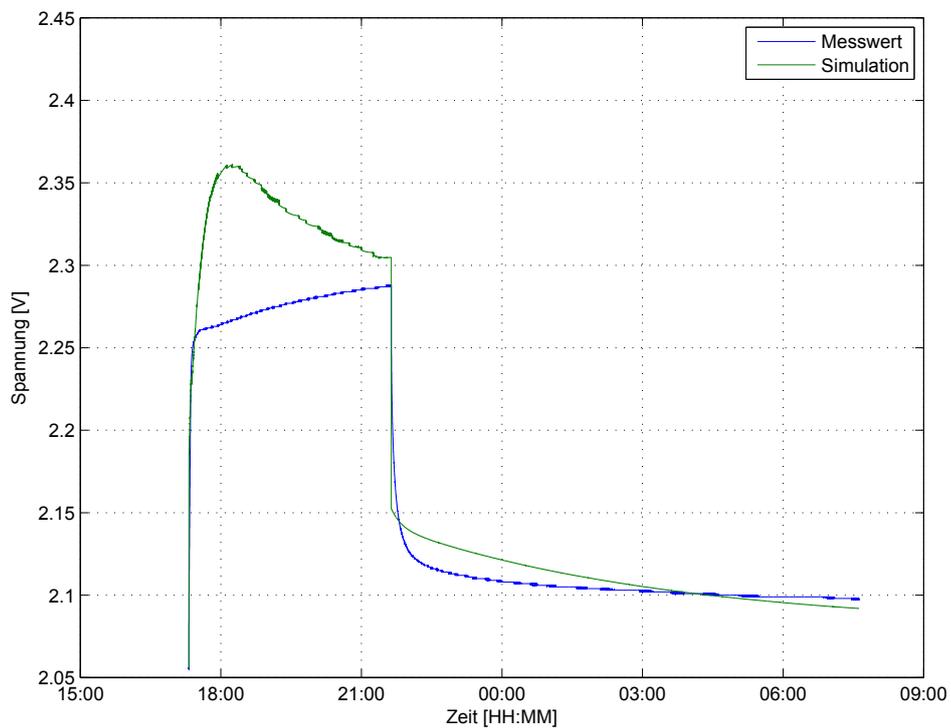


Abbildung 3.21.: Vergleich des Verlaufs des Modells mit dem Messwert

3.6.2. Auswertung und auftretende Probleme

Das angepasste Modell 4 Klasse 1 kann das dynamische Verhalten der Bleibatterie ausreichend gut darstellen, nicht nur die Ruhespannung nach der Belastung sondern auch die Spannungsabsenkung beim ersten Entladevorgang und den Spannungsverlauf beim Wechsel des Stroms. Die durchschnittliche Abweichung zwischen dem Messwert und der Simulation beträgt etwa 1 mV. Die größte Abweichung tritt ebenfalls bei Stromänderungen auf und beträgt etwa 5 mV.

Trotzdem eignet sich das angepasste Modell 4 Klasse 1 nicht zur Darstellung der kontinuierlichen Entladung. Ein konstanter großer Kondensator kann die nicht Entladungskurve vollständig nachbilden. Wird das angepasste Modell 4 Klasse 1 durch das Hinzufügen von Elementen noch erweitert, kann dieses Verhalten vielleicht optimiert werden. Der Rechenaufwand steigt dafür. Zurzeit enthält das angepasste Modell 4 Klasse 1 schon acht Elemente bzw. acht Parameter. Für die Starterbatterie ist das dynamische Verhalten der Batterie wichtiger als für andere Batterien. Normalerweise sollte die Starterbatterie nach dem Starten des Motors sofort vom Generator wieder aufgeladen werden. Selten wird die Starterbatterie ausgeschöpft oder tiefentladen.

Eine Frage zum angepassten Modell 4 Klasse 1 bleibt offen nämlich, wie die allgemeinen Parameter zur Beschreibung des Verhaltens der Batterie festgestellt werden können. Das angepasste Modell 4 Klasse 1 mit diesen allgemeinen Parametern kann unterschiedliche Entladung darstellen.

Es ist bekannt, dass die Temperatur der Umgebung und der Ladezustand das Lade- und Entladeverhalten der Batterie beeinflussen können. Die Ursache besteht darin, dass die elektrochemischen Reaktionen in den Elektroden/ Elektrolyten der Batterie von der Temperatur abhängig sind. Bei fallender Temperatur nimmt die Geschwindigkeit der Elektrodenreaktion ab, dabei verschlechtert sich der Stofftransport innerhalb der Elektrolyten. Bei steigender Temperatur liegt das Verhältnis umgekehrt. Dadurch stellt sich die Frage, wie der Einfluss der Temperatur und des Ladezustandes auf das Entlade- und Ladeverhalten der Batterie in den angewandeten Parametern des Modells integriert werden können.

3.6.3. Betrachtung des Temperatureinflusses auf das Batterieverhalten

Die Starterbatterie kann in einem weiten Temperaturbereich betrieben werden. Bei tiefen Temperaturen wird die Leistungsabgabe der Batterie begrenzt, weil die Viskosität der Säure ansteigt und die Diffusion behindert. Wird eine Batterie bei einer sehr hohen Temperatur betrieben, wird die Lebenserwartung der Batterie drastisch verkürzt.

Im folgenden Abschnitt wird der Entladungs- bzw. Aufladungsverlauf bei drei Temperaturen (-2°C , 21°C , 50°C) dargestellt. Damit wird der Temperatureinfluss auf das Verhalten der Batterie analysiert.

Bei -2°C wird die Batterie in einen Kühlschrank gelegt und bei 50°C steht die Batterie in einem Wärmeschrank. Die Temperatur wird durch einen eigenen Reglungsalgorithmus gesteuert, um eine quasikonstante Temperatur in dem Raum zu gewährleisten.

Kontinuierliche Entladung

Die Abb. 3.22 ist das Ergebnis der Messung der kontinuierlichen Entladung. Eine Zelle der Starterbatterie wird bis $1,75\text{ V}$ bei drei Temperaturen entladen. Der Entladestrom beträgt jeweils etwa fünf Ampere.

Mit ansteigender Temperatur verlängert sich die Dauer der Entladung. Je höher die Temperatur ist, desto mehr Kapazität kann aus dieser Zelle entnommen werden. Die entnehmbare Kapazität bei 50°C ist doppelt so hoch wie die bei -2°C , weil bei hoher Temperatur der Diffusionsprozess bzw. die Reaktionsgeschwindigkeit beschleunigt wird (drittes Bild in der Abb. 3.22). Der Spannungseinbruch in der Messkurve bei -2°C ist vermutlich auf die zur Ladungsbewegung in den Elektrolyten notwendigen Diffusionsprozesse zurückzuführen. Die Wärmeentwicklung reduziert alle Innenwiderstände.

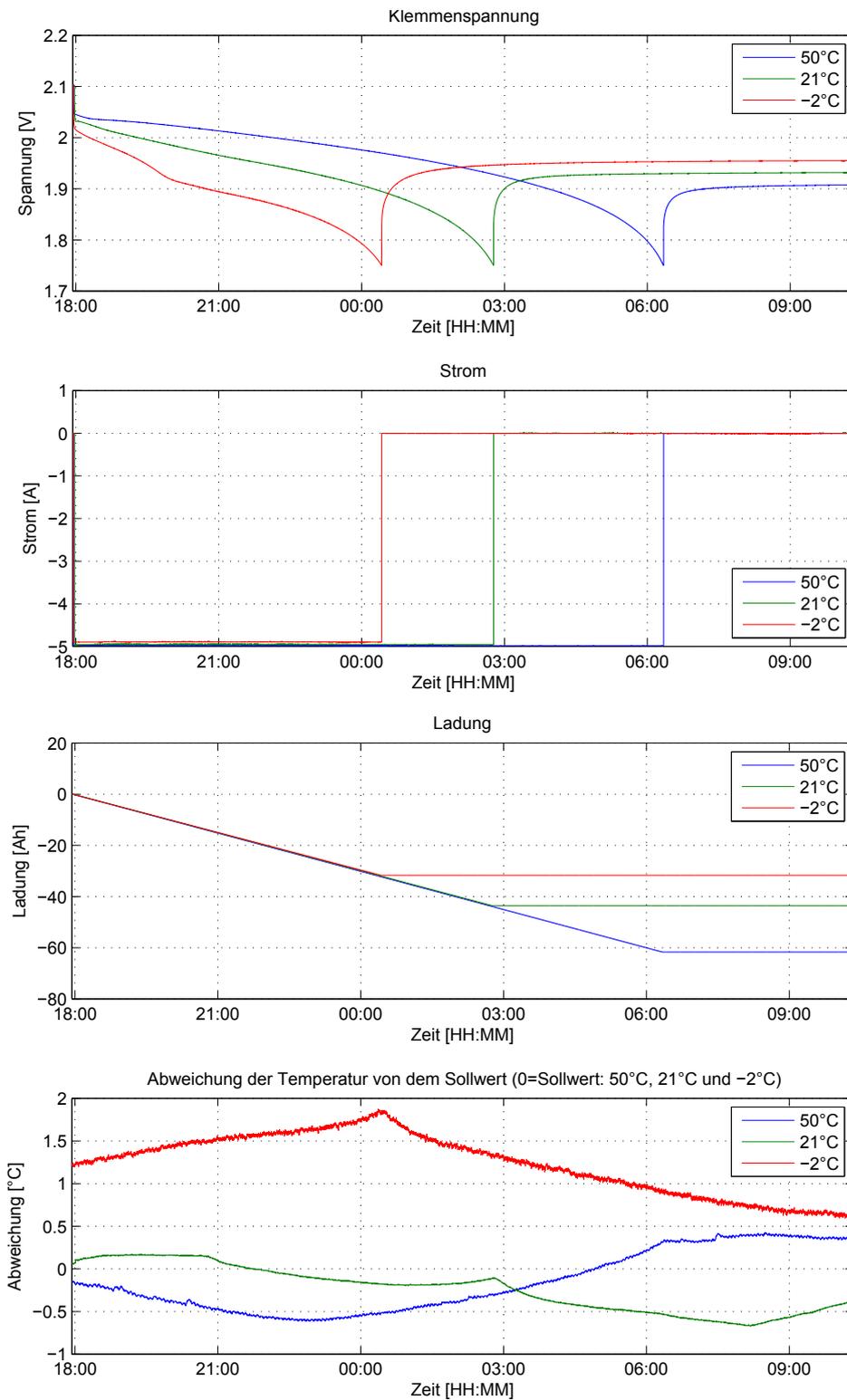


Abbildung 3.22.: kontinuierliche Entladung bei drei Temperaturen

Partielle Entladung

Ziel der Messung ist die Ruhespannung beim unterschiedlichen Ladezustand bzw. Entladegrad zu bestimmen. Später wird aus dem Spannungsabfall beim Einschalten der Last der Innenwiderstand bestimmt. Die Ruhespannung gilt häufig als der Anhaltspunkt für den Ladezustand der Batterie. Anhand der näherungsweise linearen Beziehung zwischen dem Ladezustand und der Ruhespannung kann der Ladezustand berechnet werden. Für diese Arbeit wird von Ruhespannung gesprochen, wenn die letzte Lade- bzw. Entladephase mehr als 6 Stunden zurückliegt. Das Ergebnis der Messung wird in der Abb. 3.23 angezeigt.

Mit ansteigender Temperatur kann mehr Kapazität aus der Zelle entnommen. Der Endwert vom Messwert bei -2°C und 21°C wird über die Zeitachse des Messwertes bei 50°C ausgedehnt. Prinzipiell sollte die Endspannung des Messwertes bei -2°C größer sein als des bei 21°C . Der Spannungseinbruch an der letzten Entladung wird durch die falsche Einstellung in dem Programm verursacht. In dem Programm wurde der maximale Zyklus der Belastung als zehn eingestellt. Aus der Abb. 3.23 ist schon festzustellen, dass bei 50°C es bestimmt mehr als zehn Zyklen braucht, bis die Zelle vollständig entladen ist. Für die Abschätzung des Temperatureinflusses ist dieser Fehler aber nicht wichtig. Je höher die Temperatur ist, desto größer ist der Spannungsabfall. Es bedeutet, dass mit der ansteigenden Temperatur der Innenwiderstand sich vergrößert.

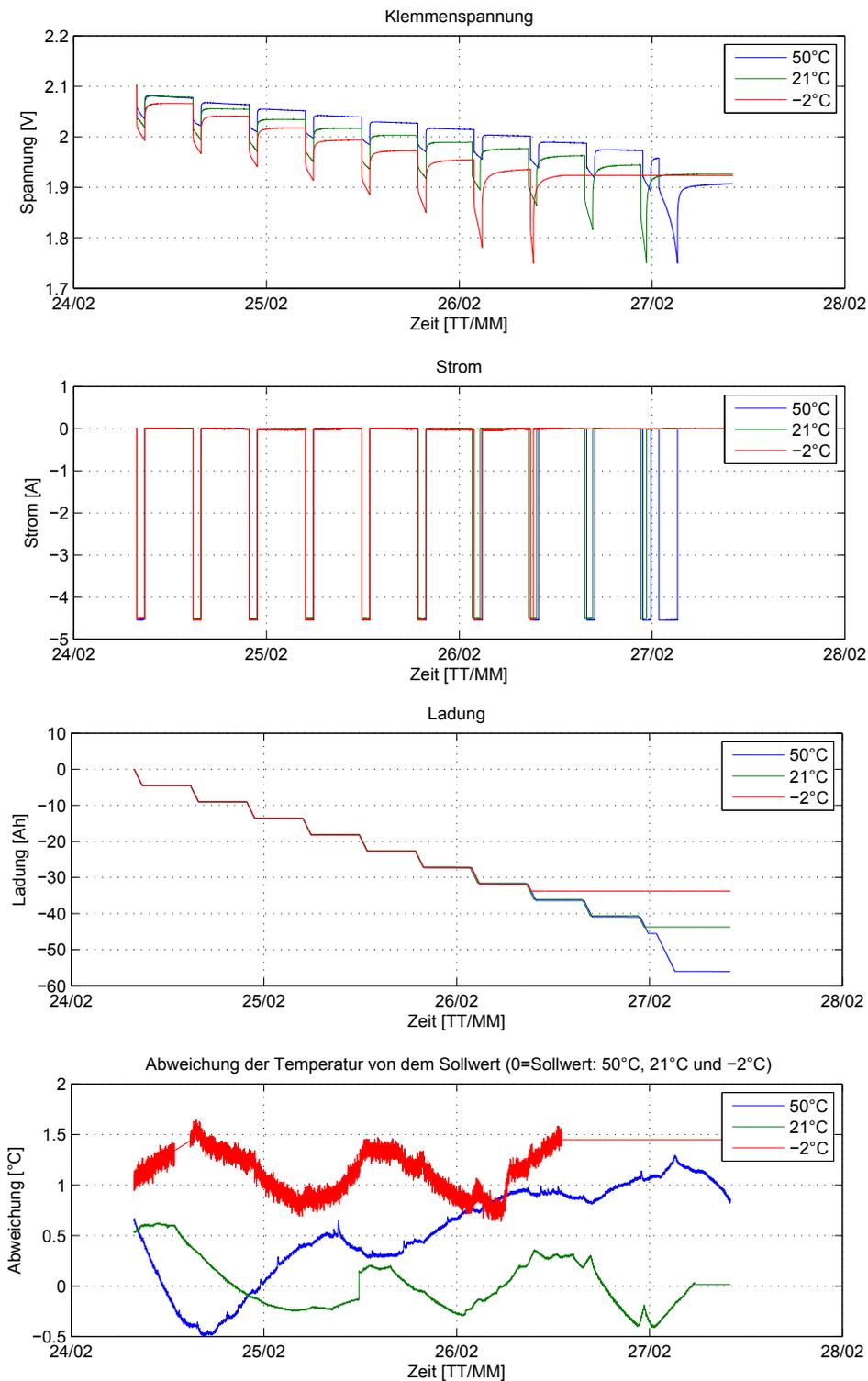


Abbildung 3.23.: partielle Entladung bei drei Temperaturen

Dynamisches Verhalten bei der Entladung

In der Abb. 3.24 ist das dynamische Verhalten bei drei Temperaturen veranschaulicht.

Es ist zu beobachten, dass der Verlauf der Ruhespannung bei 50°C langsam sinkt. Die Selbstentladung führt zu diesem Absinken. Durch Verunreinigungen hervorgerufene Nebenreaktion verursacht hauptsächlich die Selbstentladung. Die Selbstentladung steigt bei höheren Temperaturen an. Typischerweise wird die Geschwindigkeit der Selbstentladung mit jedem 10°C verdoppelt [24]. Außerdem steigt die Geschwindigkeit der Selbstentladung mit der Alterung und dem Gebrauch der Batterie. In dem vierten Bild von der Abb. 3.24 fehlt die Messkurve der Temperatur 21°C , weil damals noch kein Temperatursensor (NTC) zur Verfügung stand.

Mit ansteigender Temperatur wird schneller eine Ruhespannung erreicht. Die höhere Temperatur beschleunigt die Geschwindigkeit des Stofftransports beim chemischen Prozess.

Aufladung

Kein vergleichbarer Aufladeprozess bei den drei Temperatur wurde durchgeführt. Deswegen wird hier nicht der Temperatureinfluss auf das Ladeverhalten diskutiert.

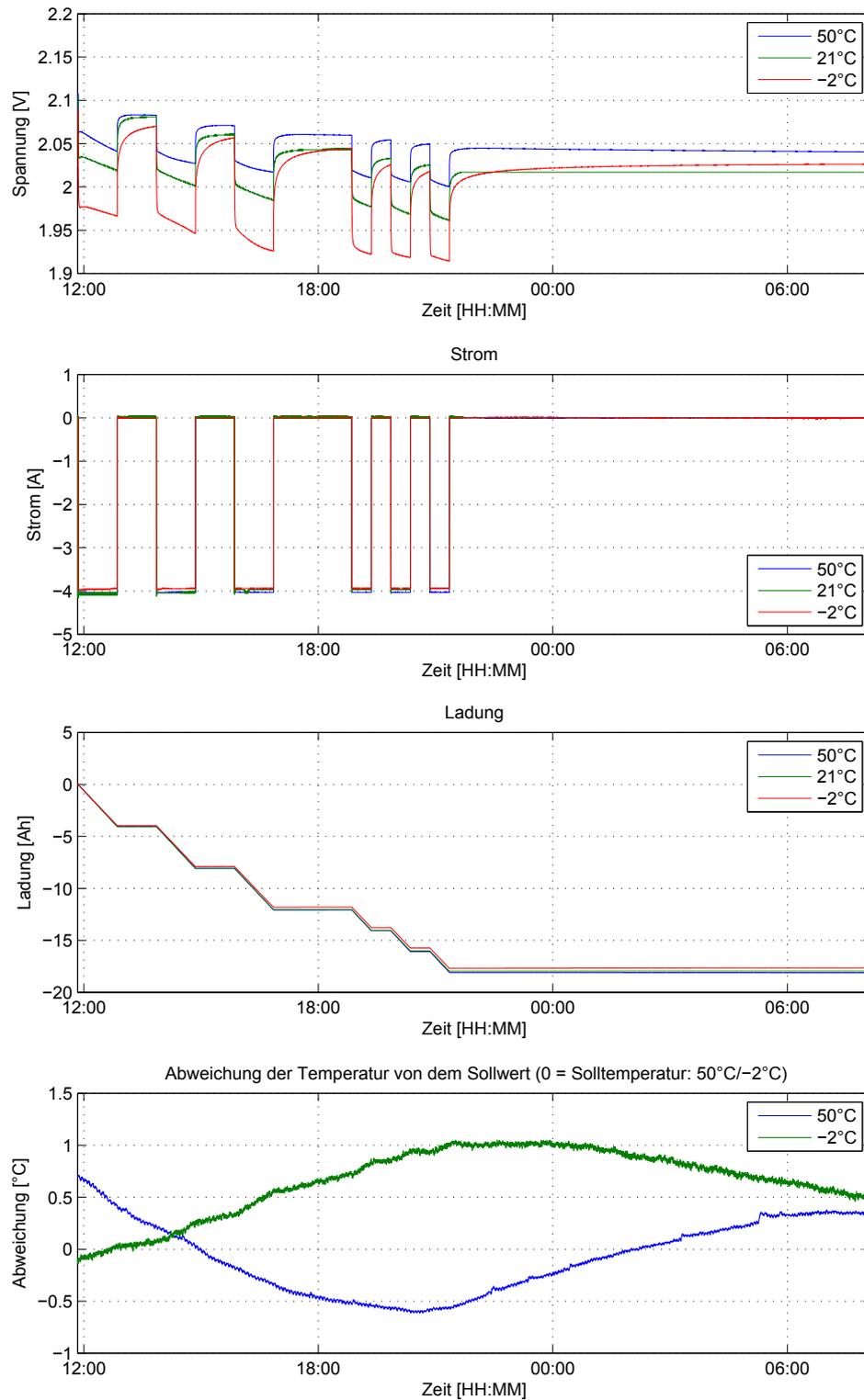


Abbildung 3.24.: dynamisches Verhalten beim Entladen bei drei Temperaturen

3.7. Das angepasste Modell 4 Klasse 2

Im vorherigen Abschnitt wird der Einfluss der Temperatur auf das Verhalten der Bleibatterie analysiert. Nun wird versucht, den Einfluss der Temperatur in einem Modell durch einige Formeln bei den angepassten Parametern einzubringen. Dieses Modell sollte von der Temperatur bzw. dem Ladezustand abhängig sein.

3.7.1. Festlegung der Ruhespannung der Starterbatterie bei drei Temperaturen

In diesem Abschnitt wird der Verlauf der Ruhespannung bei der Entladung der Batterie bei drei Temperaturen untersucht. Die Ruhespannung spielt eine wichtige Rolle zur Ermittlung des Ladezustands bzw. der noch verfügbaren Kapazität in der Batterie. Aus dem Verlauf des Messwertes wird je ein Messpunkt der Ruhespannung von jeder partiellen Entladung aufgenommen (markierte grüne Punkte in Abb. 3.25). Die verkettete Kurve der einzelnen Ruhespannungen wird in der Abb. 3.26 angezeigt. Es lässt sich aus diesen Kurven feststellen, dass der Verlauf der Ruhespannung mit der Zunahme des Entladegrads nicht rein linear verläuft. Damit wird es erklärt, warum das Modell mit einem großen Kondensator als den Hauptenergiespeicher für die Darstellung der kontinuierlichen Entladung nicht geeignet ist. In der Literatur ([7], [41]) wird häufig der Hauptenergiespeicher der Batterie durch einen Kondensator nachgebildet, weil das Modell verständlich darzustellen ist und sich ein solches Modell für die kurzzeitige Entladung eignet. Dabei wird ein Kompromiss zwischen Komplexität des Modells und der Genauigkeit geschlossen.

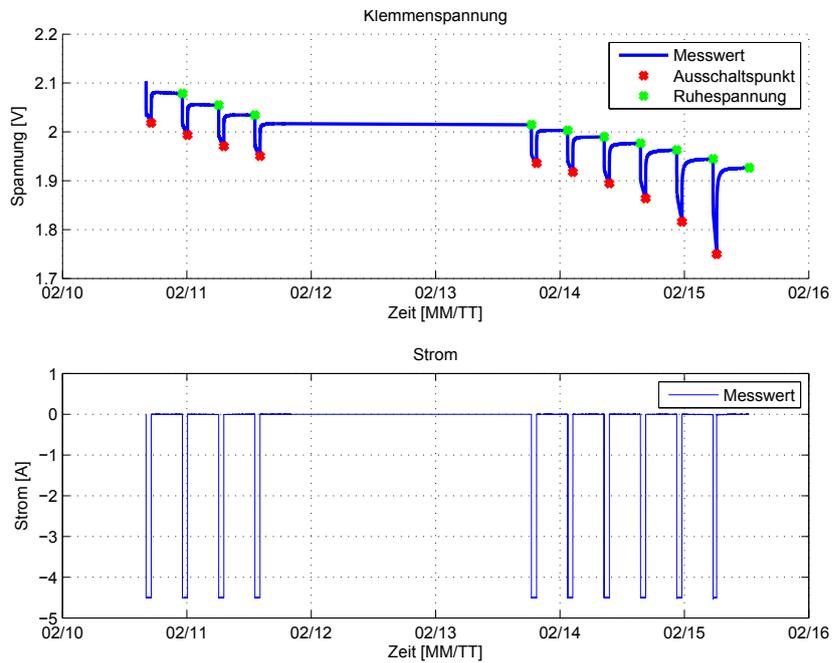


Abbildung 3.25.: Ruhespannung in der partiellen Entladung bei 20°C

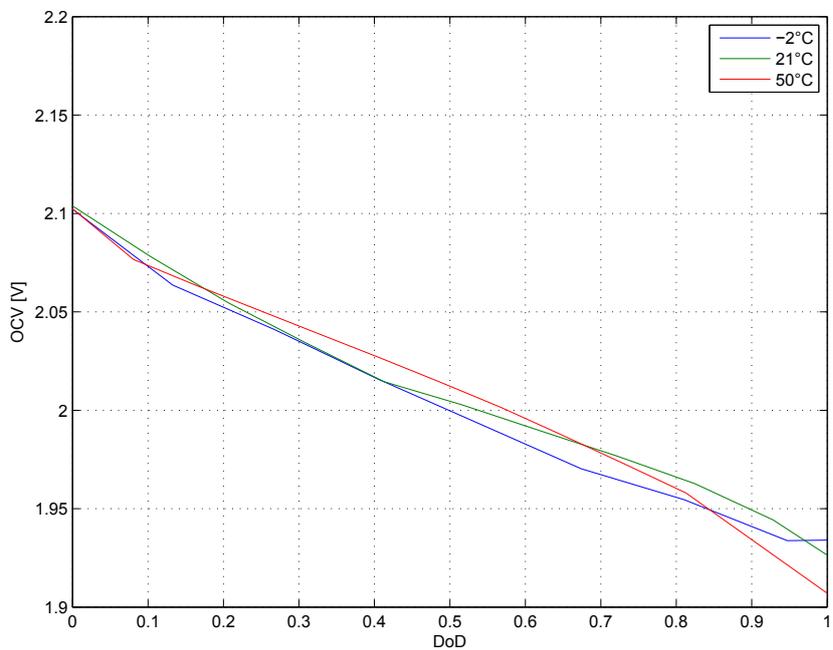


Abbildung 3.26.: Darstellung der Ruhespannung bei drei Temperaturen

Die Abb. 3.27 zeigt einen vollständigen Entladungsverlauf. Dabei werden zwei Kapazitäten für den Kondensator berechnet. Bei der grünen Kurve mit der kleinen Kapazität passt der vordere Teil der Klemmenspannung, bei der roten Kurve mit großer Kapazität passt hingegen der hintere Teil der Klemmenspannung besser. Der Verlauf der vollständigen Entladung kann nicht durch einen Kondensator ermittelt werden. Im Verlauf der vollständigen Entladung verlangsamt sich die Geschwindigkeit des Kapazitätsverlustes der Bleibatterie. Nun wird versucht, statt eines Kondensators andere Elemente für den Hauptenergiespeicher in dem Modell einzusetzen, mit denen der Verlauf der Ruhespannung gut dargestellt werden kann.

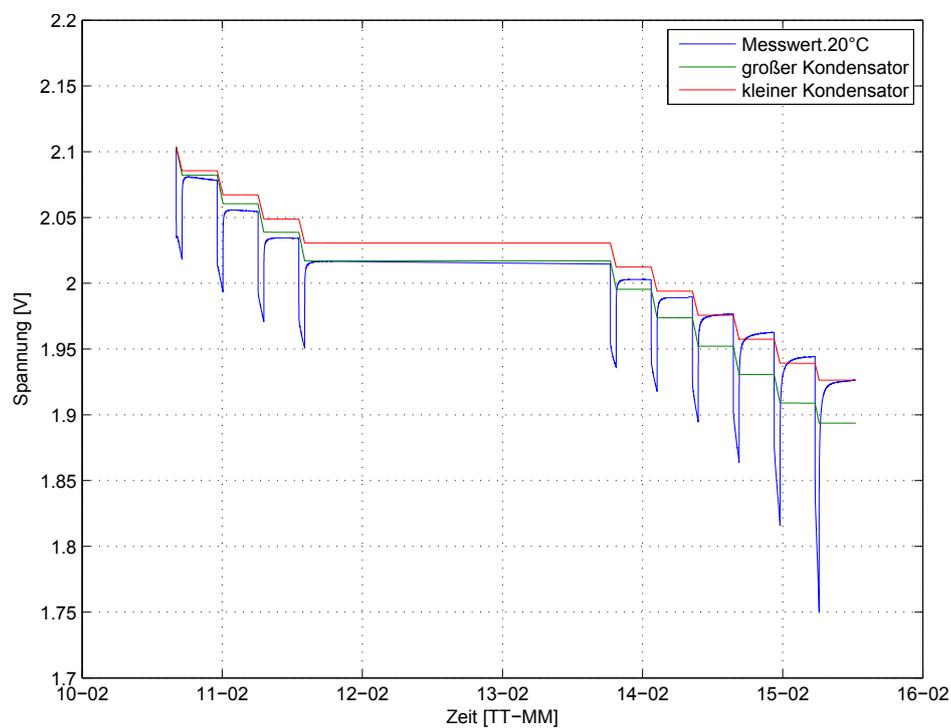


Abbildung 3.27.: Spannungsverlauf der Kondensatoren und Klemmenspannung bei 20°C

Festlegung des Temperatureinflusses in einer Formel

Aus der Abb. 3.26 geht hervor, dass der Verlauf der Ruhespannung über den Entladegrad bei einer Temperatur nichtlinear ist. Es ist schwierig, diesen in der Abb. 3.26 angezeigten Verlauf direkt anzunähern. Mittelbar wird die Abweichung zwischen dem Verlauf der gemessenen Ruhespannung und dem mit einem Kondensator berechneten Verlauf ermittelt. Die relative Abweichung zu der Kondensatorspannung wird als die Zielkurve bezeichnet (Abb. 3.29). Nach der Gl. 3.1 wird die relativ Abweichung berechnet.

Zum Vergleich mit dem Verlauf der gemessenen Ruhespannung kann im Modell als Hauptenergiespeicher entweder ein Kondensator mit kleiner oder einer mit großer Kapazität eingesetzt werden (Abb. 3.28). Die große Kapazität ist die Nennkapazität der Batterie und ist aus den Datenblättern des Herstellers bekannt. Wegen des Alterungsprozesses kann möglicherweise der angegebene Wert etwas größer sein als die aktuelle in der Batterie verbliebene Kapazität. Die kleine Kapazität des Kondensators kann in der Realität durch eine Messung mit der kurzzeitigen Belastung bestimmt werden. In dieser Arbeit wird ein Kondensator mit großer Kapazität in die Formel eingesetzt. Mit Hilfe einer Formel (Gl. 3.2) wird das Verhältnis angenähert.

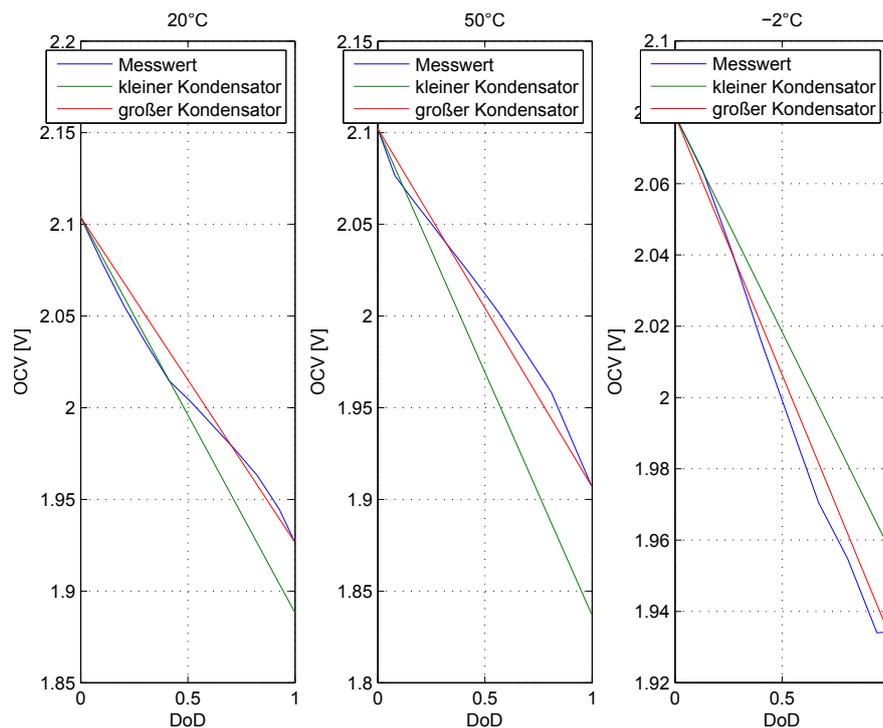


Abbildung 3.28.: Vergleich des Verlaufs der Kondensatoren bei unterschiedlichen Temperaturen

$$\text{relativeAbweichung} = \frac{\text{Ruhespannung} - \text{Kondensatorspannung}}{\text{Kondensatorspannung}} \quad (3.1)$$

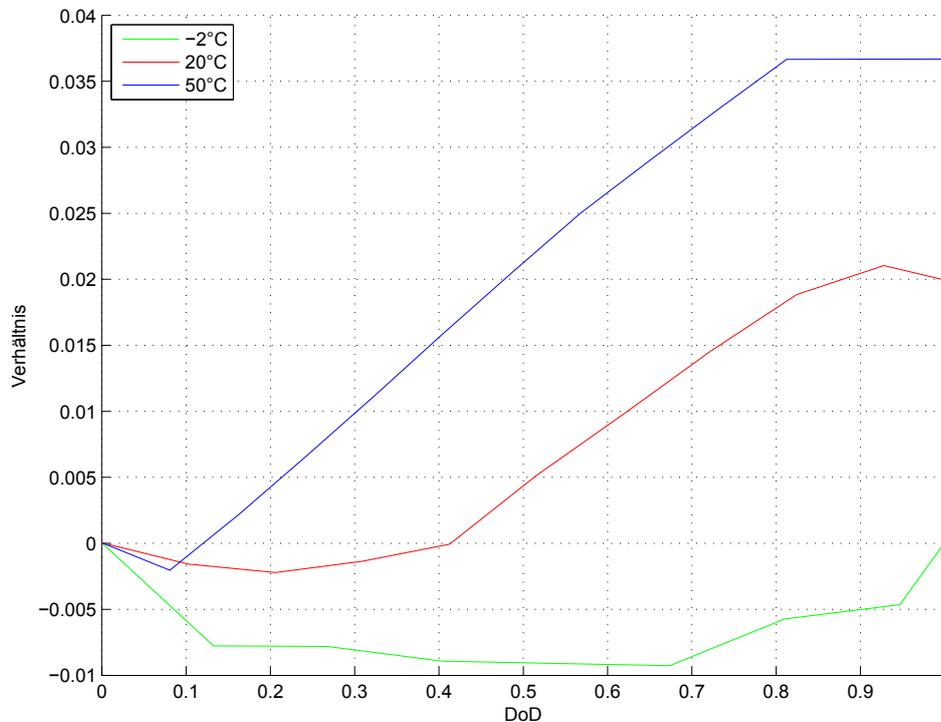


Abbildung 3.29.: Relative Abweichung gemäß Gleichung. [?]

Formel zur Beschreibung dieser relativen Abweichung:

$$a_1 * DoD * (1 - DoD)^{a_2} + a_3 * DoD^{a_4} + (T - 20) * (a_5 * DoD^{a_6} - a_7 * DoD^{a_8}) \quad (3.2)$$

Die ersten vier Parameter a_1, a_2, a_3, a_4 werden durch die Annäherung von der Messkurve bei 20°C erhalten. Die übrigen vier Parameter werden mittels der ausgegebenen Formel durch die Annäherung an die Messkurve bei 50°C berechnet. Der Vergleich der angenäherten Kurve mit der Zielkurve wird in der Abb. 3.30 dargestellt. Die Formel kann die Kurve bei 20°C und 50°C gut annähern. Die Kurve bei -2°C passt nicht (grüne Kurve in der dritten Figur der Abb. 3.30). Die Erweiterung dieser Formel wird aufgegeben, weil die Formel schon komplex genug ist. Mit den angepassten Parametern bei -2°C kann diese Formel ebenso die Messkurve bei -2°C darstellen (rote Kurve in der dritten Figur der Abb. 3.30). Es wird vorgeschlagen, dass die Messungen bei anderen Minus-Temperaturen durchgeführt werden. Damit kann das Verhalten der Batterie bei der Minus-Temperatur verglichen und zusammen-

gefasst werden. Anschließend kann eine spezielle Formel zur Anpassung des Verhaltens der Bleibatterie bei der Minus-Temperatur erstellt werden.

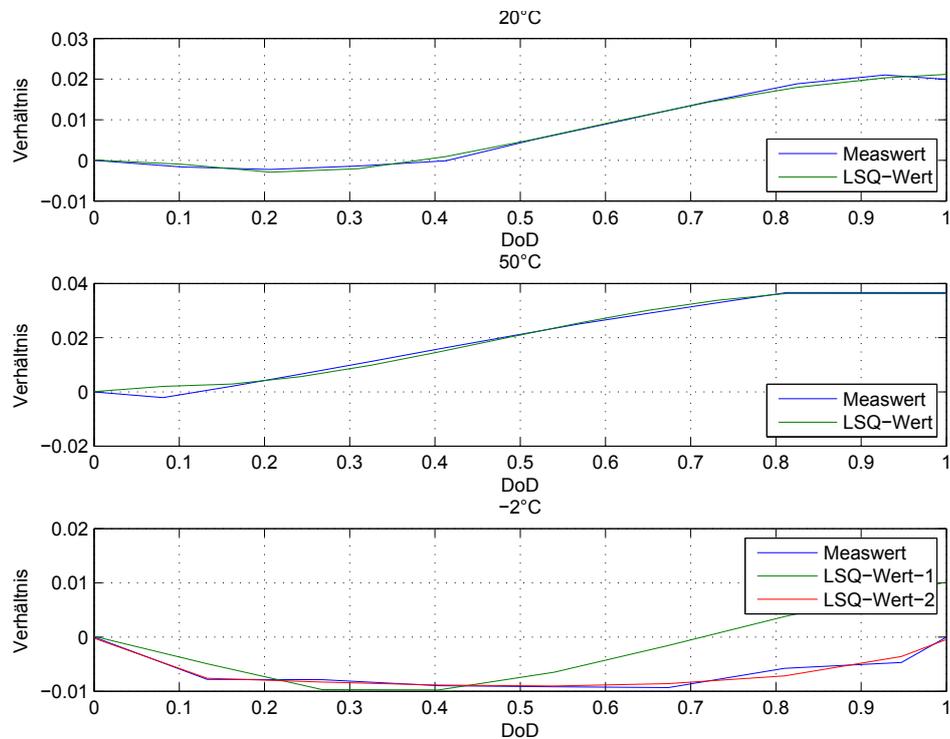


Abbildung 3.30.: Vergleich mit der Ausgabe der Formel

In den Abb. 3.31 bis Abb. 3.33 wird der Verlauf der mit der Formel berechneten Ruhespannung dargestellt. Die berechnete Ruhespannung stimmt mit dem Endwert einzelner partieller Entladungen gut überein. Der Verlauf des starken Spannungsabfalls in der Ruhephase bei 50°C kann durch die Formel nicht nachgebildet werden, weil beim Erstellen der Formel nur ein Punkt der Ruhespannung von der einzelnen partiellen Entladung aufgenommen wurde. Diese Spannungsabfall sollte mit Hilfe eines zusätzlichen Elementes im Modell nachgeholt werden, z.B. ein mit dem Hauptenergiespeicher parallelgeschalteten, temperaturabhängigen Widerstand.

Paramter	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
Wert	-0,117	2,348	0,0212	0,4096	-0,0156	1,7468	0,0161	1,6306

Tabelle 3.2.: Parameter der Formel zur Ruhespannung

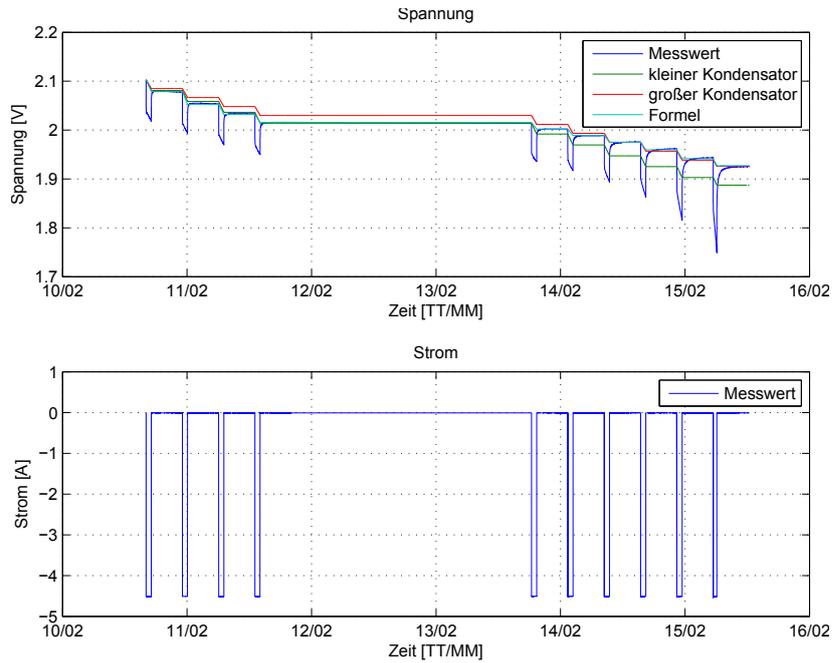


Abbildung 3.31.: Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei 20°C

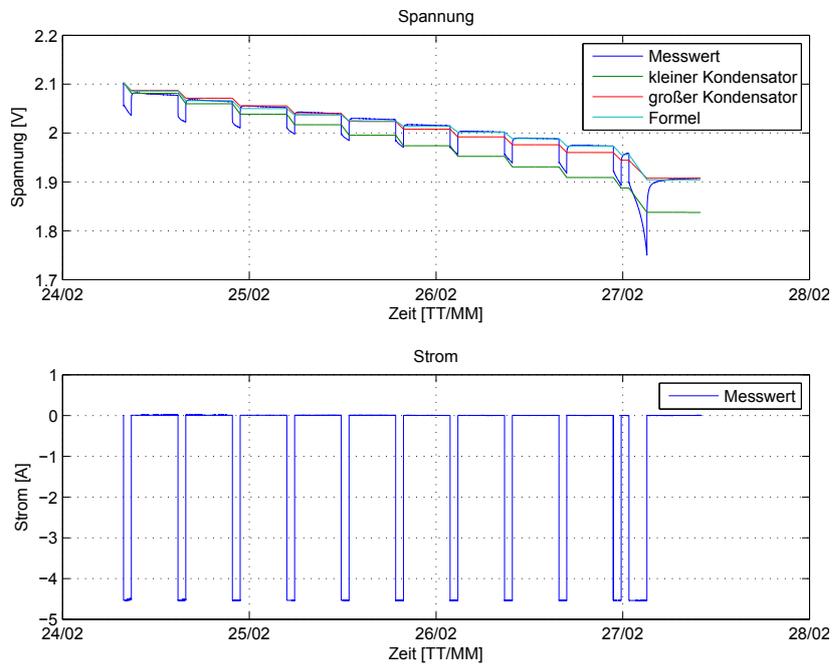


Abbildung 3.32.: Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei 50°C

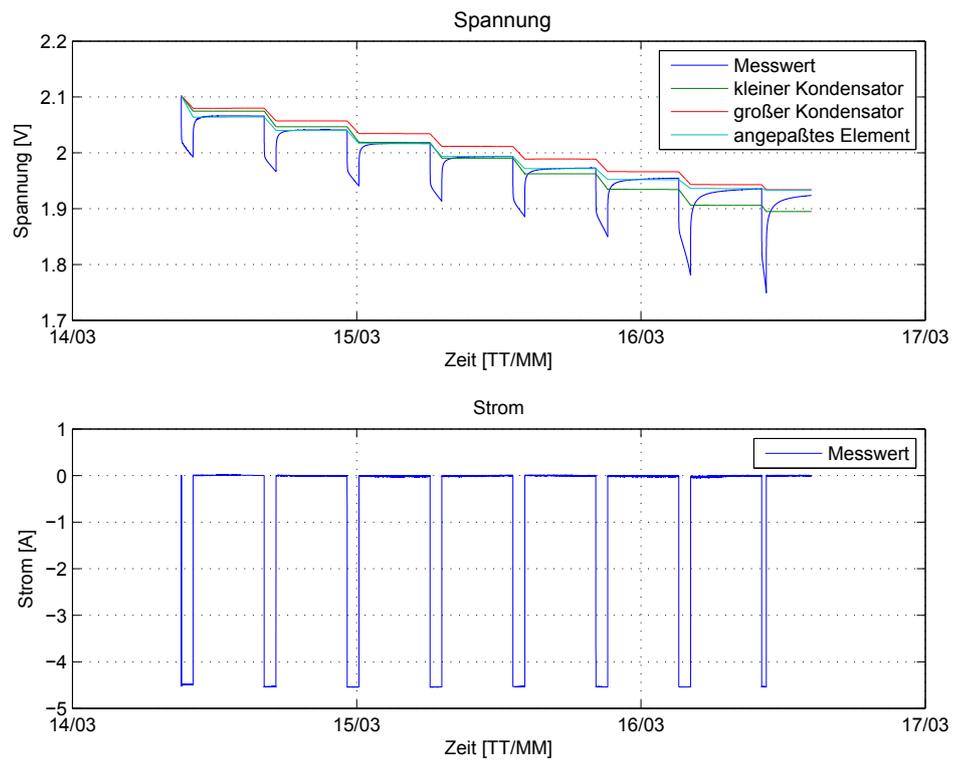


Abbildung 3.33.: Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei -2°C

3.7.2. Festlegung des Innenwiderstandes der Batterie bei drei Temperaturen

Vergleich des Innenwiderstands der Batterie bei drei Temperaturen

In dem Abschnitt 3.6.3 wird die partielle Entladung bei drei Temperaturen erwähnt. Aus dem Spannungsabfall beim Einschalten der Last wird der Innenwiderstand bestimmt und durch die GL. 3.4 berechnet (Abb. 3.25).

$$R_{in} = \frac{V_R - V_S}{\text{aktuellerStrom}} \quad (3.3)$$

V_R : Ruhespannung (grüne markierte Punkte);

V_S : Spannung kurz nach dem Einschalten (rote markierte Punkte);

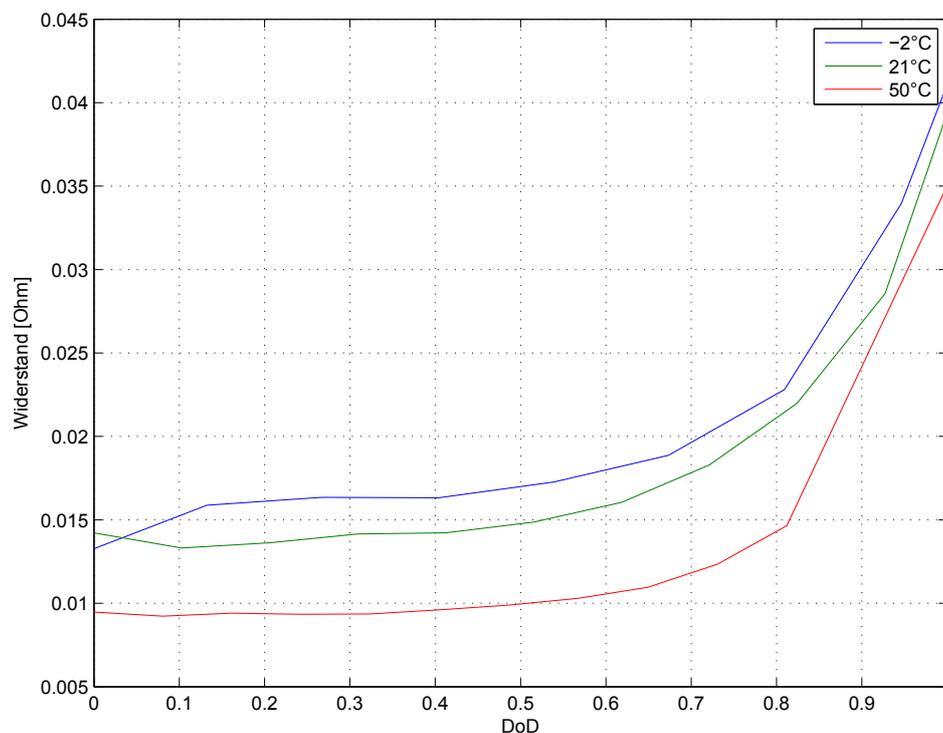


Abbildung 3.34.: Vergleich des Innenwiderstands

In der Abb. 3.34 wird der Vergleich des Innenwiderstandes veranschaulicht. Mit fallender Temperatur vergrößert sich der Innenwiderstand über den Entladegrad, weil die Leitfähigkeit des Elektrolyten bei tiefen Temperaturen stark abnimmt. Es scheint, dass der Wert des Innenwiderstandes fast proportional zur fallenden Temperatur ansteigt.

Erfassen des Temperatureinflusses in einer Formel

Versuche für die Aufstellung einer allgemeinen Gleichung wurden mehrfach unternommen. Jedoch müssen für den Innenwiderstand bei unterschiedlichen Temperaturen immer eine Vielzahl von den Konstanten bzw. Parametern bestimmt werden. Damit kann der Einfluss der Temperatur und des Ladeszustandes auf den Innenwiderstand der Bleibatterie berücksichtigt werden.

Formel des Widerstandes:

$$a_1 + a_2 * DoD^{a_3} + a_4 * (1 - DoD)^{a_5} + a_6 * (T - 20) \quad (3.4)$$

Die Parameter a_1 , a_2 und a_3 werden durch die Parameteroptimierung der Messkurve mit dem Innenwiderstand bei 20°C bestimmt. Im ersten Bild der Abb. 3.35 wird die Simulation mit der Messkurve verglichen. Die Parameter a_4 , a_5 und a_6 bezeichnen den Temperatureinfluss und werden durch die Parameteroptimierung an der Messkurve des Innenwiderstandes bei 50°C bestimmt. Im zweiten Bild der Abb. 3.35 wird die Simulation mit der Messkurve verglichen. Die Temperatur -2°C wird in diese endgültige Formel eingesetzt. Im letzten Bild der Abb. 3.35 wird das Ergebnis der Formel mit der Messkurve des Innenwiderstandes bei -2°C verglichen. Die durchschnittliche Abweichung der Formel bei drei Temperaturen liegt innerhalb 5 Milliohm. Falls noch mehr Messwerte bei anderen Temperaturen zur Verfügung stehen, wird die Allgemeingültigkeit der Formel weiter optimiert.

Parameter	a_1	a_2	a_3	a_4	a_5	a_6
Wert	-0,014	0,0246	5,9777	-0,0007	0,0447	-0,0001

Tabelle 3.3.: Parameter der Formel zum Widerstand

Auswertung der Formel des Widerstandes

Die weiteren Temperaturen werden in der endgültigen Formel eingesetzt. Das Ziel ist die Allgemeingültigkeit der Formel zu testen. Das Ergebnis wird in der Abb. 3.36 angezeigt. Durch die geringere Ionenbeweglichkeit und die kleinere Reaktionsgeschwindigkeit bei kalten Temperaturen nimmt der Innenwiderstand zu. Das Ergebnis stimmt mit der Beziehung des Innenwiderstandes und der Temperatur überein. Der Innenwiderstand steigt fast proportional mit fallender Temperatur an.

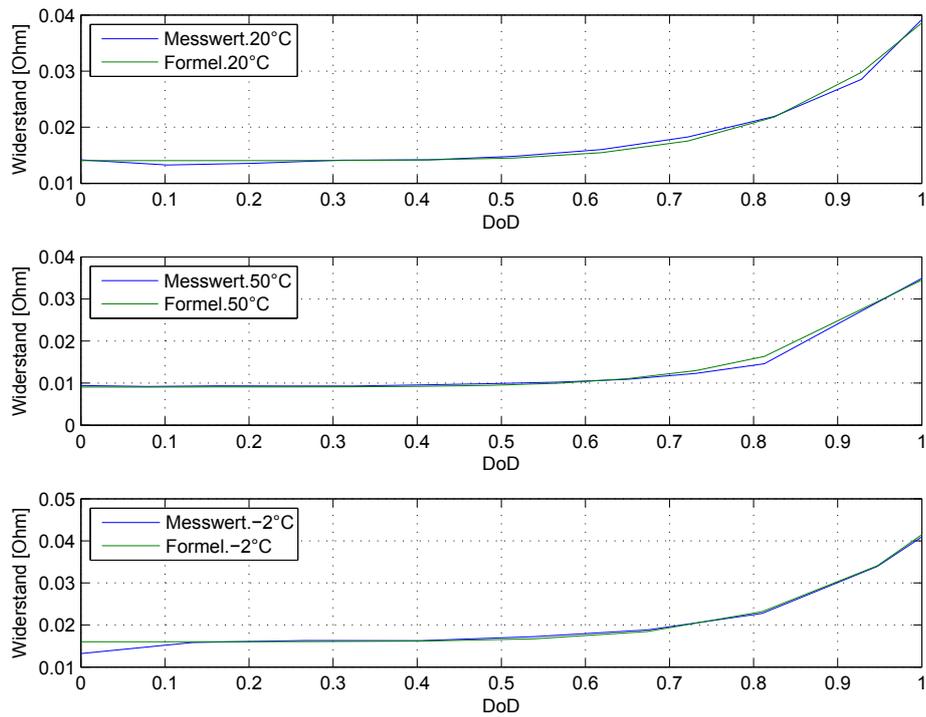


Abbildung 3.35.: Darstellung des Widerstandes aus der Formel

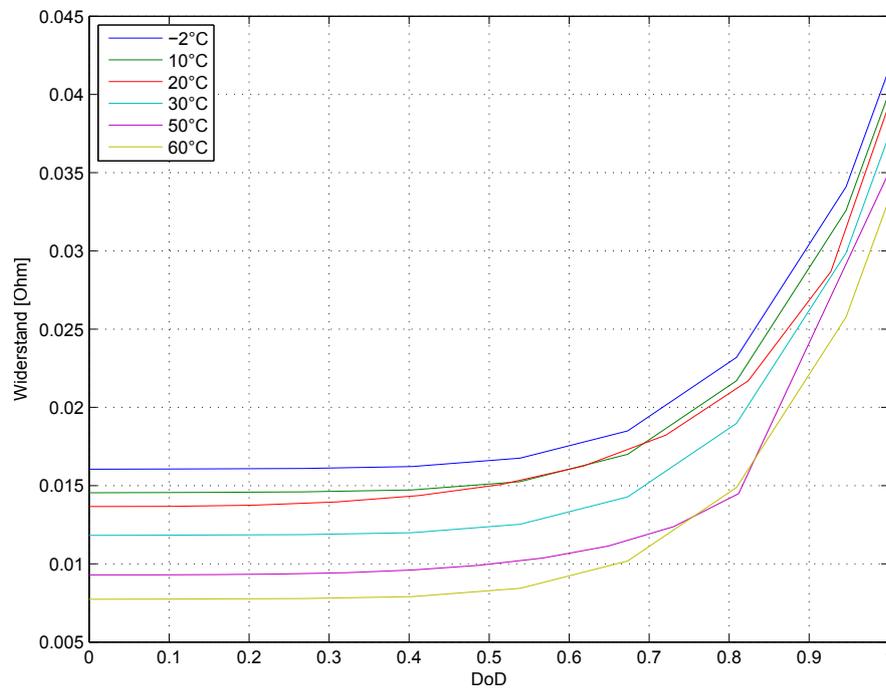


Abbildung 3.36.: Darstellung des Widerstandes bei unterschiedlichen Temperaturen aus der Formel des Widerstandes

3.8. Implementation des angepassten Modells Klasse 2

In den vorangegangenen Abschnitten ist die temperatur- und ladezustandsabhängige Ruhespannung und der Innenwiderstand der Bleibatterie in jeweils einer Formel zusammengefasst. Statt eines Kondensators wird eine von einer Formel (Gl. 3.2) gesteuerte Spannungsquelle als der Hauptenergiespeicher im Modell eingesetzt. Die Ausgabe der Formel zum Widerstand (Gl. 3.4) wird als gesamter Innenwiderstand der Batterie im Modell eingesetzt. Das dynamische Verhalten beim Ein- und Ausschalten des Stroms wird durch eine RC-Serienschaltung nachgebildet.

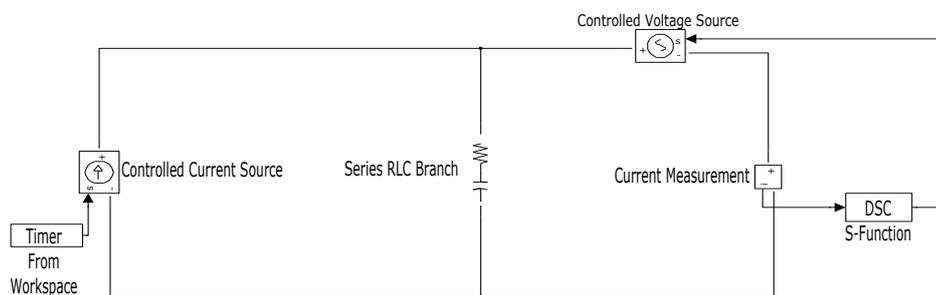


Abbildung 3.37.: das angepasste Modell Klasse 2

Das angepasste Modell 4 Klasse 2 wird in der Abb. 3.37 angezeigt. In diesem Simulink-Modell wird die Spannungsquelle durch einen Benutzer-definierten Block (*dsc*) gesteuert. Bei jedem diskreten Zeitpunkt nimmt dieser Block einen Stromwert auf und ruft die Funktion „diff_uc“ auf. In dieser Funktion wird die Ruhespannung und die Spannung an dem Innenwiderstand berechnet. Der Ablauf wird durch die Abb. 3.38 verdeutlicht. Durch die Parameteroptimierung wird der Wert von *Csurface* und *Rs* bestimmt.

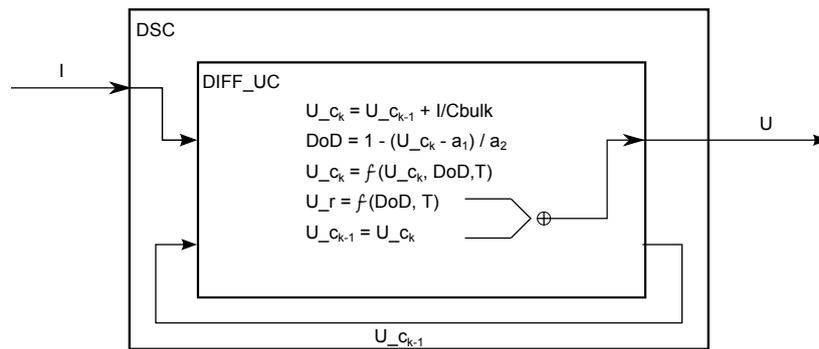


Abbildung 3.38.: Ablauf des Benutzer-definierten Blocks

3.8.1. Vergleich des simulierte Ergebnisses mit dem Messwert

In diesen Abschnitt wird das Ergebnis der Simulation des angepassten Modells 4 Klasse 2 mit dem Messwert verglichen. Damit wird das angepasste Modell 4 Klasse 2 bewertet.

Kontinuierliche Entladung

In der Abb. 3.39 und Abb. 3.40 wird das Simulationsergebnis des angepassten Modells 4 Klasse 2 mit dem Messwert verglichen. Die rote Kurve stellt die simulierte Ruhespannung der Batterie dar. Der Endwert nähert den Messwert gut an. Im Vergleich zum linearen Verlauf des Kondensators ist der Verlauf der Ruhespannung durch das mit der Formel gesteuerte Element sehr gut dargestellt. Die größte Abweichung zwischen dem simulierten Verlauf der Klemmspannung und dem Messwert entsteht beim Ein- und Ausschalten des Stroms. Die Spannungsspitze beim Ausschalten des Stroms wird von der niedrigen Abtastrate verursacht.

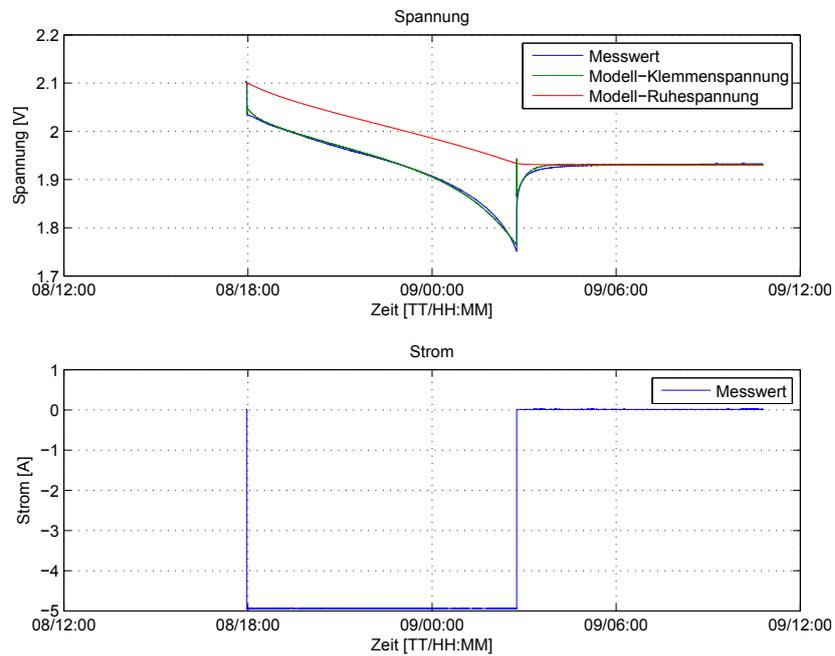


Abbildung 3.39.: Vergleich mit der kontinuierlichen Entladung bei 20°C

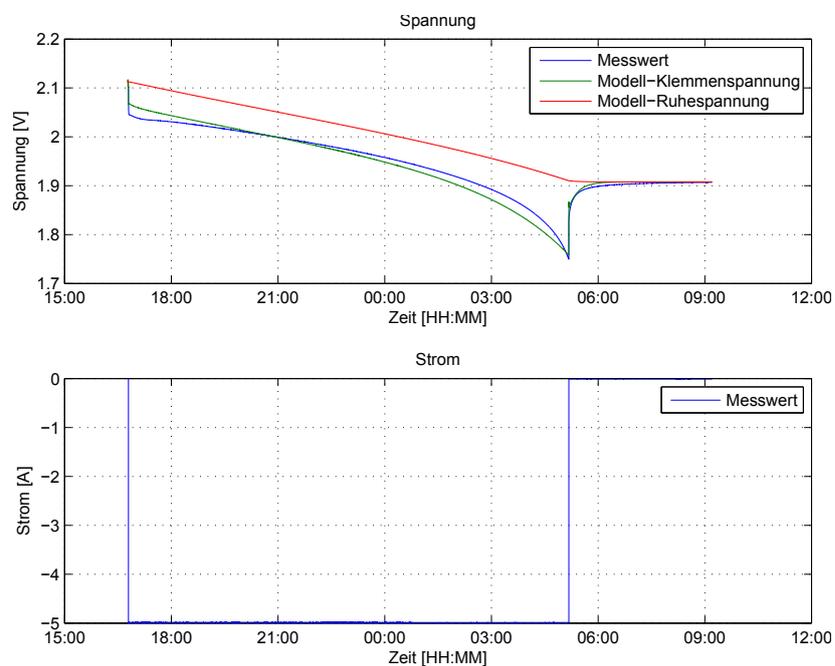


Abbildung 3.40.: Vergleich mit der kontinuierlichen Entladung bei 50°C

Partielle Entladung

In der Abb. 3.41 und Abb. 3.42 wird die Simulationsausgabe des angepassten Modells Klasse 2 dargestellt. Die Abweichung in der langen Ruhephase bei 20°C wird mit der unvollständigen Messung begründet. Die Messung wird kurz nach dem Ausschalten des Stroms abgebrochen. In der Datenverarbeitung wird nur der Endwert der Klemmenspannung entlang der Zeitachse extrapoliert. Die Klemmenspannung sollte in der Ruhephase noch langsam ansteigen. Dabei wird diese Abweichung etwa verkleinert. Bei der höheren Temperatur ist der von der Selbstentladung verursachte Spannungsabfall in der Ruhephase bemerkbar (Abb. 3.42). Der Spannungsabfall kann nicht direkt vom dem angepassten Modell 4 Klasse 2 nachgebildet werden, weil es an Messpunkten der Ruhephase in der Formel mangelt. Ansonsten kann das angepasste Modell 4 Klasse den Verlauf der partiellen Entladung gut darstellen.

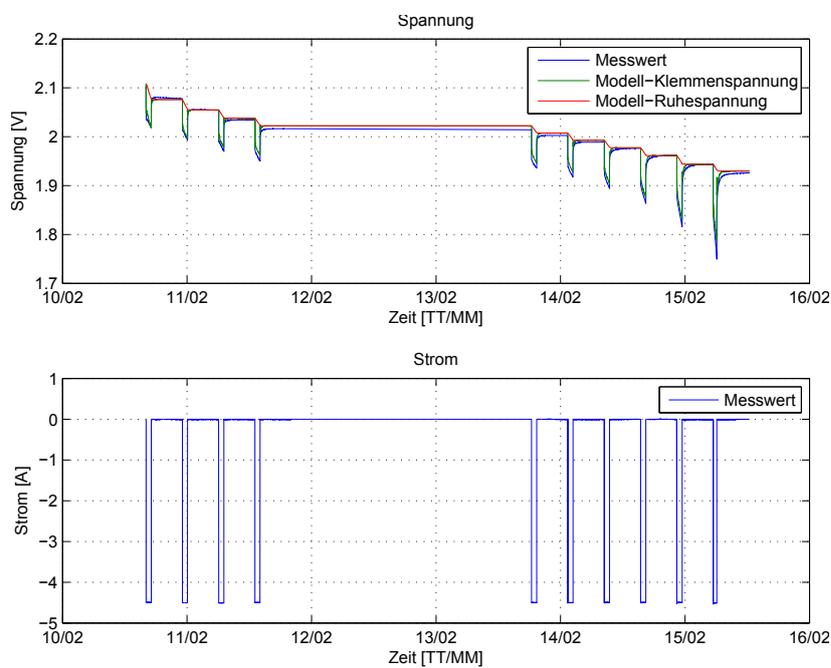


Abbildung 3.41.: Vergleich mit der partiellen Entladung bei 20°C

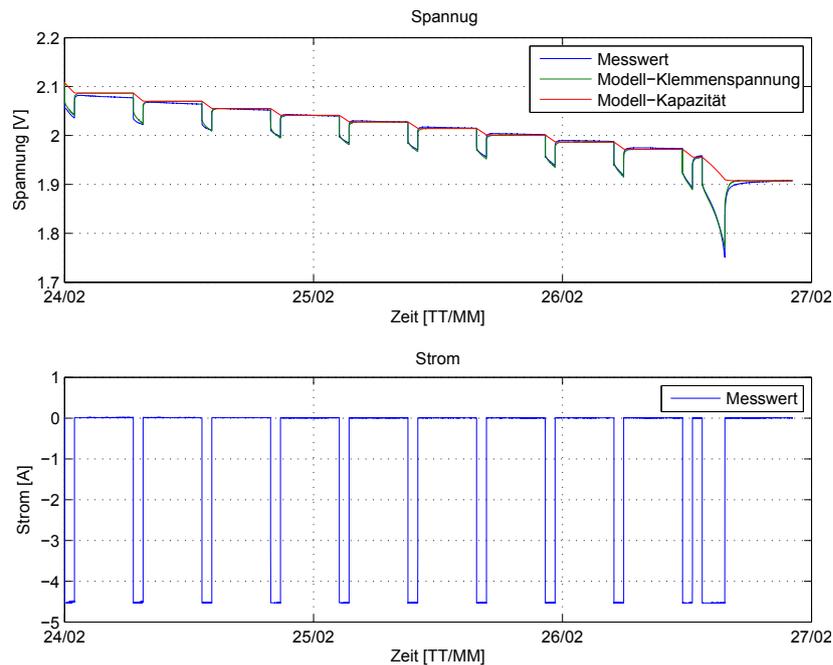


Abbildung 3.42.: Vergleich mit der partiellen Entladung bei 50°C

Dynamisches Verhalten beim Entladen

In der Abb. 3.43 und Abb. 3.44 wird die Berechnung des angepassten Modells 4 Klasse 2 dargestellt. Der simulierte Verlauf stimmt mit Teilen des Messwertes überein. Im Vergleich mit dem angepassten Modell 4 Klasse 1 hat das angepasste Modell 4 Klasse 2 eine einfache Schaltung, besonders beim Aufbau des Zweiges, der das dynamische Verhalten der Batterie beschreibt. Wegen des begrenzten Rechenaufwandes des Mikrocontrollers wird hier im angepassten Modell 4 Klasse 2 auf den komplexen Zweige verzichtet. Deshalb tritt am Beginn der Entladung ein Fehler auf. Es fehlen einige kleine Energiespeicher, um die dynamische Änderung nachzubilden. Dieser Fehler kann später durch das Kalman-Filter korrigiert werden.

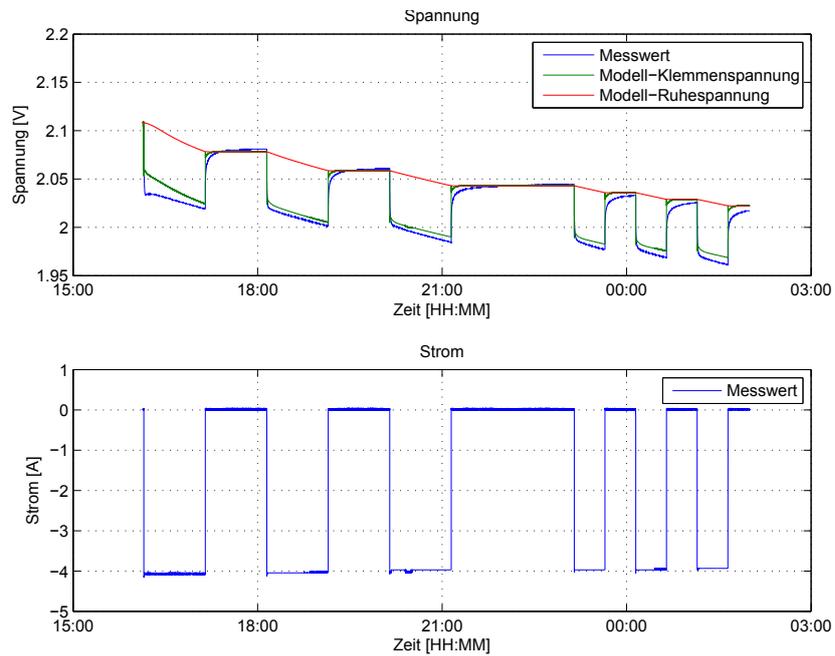


Abbildung 3.43.: Vergleich mit dem dynamischen Verhalten bei 20°C

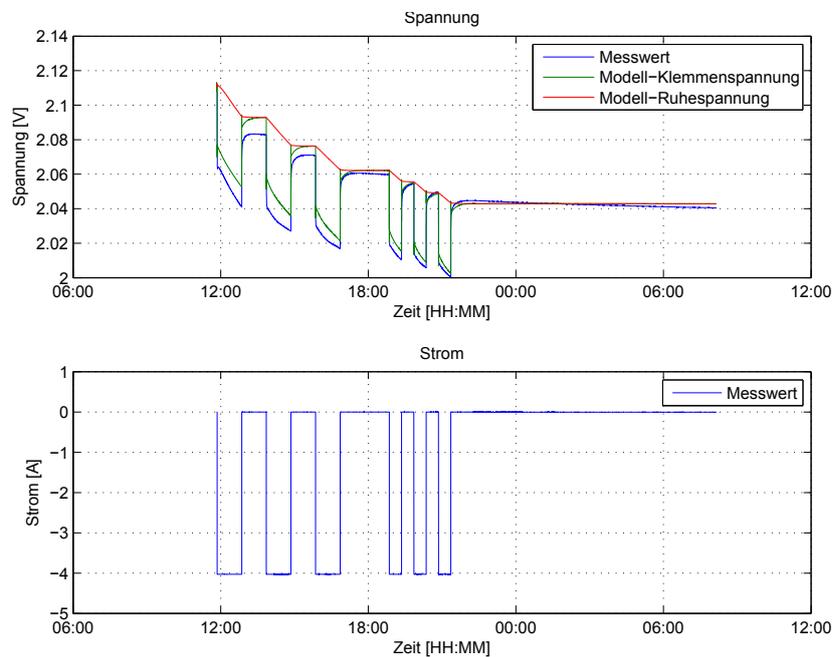


Abbildung 3.44.: Vergleich mit dem dynamischen Verhalten bei 50°C

3.9. Auswertung des angepassten Modells 4 Klasse 2

Das angepasste Modell 4 Klasse 2 besteht aus einem mit der angepassten Formel gesteuerten Energiespeicher, einem mit der angepassten Formel eingestellten Innenwiderstand und einer RC-Parallelschaltung, die das dynamische Verhalten der Batterie nachbildet.

Mit Hilfe der zwei Formeln eignet sich das angepasste Modell 4 Klasse 2 zur Ermittlung des Verlaufs der Ruhespannung und zur Beschreibung des temperaturabhängigen Verhaltens der Bleibatterie. Ein Modell, welches nur einen konstanten Kondensator als Energiespeicher enthält kann das Verhalten der Batterie nicht so gut nachbilden.

Das angepasste Modell 4 Klasse 2 kann das Verhalten der Entladung unter unterschiedlichen Lastprofilen bei 20°C und 50°C darstellen, d.h. sowohl bei der kurzzeitigen Entladung als auch bei der langen Entladung. Die Abweichung zwischen dem Messwert und dem simulierten Wert ist je nach eingesetzten Lastprofile unterschiedlich. Die Abweichung bei dynamischer Entladung ist beträchtlich größer als bei anderer Entladungen. Die Ungenauigkeit des Modells und die unbekannt verfügbare Kapazität verursacht hauptsächlich diese Abweichung. Eine falsche Eingabe des Ladezustands in der Formel führt zur fehlerhaften Kalibrierung der Ruhespannung bzw. des Innenwiderstandes. Um den initialen Ladezustand bzw. die noch verfügbare Kapazität in der Batterie zu bestimmen, kann man vor der Belastung durch ein spezielles Messgerät die Säuredichte messen, z.B mittels Säurenheber und Refraktormeter. Das Messergebnis der Säuredicht wird im Anhang [B](#) dargestellt.

3.10. Das angepasste Modell 4 Klasse 3

Das angepasste Modell 4 Klasse 1 eignet sehr gut zur Darstellung des dynamischen Verhaltens der Batterie. Das angepasste Modell 4 Klasse 2 kann eher besser die langzeitige Entladung darstellen. Das angepasste Modell 4 Klasse 3 verbindet die Vorteile beider Modelle, den zur Beschreibung des dynamischen Verhaltens verwendeten Zweig im angepassten Modell 4 Klasse 1 und den angepassten Hauptenergiespeicher und Innenwiderstand im angepassten Modell 4 Klasse 2. Das angepasste Modell 4 Klasse 3 wird in der Abb. 3.45 gezeigt. Ein beträchtlicher Nachteil des angepassten Modells 4 Klasse 3 ist der hohe Rechenaufwand. In dem Forschungsprojekt „BATSEN“ wird in der Zukunft mit Hilfe eines Modells der Ladezustand der Batterie via Mikrocontroller analysiert und bestimmt. Dabei sollte unbedingt bei dem Design des Modells der Rechenaufwand bzw. die Komplexität des Modells berücksichtigt werden. Deswegen wird das angepasste Modell 4 Klasse 3 nicht im Rahmen dieser Arbeit implementiert. Es ist zu erwarten, mit Hilfe des angepassten Modells 4 Klasse 3 sowohl den Verlauf der kurzzeitigen Entladung als auch den der längeren Entladung sehr gut darzustellen.

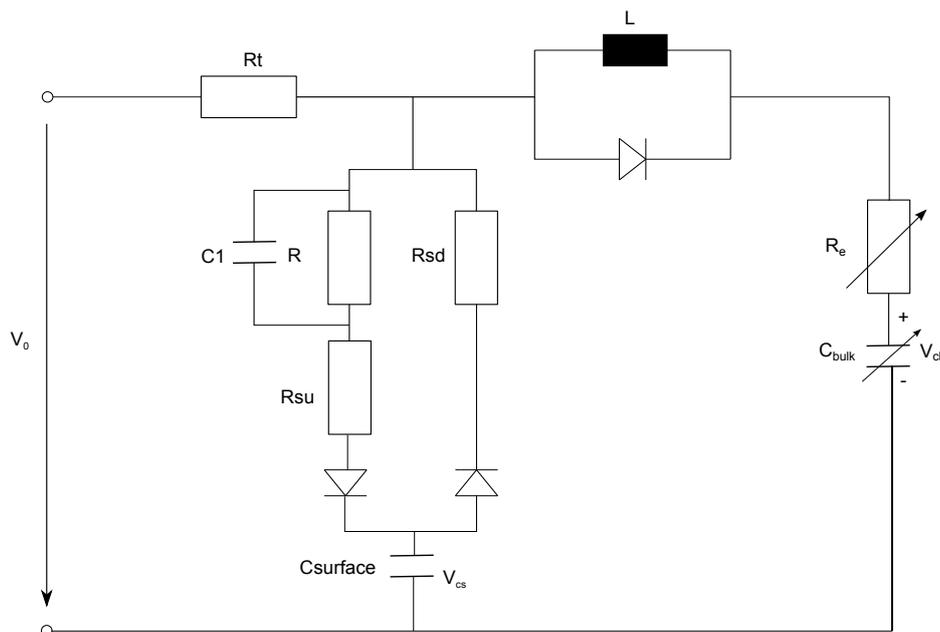


Abbildung 3.45.: Das angepasste Modell 4 Klasse 3

3.11. Auswertung aller implementierten Modelle

In diesem Kapitel werden drei Modelle in Matlab /Simulink implementiert. Dann werden entsprechende Simulationsergebnisse dargestellt und mit dem Messwert verglichen. Das Modell 4 hat die wenigsten Parameter und kann den Verlauf der Entladung bzw. Ladung nur unzureichend darstellen.

Das angepasste Modell 4 Klasse 1 enthält eine komplizierte elektrische Schaltung. Damit kann dieses Modell mit angepassten Parametern das dynamische Verhalten der Bleibatterie sehr gut beschreiben. Aber dieses Modell eignet sich nicht für die Darstellung der kontinuierlichen Entladung. Bei der Aufladung kann das angepasste Modell 4 Klasse 1 den Verlauf außerhalb der Ruhephase gut darstellen.

Das angepasste Modell 4 Klasse 2 besitzt eine einfache elektrische Schaltung, aber die meisten Parameter. Eine Vielzahl von Parametern werden in den Formeln eingesetzt. Statt eines linearen Kondensators und eines konstanten Innenwiderstandes kann mit Hilfe der Formeln die Nichtlinearität des Verlaufs der Ruhespannung und des Innenwiderstandes nachgebildet werden. Dieses angepasste Modell 4 Klasse 2 berücksichtigt den Einfluss der Temperatur und des Ladezustandes auf das Verhalten der Bleibatterie. Solche Betrachtung wird in den Formeln zusammengefasst. Das angepasste Modell 4 Klasse 2 kann mit gleichen Parametern unterschiedliche Entladungsverhalten bei 20°C und 50°C darstellen. Es eignet sich gut für die Beschreibung der kontinuierlichen Entladung. Mit angepassten Parametern kann das Verhalten unter Minustemperaturen ebenso nachgebildet werden.

Das angepasste Modell 4 Klasse 3 verknüpft das angepasste Modell 4 Klasse 1 mit dem angepassten Modell 4 Klasse 2 und besitzt komplexe Schaltungen. Dabei könnte das angepasste Modell 4 Klasse 3 alle Verläufe der Batterie gut darstellen.

Die Vor- und Nachteile der genannten Modelle sind unterschiedlich und werden durch die Tabelle 3.4 verdeutlicht.

	Modell 4	Modell 4 Klasse 1	Modell 4 Klasse 2	Modell 4 Klasse 3 *
Hauptenergiespeicher	Kondensator	Kondensator	nichtlineare angepasste Elemente	nichtlineare angepasste Elemente
Innenwiderstand	konstanter Widerstand	konstanter Widerstand	einstellbarer Widerstand	einstellbarer Widerstand
Temperaturabhängigkeit	/	/	+	+
SOC-Abhängigkeit	/	/	+	+
Induktivität	/	+	/	+
lange Entladung	-	-	+++	+++
kurzzeitige Entladung	+	+++	++	++
Aufladung	+	++	++ ¹	++
Komplexität ²	normal	hoch	normal	sehr hoch
Erweiterungsmöglichkeit	+++	+	++	+
Anzahl der Parameter	5	9	2 + 14 ³	9 + 14
Rechenaufwand	niedrig	normal	hoch	sehr hoch

Tabelle 3.4.: Vergleich der implementierten Modelle

Bemerkung zur Tabelle:

Das Zeichen „+“ bedeutet, dass das Modell dieses Kriterium erfüllt.

Das Zeichen „++“ bedeutet, dass mit diesem Modell diese Eigenschaft mit begrenztem Fehler gut dargestellt werden kann.

Das Zeichen „+++“ bedeutet, dass mit diesem Modell diese Eigenschaft ausreichend gut dargestellt werden kann.

Das Zeichen „/“ bedeutet, dass das Modell dieses Kriterium nicht erfüllt.

Das Zeichen „*“ bedeutet, dass das angepasste Modell 4 Klasse 3 nach der Erfahrung bewertet wird. Das Zeichen (1) bedeutet, dass durch angepasste Parameter das angepassten Modell 4 Klasse 2 den Verlauf der Aufladung nachbilden kann.

Das Zeichen (2) bedeutet, dass die Komplexität des Modells durch die Anzahl der im Modell eingesetzten elektrischen Elemente bezeichnet wird.

Das Zeichen (3) bedeutet, dass 2 für die Anzahl der verwendeten elektrischen Elemente steht und 14 die Anzahl der in der Formel eingesetzten Parameter ist.

4. Implementation des Batteriemodells in das Kalman-Filter

In dem vorliegenden Kapitel wurde die Bleibatterie mit verschiedenen Lastprofilen gemessen. Dabei wurde versucht, ein Modell aufzubauen, das das Entlade- bzw. Aufladeverhalten der Bleibatterie so genau wie möglich darstellen kann. Das angepasste Modell 4 Klasse 2 kann die unterschiedlichen Messwerte gut darstellen. Trotzdem erscheinen Fehler in der Simulation. Die Abweichung ist von dem eingesetzten Lastprofil abhängig. In diesem Kapitel wird das angepasste Modell 4 Klasse 2 im Kalman-Filter eingesetzt. Aufgrund der Eigenschaften des Kalman-Filters (Abschnitt. 2.1.1) ist es möglich Fehler aufgrund verrauschter Messungen oder Ungenauigkeiten im Modell zu korrigieren.

4.1. Umsetzung des angepassten Modells 4 Klasse 2

Der grundlegende Algorithmus des Kalman-Filters wurde in dem Abschnitt. 2.3.1 erwähnt. In das Kalman-Filter sollte das eingesetzte Modell in der Form von Zustandsdifferentialgleichungen beschrieben werden. Deshalb sollte das angepasste Modell 4 Klasse 2 zunächst in die Zustandsdifferentialgleichung umgesetzt werden. Das angepasste Modell 4 Klasse 2 (Abb. 4.1) wird durch die Zustandsvariablen V_{cb} , $V_{surface}$, V_0 beschrieben.

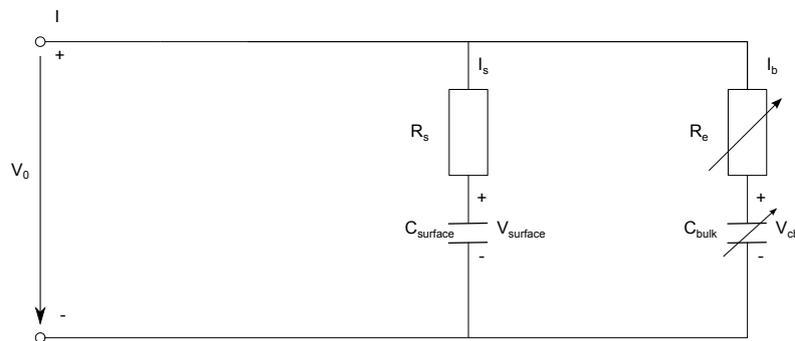


Abbildung 4.1.: Angepasste Modell 4 Klasse 2

Die Beziehung zwischen den Zustandsvariablen lautet:

$$V_0 = I_b * R_e + V_{cb} \quad (4.1)$$

$$V_0 = I_s * R_s + V_{cs} \quad (4.2)$$

Laut der Kirchhoffschen Regel gilt, $I = I_b + I_s$, somit gilt auch:

$$I_b * (R_e + R_s) = I * R_s + V_{cs} - V_{cb} \quad (4.3)$$

Aufgrund $I_b = \dot{V}_{cb} * C_{bulk}$ wird Gl. 4.3 neu geordnet:

$$\dot{V}_{cb} = -\frac{V_{cb}}{C_{bulk} * (R_e + R_s)} + \frac{V_{cs}}{C_{bulk} * (R_e + R_s)} + \frac{I * R_s}{C_{bulk} * (R_e + R_s)} \quad (4.4)$$

Durch eine ähnliche Umformung wird Gl 4.5 erzeugt.

$$\dot{V}_{cs} = -\frac{V_{cs}}{C_{surface} * (R_e + R_s)} + \frac{V_{cb}}{C_{surface} * (R_e + R_s)} + \frac{I * R_e}{C_{surface} * (R_e + R_s)} \quad (4.5)$$

Die Klemmspannung wird in der Gl. 4.6 berechnet.

$$V_0 = \begin{bmatrix} \frac{R_s}{R_e + R_s} & \frac{R_e}{R_e + R_s} \end{bmatrix} \begin{bmatrix} V_{cb} \\ V_{cs} \end{bmatrix} + \frac{R_e * R_s}{R_e + R_s} * I \quad (4.6)$$

Die Ableitung der Klemmspannung lautet:

$$\begin{aligned} \dot{V}_0 &= \left[-\frac{R_s}{C_{bulk} * (R_e + R_s)^2} + \frac{R_e}{C_{surface} * (R_e + R_s)^2} \right] * V_{cb} \\ &+ \left[\frac{R_s}{C_{bulk} * (R_e + R_s)^2} - \frac{R_e}{C_{surface} * (R_e + R_s)^2} \right] * V_{cs} \\ &+ \left[\frac{R_s^2}{C_{bulk} * (R_e + R_s)^2} + \frac{R_e^2}{C_{surface} * (R_e + R_s)^2} \right] * I \end{aligned} \quad (4.7)$$

In der Literatur, z.B. [7], [21] wird $\frac{dI}{dt} \approx 0$ angenommen, weil der Strom meistens konstant oder die Veränderung sehr klein ist. Dieser einfache Ersatz der Ableitung des Stroms verursacht den kurzzeitigen hohen Spannungssprung beim Stromwechsel. Um diesen Fehler zu beheben, wird der Strom ebenfalls als eine Zustandsvariable angenommen. Die Ableitung des Stroms wird aus der Differenz des vorherigen und des aktuellen Stromwertes geschätzt. Dieser Einsatz der Stromdifferenz kann den hohen Spannungssprung nur dämpfen, aber nicht vollständig entfernen.

Die neue Zustandsmatrix lautet wie Gl. 4.8.

$$\begin{aligned}
 \begin{bmatrix} \dot{V}_{cb} \\ \dot{V}_{cs} \\ \dot{V}_0 \\ i \end{bmatrix} &= \begin{bmatrix} -\frac{1}{C_{bulk}*(R_e+R_s)} & \frac{1}{C_{bulk}*(R_e+R_s)} & 0 & 0 \\ \frac{1}{C_{surface}*(R_e+R_s)} & -\frac{1}{C_{surface}*(R_e+R_s)} & 0 & 0 \\ A(3,1) & 0 & A(3,3) & -\frac{R_e*R_s}{R_e+R_s} \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} V_{cb} \\ V_{cs} \\ V_0 \\ i \end{bmatrix} \\
 &+ \begin{bmatrix} \frac{R_s}{C_{bulk}*(R_e+R_s)} \\ \frac{R_s}{C_{surface}*(R_e+R_s)} \\ \frac{R_e^2}{C_{surface}*(R_e+R_s)^2} + \frac{R_e*R_s}{C_{surface}*(R_e+R_s)^2} \\ 1 \end{bmatrix} * i \\
 A(3,1) &= -\frac{R_s}{C_{bulk} * (R_e + R_s)^2} + \frac{R_e}{C_{surface} * (R_e + R_s)^2} \\
 &-\frac{R_s^2}{C_{bulk} * R_e * (R_e + R_s)^2} + \frac{R_s}{C_{surface} * (R_e + R_s)^2} \\
 A(3,3) &= \frac{R_s}{C_{bulk} * R_e * (R_e + R_s)} - \frac{1}{C_{surface} * (R_e + R_s)} \tag{4.8}
 \end{aligned}$$

Ein kontinuierliches zeitinvariantes lineares System kann in der Zustandsvariablen-Form beschrieben werden:

$$\dot{x}(t) = A * x(t) + B * u(t) \tag{4.9}$$

$$y(t) = C * x(t) \tag{4.10}$$

wobei

A : Zustandsübergangsmatrix

B : Eingabematrix

C : Ausgabematrix

$x(t)$: Zustandsmatrix

$\dot{x}(t)$: folgende Zustandsmatrix

$u(t)$: Eingabe

$y(t)$: Ausgabe

Ein zeitdiskretes System wird durch die folgenden Gl. 4.13 beschrieben.

$$x_{k+1} = A_d * x_k + B_d * u_k \quad (4.11)$$

$$y_{k+1} = C_d * x_{k+1} \quad (4.12)$$

wobei

$$A_d = I + A * T_c; \quad I : \text{Einheitsmatrix}$$

$$B_d = B * T_c; \quad C_d = C;$$

Das vollständige zeitdiskrete System wird mit Gl 4.14 berechnet.

$$\begin{aligned} \begin{bmatrix} \dot{V}_{cb} \\ \dot{V}_{cs} \\ \dot{V}_0 \\ i \end{bmatrix} &= \begin{bmatrix} 1 - \frac{T_c}{C_{bulk} * (R_e + R_s)} & \frac{1}{C_{bulk} * (R_e + R_s)} & 0 & 0 \\ \frac{1}{C_{surface} * (R_e + R_s)} & 1 - \frac{T_c}{C_{surface} * (R_e + R_s)} & 0 & 0 \\ A_d(3, 1) & 0 & A_d(3, 3) & -\frac{R_e * R_s}{R_e + R_s} \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &* \begin{bmatrix} V_{cb} \\ V_{cs} \\ V_0 \\ I \end{bmatrix} \\ &+ \begin{bmatrix} \frac{R_s}{C_{bulk} * (R_e + R_s)} \\ \frac{R_e}{C_{surface} * (R_e + R_s)} \\ \frac{R_e^2}{C_{surface} * (R_e + R_s)^2} + \frac{R_e * R_s}{C_{surface} * (R_e + R_s)^2} \\ 1 \end{bmatrix} * I \\ A_d(3, 1) &= -\frac{R_s}{C_{bulk} * (R_e + R_s)^2} + \frac{R_e}{C_{surface} * (R_e + R_s)^2} \\ &- \frac{R_s^2}{C_{bulk} * R_e * (R_e + R_s)^2} + \frac{R_s}{C_{surface} * (R_e + R_s)^2} \\ A_d(3, 3) &= 1 + \frac{R_s * T_c}{C_{bulk} * R_e * (R_e + R_s)} - \frac{T_c}{C_{surface} * (R_e + R_s)} \end{aligned} \quad (4.13)$$

$$C = [0 \ 0 \ 1 \ 0] \quad (4.14)$$

4.2. Implementation des angepassten Modells 4 Klasse 2 in das Kalman-Filter

Die Umsetzung des angepassten Modells 4 Klasse 2 in das Kalman-Filter wird durch den folgenden Code implementiert. Im ersten Schritt wird die Zustandsübergangsmatrix bzw. Eingabematrix nach Gl. 4.14 erstellt. In jeder Iteration wird diese Matrix mit dem aktuell berechneten Wert neu zugewiesen (List. 4.1).

Listing 4.1: Zuordnung der Matrix

```

1
2   for k=1:num_samples
3       % aktuelles Sampleintervall aus Vektor nehmen
4       Tc = Tc_vec(k);
5       % Berechnung des aktuellen Widerstands...
6       % mit der angepassten Formel
7       Re = 0.014 + 0.0246*dod^5.9777 - 0.0007*(1-dod)^0.0447...
8           - 0.0001*(T-20);
9       % Zuordnung der Zustandsübergangsmatrix
10      a11 = 1 - Tc / (Cbulk * (Re + Rs));
11      a12 = 1 / (Cbulk * (Re + Rs));
12      a13 = 0;
13      a14 = 0;
14      a21 = Tc / (Csurface * (Re + Rs));
15      a22 = 1 - Tc / (Csurface * (Re + Rs));
16      a23 = 0;
17      a24 = 0;
18      a31 = (-1) * Tc * Rs / (Cbulk * (Re + Rs)^2) + ...
19            Tc * Re / (Csurface * (Re + Rs)^2) ...
20            - Tc * Rs^2 / (Cbulk * Re * (Re + Rs)^2) ...
21            + Tc * Rs / (Csurface * (Re + Rs)^2);
22      a32 = 0;
23      a33 = 1 + Tc * Rs / (Cbulk * Re * (Re + Rs)) ...
24            - Tc / (Csurface * (Re + Rs));
25      a34 = -1*((Re+Rs)/(Re+Rs));
26      a41 = 0;
27      a42 = 0;
28      a43 = 0;
29      a44 = 0;
30
31      A = [a11 a12 a13 a14;a21 a22 a23 a24;...
32           a31 a32 a33 a34;a41 a42 a43 a44];
33      % Zuordnung der Ausgabematrix
34      b1 = Tc/Cbulk;

```

```

35 b2 = Tc * Re / (Csurface * (Re + Rs));
36 b3 = Tc * Re^2 / (Csurface * (Re + Rs)^2) ...
37 + Tc * Re * Rs / (Csurface * (Re + Rs)^2)...
38 + (Re*Rs/(Re+Rs));
39 b4 = 1;
40
41 B= [b1;b2;b3;b4];

```

In zweiten Schritt wird der Zustandsraum aktualisiert. Die Ruhespannung wird anhand der angepassten Formel kalibriert. Anschließend werden der aktuelle Ladezustand und die Fehlerkovarianz neu berechnet (List. 4.2).

Listing 4.2: Prädiktion

```

1 % Berechnung der Spannung des Kondensators
2 x(1) = (A(1,:) * x + B(1) * i_im(k)) ;
3
4 % Multiplikation der Spannung des Kondensators...
5 % mit der angepassten Formel
6 m = x(1) * (1 + (-0.117 * dod * (1 - dod)^(2.348) + 0.0212 * dod^0.4096 ...
7 + (T - 20) * (-0.0156 * dod^1.7468 + 0.0161 * dod^1.6306)) - k1);
8
9 % Berechnung der übrigen Zustandsvariablen
10 n = [m; x(2:4)];
11 x(2:4) = A(2:4,:) * n + B(2:4) * i_im(k);
12
13 % Berechnung des aktuellen Ladezustands
14 dod = 1 - (m - u_anf) / u2_end;
15
16 % Übergabe der aktuell geschätzten Ausgabe
17 z1 = x(3);
18
19 % Berechnung der Fehlerkovarianzmatrix
20 P = A * P * A' + Q;
21
22 % Abspeichern der Zustandsvariablen
23 sV(:, k) = [x(1); m; x(2:4)];
24
25 % Abspeichern der Schätzung
26 zV(k) = z1;

```

Wenn der Abstand der Korrektur erreicht, wird die Korrektur durchgeführt. Mit Hilfe des Messwertes wird die Schätzung des Systemmodells korrigiert. Dabei wird das Kalman Gain berechnet und die Fehlerkovarianzmatrix aktualisiert (List. 4.3).

Listing 4.3: Korrektur

```
1 %% Phase der Korrektur
2 if (rem(k,iter) == 0)
3     % Übergabe der aktuellen Zustandsvariablen
4     x1 = x;
5
6     % Berechnung des Kalman Gains
7     K = (P*C') / (C*P*C' + R);
8     % Korrektur der Zustandsvariablen
9     x = x1 + K*(u_im(k) - z1);
10
11     % Übergabe des korrigierten Zustandsvariablen
12     x1 = x;
13
14     % Korrektur der Fehlerkovarianzmatrix
15     P = P - K*C*P;
16
17     % Abpeicherung des korrigierten Zustandsvariablen
18     xV(:,k) = x1;
19
20 else
21     % Abpeicherung des letzt korrigierten Zustandsvariablen
22     xV(:,k) = x1;
23 end
```

4.3. Messrauschen des Sensors

Das Kalman-Filter ist bekanntlich sehr gut geeignet, Messrauschen oder ähnliche Fehler mit der statistischen Eigenschaft von Messwerten zu entfernen. Das sollte im Folgenden für die im Forschungsprojekt „BATSEN“ verfügbaren Zellsensoren als Beispiel untersucht werden.

In diesem Abschnitt wird ein angenommenes Messrauschen des Sensors zu dem von dem Tischmultimeter „Fluke 45“ gemessenen vergleichsweise genauen Wert addiert. In dem Forschungsprojekt „BATSEN“ wird die Batterie durch einen drahtlosen Zellsensor überwacht (siehe Abschnitt. 1.2.2). Dabei steht kein Messwert so genau wie der Messwert vom Tischmultimeter „Fluke 45“ zur Verfügung. Deshalb ist es sinnvoll, dass der verrauschte Messwert in das Kalman-Filter eingegeben wird.

Als Versuche messen das Tischmultimeter „Fluke 45“ und ein Zellsensor gleichzeitig den Spannungsgenerator. Die beiden Messwerte werden verglichen. Die Abweichungsverteilung einzelner Sensoren wird in der Abb. 4.2 gezeigt.

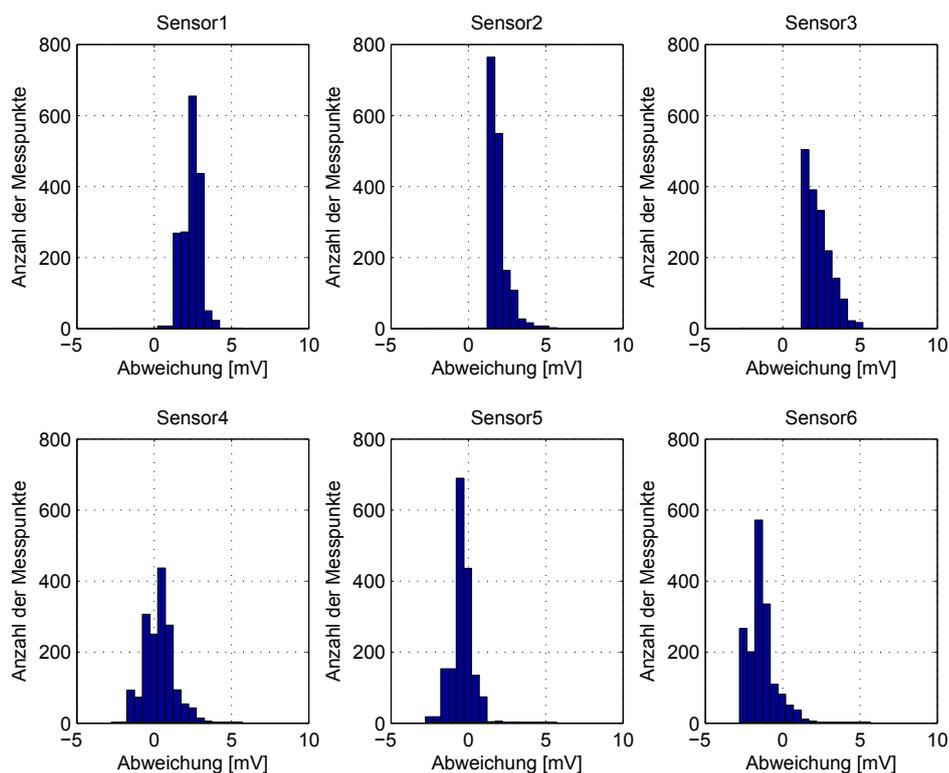


Abbildung 4.2.: Rauschenverteilung einzelner Sensoren

Die von einer Normalverteilung abweichende Verteilung des Rauschens lässt sich durch die Quantierungsfehler der beiden Messgeräte und die ungleiche Genauigkeit zwischen dem

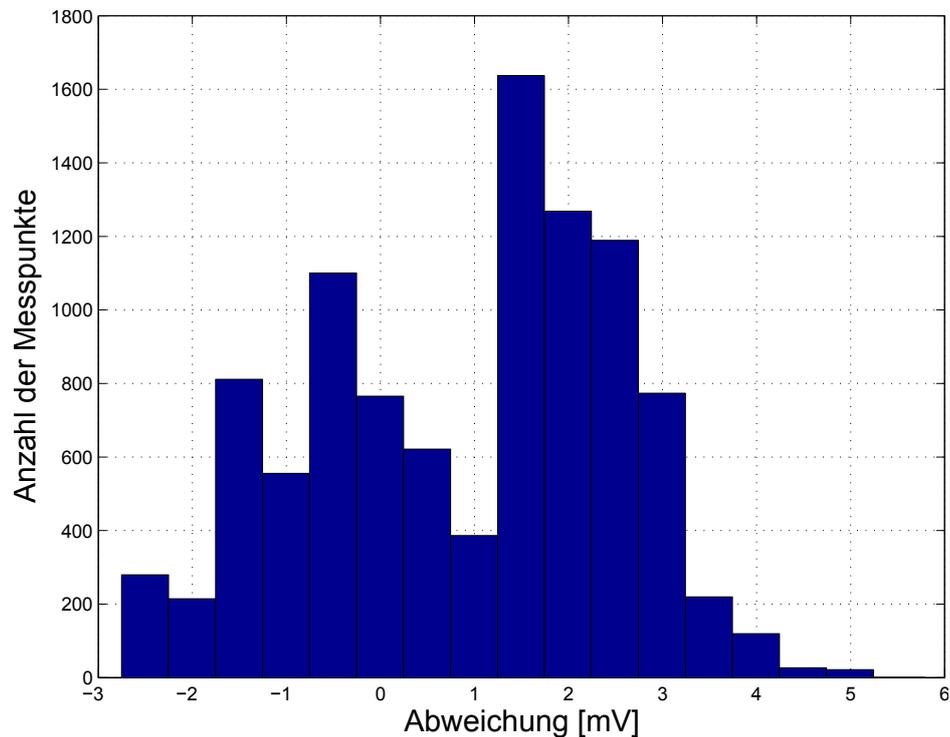


Abbildung 4.3.: Rauschenverteilung aller Sensoren

Sensor und dem Tischmultimeter „Fluke 45“ begründen. Das Rauschen liegt allerdings zwischen -3mV und 6mV . Damit erweist sich der Messwert des Sensors als zuverlässig und anwendbar. Mit Hilfe des zuverlässigen Messwertes kann das Kalman-Filter so oft wie möglich die Schätzung korrigieren.

In der Abb. 4.3 wird die gesamte Verteilung des Rauschens aller Sensoren dargestellt.

Für die Kalman-Filter-Simulationen mangelt es an den zu den Entlade- und Ladevorgängen passenden Messwerten der Sensoren. Es sind also sehr genaue Werte von dem Tischmultimeter „Fluke 45“ verfügbar. Zur Rekonstruktion des Messwertes des Sensors wird versucht, dass das gemessene Rauschen des Sensors zu dem Messwert des Tischmultimeter „Fluke 45“ addiert wird. Das Rauschen wird gleich wie in der Abb. 4.3 verteilt. Der verrauschte Messwert wird in das Kalman-Filter eingegeben. Die Varianz des gesamten Rauschens liegt bei 3,48 und wird als Rauschenvarianz in das Kalman-Filter eingesetzt. Ziel ist zu prüfen, wie gut das Kalman-Filter beim verrauschten Messwert funktioniert.

4.4. Vergleich des Ergebnisses des Kalman-Filters mit dem verrauschten Messwert

In diesem Abschnitt wird das Ergebnis des Kalman-Filters dargestellt. Die Testumgebung werde die gleiche wie die bei der Messung mit dem Sensor eingestellt. Jede Sekunde (Iteration) nimmt das Kalman-Filter einen Stromwert auf. Der Stromwert dient zur Berechnung der aktuellen Schätzung. Jede zwei Sekunden wird ein verrauschter Spannungswert aufgenommen. Der Spannungswert dient zur Korrektur der Schätzung. Die Varianz des gesamten Rauschens beträgt 3,48.

Kontinuierliche Entladung

In der Abb. 4.4 und Abb. 4.5 wird der Vergleich mit dem entsprechenden verrauschten Messwert dargestellt. Im Vergleich mit dem verrauschten Messwert (grüne Kurve) verläuft die Klemmspannung des Kalman-Filters viel stabiler und steht mittig zur verrauschten Messkurve. Nach der Korrektur kann der Verlauf der Klemmspannung vollständig dargestellt werden. Wegen der unvollständigen Ableitung des Stroms tritt ein kleiner Spannungssprung beim Stromwechsel auf. Der nichtlineare Verlauf der Ruhespannung kann deutlich dargestellt werden.

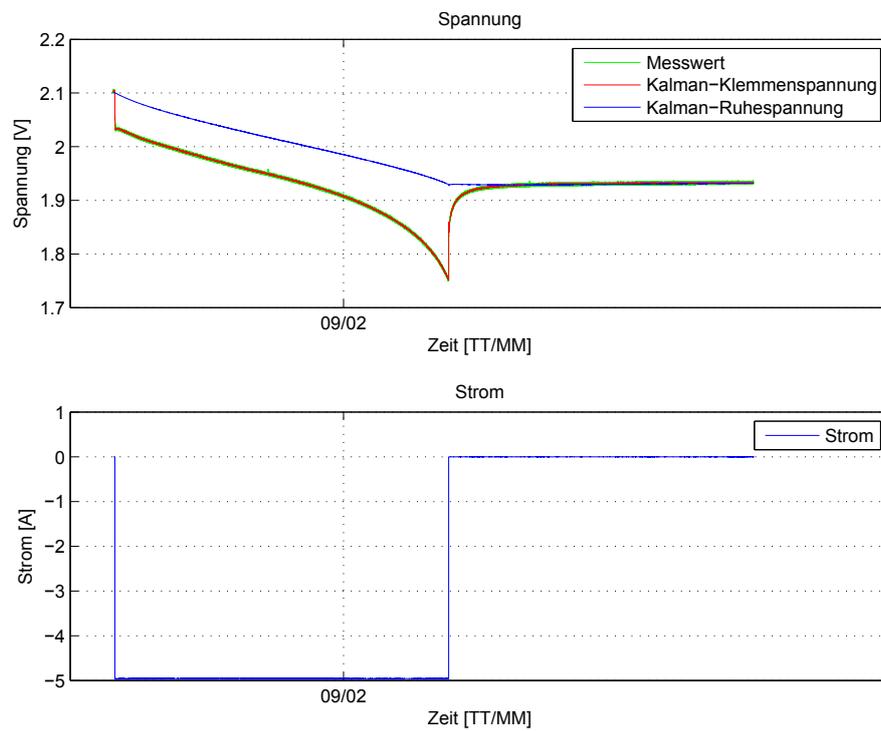


Abbildung 4.4.: Vergleich der kontinuierlichen Entladung bei 20°C

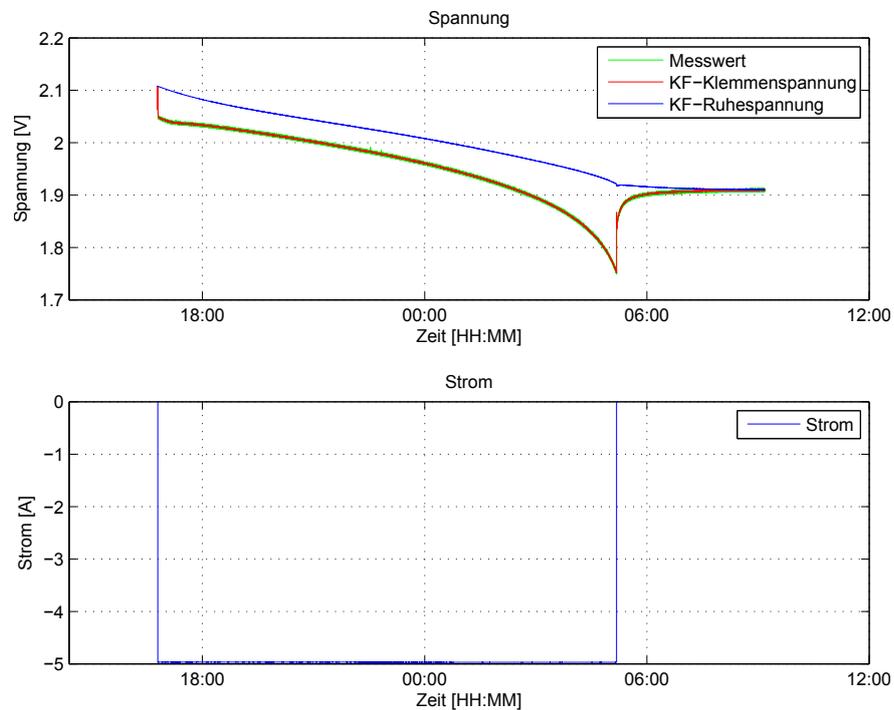


Abbildung 4.5.: Vergleich der kontinuierlichen Entladung bei 50°C

Partielle Entladung

Die Abb. 4.6 und die Abb. 4.7 zeigen den Verlauf des vom Kalman-Filter berechneten Wertes im Vergleich mit dem verrauschten Messwert. Der Verlauf des Klemmenspannungswertes oder Ruhespannungswertes des Kalman-Filters stimmt mit dem Messwert gut überein.

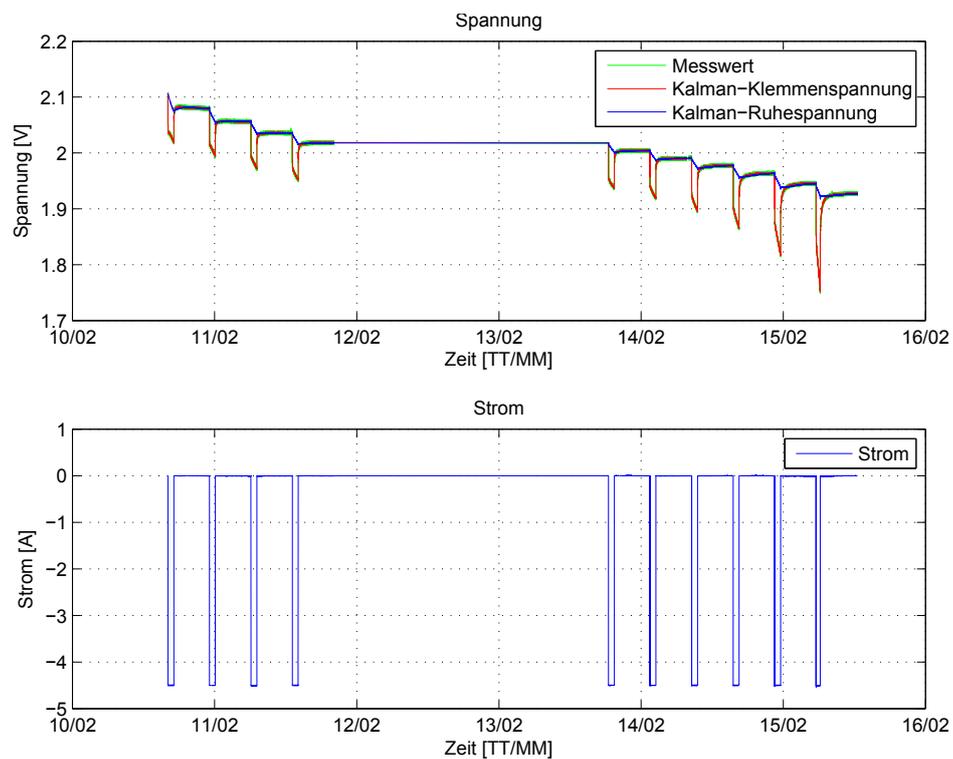


Abbildung 4.6.: Vergleich der partiellen Entladung bei 20°C

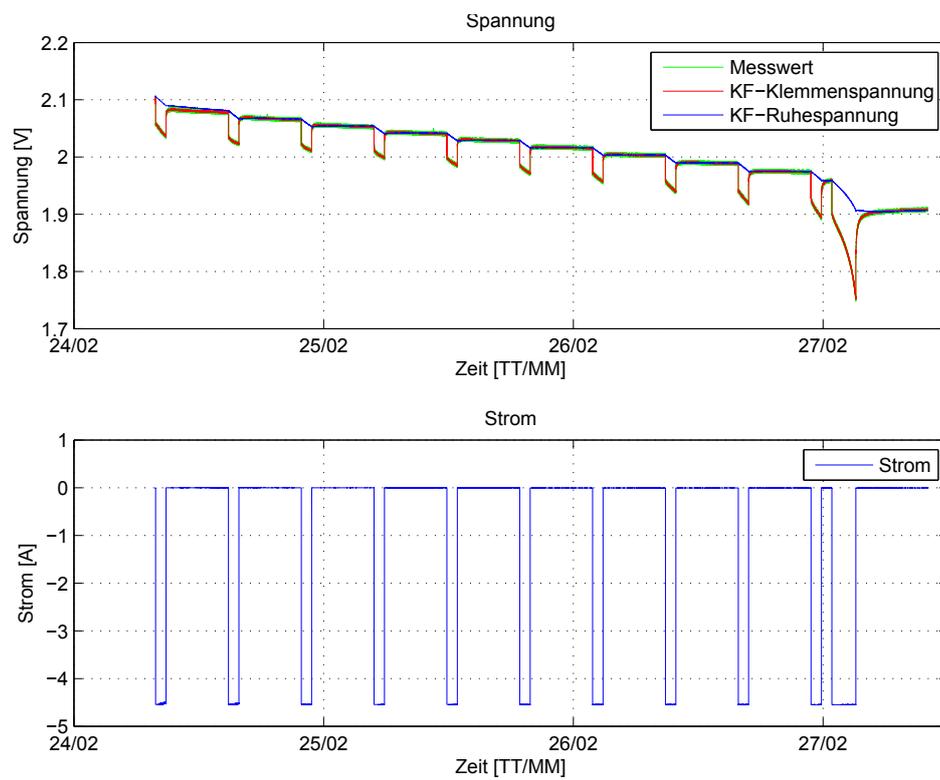


Abbildung 4.7.: Vergleich der partiellen Entladung bei 50°C

Dynamisches Verhalten der Entladung

In der Abb. 4.8 und der Abb. 4.9 wird der Verlauf des Ergebnisses des Kalman-Filters mit der verrauschten Messkurve dargestellt.

Im Vergleich mit dem Verlauf der Klemmspannung in der Abb. 3.43 und der Abb. 3.44 stellt der vom Kalman-Filter korrigierte Verlauf die Klemmspannung viel besser dar, als das Simulationsergebnis. Der wegen der Ungenauigkeit des Modells und des unbekanntes initialen Ladezustandes verursachte Fehler werden mit Hilfe der Korrektur im Kalman-Filter korrigiert. Der mit Hilfe des angepassten Modells 4 Klasse 2 nicht realisierbare Effekt der Selbstentladung wird durch die Korrektur mit dem Messwert im Kalman-Filter verwirklicht (Abb. 4.9).

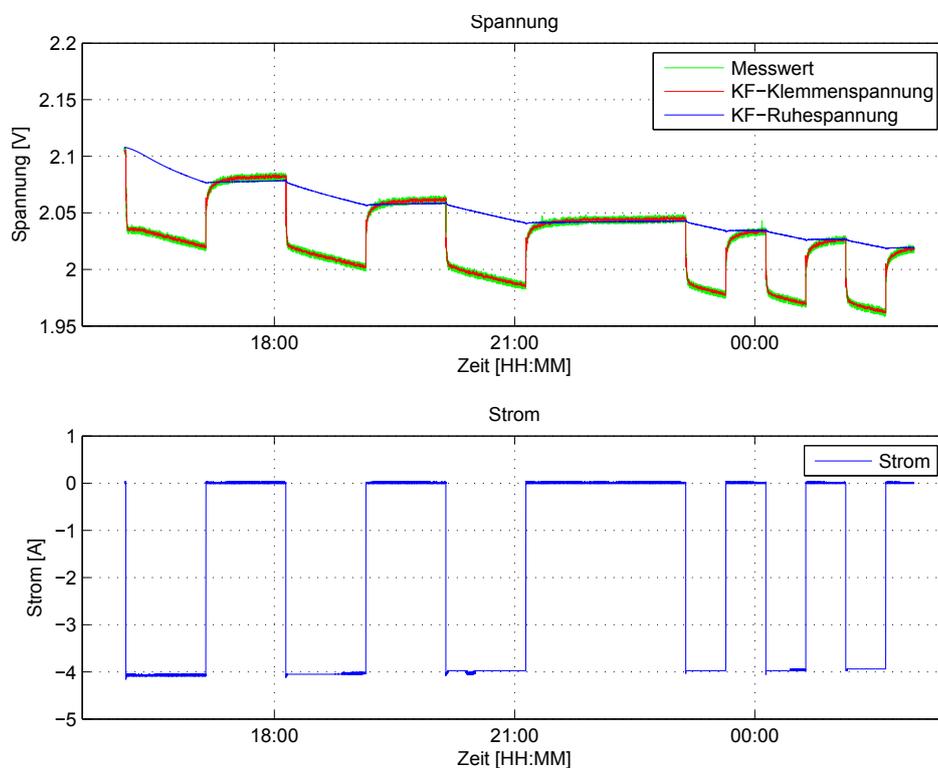


Abbildung 4.8.: Vergleich des dynamischen Verhaltens bei 20°C

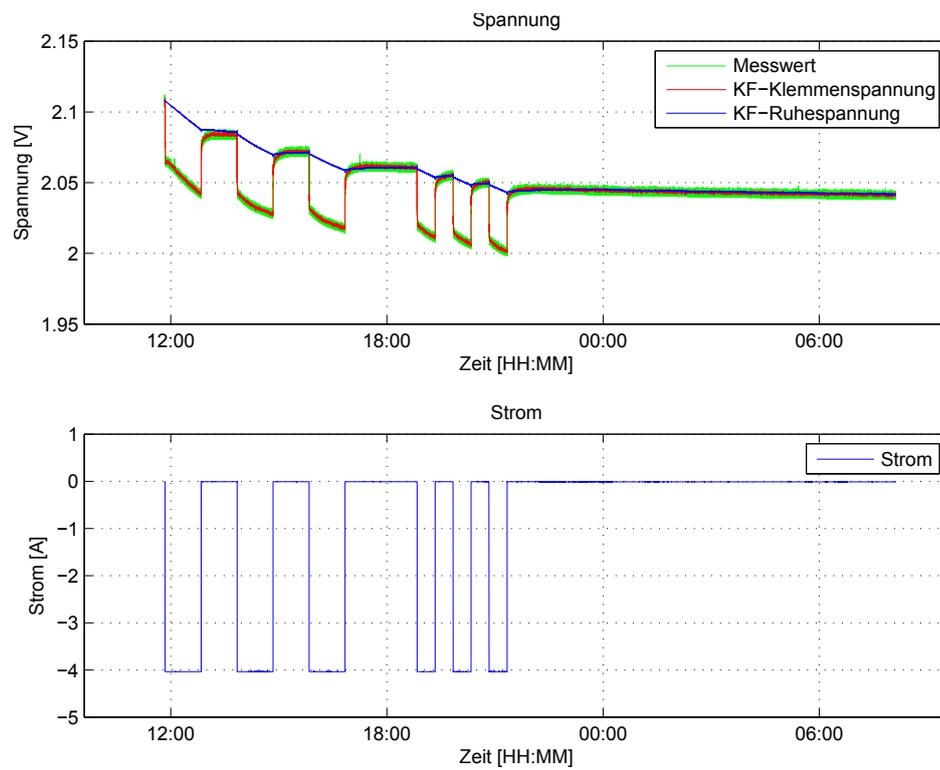


Abbildung 4.9.: Vergleich des dynamischen Verhaltens bei 50°C

4.5. Systemrauschen, Messrauschen und Iteration

Es ist in dem Abschnitt 2.1 erwähnt, dass die Genauigkeit des Kalman-Filters eng mit der Einstellung der Varianz des Systemrauschens, des Messrauschens und dem Abstand der Korrektur verbunden ist. Je größer die Varianz des Messrauschens ist, desto mehr vertraut das Kalman-Filter auf das Systemmodell. Umgekehrt, falls die Varianz des Systemrauschens einen relativen großen Wert annimmt, vertraut das Kalman-Filter mehr auf den Messwert, d.h. die Korrektur im Kalman-Filter richtet sich nach dem Messwert.

Bei jeder Iteration wird das Systemrauschen Q in der Gl. 4.15 zur Fehlerkovarianzmatrix P_k addiert. Das Systemrauschen besitzt einen kleinen Wert. Die bis zur Korrektur aufsummierte Zahl ist von dem eigenen Wert und dem Abstand der Korrektur unterschiedlich, aber größer als der originale Wert. Das Messrauschen wird nur bei jeder Korrektur zur Berechnung des Kalman-Gains in dem Nenner der Gl. 4.16 eingesetzt. Der Verlauf richtet sich bei der Korrektur nach dem Messwert, obwohl das Systemrauschen viel kleiner als das Messrauschen eingestellt wird.

$$P_k^- = A * P_{k-1} * A^T + Q \quad (4.15)$$

$$K_k = P_k^- * H^T * (H * P_k^- * H^T + R)^{-1} \quad (4.16)$$

Der Abstand der Korrektur bedeutet die Häufigkeit der Korrektur bzw. die Häufigkeit der Aufnahme des Spannungswertes, der zur Korrektur der Schätzung dient.

In dem vorliegenden Abschnitt werden die Varianz des Messrauschens und der Abstand zur Korrektur so wie die in der Realität bei der Messung mit dem Sensor eingestellt. Dabei kann das Kalman-Filter alle Verläufe gut darstellen.

Nun wird getestet, wie weit das Kalman-Filter dem Systemmodell vertrauen kann, d.h. wie selten die Korrektur benötigt wird, um den Verlauf mit begrenztem Fehler gut darzustellen.

Das Systemrauschen wird konstant eingestellt und das Messrauschen variiert von 1 bis 1000. Der Abstand der Korrektur steigt jeweils von 2 Sekunden bis 300 Sekunden.

Der durchschnittliche Fehler wird nach der Gl. 4.17 in Prozent angegeben. Häufig wird 2.10 V als die Ruhespannung der vollgeladenen Bleibatterie zelle angenommen, 1.95 V als die der vollständigen entladenen Bleibatterie zelle. d.h. der maximale Spannungsabfall zwischen beiden Ladezuständen beträgt 150 mV. Als Normierungsgröße im Nenner der Gl. 4.17 wird der maximale Spannungsabfall von 150 mV eingesetzt. Das Ziel ist der Versuch, eine passende Beziehung zwischen dem Messrauschen und dem Abstand der Korrektur zu finden, womit eine Fehlerrate von 3% erreicht wird.

$$Fehlerrate = \frac{\text{Messwert} - \text{Kalman_Filter_Wert}}{\text{Normierungsgrö}} \quad (4.17)$$

Die durchschnittliche Fehlerrate bei unterschiedlichen Einstellungen des Messrauschens und des Abstandes der Korrektur wird in den folgenden 3d-Plots dargestellt (Abb. 4.11...Abb. 4.15).

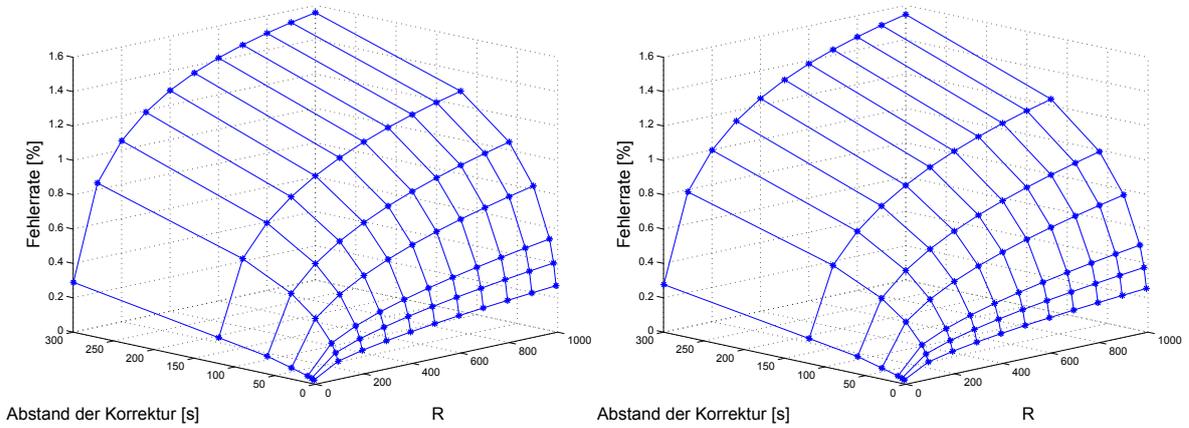


Abbildung 4.10.: Fehlerrate bei der partiellen Entladung 20°C

Abbildung 4.11.: Fehlerrate bei der partiellen Entladung 50°C

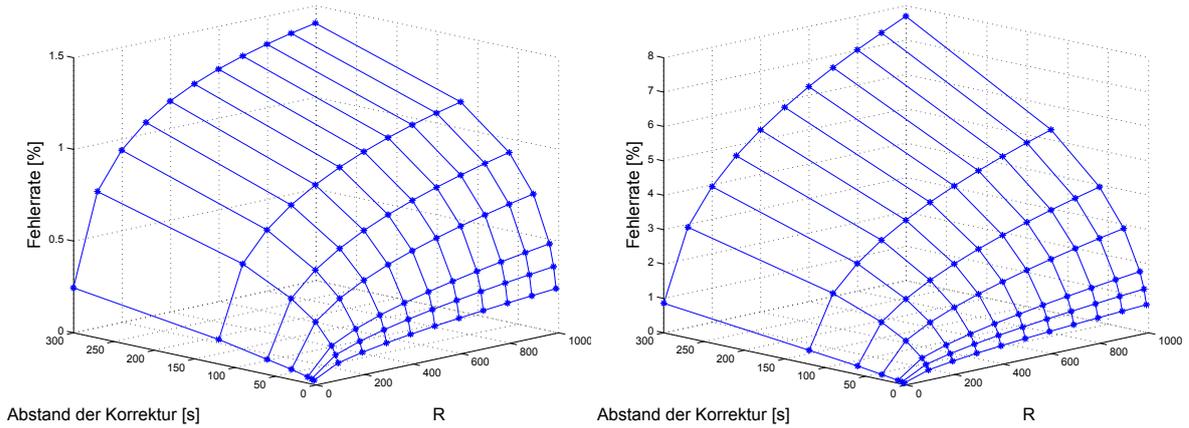


Abbildung 4.12.: Fehlerrate bei der kontinuierlichen Entladung 20°C

Abbildung 4.13.: Fehlerrate bei der kontinuierlichen Entladung 50°C

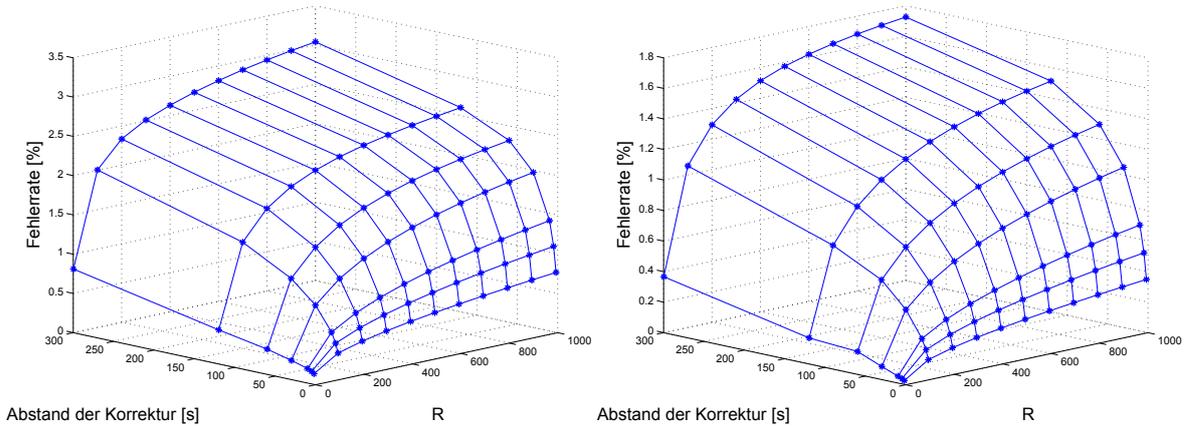


Abbildung 4.14.: Fehlerrate beim dynamischen Verhalten 20°C

Abbildung 4.15.: Fehlerrate beim dynamischen Verhalten 50°C

Mit dem Anstieg der Korrektur bzw. dem ansteigenden Messrauschen nimmt die Fehlerrate zu. Die maximale Fehlerrate beträgt bei der kontinuierlichen Entladung 50°C etwa 5% (Abstand der Korrektur 300s, Messrauschen 1000). Bei den anderen Entladungen liegt die Fehlerrate unter 3%. Wird das Messrauschen mit 100 und der Abstand der Korrektur mit 300s eingesetzt, kann das Kriterium erfüllt werden, dass die durchschnittliche Fehlerrate unter 3% des gesamten Spannungsbereichs liegt. In den folgenden Abbildungen (Abb. 4.16...Abb. 4.21) wird das zugehörige Ergebnis des Kalman-Filters dargestellt.

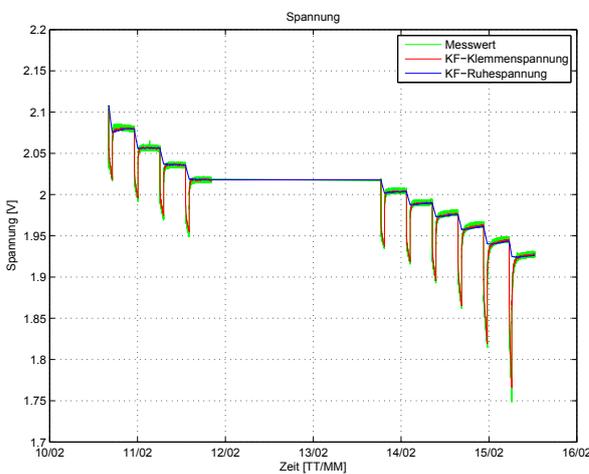


Abbildung 4.16.: Vergleich mit der verrauschten partiellen Entladung 20°C

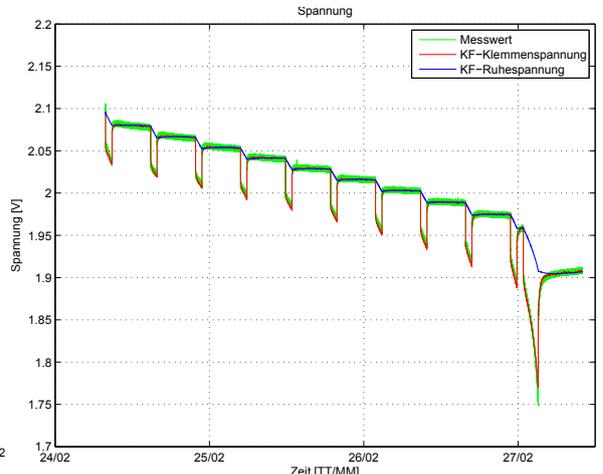


Abbildung 4.17.: Vergleich mit der verrauschten partiellen Entladung 50°C

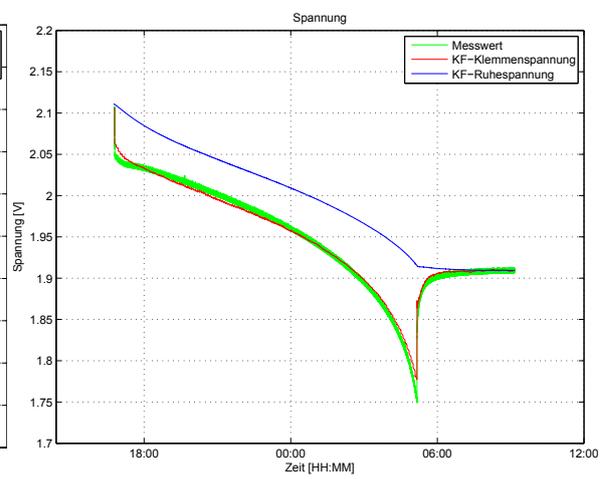
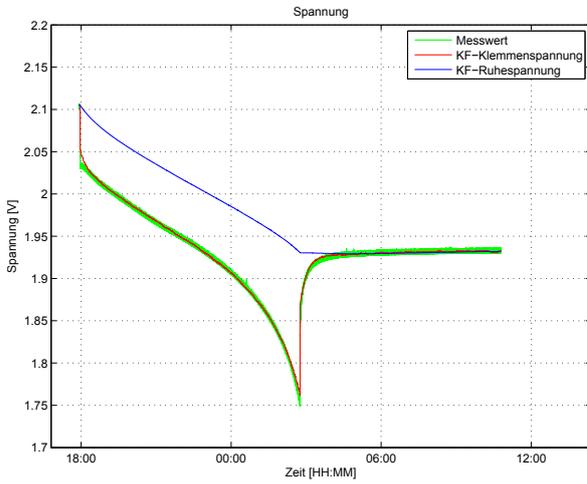


Abbildung 4.18.: Vergleich mit der verrauschten kontinuierlichen Entladung 20°C

Abbildung 4.19.: Vergleich mit der verrauschten kontinuierlichen Entladung 50°C

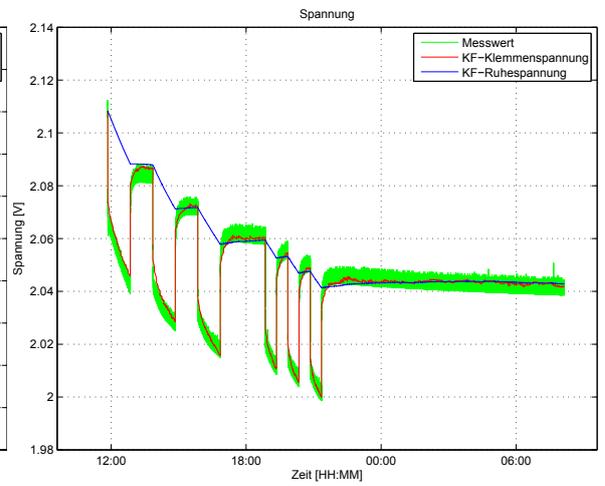
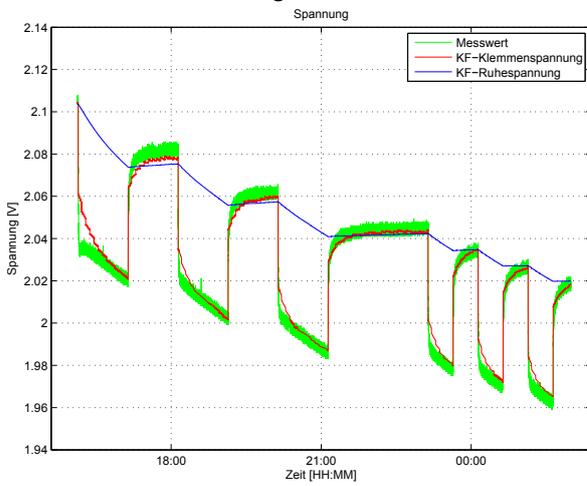


Abbildung 4.20.: Vergleich beim stark verrauschten dynamischen Verhalten 20°C

Abbildung 4.21.: Vergleich beim stark verrauschten dynamischen Verhalten 50°C

Das Kalman-Filter korrigiert alle 300 Sekunden die Schätzung des angepassten Modells 4 Klasse 2. Der Fehler ist größer als der mit der realen Einstellung, besonders bei dem dynamischen Verhalten der Entladung ist die stückweise Korrektur auffällig. Die Ungenauigkeit des Modelles verursacht diese Abweichung. Die häufigen Korrekturen kompensieren diese Ungenauigkeit.

4.6. Vergleich des Ergebnisses des Kalman-Filters mit dem stark verrauschten Messwert

In dem Abschnitt. 4.4 ist die Eingabe des Kalman-Filters leicht verrauscht. Das Kalman-Filter kann so oft wie möglich durch den Messwert die Schätzung korrigieren. Dabei ist ein sehr gutes Ergebnis gezeigt worden. Diese Untersuchung soll in diesem Abschnitt erweitert werden.

Nun wird ein künstlich erstelltes sehr starkes Rauschen mit der Normalverteilung und der Standardabweichung von „20 mV“ zu dem idealen Messwert addiert und in das Kalman-Filter eingegeben. Die Verteilung des starken Rauschens wird in der Abb. 4.22 dargestellt. Wegen des unzuverlässigen Messwertes sollte das Kalman-Filter viel mehr dem Systemmodell vertrauen. Es wird erwartet, mit Hilfe des Kalman-Filters das starke Rauschen zu dämpfen.

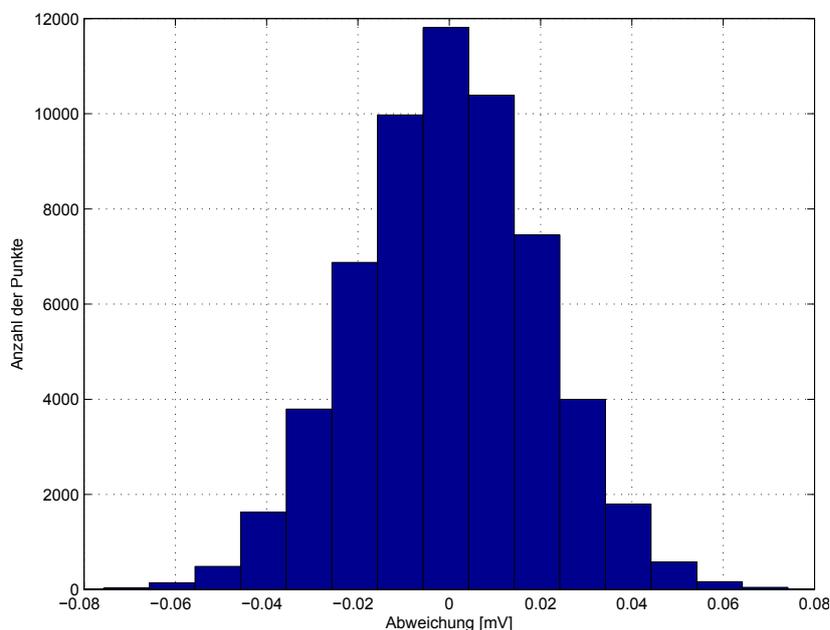


Abbildung 4.22.: Verteilung des künstlichen Rauschens

Das Ergebnis des Kalman-Filter-Modells wird mit dem verrauschten Messwert verglichen. Dabei wird der ideale Messwert ebenso in dem Plot als auch der Referenzwert dargestellt (Abb. 4.23....Abb. 4.28).

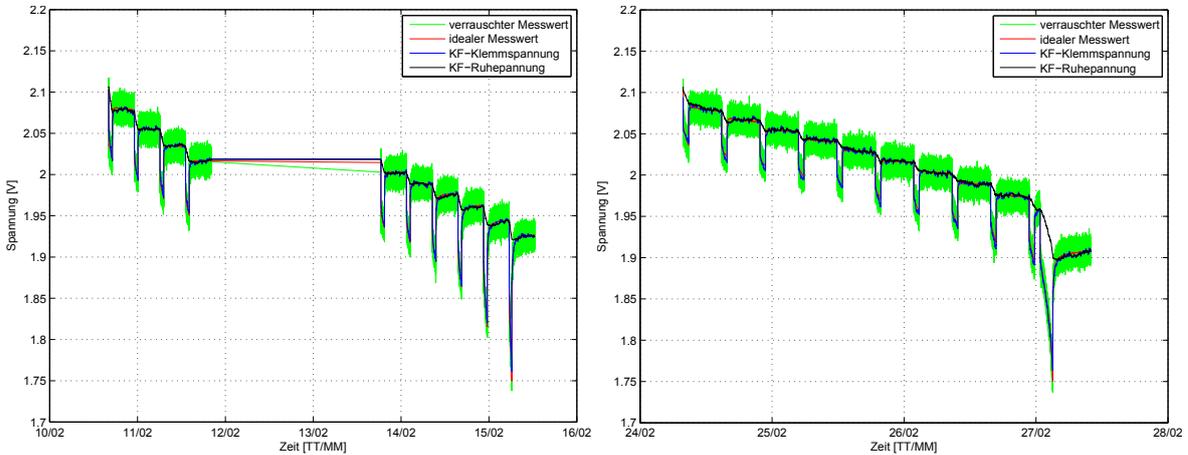


Abbildung 4.23.: Vergleich mit der stark ver-
rauschten partiellen Entla-
dung 20°C

Abbildung 4.24.: Vergleich mit der stark ver-
rauschten partiellen Entla-
dung 50°C

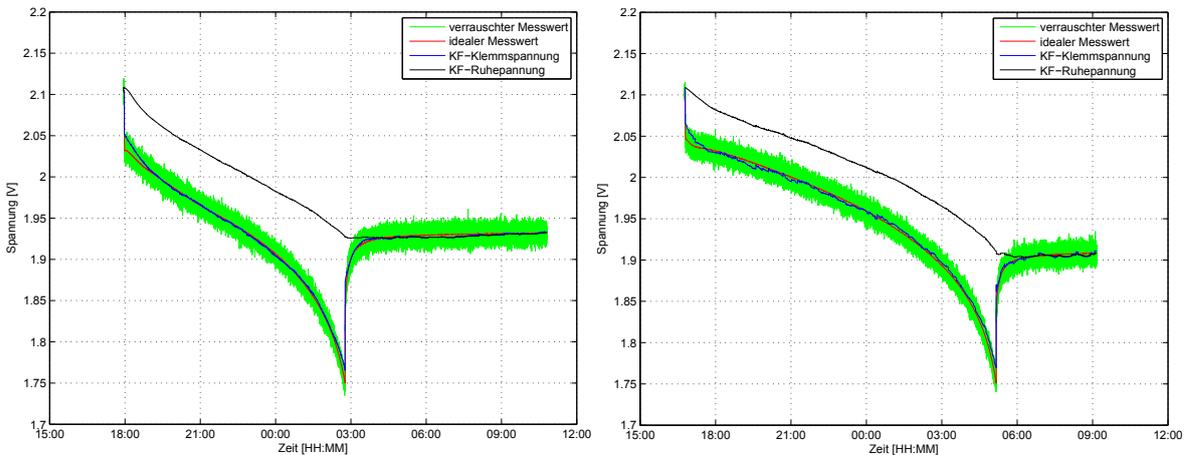


Abbildung 4.25.: Vergleich mit der stark ver-
rauschten Entladung 20°C

Abbildung 4.26.: Vergleich mit der stark ver-
rauschten Entladung 50°C

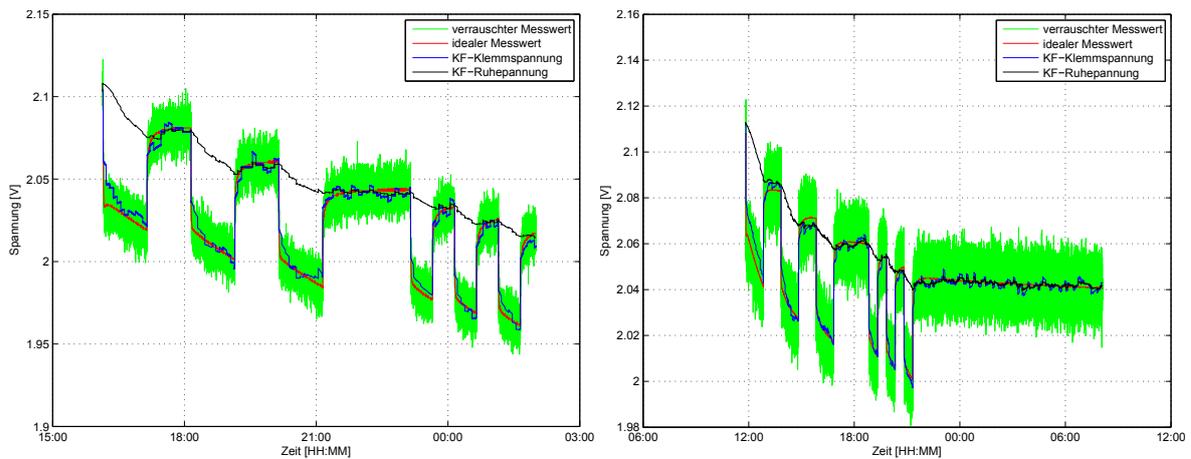


Abbildung 4.27.: Vergleich beim stark ver-
rauschten dynamischen Ver-
halten 20°C

Abbildung 4.28.: Vergleich beim stark ver-
rauschten dynamischen Ver-
halten 50°C

Nach dem Kalman-Filter ist das starke Rauschen gut gefiltert. Das in das Kalman-Filter eingesetzte Modell hilft dabei, ein vergleichsweise vernünftiges Ergebnis zu bekommen. Das Ergebnis liegt in der Nähe der idealen Messkurve. Trotzdem schwankt die Kurve des Kalman-Filters leicht, weil das Kalman-Filter bei der Korrektur immer mit dem starken Rauschen „gekämpft“ hat.

4.7. Auswertung der Korrektur mit Hilfe des Kalman-Filters

Mit Hilfe des Kalman-Filters kann das angepasste Modell 4 Klasse 2 unterschiedliche Entladungen bei 20°C und 50°C vollständig darstellen.

Wenn ein zuverlässiger Messwert mit beschränkten Fehlern zur Verfügung steht, kann das Kalman-Filter-Modell durch die Korrektur mit dem Messwert ein vollständiges und genaues Ergebnis erreichen (Abschnitt. 4.4). Die Bleibatterie ist ein komplexes System. Es ist ziemliche schwer, mit den begrenzten Elementen das System vollständig nachzubilden. Aber in dem Kalman-Filter kann durch die häufige Korrektur mit dem zuverlässigen Messwert eventuell die Ungauigkeit des eingesetzten Modells kompensiert werden. Dabei wird nicht nur der niedrigen Rechenaufwand sondern auch die hohe Genauigkeit des Ergebnisses erreicht.

Die durchschnittliche Abweichung liegt unter 3% der Normierungsgröße, d.h. 4, 5 mV. Damit wird das Kalman-Filter-Modell als geeignet bewertet.

Wenn ein stark verrauschter Messwert zur Verfügung steht, kann das Kalman-Filter-Modell trotzdem ein vergleichsweise genaues Ergebnis erhalten, vorausgesetzt, dass das im Kalman-Filter eingesetzte Modell das System gut nachbilden kann (Abschnitt. 4.6). Das in Kalman-Filter eingesetzte Modell berechnet selber das Ergebnis. Dieses Ergebnis könnte nach der Einstellung bei der Korrektur höher gewichtet sein, als der angewendete Messwert. Deswegen wird bei der Korrektur das Ergebnis eher nach der berechneten Kurve des Modells gerichtet. Dabei wird das Messrauschen gefiltert.

Das Kalman-Filter-Modell kann bei verrauschten Messwerten trotzdem sinnvolle Ergebnisse darstellen. Auf dieser Grundlage kann das angepasste Modell 4 Klasse 2 als geeignet bewertet werden, um das Verhalten zu bewerten.

Im Projekt liefert der Sensor einen Messwert mit dem Fehler innerhalb eines Bereiches von 6 mV, daher wird vorgeschlagen, dass das Kalman-Filter mehr auf den Messwert vertraut, als auf das Systemmodell. Der zuverlässige Messwert hilft dabei, die Schätzung des Modells zu verbessern. Durch den Versuch ergibt sich, dass alle 300 Sekunden ein Messwert zur Korrektur vorliegen muss. Das entspricht einer Abtastrate von 0,0033 Hz bzw. einer Datenrate von 0,04 Byte/s (aktuelles Klasse 1 Protokoll, 12 Byte, ohne Manchester-Codierung).

5. Zusammenfassung und Ausblick

Die vorliegende Arbeit beschäftigte sich mit der Modellierung einer Bleibatterie, um den Ladezustand der Bleibatterie zu bestimmen. Dabei bestand die Aufgabe darin, dass unter Laborbedingungen Messungen an der eingesetzten Batterie durchgeführt wurden. Damit wurde das Entlade- und Ladeverhalten der Bleibatterie analysiert. Desweiteren bestand die Aufgabe, ein Modell mit einem elektrischen Ersatzschaltbild zu entwickeln, wodurch unterschiedliche Messkurven so genau wie möglich dargestellt werden können. Das entwickelte Modell wurde im Kalman-Filter implementiert. Das Kalman-Filter korrigiert die Schätzung des Systemsmodelles. Weiterhin wurde die Genauigkeit des Kalman-Filter-Modells untersucht.

Die Aufgabenstellung der Masterarbeit wurde erfolgreich umgesetzt. Die Arbeit gliedert sich in theoretische und experimentelle Teilaufgaben.

Für die Untersuchungen wurde ein Messplatz aufgebaut, um das Verhalten der Batterie unter unterschiedlichen Lastprofilen zu ermitteln. Die Messungen wurden in Form der Konstantstromentladung bzw. dem IU-Ladeverfahren bei unterschiedlichen Strömen und unterschiedlichen Temperaturen durchgeführt. Die Klemmenspannung, Temperatur und Stromwert sollten zur weiteren Untersuchung des Modells zur Verfügung stehen.

Im theoretischen Teil wurde nach der umfangreichen Recherche ein Modell als Grundmodell und Ausgangspunkt der Implementation genutzt. Zur präzisen Darstellung des dynamischen Verhaltens der Batterie wird das Grundmodell durch Hinzufügen zusätzlicher Elemente erweitert. Die benötigten Parameter des Modells werden mit Hilfe des Least-Square-Optimierungsverfahrens durch die Annäherung der Messkurve bestimmt. Das angepasste Modell 4 Klasse 1 eignet sich gut zur Darstellung der kurzzeitigen Entladung und kann die Messkurve vollständig darstellen. Weiterhin ist bekannt, dass das Verhalten der Batterie stark von der Temperatur und dem Ladezustand abhängt. Der Temperatur- und akute Ladezustandseinfluss werden in einer Formel zusammengefasst. Statt eines Kondensators wird eine von dieser Formel gesteuerte Spannungsquelle als Hauptenergiespeicher im Modell eingesetzt. Der von der Temperatur und dem Ladezustand abhängige Innenwiderstand wird ebenso durch eine Formel zusammengefasst. Das angepasste Modell 4 Klasse 2 kann damit die erwarteten Messwerte bei unterschiedlichen Temperaturen mit den gleichen Parametern darstellen.

Als nächster Schritt wird das angepasste Modell 4 Klasse 2 im Kalman-Filter eingesetzt. Das Kalman-Filter hilft dabei, das Modell zu korrigieren. Dabei wird das von dem Messgerät verursachte Rauschen und die Ungenauigkeit des Modells kompensiert.

Im Kalman-Filter wurde die Abweichung zwischen der von dem Modell berechneten Klemmenspannung und dem Messwert ermittelt. Dabei wurde das Kalman-Gain berechnet. Mit Hilfe des Kalman-Gains und der aktuell berechneten Fehlervarianz wurde die nicht messbare, aber vom dem Modell berechnete Spannung an dem Hauptenergiespeicher und dem Nebenenergiespeicher kalibriert. Die Spannung an dem Hauptenergiespeicher in dem Modell spiegelt direkt die Ruhespannung wieder. Wegen der linearen Beziehung zwischen dem Ladezustand und der Ruhespannung erfolgt die Ladezustandsbestimmung über die Bestimmung der Ruhespannung der Bleibatterie.

Die angepasste Formel der Ruhespannung eignet sich zur Darstellung des nichtlinearen Verlaufs. Nach der Kalibrierung durch das Kalman-Filter wird eine zuverlässige Aussage des Ladezustandes getroffen. Das Kalman-Filter-Modell benötigt maximal jede 300 Sekunde einen Messwert, der zur Korrektur dient. Dabei beträgt der durchschnittliche Fehler weniger als 4,5 mV. Das Kalman-Filter-Modell erweist sich als geeignet.

Die wichtigsten, durch die Arbeit gesammelte Erfahrungen sind:

- Ein gutes Modell entsteht nicht mal eben so. Aus diesem Grunde ist der Modellierer sich im Klaren, was für ein Ziel das zu entwickelnde Modell unbedingt erreichen sollte.
- Eine weitere Herausforderung bildet die Schätzung der Genauigkeit, d.h. wie genau ein Modell sein muss. Eine zu hohe Genauigkeit steigert die zur Verfügung stehenden Rechnerressourcen, eine zu geringe Genauigkeit führt nur zu einem ungenügenden und fehlerhaften Modell.
- Ein weitere Frage besteht darin, welche Messungen zur Darstellung des Systems sinnvoll sind, welche Simulationsumgebungen für das Modell zielführend sind. Modellierungen sind immer eng mit der Simulation und Messung verbunden. Anhand des Simulationsergebnisses wird das Modell bewertet und möglicherweise weiter entwickelt. Das neue angepasste Modell sollte durch weitere Simulationen bewertet werden. Damit wird ein zuverlässigeres Modell mit einem eingeschränkten Fehler entwickelt.
- Eine andere Schwierigkeit liegt in der Parameterbestimmung, wie eine Menge von den Parametern optimal approximiert werden kann. In der Arbeit wurde eine Fehlerfunktion formuliert. Durch die Optimierungsaufgabe werden die passenden Parameter gefunden, mit denen der minimale Fehler erreicht ist. Die Güte der Funktionsapproximation ergibt eine Aussage über die Gültigkeit der Parametern.

- Außerdem ist es manchmal schwierig, ein tatsächliches System (z.B. eine Bleibatterie) direkt durch ein Modell komplett und fehlerfrei nachzubilden. Deshalb sollte über Verfahren bzw. Methoden nachgedacht werden, mit denen der von der Ungenauigkeit des Modells verursachten Fehler gedämpft oder idealerweise vollständig entfernt werden kann. Das in dieser Arbeit implementierte Kalman-Filter ist dazu eine gute Auswahl. Das Kalman-Filter hilft, die Schätzung des Modells zu korrigieren und das Messrauschen zu dämpfen. Bei der Einstellung des Kalman-Filters wird immer die Beziehung zwischen dem Messrauschen, Systemrauschen und der Abstand der Korrektur getestet, um eine effiziente Kombination dazwischen zu erreichen. Damit ergibt das Kalman-Filter-Modell eine niedrige Korrekturrate mit definiert begrenztem Fehler.

Nach sehr vielen Details soll nun am Schluss der Arbeit das Zitat von Albert Einstein gestellt werden: „Das Problem zu erkennen ist wichtiger, als die Lösung zu erkennen, denn die genaue Darstellung des Problems führt zur Lösung.“

Die erzielten Erkenntnisse als Teilbeiträge für das Forschungsprojekt „BATSEN“ sind:

- Ein Messplatz wurde erfolgreich aufgebaut. Die Schaltzeit der Ladung bzw. Entladung wurde durch eine Relais-Karte gesteuert.
- Das Entladeverhalten der Bleibatterie bei unterschiedlichen Temperaturen wurde zusammengefasst.
- Das angepasste Modell 4 Klasse 1 eignet sich zur Darstellung des dynamischen Verhaltens der Bleibatterie.
- Das angepasste Modell 4 Klasse 2 stellt nicht nur den Verlauf der kurzzeitigen Entladung dar, sondern auch den der langzeitigen Entladung.
- Das auf das angepasste Modell 4 Klasse 2 basierte Kalman-Filter-Modell kann den Verlauf unterschiedlicher Entladung vollständig darstellen. Dabei kann es den von der Ungenauigkeit des Modells verursachten Fehler und das von dem Messgerät verursachten Rauschen dämpfen.
- Das Kalman-Filter-Modell kann eine zuverlässige Aussage zur Bestimmung des Ladezustandes treffen. Im Vergleich mit dem Coloumb-Counting-Verfahren kann das Kalman-Filter-Modell den durch das Integral des Messrauschens aufsummierten Fehler korrigieren. Außerdem bestimmt das Kalman-Filter-Modell einen nichtlinearen Verlauf des Ladezustandes, genau wie den in der Realität.
- Der in dem Forschungsprojekt „BATSEN“ eingesetzte Sensor kann in der Zukunft mit einer niedrigen Datenrate im Sekundenbereich betrieben werden, weil das Kalman-Filter-Modell die Fähigkeit zur zuverlässigen Schätzung der Klemmenspannung bzw. Ruhespannung besitzt.

Die vorliegende Arbeit basiert im Wesentlichen auf der Theorie und Simulationen. Experimente waren nur als geringer Umfang in dieser Arbeit vorgesehen. Sie sollten als Beispieldienste und Parameter liefern. In der Zukunft sollten umfangreiche realitätsnahe Experimente mit dem vorhandenen Kalman-Filter-Modell durchgeführt werden, d.h. das Kalman-Filter-Modell sollte in sehr vielen Betriebssimulationen unter realen Bedingungen betrieben werden. Außerdem ist es sinnvoll, anhand des gegenwärtigen Modells der SOH (State of Health) der Batterie zu untersuchen, da SOH immer eine wichtigere Rolle für die Batterie spielt. Weitere Untersuchungen auf diesem Gebiet sind folglich empfohlen.

Literaturverzeichnis

- [1] <http://de.wikipedia.org>.
- [2] Robyn A.Jackey. A Simple, Effective Lead-Acid Battery Modeling Process for Electrical System Component Selection.
- [3] Anne Angermann, Michael Beuschel, Martin Rau, and Ulrich Wohlfarth. *Matlab - Simulink - Stateflow - Grundlagen - Toolboxen - Beispiele*. Oldenbourg Verlag, 2011.
- [4] Stefano Barsail and Massimo Ceraolo. Dynamical Models of Lead-Acid Batteries: Implementation Issues. *IEEE TRANSACTIONS ON ENERGY CONVERSION*, 2002.
- [5] Oliver Sven Bohlen. *Impedence-based battery monitoring*. Dissertation, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2008.
- [6] Robert Grover Brown and Patrick Y.C HWANG. *INTRODUCTION TO RANDOM SIGNALS AND APPLIED KALMAN FILTERING with Matlab exercises and solutions*. JOHN WILEY & SONS, 1977.
- [7] B.S.Bhangu. Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid - Electric Vehicles. *IEEE*, 2005.
- [8] John Chiasson and Baskar Vairamohan. Estimating the State of Charge of a Battery. *IEEE Transactions on control systems technology*, 2005.
- [9] Ing.J.C.F.Van der Meer. Technischer Lehrgang Starterbatterien. Braunschweig Wiesbaden.
- [10] Matthias Dürr, Andrew Cruden, Sinclair Gair, and J.R. McDonald. Dynamic model of a lead acid battery for use in a domestic fuel cell system. *Journal of Power Sources*, 2006.
- [11] Lijun Gao, Shengyi Liu, and Roger A.Dougal. Dynamic Lithium-Ion Battery Model for System Simulation. *IEEE TRANSACTIONS ON COMPONENTS AND PACKAGING TECHNOLOGIES*, 2002.
- [12] H.A.Kiehne. *BATTERY TECHNOLOGY HANDBOOK*. MARCEL DEKKER,INC, 2003.

- [13] Jaehyun Han, Dongchul Kim, and Myoungcho Sunwoo. State-of-charge estimation of lead-acid batteries using an adaptive extended kalman filter. *Journal of Power Sources*, 2009.
- [14] Hongwen He, Rui Xiong, and Xiaowei Zhang. State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 2011.
- [15] Schlitt Herbert. *Systemtheorie für stochastische Prozesse* -. Springer, Berlin, Heidelberg, 1992.
- [16] Valerie H.Johnson, Ahmad A.Pesaran, and Thomas Sack. Temperature-Dependent Battery Models for High-Power Lithium-Ion Batteries. *17th Electric Vehicle Symposium*, 2000.
- [17] J.Appelbaum. an electrical model of the lead - acid battery. *IEEE*, 1982.
- [18] Helmut Kopka. *Latex eine Einführung*. ADDISON-WESLEY PUBLISHING COMPANY, 1989.
- [19] Julia Kowal. *Spatially-resolved impedance of nonlinear inhomogeneous devices -using the example of lead-acid batteries-*. Dissertation, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2010.
- [20] K.W.E.Cheng, B.P.Divakar, Hongjie Wu, Kai Ding, and Ho Fai Ho. Battery-management System (BMS) and SOC Development for Electrical Vehicles. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 2011.
- [21] J.F. Araujo Leao, L.V.Hartmann, and M.B.R:Correa. Lead-Acid Battery Modeling and State of Charge monitoring. *IEEE*, 2010.
- [22] Jaemoon Lee, Oanyong Nam, and H.H.Cho. Li-ion battery SOC estimation method based on the reduced order extended Kalman filtering. *Journal of Power Sources*, 2007.
- [23] Seongjun Lee, Jonghoon Kim, and Jaemoon Lee. State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge. *Journal of Power Sources*, 2008.
- [24] David Linden. *Handbook of Batteries*. McGraw-Hill, United States of America, 2001.
- [25] Gregory L.Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part1. Background. *Journal of Power Sources*, 2004.
- [26] Gregory L.Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part2. Modelung and identification. *Journal of Power Sources*, 2004.

- [27] Gregory L.Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part3. State and parameter estimation. *Journal of Power Sources*, 2004.
- [28] Nazih Moubayed, Janine Kouta, and Ali El-Ali. PARAMETER IDENTIFICATION OF THE LEAD-ACID BATTERY MODEL. *IEEE*, 2008.
- [29] O.Erdinc, B.Vural, and M.Uzunoglu. A dynamic lithium-ion battery model considering the effects of temperature and capacity fading. *IEEE*, 2009.
- [30] Panasonic. Valve regulated Lead-Acid Batteries – Technical Handbook. Braunschweig Wiesbaden.
- [31] Sabine Piller, Marion Perrin, and Andreas Jossen. Methods for state-of-charge determination and their applications. *Journal of Power Sources*, 2001.
- [32] Konrad Reif. *Batterien, Bordnetze und Vernetzung*. Vieweg + Teuber Verlag | Springer Fachmedien Wiesbaden GmbH, 2010.
- [33] Prof. Dr.-Ing. Karl-Ragmar Riemschneider. Drahtlose Zellen Sensoren für Fahrzeugbatterie BATSEN. Förderung beim BMVF.
- [34] Püttjer Simon. Diagnosefunktionen für Automobil-Starterbatterie mit drahtlosen Zellen-sensoren, 2011.
- [35] Marc Thele. *A contribution to the modeling of the charge acceptance of lead-acid batteries -using frequency and time domain based concepts*. Dissertation, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2008.
- [36] Battery University. <http://batteryuniversity.com/index-german.htm>.
- [37] A. Vasebi, S.M.T. Bathaee, and M. Partovibakhsh. Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended Kalman filter. *Journal of Power Sources*, 2007.
- [38] Amir Vasebi, Maral Partovibakhsh, and S.Mohammad Taghi Bathaee. A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications. *Journal of Power Sources*, 2007.
- [39] XueZhe Wei, XiaoPeng Zhao, and YongJun Yuan. Study of Equivalent Circuit Model for Lead-acid Batteries in Electric Vehicle. *International Conference on Measuring Technology and Mechatronics Automation*, 2009.
- [40] Dr. Heinz Wenzel. Beratung für batterien und energietechnik.
- [41] Fei Zhang, Guangjun Liu, and Lijin Fang. A Battery State of Charge Estimation Method with Extended Kalman Filter. *IEEE*, 2008.

A. Modell

Hier wird die Ersatzschaltbild der Modelle, die in den Rahmen dieser Arbeit untersucht werden, bezeichnet.

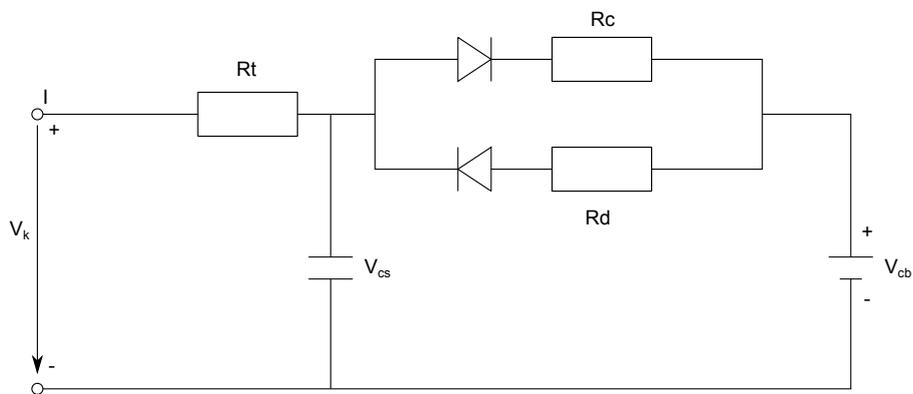


Abbildung A.1.: Modell A1

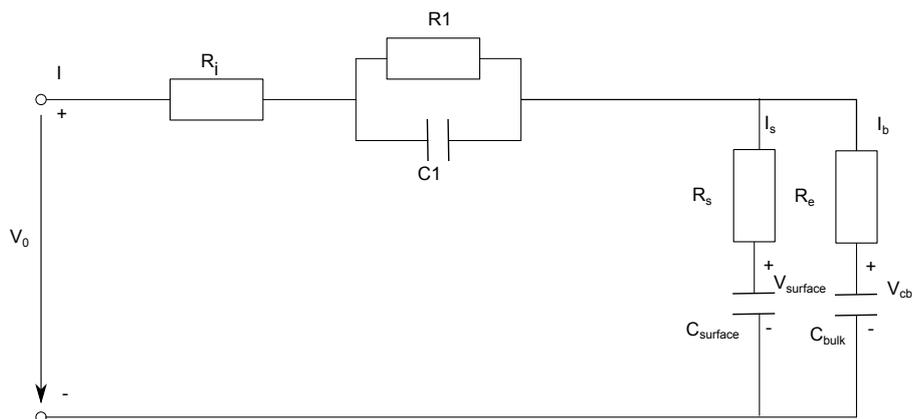


Abbildung A.2.: Modell A2

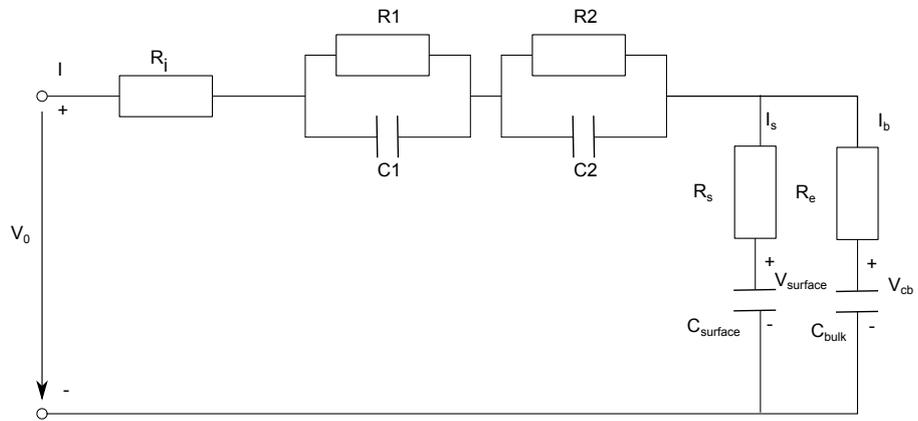


Abbildung A.3.: Modell A3

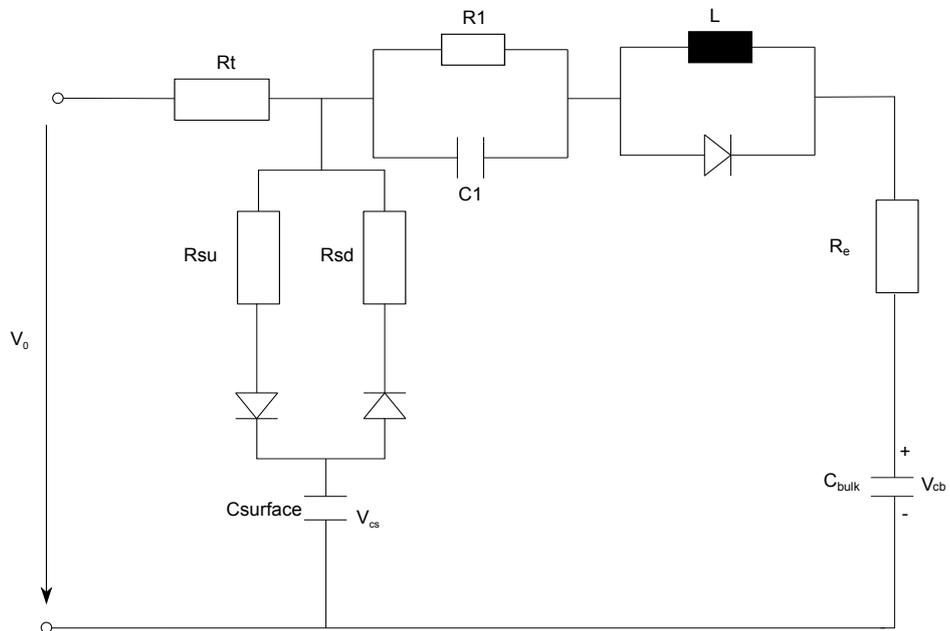


Abbildung A.4.: Modell A4

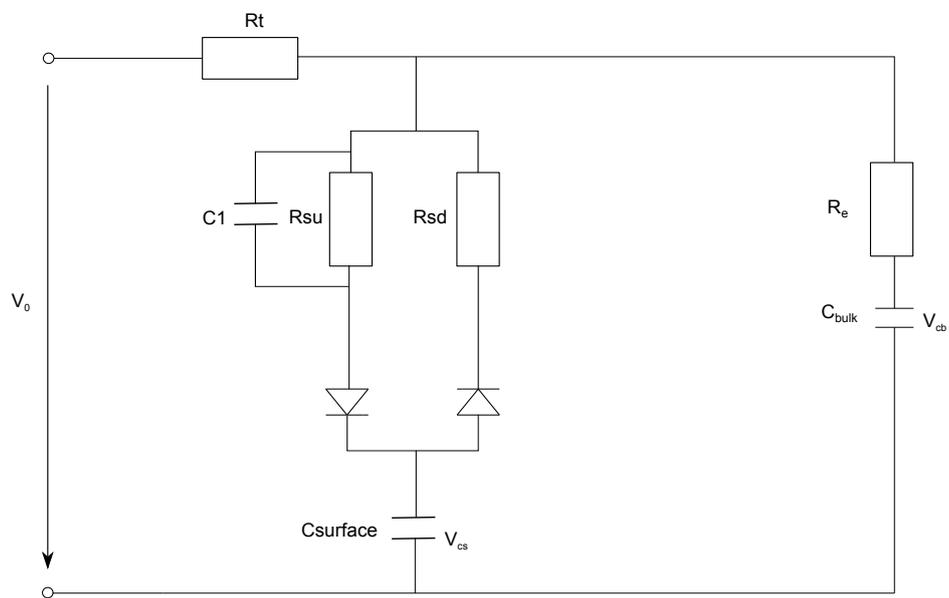


Abbildung A.5.: Modell A5

B. Säuredichtemessung

Die Säuredichte der Bleibatterie wird durch das Refraktormeter und Hydrometer gemessen. Dabei wird gleichzeitig die Klemmspannung durch das Tischmultimeter „Fluke 45“ gemessen. Das Ergebnis lässt in den folgenden Tabelle veranschaulichen.

	Fluke 45	Refraktormeter(Extech)	Hydrometer
Zelle 1	1,9705 V	1,12 kg/l	1,14 kg/l
Zelle 2	1,9441 V	<1,10 kg/l	1,11 kg/l
Zelle 3	1,9298 V	<1,10 kg/l	<1,10 kg/l
Zelle 4	1,9936 V	1,1148 kg/l	1,15 kg/l
Zelle 5	2,1197 V	1,158 kg/l	1,17 kg/l
Zelle 6	1,9881 V	1,142 kg/l	1,16 kg/l

Tabelle B.1.: Messung der Ruhespannung bzw. des Ladezustandes 1

	Fluke 45	Refraktormeter(Extech)	Hydrometer
Zelle 1	1,9678 V	1,11 kg/l	1,13 kg/l
Zelle 2	1,9423 V	<1,10 kg/l	1,10 kg/l
Zelle 3	1,9283 V	<1,10 kg/l	<1,10 kg/l
Zelle 4	1,9909 V	1,14 kg/l	1,15 kg/l
Zelle 5	2,1277 V	1,17 kg/l	1,19 kg/l
Zelle 6	1,9858 V	1,14 kg/l	1,15 kg/l

Tabelle B.2.: Messung der Ruhespannung bzw. des Ladezustandes 2

	Fluke 45	Refraktormeter(Extech)	Hydrometer
Zelle 1	1,9683 V	1,112 kg/l	1,13 kg/l
Zelle 2	1,9426 V	<1,10 kg/l	1,10 kg/l
Zelle 3	1,9285 V	<1,10 kg/l	<1,10 kg/l
Zelle 4	1,9911 V	1,145 kg/l	1,15 kg/l
Zelle 5	2,0186 V	1,155 kg/l	1,16 kg/l
Zelle 6	1,9682 V	1,14 kg/l	1,15 kg/l

Tabelle B.3.: Messung der Ruhespannung bzw. des Ladezustandes 3

C. Quellcode

C.1. kontinuierliche Entladung bei 3 Temperatur

Listing C.1: kontinuierliche Entladung bei 3 Temperatur

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Vergleich der Klemmspannung bei der kontinuierlichen Entladung
3 % In diesem Code wird die kontinuierliche Entladung
4 % bei 3 Temperatur verglichen und dargestellt
5
6 clear all;
7 % -2°C messwert
8 load('C5_discharge_m2deg_5A_1V75_all.mat')
9
10 % Datenverarbeitung Intepolation
11 % -2°C
12 time_m2deg = (comb_cur_data(:,1) - comb_cur_data(1,1)) * 24 * 3600;
13 duration_m2deg = round(time_m2deg(end)) - 1;
14 lange = 1:1:duration_m2deg;
15 ccd_m2deg = interp1(time_m2deg, comb_cur_data(:,2), lange);
16 cvd_m2deg = interp1(time_m2deg, comb_volt_data(:,2), lange);
17 ctd_m2deg = interp1(time_m2deg, comb_temp_data(:,2), lange);
18 cchd_m2deg = interp1(time_m2deg, comb_charge_data, lange);
19 time_m2deg = interp1(time_m2deg, comb_cur_data(:,1), lange);
20 t_m2=time_m2deg; % -time(1);
21
22 % 50°C Messwert
23 load('C5_discharge_50deg_5A_all.mat')
24 time_50deg = (comb_cur_data(:,1) - comb_cur_data(1,1)) * 24 * 3600;
25 duration_50deg = round(time_50deg(end)) - 1;
26 %lange = 1:1:duration_50deg;
27 ccd_50deg = interp1(time_50deg, comb_cur_data(:,2), lange);
28 cvd_50deg = interp1(time_50deg, comb_volt_data(:,2), lange);
29 ctd_50deg = interp1(time_50deg, comb_temp_data(:,2), lange);
30 cchd_50deg = interp1(time_50deg, comb_charge_data, lange);
31 time_50deg = interp1(time_50deg, comb_cur_data(:,1), lange);
```

```
32 t_50=time_50deg;%-time(1);
33
34 % 20°C Messwert
35 load('C5_discharge_5A_1V75_all.mat')
36 time_21deg = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
37 duration_21deg = round(time_21deg(end))-1;
38 %lange = 1:1:duration_21deg;
39 charge_21deg = zeros(1,length(comb_cur_data(:,1)));
40 charge_21deg(1)=0;
41 for k =2:length(comb_cur_data(:,1))
42     charge_21deg(k) = charge_21deg(k-1) + comb_cur_data(k,2)...
43     * (comb_cur_data(k,1) - comb_cur_data(k-1,1))*24;
44 end
45
46 ccd_21deg = interp1(time_21deg,comb_cur_data(:,2),lange);
47 cvd_21deg = interp1(time_21deg,comb_volt_data(:,2),lange);
48 ctd_21deg = interp1(time_21deg,comb_temp_data(:,2),lange);
49 cchd_21deg = interp1(time_21deg,charge_21deg,lange);
50 time_21deg = interp1(time_21deg,comb_cur_data(:,1),lange);
51
52 t=time_21deg;
53 %% Plot
54 figure(1)
55 subplot(2,1,1)
56 % Klemmenspannung
57 plot(t,cvd_50deg,t,cvd_21deg,t,cvd_m2deg)
58 datetick('x',15)
59 xlim([t(1),t(end)])
60 % title({'\fontsize{14}Vergleich der Klemmenspannung...
61 % bei 3 Temperaturen';'Messwert: Entladen...
62 % mit konstanten Strom (5A) von 2.1V bis 1.75V ';..
63 %;\fontsize{10}Klemmenspannung'});
64 title('\fontsize{10}Klemmenspannung')
65 xlabel('Zeit [HH:MM]');
66 ylabel('Spannung [V]')
67 legend('50°C','21°C','-2°C')
68 grid
69
70 % Strom
71 subplot(2,1,2)
72 plot(t,ccd_50deg,t,ccd_21deg,t,ccd_m2deg)
73 legend('50°C','21°C','-2°C','Location','SouthEast');
74 datetick('x',15)
75 xlim([t(1),t(end)])
```

```
76 title('Strom');
77 xlabel('Zeit [HH:MM]');
78 ylabel('Strom [A]');
79 grid
80 saveas(gcf,'1','pdf')
81 figure
82 % subplot(4,1,3)
83 % Ladung
84 subplot(2,1,1)
85 plot(t,cchd_50deg,t,cchd_21deg,t,cchd_m2deg)
86 legend('50°C','21°C','-2°C');
87 datetick('x',15)
88 xlim([t(1),t(end)])
89 title('Ladung');
90 xlabel('Zeit [HH:MM]');
91 ylabel('Ladung [Ah]');
92 grid
93
94 delta_temp_21 = ctd_50deg(1)-ctd_21deg(1);
95 delta_temp_m2 = ctd_50deg(1)-ctd_m2deg(1);
96
97 % Temperatur
98 subplot(2,1,2)
99 % subplot(4,1,4)
100 plot(t,ctd_50deg-50,t,ctd_21deg-21.5,t,ctd_m2deg+2.1)
101 legend('50°C','21°C','-2°C');
102 datetick('x',15)
103 xlim([t(1),t(end)])
104 title('Abweichung der Temperatur von dem Sollwert...
105 (0=Sollwert: 50°C, 21°C und -2°C)');
106 xlabel('Zeit [HH:MM]');
107 ylabel('Abweichung [°C]');
108 grid
109 % saveas(gcf,'Vergleich beim Entladen_3T','pdf')
110 saveas(gcf,'2','pdf')
111
112 %% Endwert vergleich
113 C_21deg = 0.92e6;
114 C_50deg = 1.132e6;
115 C_m2deg = 0.7728e6;
116
117 cvd_21deg_C =zeros(1,lange);
118 cvd_50deg_C =zeros(1,lange);
119 cvd_m2deg_C =zeros(1,lange);
```

```

120
121 cvd_21deg_C(1)=cvd_21deg(1);
122 cvd_50deg_C(1)=cvd_50deg(1);
123 cvd_m2deg_C(1)=cvd_m2deg(1);
124
125
126 for k=2:length(lange)
127     cvd_21deg_C(k) = cvd_21deg(1) +cchd_21deg(k)*3600/C_21deg;
128     cvd_50deg_C(k) = cvd_50deg(1) +cchd_50deg(k)*3600/C_50deg;
129     cvd_m2deg_C(k) = cvd_m2deg(1) +cchd_m2deg(k)*3600/C_m2deg;
130 end
131
132 % figure(3)
133 % subplot(2,1,1)
134 % plot(t,cvd_50deg,t,cvd_50deg_C)
135 % datetick('x',15);
136 % legend('50deg')
137 % subplot(2,1,2)
138 % plot(t,cvd_21deg,t,cvd_21deg_C)
139 % legend('21deg')
140 % datetick('x',15);
141 %%
142 %% Plot
143 figure(4)
144 subplot(2,1,1)
145 plot(t,cvd_50deg,t,cvd_50deg_C,t,cvd_21deg,...
146 t,cvd_21deg_C,t,cvd_m2deg,t,cvd_m2deg_C)
147 datetick('x',15);
148 legend('Voltage.50deg','Voltage.Kondensator(1.132e6F).50deg',...
149 'Voltage.21deg','Voltage.Kondensator(0.92e6F).21deg',...
150 'Voltage.m2deg','Voltage.Kondensator(0.7728e6F).m2deg')
151 title({'\fontsize{14} Comparision with Terminal Voltage...
152 in Ambient and Thermostat';'Discharge with constant last...
153 (5A) from 2.1V to 1.75V';'\fontsize{10}Terminal Voltage'});
154 xlabel('Time in [HH:MM]');
155 ylabel('Voltage in [V]');
156 subplot(2,1,2)
157 plot(t,ccd_50deg,'b',t,ccd_21deg,'r',t,ccd_m2deg,'m')
158 legend('Current.50deg','Current.21deg','Current.m2deg')
159 title('Current');
160 xlabel('Time in [HH:MM]');
161 ylabel('Current in [A]');
162 %%
163 soc50deg = (cchd_50deg(end) - cchd_50deg(2.966e4:4.966e4))./...

```

```

164 cchd_50deg(end);
165 soc21deg = (cchd_21deg(end) - cchd_21deg(1.682e4:3.682e4))./...
166 cchd_21deg(end);
167 socm2deg = (cchd_m2deg(end) - cchd_m2deg(0.8361e4:2.8361e4))./...
168 cchd_m2deg(end);
169
170 charge1_50deg = (cchd_50deg(4.966e4)-cchd_50deg(2.966e4))*-1;
171 charge1_21deg = (cchd_21deg(3.682e4)- cchd_21deg(1.682e4))*-1;
172 charge1_m2deg = (cchd_21deg(2.8361e4)- cchd_21deg(0.8361e4))*-1;
173
174 t1=t(2.966e4:4.966e4)-t(2.966e4);
175 figure(5)
176 plot(t1,cvd_50deg(2.966e4:4.966e4),t1,soc50deg,...
177 t1,cvd_21deg(1.682e4:3.682e4),t1,soc21deg,...
178 t1,cvd_m2deg(0.8361e4:2.8361e4),t1,socm2deg)
179 datetick('x',15);
180 legend('Voltage.50deg',strcat('SOC.50deg.Ladung.',...
181 num2str(charge1_50deg),'Ah '))...
182     ,'Voltage.21deg',strcat('SOC.21deg.Ladung.',...
183     num2str(charge1_21deg),'Ah '))...
184     ,'Voltage.m2deg',strcat('SOC.m2deg.Ladung.',...
185     num2str(charge1_m2deg),'Ah '))
186 title(['\fontsize{14} Voltage / SOC']);
187 xlabel('Time in [HH:MM]');
188 ylabel('Voltage in [V] / SOC ');

```

C.2. partielle Entladung bei 3 Temperatur

Listing C.2: partielle Entladung bei 3 Temperatur

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Vergleich der Klemmspannung bei der partiellen Entladung
3 % In diesem Code wird die partielle Entladung
4 % bei 3 Temperatur verglichen und dargestellt
5
6 clear;clc;close all;
7 load('C5_int_dis_4A5_1V75_all.mat')
8 load('C5_discharge_20deg_stufe.mat')
9 load('C5_discharge_m2deg_4A5_stage_all_in_all.mat')
10 %
11 % Datenverarbeitung Intepolation
12 % 50°C

```

```
13 time_50deg_1 = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
14 duration_50deg_1 = round(time_50deg_1(end))-1;
15 lange = 1:1:duration_50deg_1;
16 ccd_50deg_1 = interp1(time_50deg_1,comb_cur_data(:,2),lange);
17 cvd_50deg_1 = interp1(time_50deg_1,comb_volt_data(:,2),lange);
18 ctd_50deg_1 = interp1(time_50deg_1,comb_temp_data(:,2),lange);
19 cchd_50deg_1 = interp1(time_50deg_1,comb_charge_data,lange);
20 time_50deg_1 = interp1(time_50deg_1,comb_cur_data(:,1),lange);
21
22 % Datenlänge einigen
23 cvd_m2deg = [cvd_m2deg ones(1,length(cvd_50deg_1)...
24 -length(cvd_m2deg))*cvd_m2deg(end)];
25 cvd_20deg_stufe = [cvd_20deg_stufe ones(1,length(cvd_50deg_1)...
26 -length(cvd_20deg_stufe))*cvd_20deg_stufe(end)];
27 ccd_m2deg = [ccd_m2deg zeros(1,length(ccd_50deg_1)...
28 -length(ccd_m2deg))];
29 ccd_20deg_stufe = [ccd_20deg_stufe zeros(1,length(ccd_m2deg)...
30 -length(ccd_20deg_stufe))];
31 cchd_20deg_stufe = [cchd_20deg_stufe ones(1,length(cchd_50deg_1)...
32 -length(cchd_20deg_stufe))*cchd_20deg_stufe(end)];
33 cchd_m2deg = [cchd_m2deg ones(1,length(cchd_50deg_1)..
34 -length(cchd_m2deg))*cchd_m2deg(end)];
35 ctd_m2deg = [ctd_m2deg ones(1,length(ctd_50deg_1)...
36 -length(ctd_m2deg))*ctd_m2deg(end)];
37 ctd_20deg_stufe = [ctd_20deg_stufe ones(1,length(ctd_50deg_1)...
38 -length(ctd_20deg_stufe))*ctd_20deg_stufe(end)];
39
40
41 %% Plot
42 %t=1:length(cvd_m2deg);
43 t = time_50deg_1;
44 figure
45 % Klemmenspannung
46 subplot(2,1,1)
47 plot(t, cvd_50deg_1,t,cvd_20deg_stufe,t,cvd_m2deg)
48 datetick('x','dd/mm')
49 title('Klemmenspannung')
50 legend('50°C','21°C','-2°C')
51 xlabel('Zeit [TT/MM]')
52 ylabel('Spannung [V]')
53 grid
54
55 % Strom
56 subplot(2,1,2)
```

```
57 plot(t,ccd_50deg_1,t,ccd_20deg_stufe,t,ccd_m2deg)
58 datetick('x','dd/mm')
59 legend('50°C','21°C','-2°C')
60 title('Strom')
61 xlabel('Zeit [TT/MM]')
62 ylabel('Strom [A]')
63 grid
64 % saveas(gcf,'111','pdf')
65
66 % Ladung
67 figure
68 subplot(2,1,1)
69 plot(t,cchd_50deg_1,t,cchd_20deg_stufe,t,cchd_m2deg)
70 datetick('x','dd/mm')
71 title('Ladung')
72 legend('50°C','21°C','-2°C')
73 xlabel('Zeit [TT/MM]')
74 ylabel('Ladung [Ah]')
75 grid
76 delta_temp_20 = ctd_50deg_1(1)-ctd_20deg_stufe(1);
77 delta_temp_m2 = ctd_50deg_1(1)-ctd_m2deg(1);
78 %
79 % subplot(3,1,3)
80 % plot(t,ctd_50deg,t,ctd_21deg+ delta_temp_21...
81 %,t,ctd_m2deg+ delta_temp_m2)
82 % legend('Temperatur.50deg',strcat('Temperature.21deg + '...
83 %, [num2str(delta_temp_21)],'°C'),strcat('Temperature.m2deg + '...
84 %, [num2str(delta_temp_m2)],'°C'));
85
86 % Temperatur
87 subplot(2,1,2)
88 plot(t,ctd_50deg_1-50,t,ctd_20deg_stufe -21.5,t,ctd_m2deg+2.1)
89 datetick('x','dd/mm')
90 legend('50°C','21°C','-2°C')
91 title('Abweichung der Temperatur von dem Sollwert...
92 (0=Sollwert: 50°C, 21°C und -2°C)')
93 xlabel('Zeit [TT/MM]')
94 ylabel('Abweichung [°C]')
95 grid
96 % saveas(gcf,'112','pdf')
```

C.3. Dynamische Entladung bei 3 Temperatur

Listing C.3: Dynamische Entladung bei 3 Temperatur

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Vergleich der Klemmspannung bei der kurzzeitigen Entladung
3  % In diesem Code wird die kurzzeitige Entladung
4  % bei 3 Temperatur verglichen und dargestellt
5
6  clear all;clc;close all;
7  % Datenverarbeiten, Intepolation
8  % 50°C
9  load('C5_discharge_4A_9H5_all.mat')
10 time_50deg = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
11 duration_50deg = round(time_50deg(end))-1;
12 lange = 1:1:duration_50deg;
13 ccd_50deg = interp1(time_50deg,comb_cur_data(:,2),lange);
14 cvd_50deg = interp1(time_50deg,comb_volt_data(:,2),lange);
15 ctd_50deg = interp1(time_50deg,comb_temp_data(:,2),lange);
16 cchd_50deg = interp1(time_50deg,comb_charge_data,lange);
17 time_50deg = interp1(time_50deg,comb_cur_data(:,1),lange);
18 t_50=time_50deg;%-time(1);
19
20 cvd_50deg(duration_50deg:end)=cvd_50deg(duration_50deg);
21 ccd_50deg(duration_50deg:end)=ccd_50deg(duration_50deg);
22 cchd_50deg(duration_50deg:end)=cchd_50deg(duration_50deg);
23
24 % -2°C
25 load('C5_discharge_m2degree3_9H5_all.mat')
26 time_m2deg = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
27 duration_m2deg = round(time_m2deg(end))-1;
28 %lange = 1:1:duration_m2deg;
29 ccd_m2deg = interp1(time_m2deg,comb_cur_data(:,2),lange);
30 cvd_m2deg = interp1(time_m2deg,comb_volt_data(:,2),lange);
31 ctd_m2deg = interp1(time_m2deg,comb_temp_data(:,2),lange);
32 cchd_m2deg = interp1(time_m2deg,comb_charge_data,lange);
33 time_m2deg = interp1(time_m2deg,comb_cur_data(:,1),lange);
34 t_m2=time_m2deg;%-time(1);
35
36 % 20°C
37 load('C5_discharge2_0R5_9H5_all.mat')
38 time_21deg = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
39
40 duration_21deg = round(time_21deg(end))-1;

```

```

41 charge_21deg=zeros(1,length(comb_cur_data(:,1)));
42 charge_21deg(1)=0;
43 for k =2:length(comb_cur_data(:,1))
44     charge_21deg(k) = charge_21deg(k-1) + comb_cur_data(k,2)...
45     * (comb_cur_data(k,1) - comb_cur_data(k-1,1))*24;
46 end
47
48 ccd_21deg = interp1(time_21deg,comb_cur_data(:,2),lange);
49 cvd_21deg = interp1(time_21deg,comb_volt_data(:,2),lange);
50 %ctd_21deg = interp1(time_21deg,comb_temp_data(:,2),lange);
51 cchd_21deg = interp1(time_21deg,charge_21deg,lange);
52 time_21deg = interp1(time_21deg,comb_cur_data(:,1),lange);
53
54 cvd_21deg(35536:end)=cvd_21deg(35535);
55 ccd_21deg(35536:end)=ccd_21deg(35535);
56 cchd_21deg(35536:end)=cchd_21deg(35535);
57
58 %Plot
59 % Klemmenspannung
60 t=time_50deg;
61 figure(1)
62 subplot(2,1,1)
63 k1=0;cvd_50deg(1)-cvd_21deg(1);
64 k2=0;cvd_50deg(1)-cvd_m2deg(1);
65 plot(t, cvd_50deg,t, cvd_21deg+k1,t, cvd_m2deg+k2)
66 datetick('x','HH:MM')
67 xlim([t(1),t(end)]);
68 % % % title({'\fontsize{14}Vergleich der Klemmenspannung...
69 % bei unterschiedlichen Temperaturen';...
70 % % 'Messwert: Entladen mit konstanten Strom (4A)';...
71 % % 'Lastzyklus = Lastzyklus1 * 3 + 1 Std (ohne Last) + ...
72 % Lastzyklus2 * 3';;' Lastzyklus1 = 1 Std;...
73 % Lastzyklus2 = 0.5 Std ';'\fontsize{10}Klemmenspannung'});
74 xlabel('Zeit [HH:MM]');
75 ylabel('Spannung [V]')
76 legend('50°C','21°C','-2°C')
77 grid
78 %
79
80 % h=subplot(3,1,2)
81 % pos=get(h,'Position')
82 % set(h,'Position',[pos(1) pos(2) pos(3) 0.8*pos(4)]);
83 % Strom
84 subplot(2,1,2)

```

```
85 plot(t,ccd_50deg,t,ccd_21deg,t,ccd_m2deg)
86 legend('50°C','21°C','-2°C','Location','SouthEast');
87 datetick('x','HH:MM')
88 xlim([t(1),t(end)]);
89 title('Strom');
90 xlabel('Zeit [HH:MM]');
91 ylabel('Strom [A]');
92 grid
93 saveas(gcf,'11','pdf')
94
95 %delta_temp=ctd_50deg(1)-ctd_21deg(1);
96 % h=subplot(3,1,3)
97 % pos=get(h,'Position')
98 % set(h,'Position',[pos(1) pos(2) pos(3) 0.8*pos(4)]);
99 % Ladung
100 figure
101 subplot(2,1,1)
102
103 plot(t,cchd_50deg,t,cchd_21deg,t,cchd_m2deg)
104 legend('50°C','21°C','-2°C');
105 datetick('x','HH:MM')
106 xlim([t(1),t(end)]);
107 title('Ladung');
108 xlabel('Zeit [HH:MM]');
109 ylabel('Ladung [Ah]');
110 grid
111 % Temperatur
112 subplot(2,1,2)
113 plot(t,ctd_50deg-50,t,ctd_m2deg+2.2)
114 legend('50°C','-2°C ');
115 datetick('x','HH:MM')
116 xlim([t(1),t(end)]);
117 title('Abweichung der Temperatur von dem Sollwert...
118 (0 = Solltemperatur: 50°C/-2°C)');
119 xlabel('Zeit [HH:MM]');
120 ylabel('Abweichung [°C]');
121 grid
122 saveas(gcf,'12','pdf')
123
124 % saveas(gcf,'Vergleich der Klemmenspannung...
125 % dynamischen Verhalten','pdf')
126 %%
127 deltav=cvd_50deg(1)-cvd_21deg(1);
128 deltav_m2=cvd_50deg(1)-cvd_m2deg(1);
```

```
129
130 % Endwert Vergleich
131 C_21deg = 0.8e6;
132 C_50deg = 1.04e6;
133 C_m2deg = 1.45e6;
134 %min(cchd_m2deg) / (cvd_m2deg(end) - cvd_m2deg(1)) * 24 * 3600;
135
136 cvd_21deg_C = zeros(1, length(lange));
137 cvd_50deg_C = zeros(1, length(lange));
138 cvd_m2deg_C = zeros(1, length(lange));
139
140 cvd_21deg_C(1) = cvd_21deg(1) + deltav;
141 cvd_50deg_C(1) = cvd_50deg(1);
142 cvd_m2deg_C(1) = cvd_m2deg(1);
143
144 for k = 2:length(lange)
145     cvd_21deg_C(k) = cvd_21deg(1) + cchd_21deg(k) * 3600 / C_21deg;
146     cvd_50deg_C(k) = cvd_50deg(1) + cchd_50deg(k) * 3600 / C_50deg;
147     cvd_m2deg_C(k) = cvd_m2deg(1) + cchd_m2deg(k) * 3600 / C_m2deg;
148
149 end
150 % figure(3)
151 % subplot(2,1,1)
152 % plot(t, cvd_50deg, t, cvd_50deg_C)
153 % datetick('x', 15);
154 % legend('50deg')
155 % subplot(2,1,2)
156 % plot(t, cvd_21deg, t, cvd_21deg_C)
157 % legend('21deg')
158 % datetick('x', 15);
159
160 % Plot
161 figure(4)
162 subplot(2,1,1)
163 plot(t, cvd_50deg, t, cvd_50deg_C, t, cvd_21deg + deltav, ...
164 t, cvd_21deg_C, t, cvd_m2deg + deltav_m2, t, cvd_m2deg_C)
165 datetick('x', 15);
166 legend('Voltage.50deg', strcat('Voltage.Kondensator...',
167 ('', num2str(C_50deg, '%1.3e'), 'F).50deg'), ...
168 ... strcat('Voltage.21deg + ', num2str(deltav), 'V'), strcat...
169 ('Voltage.Kondensator(', num2str(C_21deg, '%1.3e'), 'F).21deg') ...
170 , strcat('Voltage.m2deg + ', num2str(deltav_m2), 'V') ...
171 , strcat('Voltage.Kondensator(', num2str(C_m2deg, '%1.3e') ...
172 , 'F).21deg'))
```

```

173 title({'\fontsize{14}Comparsion with Terminal Voltage...
174 in Ambient and Thermostat';'Discharge with constant last...
175 (4A) for 9.5 Hours ';'\fontsize{10}Terminal Voltage'});
176 xlabel('Time in [HH:MM]');
177 ylabel('Voltage in [V]');
178 subplot(2,1,2)
179 plot(t,ccd_50deg,t,ccd_21deg,t,ccd_m2deg)
180 datetick('x',15);
181 legend('Current.50deg','Current.21deg','Current.m2deg')
182 title('Current');
183 xlabel('Time in [HH:MM]');
184 ylabel('Current in [A]');

```

C.4. Beispiel der Simulation des Modells 4

Listing C.4: Beispiel der Simulation des Modells 4

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In diesem Code wird die Simulation des
3 % Modells 4 durchgeführt. Dabei wird durch
4 % LSQ die Parameter des Modells optimiert.
5 % Anschließend wird das Ergebnis angezeigt.
6 clear all;
7 clc;
8 close all;
9 % globale Variable
10 global u_im;
11 global num_samples;
12 global Cbulk;
13 global Csurface;
14 global Re;
15 global Rt;
16 global Rs;
17 global i_im;
18 global tt4;
19
20 % Datenverarbeiten, Intepolation
21
22 load('09120401_1.mat')
23 zeit = (current_data(:,1) - current_data(1,1)) * 24 * 60 *60;
24 num_samples = round(zeit(length(zeit)))-1;
25 lange = 1: num_samples;

```

```
26 t=interp1(zeit, voltage_data(:,1),lange);
27 u_im=interp1(zeit,voltage_data(:,2),lange);
28 i_im = (current_data(:,2)-current_data(1,2))/26.7*1000;
29 i_im=interp1(zeit,i_im,lange)*(-1);
30 t2 = t - t(1);
31
32 % i_im(isnan(i_im))=0;
33 % Parameter der gesteuerten Stromquelle einstellen
34 t1=1:num_samples;
35 timer.dimensions = 1;
36 timer.time = t1';
37 timer.signals.values = i_im';
38
39 figure
40 plot(t2,u_im)
41 datetick('x',15)
42 tt4=num_samples;
43 %% Versuch
44     load('parameter_f.mat')
45     a = parameter_f;
46     Cbulk = a(1)*1e3;
47     Csurface = a(2)*1e3;
48     Re = a(3)/1e3;
49     Rt = a(4)/1e3;
50     Rs = a(5)/1e3;
51
52 % Simulationsaufruf
53 [t_sim,zustand,out]= sim('Modell14',[1,tt4]);
54
55 u_im_test = out(1:num_samples,1);
56 i_im_test = out(1:num_samples,2);
57 Vcb_xv_test=out(1:num_samples,3);
58 Vcs_xv_test=out(1:num_samples,4);
59
60 figure
61 plot(1:num_samples,u_im_test,1:num_samples,u_im)
62 legend('uim','Messwert'),'Vcb','Vcs'
63 title('Simulink mit gegebenen Parameter')
64 datetick('x',15)
65 %%
66 %Anfangswert der Optimierungsparametern
67 a=[1e3; 3e2; 1e2; 1e-1; 1];
68 % Cbulk csurface Re Rt Rs
69 % Ober- und Untergrenze der Parametern
```

```
70 lb=[1e2; 1e2; 1e-2; 1e-3; 1e-3];
71 ub=[5e3; 1e3; 1e2; 1e2; 1e2];
72
73 % Optimierungsoptions
74 options = optimset('Algorithm','interior-point','TolFun',...
75 1e-8,'TolX',1e-6,'MaxFunEvals',1e5,'MaxIter',1e8)...
76 %,'OutputFcn',@plotfcn);
77 % FMINCON
78 % In dem Problem wird Funktion 1 aufgerufen
79 % Optimierungsproblem erstellen
80 problem = createOptimProblem('fmincon','objective',...
81 @KFCN1P_Diode_Spule,'x0',a,'lb',lb,'ub',ub,'options',options);%
82 %% Fmincon
83 t_1=now;
84 %[parameter_f,resnorm,residual,exitflag,iterations]...
85 %= fmincon(problem)
86 parameter_f = fmincon(problem)
87 t_f = datevec(now)-datevec(t_1);
88 %save('LSQ.mat','parameter_f')
89
90 %% GLOBALSEARCH
91 t_2 =now;
92 gs = GlobalSearch;%( 'StartPointsToRun','bounds','TolX',0);
93 %('NumTrialPoints',1e5,'Display','iter');
94 [parameter_g,fmin,exitflag,outptm,someminsm]= run(gs,problem)
95 % save('globalsearch.mat','parameter_g')
96 t_g = datevec(now)-datevec(t_2);
97
98 % MULTISEARCH
99 ms = MultiStart('Display','iter','StartPointsToRun','bounds');
100 %,'TolX',0,'TolFun',1e-8
101 t_3 = now;
102 [parameter_m,f1,flag1,out1,some] = run(ms,problem,20)
103 t_m =datevec(now)-datevec(t_3);
104 save('Multistart.mat','parameter_g')
105
106 %% KURVE BERECHNEN
107 %[vo1 vb1 vs1]= KFCN_1_Diode(parameter_f);
108 % Aufruf der Funktion 2
109 [vo2 vb2 vs2 i_im_test] = KFCN_1_Diode_Spule(parameter_g);
110 %[vo4 vb4 vs4] = KFCN_1_Diode_Spule(parameter_g);
111 % [vo3 vb3 vs3] = KFCN_1_Diode_Spule(parameter_m);
112
113 %% Plot
```

```

114 figure(3)
115 subplot(2,1,1)
116 %plot(1:num_samples,u_im,1:num_samples,vo2)
117 plot(t2,u_im,t2,vo2)%,t2,vo3
118 legend('Messwert','Simulation' )
119 title('Modell4')
120 datetick('x','MM')
121 xlabel('Zeit [MM]')
122 ylabel('Spannung [V]')
123 grid
124 title({'\fontsize{14}Vergleich des Simulationsergebnises...
125 des Modells4 mit dem Messwert';;'Messwert: Last von...
126 konstanten Strom (~10A) Lastzyklus : ~9 min';...
127 ;'\fontsize{12}Klemmspannung' })
128 %saveas(gcf,'Modell4_Diode_RC_CS_Current_g_1','png')
129
130 %plot(1:num_samples,u_im,1:num_samples,vo2)
131 subplot(2,1,2)
132 plot(t2,i_im)%,t2,vo3
133 legend('Messwert')
134 datetick('x','MM')
135 title('\fontsize{12}Strom')
136 xlabel('Zeit [MM]')
137 ylabel('Strom [A]')
138 grid
139 saveas(gcf,'Modell4','pdf')
140 %%
141 % save('Modell4_Diode_RC_CS_Current_g_1.mat','parameter_g')

```

C.5. Angerufene Funktion 1 in der Simulation des Modells 4

Listing C.5: Beispiel der Simulation des Modells 4

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In der Funktion 1 wird die Simulation aufgerufen.
3 % Dabei wird die quadratische Abweichung zwischen
4 % dem Messwert und dem Ergebnis der Simulation
5 % berechnet.
6 % Die quadratische Abweichung kann im bestimmten
7 % Bereich (Ruhephase) höhe gewichtet werden,

```

```
8 % als die in anderen Bereiche.
9 function fehler = KFCN1P_xV(a)
10 global u_im;
11 global num_samples;
12 global Cbulk;
13 global Csurface;
14 global Re;
15 global Rt;
16 global Rs;
17 global i_im;
18 global tt4;
19
20 Cbulk = a(1)*1e3;
21 Csurface = a(2)*1e3;
22 Re = a(3)/1e3;
23 Rt = a(4)/1e3;
24 Rs = a(5)/1e3;
25
26 % Simulation durchführen
27 [t,zustand,out]= sim('Modell14',[0,tt4-1]);
28 %%
29 % % % GewichteFCN
30 % ungewicht_up_grenz = 4655;
31 % ungewicht_down_grenz = 1216;
32 % fe_ungewicht = 0.1*((out(ungewicht_down_grenz:...
33 % ungewicht_up_grenz,1)-u_im(ungewicht_down_grenz:
34 % ...ungewicht_up_grenz)).^2 );%
35 %
36 % fe_gewicht_down = 100* ((out(1:ungewicht_down_grenz,1)...
37 % -u_im(1:ungewicht_down_grenz)).^2);
38 % fe_gewicht_up = 100 * ((out(ungewicht_up_grenz:num_samples,1)...
39 % -u_im(ungewicht_up_grenz:num_samples)).^2);
40 %
41 % fehler = sum(fe_ungewicht)+sum(fe_gewicht_down)...
42 % + sum(fe_gewicht_up);
43
44 % gewicht = 3.426e4;
45 % fe_ungewicht = (out(2:gewicht+1,1) -u_im(1:gewicht)').^2;
46 % fe_gewicht= 3 * ((out(gewicht+1:num_samples+1,1)...
47 % -u_im(gewicht:num_samples)').^2);
48 % fehler = sum(fe_ungewicht) + sum(fe_gewicht);
49
50 %%
51 % UngewichteFCN
```

```
52 fe = (out(:,1) -u_im').^2;
53 fehler= sum(fe);
```

C.6. Angerufene Funktion 2 in der Simulation des Modells 4

Listing C.6: Beispiel der Simulation des Modells 4

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In dieser Funktion 2 wird die Simulation des Modell 4
3 % aufgerufen. Das Ergebnis steht zur weiteren Verarbeitung
4 % zur Verfügung.
5
6 function [out1 out2 out3 out4] = KFCN_1(a)
7 % global variable
8 global u_im;
9 global num_samples;
10 global Cbulk;
11 global Csurface;
12 global Re;
13 global Rt;
14 global Rs;
15 global i_im;
16 global tt4;
17
18 Cbulk = a(1)*1e3;
19 Csurface = a(2)*1e3;
20 Re = a(3)/1e3;
21 Rt = a(4)/1e3;
22 Rs = a(5)/1e3;
23 % Simulation durchführen
24 [t,zustand,out]= sim('Modell14',[0,tt4-1]);
25
26 out1=out(1:num_samples,1);
27 out2=out(1:num_samples,3);
28 out3=out(1:num_samples,4);
29 out4=out(1:num_samples,2);
```

C.7. Simulation des angepassten Modell 4 Klasse 2

Listing C.7: Beispiel der Simulation des Modells 4

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % In diesem Code wird die Simulation des
3  % angepassten Modells 4 Klasse 2 durchgeführt.
4  % Dabei wird durch LSQ die Parameter des Modells
5  % optimiert. Anschließend wird das Ergebnis angezeigt.
6  %
7  clear all;
8  clc;
9  close all;
10 % globale Variable
11 global u_im;
12 global i_im;
13 global num_samples;
14
15 global Cbulk;
16 global Csurface;
17 global Re;
18 global Rt;
19 global Rsu;
20 global Rsd;
21 global L;
22 global Rl;
23 global Cl;
24
25 global tt4;
26 global Rl;
27 global V1;
28
29 % Daten von Messwert importieren
30 load('09120401_1.mat')
31 zeit = (current_data(:,1) - current_data(1,1)) * 24 * 60 *60;
32 num_samples = round(zeit(length(zeit)))-1;
33 lange = 1: num_samples;
34 %Datenverarbeitung, Intepolation
35 t=interp1(zeit,voltage_data(:,1),lange);
36 u_im=interp1(zeit,voltage_data(:,2),lange);
37 i_im=interp1(zeit,(current_data(:,2) -current_data(1,2))...
38 /26.7*1e3,lange)*(-1);
39 t2 = t - t(1);
40
```

```
41 % Parameter der gesteruten Stromquelle einstellen
42 for k=1:num_samples
43     if(abs(i_im(k))>1)
44         Y1(k)=1;
45     else
46         Y1(k)=0;
47     end
48 end
49
50 t1=1:num_samples;
51 timer.dimensions = 1;
52 timer.time = t1';
53 timer.signals.values = Y1';
54
55 figure
56 plot(t2,u_im)
57 datetick('x',15)
58
59 figure
60 plot(t2,i_im)
61 datetick('x',15)
62 % Last
63 Rl = 0.185;
64 V1= u_im(1);
65 tt4= num_samples;
66 %% Versuch
67     load('parameter_g_RC-Cs_2.mat')
68     a = parameter_g;
69     Cbulk = a(1)*1e3;
70     Csurface = a(2)*1e3;
71     Re = a(3)/1e3;
72     Rt = a(4)/1e3;
73     Rsu = a(5)/1e3;
74     Rsd = a(6)/1e3;
75     L = a(7);
76     Rl = a(8)/1e3;
77     C1 = a(9)*1e3;
78 % Simualtion aufrufen
79 [t_sim,zustand,out]= sim('Modell4_Diode_Spule_RC_CS',[0,tt4]);
80
81 u_im_test = out(2:num_samples+1,1);
82 i_im_test = out(2:num_samples+1,2);
83 Vcb_xv_test=out(2:num_samples+1,3);
84 Vcs_xv_test=out(2:num_samples+1,4);
```

```

85
86     figure(1)
87     subplot(2,1,1)
88     plot(t,u_im,t,u_im_test)
89     legend('Messwert','Simulation' )
90     title('Modell14')
91     datetick('x','MM')
92     xlabel('Zeit [MM]')
93     ylabel('Spannung [V]')
94     grid
95     title({'\fontsize{14}Vergleich des Simulationsergebnises des...
96     erweiteren Modells4\_2 mit dem Messwert';;'Messwert: Last...
97     von konstanten Strom (~10A) Lastzyklus : ~9 min';...
98     ;'\fontsize{12}Klemmspannung' })
99
100    subplot(2,1,2)
101    plot(t,i_im)
102    legend('Messwert')
103    datetick('x','MM')
104    title('\fontsize{12}Strom')
105    xlabel('Zeit [MM]')
106    ylabel('Strom [A]')
107    grid
108    saveas(gcf,'erweitertesModell14_2','pdf')
109
110    figure(2)
111    plot(1:num_samples,i_im,1:num_samples,i_im_test)
112    legend('i (Messwert)','i (Simulink)')
113
114    %% künstlich erstelltes Rauschen
115    Abweichung_mean=mean(abs(u_im-Vcb_xv_test'));
116    Abweichung_max=max(abs(u_im-Vcb_xv_test'));
117    Abweichung_Endpunkt1000= mean(abs(u_im(end-1000:end)...
118    -Vcb_xv_test(end-1000:end)'));
119    save('Abweichung.mat','Abweichung_mean','Abweichung_max'...
120    ,'Abweichung_Endpunkt1000')
121    %%
122    %Anfangswert der optimierungsparametern
123    a=[1e3; 4e2; 3; 1e-1; 1; 5; 1; 10; 10];
124    % Cbulk csurface Re Rt Rsu Rsd L R1 C1
125    % Ober- und Untengrenze der Paramenter
126    lb=[1e3; 1e2; 1; 1e-3; 1; 1; 1e-1; 1; 1];
127    ub=[1e4; 1e3; 1e3; 1; 1e2; 1e2; 1e3; 1e3; 1e2];
128    % Optimierungsoptions

```

```

129 options = optimset('Algorithm','interior-point',...
130 'TolFun',1e-8,'TolX',1e-6,...
131 'MaxFunEvals',1e5,'MaxIter',1e8);%,'OutputFcn',@plotfcn
132 % Optimierungsproblem erstellen
133 % FMINCON
134 problem = createOptimProblem('fmincon','objective',...
135 @KFCN1P_Diode_Spule,'x0',a,...
136 'lb',lb,'ub',ub,'options',options);%
137 %%
138 t_1=now;
139 %[parameter_f,resnorm,residual,exitflag,iterations]...
140 %= fmincon(problem)
141 parameter_f = fmincon(problem)
142 t_f = datevec(now)-datevec(t_1);
143 % save('LSQ.mat','parameter_f')
144 % GLOBALSEARCH
145 %%
146 t_2 =now;
147 gs = GlobalSearch;%( 'StartPointsToRun','bounds','TolX',0);
148 %('NumTrialPoints',1e5,'Display','iter');
149 [parameter_g,fmin,exitflag,outptm,someminsm] = run(gs,problem)
150 % save('globalsearch.mat','parameter_g')
151 t_g = datevec(now)-datevec(t_2);
152 %%
153 % MULTISEARCH
154 ms = MultiStart('Display','iter','StartPointsToRun','bounds');
155 %,'TolX',0,'TolFun',1e-8
156 t_3 = now;
157 [parameter_m,f1,flag1,out1,some] = run(ms,problem,20)
158 t_m =datevec(now)-datevec(t_3);
159 %save('Multistart.mat','parameter_g')
160 %% KURVE BERECHNEN
161 %[vo1 vb1 vs1]= KFCN_1_Diode(parameter_f);
162 % Funktion 2 aufrufen
163 [vo2 vb2 vs2] = KFCN_1_Diode_Spule(parameter_g);
164 % [vo3 vb3 vs3] = KFCN_1_Diode_Spule(parameter_m);
165 %%
166 figure(441)
167 %plot(1:num_samples,u_im,1:num_samples,vo2)
168 plot(t2,u_im,t2,vo2)% ,t2,vo3
169 legend('Messwert','Optimal.g','Optimal.m')
170 title('Modell4.Diode.Spule.RC.CS')
171 datetick('x',15)
172 % Save

```

```
173 saveas(gcf,'Aufladung_Modell4_Diode_Spule_RC_CS_g_1','png')
174 %%
175 save('Aufladung_Modell4_Diode_Spule_RC_CS_g_1.mat','parameter_g')
```

C.8. Angerufene Funktion 1 in der Simulation des angepassten Modell 4 Klasse 2

Listing C.8: Angerufene Funktion 1 in der Simulation des Modells 4

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In diesen Funktion wird das Simulation des angepassten
3 % Modells 4 Klasse 2 aufgerufen.
4 % Dabei wird die quadratische Fehler berechnet.
5 % In bestimmten Bereich wird die Abweichung hoch gewichtet.
6
7 function fehler = KFCN1P_Diode_Spule(a)
8 % Globale Variable
9 global u_im;
10 global num_samples;
11 global Cbulk;
12 global Csurface;
13 global Re;
14 global Rsd;
15 global Rt;
16 global tt4;
17 global Rl;
18 global V1;
19 global i_im;
20 global Rsu;
21 global L;
22 global C1;
23 global R1;
24
25 Cbulk = a(1)*1e3;
26 Csurface = a(2)*1e3;
27 Re = a(3)/1e3;
28 Rt = a(4)/1e3;
29 Rsu = a(5)/1e3;
30 Rsd = a(6)/1e3;
31 L = a(7);
32 R1 = a(8)/1e3;
```

```

33 C1 = a(9)*1e3;
34 % Angepasste Modell 4 Klasse 1 Simulation durchführen
35 [t,zustand,out]= sim('Modell4_Diode_Spule_RC_CS',[0,tt4]);
36 % % % GewichteFCN
37 % ungewicht_up_grenz = 4655;
38 % ungewicht_down_grenz = 1216;
39 % fe_ungewicht = 0.1*((out(ungewicht_down_grenz:...
40 % ungewicht_up_grenz,1)
41 % -u_im(ungewicht_down_grenz:ungewicht_up_grenz)).^2 );%
42 %
43 % fe_gewicht_down = 100* ((out(1:ungewicht_down_grenz...
44 % ,1) -u_im(1:ungewicht_down_grenz)).^2);
45 % fe_gewicht_up = 100 * ((out(ungewicht_up_grenz:num_samples,...
46 % 1) -u_im(ungewicht_up_grenz:num_samples)).^2);
47 %
48 % fehler = sum(fe_ungewicht) + sum(fe_gewicht_down)...
49 % + sum(fe_gewicht_up);
50 gewicht = 5e3;
51 fe_ungewicht = (out(2:gewicht+1,1) -u_im(1:gewicht)').^2;
52 fe_gewicht= 5 * ((out(gewicht+1:num_samples+1,1)...
53 -u_im(gewicht:num_samples)').^2);
54 fehler = sum(fe_ungewicht) + sum(fe_gewicht);
55
56 %%
57 % UngewichteFCN
58 % fe = (out(2:num_samples+1,1) -u_im').^2;
59 % fehler= sum(fe);

```

C.9. Angerufene Funktion 2 in der Simulation des angepassten Modell 4 Klasse 2

Listing C.9: Angerufene Funktion 2 in der Simulation des Modells 4

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In dieser Funktion wird die Simulation des
3 % angepassten Modell 4 Klasse 2 aufgerufen.
4 % Das Ergebnis steht zur weiteren Verarbeitung
5 % zur Verfügung.
6
7 function [out1 out2 out3 out4] = KFCN_1_Diode_Spule(a)
8 % Globale Variable

```



```

2  % Diese Funktion steuert die Spannungsquelle,
3  % die Spannung an der "'Kondensator"' und Innenwiderstand.
4  % In diesen Funktion wird im jeden diskreten Zeitpunkt
5  % einem Stromwert aufgenommen und weitere Funktion aufgerufen
6
7  function [sys,x0,str,ts,simStateCompliance] = sfundsc2(t,x,u,flag)
8
9  %SFUNTMPL General MATLAB S-Function Template
10 %With MATLAB S-functions, you can define you own ordinary
11 %differential equations (ODEs), discrete system equations,
12 %and/or just about any type of algorithm to be used
13 %within a Simulink block diagram.
14 %
15 %The general form of an MATLAB S-function syntax is:
16 %[SYS,X0,STR,TS,SIMSTATECOMPLIANCE] = SFUNC(T,X,U,FLAG,P1,...,Pn)
17 %
18 %What is returned by SFUNC at a given point in time, T, depends
19 %on the value of the FLAG, the current state vector, X,
20 %and the current
21 %input vector, U.
22 %
23 %FLAG RESULT          DESCRIPTION
24 %-----
25 %0      [SIZES,X0,STR,TS] Initialization, return system sizes in SYS,
26 %          initial state in X0, state ordering strings
27 %          in STR, and sample times in TS.
28 %1      DX              Return continuous state derivatives in SYS.
29 %2      DS              Update discrete states SYS = X(n+1)
30 %3      Y               Return outputs in SYS.
31 %4      TNEXT           Return next time hit for variable step sample
32 %          time in SYS.
33 %5                      Reserved for future (root finding).
34 %9      []              Termination, perform any cleanup SYS=[].
35 %
36 %
37 %The state vectors, X and X0 consists of continuous states
38 %followed by discrete states.
39 %
40 %Optional parameters, P1,...,Pn can be provided to the S-function
41 %and used during any FLAG operation.
42 %
43 %When SFUNC is called with FLAG = 0, the following information
44 %should be returned:
45 %

```

```
46 %SYS(1) = Number of continuous states.
47 %SYS(2) = Number of discrete states.
48 %SYS(3) = Number of outputs.
49 %SYS(4) = Number of inputs.
50 % Any of the first four elements in SYS can be specified
51 % as -1 indicating that they are dynamically sized. The
52 % actual length for all other flags will be equal to the
53 % length of the input, U.
54 %SYS(5) = Reserved for root finding. Must be zero.
55 %SYS(6) = Direct feedthrough flag (1=yes, 0=no). The s-function
56 % has direct feedthrough if U is used during the FLAG=3
57 % call. Setting this to 0 is akin to making a promise that
58 % U will not be used during FLAG=3. If you break the promise
59 % then unpredictable results will occur.
60 %SYS(7) = Number of sample times. This is the number of rows in TS.
61 %
62 %
63 %X0 = Initial state conditions or [] if no states.
64 %
65 %STR = State ordering strings which is generally specified as [].
66 %
67 %TS = An m-by-2 matrix containing the sample time
68 % (period, offset) information. Where m = number of sample
69 % times. The ordering of the sample times must be:
70 %
71 % TS = [0 0, : Continuous sample time.
72 % 0 1, : Continuous, but fixed in minor step
73 % sample time.
74 % PERIOD OFFSET, : Discrete sample time where
75 % PERIOD > 0 & OFFSET < PERIOD.
76 % -2 0]; : Variable step discrete sample time
77 % where FLAG=4 is used to get time of
78 % next hit.
79 %
80 % There can be more than one sample time providing
81 % they are ordered such that they are monotonically
82 % increasing. Only the needed sample times should be
83 % specified in TS. When specifying more than one
84 % sample time, you must check for sample hits explicitly by
85 % seeing if
86 % abs(round((T-OFFSET)/PERIOD) - (T-OFFSET)/PERIOD)
87 % is within a specified tolerance, generally 1e-8. This
88 % tolerance is dependent upon your model's sampling times
89 % and simulation time.
```

```

90 %
91 % You can also specify that the sample time of the S-function
92 % is inherited from the driving block. For functions which
93 % change during minor steps, this is done by
94 % specifying SYS(7) = 1 and TS = [-1 0]. For functions which
95 % are held during minor steps, this is done by specifying
96 % SYS(7) = 1 and TS = [-1 1].
97 %
98 %SIMSTATECOMPLIANCE = Specifies how to handle this block when
99 %     saving and restoring the complete simulation
100 %     state of the model. The allowed values are:
101 %     'DefaultSimState','HasNoSimState'
102 %     or 'DisallowSimState'. If this value
103 %     is not specified, then the block's compliance with
104 %     simState feature is set to 'UnknownSimState'.
105 %
106 %
107 %Copyright 1990-2010 The MathWorks, Inc.
108 %$Revision: 1.18.2.5 $
109 %
110 %
111 %The following outlines the general structure of an S-function.
112 %
113 switch flag,
114
115     %%%%%%%%%%%
116     % Initialization %
117     %%%%%%%%%%%
118     case 0,
119         [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes;
120
121     %%%%%%%%%%%
122     % Update %
123     %%%%%%%%%%%
124     case 2,
125         sys = diff_uc(u);
126
127     %%%%%%%%%%%
128     % Output %
129     %%%%%%%%%%%
130     case 3,
131         sys = mdlOutputs(t,x,u);
132
133

```

```
134 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
135 % GetTimeOfNextVarHit %
136 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
137 case 4,
138     sys=mdlGetTimeOfNextVarHit(t,x,u);
139
140 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
141 % Terminate %
142 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
143 %
144     case { 1, 4, 9 },
145         sys = [];
146
147     otherwise
148         DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
149 end
150
151 %end sfundsc2
152
153 %
154 %=====
155 % mdlInitializeSizes
156 % Return the sizes, initial conditions,
157 % and sample times for the S-function.
158 %=====
159 %
160 function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes
161 % (u_anfang,Cbulk)
162
163 sizes = simsizes;
164
165 sizes.NumContStates = 0;
166 sizes.NumDiscStates = 1;
167 sizes.NumOutputs = 1;
168 sizes.NumInputs = 1;
169 sizes.DirFeedthrough = 1;
170 sizes.NumSampleTimes = 1;
171
172 sys = simsizes(sizes);
173
174 % Cbulk = Cbulk;
175 % x0 = 2.1040;
176 x0 = 1.9088;
177 % u_old = u_anfang;
```

```
178 str = [];  
179 ts = [-1 0]; % Sample period of 0.1 seconds (10Hz)  
180  
181 % speicfy that the simState for this s-function is...  
182 % same as the default  
183 simStateCompliance = 'DefaultSimState';  
184  
185 % end mdlInitializeSizes  
186  
187 %  
188 %=====  
189 % mdlUpdate  
190 % Handle discrete state updates, sample time hits  
191 % , and major time step, requirements.  
192 %=====  
193 %  
194 % function sys = mdlUpdate(t,x,u)  
195 % sys = u(1);  
196 %  
197 % %end mdlUpdate  
198  
199 %=====  
200 % mdlOutputs  
201 % Return the output vector for the S-function  
202 % u(1) = u_im  
203 % u(2) = i_im  
204 %=====  
205 %  
206 function sys = mdlOutputs(t,x,u)%,u_anfang,Cbulk)  
207  
208 % u_c1 = x0 + u(2)*1/Cbulk  
209 % dod = 1 - (u_c - 1.95)/0.2;  
210 % u_c = u_c1 * (-0.0620*dod*(1-dod)^2.2668+0.0225*dod)  
211 % re = 0.0204/(1-dod^2.8183+0.5098);  
212 % x0 = u(1);  
213 %  
214 % sys = re*u(1) + u_c;  
215 sys = x;  
216  
217 %end mdlOutputs  
218  
219 %  
220 %=====  
221 % mdlGetTimeOfNextVarHit
```

```

222 % Return the time of the next hit for this block.
223 % Note that the result is absolute time.
224 %=====
225 %
226 function sys=mdlGetTimeOfNextVarHit (t,x,u)
227
228 sys = t + 1;
229
230 % end mdlGetTimeOfNextVarHit

```

C.11. Angerufene Funktion 4 in der Simulation des angepassten Modell 4 Klasse 2

Listing C.11: Angerufene Funktion 4 in der Simulation des angepassten Modell 4 Klasse 2

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Diese Funktion wird von obener Funktion aufgerufen.
3 % In diesen Funktion wird die Spannung am Kondensator und
4 % Innenwiderstand berechnet.
5
6 function u_cap = diff_uc(i)
7 %Globale Variable
8 global x0;
9 global Cbulk;
10 global dod;
11 global u_c;
12 global u_r;
13 global uc;
14 global dod1;
15 global r;
16
17 % Temperatur
18 T=20;
19
20 % calculate the voltage of capacitor
21 u_c1 = x0 + i*1/Cbulk;
22
23 dod = 1 - (u_c - 1.93)/0.19; %% 20°C ENTLADEN
24
25 % calibrate the voltage of capacitor
26 u_c = u_c1 * (1+(-0.117*dod*(1-dod)^(2.348) + 0.0212*dod^0.4096...

```

```

27 + (T-20)*(-0.0156*dod^1.7468+0.0161*dod^1.6306));
28
29 % u_c = u_c1; %charge
30
31 % calculate the resistance according to formel
32 re = 0.014 +0.0246*dod^5.9777 -0.0007*(1-dod)^0.0447...
33     - 0.0001*(T-20);
34 % store the current voltage of capacitor for next iteration
35 x0 = u_c1;
36
37 % calculate the voltage of the capacity and resistance
38 u_cap = re*i + u_c;
39
40 %% diagnose the vaule during the iteration
41 % u_r = [u_r ;re*i];
42 uc = [uc; u_c];
43 % dod1 = [dod1 dod];
44 % r=[r re];
45 end

```

C.12. Vergleich der Ruhespannung bzw. des Widerstandes

Listing C.12: Vergleich der Ruhespannung bzw. des Widerstandes

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In diesem Code wird die Ruhespannung bzw.
3 % der Innenwiderstand bei 3 Temperatur verglichen.
4 % Dabei wird durch jeweils eine Formel der Verlauf
5 % der Ruhespannung bzw. des Innenwiderstands zusammen-
6 % gefasst. Das Ergebnis wird angezeigt.
7
8
9 clear; clc; close all;
10 % Ruhespannung bzw. Innenwiderstand berechnen
11 load('volt_char_data_20_deg')
12 r_20deg = (q_ocv_table_20deg(:,2)-q_v_table_20deg(:,2))./4.5;
13 dod_20deg = q_ocv_table_20deg(:,1)./q_ocv_table_20deg(end,1);
14 r_20deg = (q_ocv_table_20deg(:,2)-q_v_table_20deg(:,2))./4.5;
15
16 C_try_20 = -(q_ocv_table_20deg(end,1)-q_ocv_table_20deg(1,1))...

```

```

17 *3600/(q_ocv_table_20deg(1,2)-q_ocv_table_20deg(end,2));
18 %
19 load('volt_char_data_50_deg')
20 r_50deg = (q_ocv_table_50deg(:,2)-q_v_table_50deg(:,2))./4.5;
21 dod_50deg = q_ocv_table_50deg(:,1)./q_ocv_table_50deg(end,1);
22 r_50deg = (q_ocv_table_50deg(:,2)-q_v_table_50deg(:,2))./4.5;
23
24 C_try_50 = -(q_ocv_table_50deg(end,1)-q_ocv_table_50deg(1,1))...
25 *3600/(q_ocv_table_50deg(1,2)-q_ocv_table_50deg(end,2));
26
27 load('volt_char_data_m2_deg_1')
28 r_m2deg = (q_ocv_table_m2deg(:,2)-q_v_table_m2deg(:,2))./4.5;
29 dod_m2deg = q_ocv_table_m2deg(:,1)./q_ocv_table_m2deg(end,1);
30 r_m2deg = (q_ocv_table_m2deg(:,2)-q_v_table_m2deg(:,2))./4.5;
31
32 C_try_m2 = -(q_ocv_table_m2deg(end,1)-q_ocv_table_m2deg(1,1))..
33 *3600/(q_ocv_table_m2deg(1,2)-q_ocv_table_m2deg(end,2));
34
35 % Plot
36 figure(1)
37 plot(q_ocv_table_m2deg(:,1)/q_ocv_table_m2deg(end,1),r_m2deg,...
38      q_ocv_table_20deg(:,1)/q_ocv_table_20deg(end,1),r_20deg,...
39      q_ocv_table_50deg(:,1)/q_ocv_table_50deg(end,1),r_50deg)
40 legend('-2°C','21°C','50°C')
41 % title('\fontsize{14}internal resistance versus DoD..
42 % under different temperatur')
43 xlabel('DoD')
44 ylabel('Widerstand [Ohm]')
45 grid
46 % saveas(gcf,'Widerstand des Messwertes','pdf')
47
48 figure(2)
49 plot(q_ocv_table_m2deg(:,1)/q_ocv_table_m2deg(end,1),...
50      q_ocv_table_m2deg(:,2),q_ocv_table_20deg(:,1)/...
51      q_ocv_table_20deg(end,1),q_ocv_table_20deg(:,2)...
52      ,q_ocv_table_50deg(:,1)/q_ocv_table_50deg(end,1),...
53      q_ocv_table_50deg(:,2))
54 legend('-2°C','21°C','50°C')
55 %title('\fontsize{14}Open Circuit Voltage(OCV)...
56 %versus DoD under different temperatur')
57 xlabel('DoD')
58 ylabel('OCV [V]')
59 grid
60 % saveas(gcf,'OCV des Messwertes','pdf')

```

```
61
62 %%
63 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
64 % Formel 2 Resistance %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
66 % Formel 1
67 n= [0.0133 9 0.5];
68 % Funktion erstellen
69 resist = @(a,xdata) (a(1)./(1-xdata.^a(2)+a(3)));
70
71 % Parameter nach der Optimierung LSQ
72 parameter_r_20deg = lsqcurvefit(resist,n,dod_20deg,r_20deg);% 20deg
73 parameter_r_50deg = lsqcurvefit(resist,n,dod_50deg,r_50deg);% 50deg
74 parameter_r_m2deg = lsqcurvefit(resist,n,dod_m2deg,r_m2deg);% m2deg
75
76 % Widerstand berechnen
77 r_20T = resist(parameter_r_20deg,dod_20deg);
78 r_50T = resist(parameter_r_50deg,dod_50deg);
79 r_m2T = resist(parameter_r_m2deg,dod_m2deg);
80
81 % Formel 2
82 n0=[0.0153 0.14 20];% 0.002 0.54];
83 resist_1 = @(a,xdata) (a(1)+ a(2)*xdata.^a(3) );
84 parameter_r_20deg_1 = lsqcurvefit(resist_1,n0,dod_20deg,r_20deg);
85 parameter_r_50deg_1 = lsqcurvefit(resist_1,n0,dod_50deg,r_50deg);
86 parameter_r_m2deg_1 = lsqcurvefit(resist_1,n0,dod_m2deg,r_m2deg);
87
88 r1 = resist_1(parameter_r_20deg_1,dod_20deg);
89 r2 = resist_1(parameter_r_50deg_1,dod_50deg);
90 r3 = resist_1(parameter_r_m2deg_1,dod_m2deg);
91
92 figure(6)
93 subplot(3,1,1)
94 plot(dod_20deg,r_20deg,dod_20deg,r1)
95 legend('Messwert.20°C','Formel.20°C','Location','Northwest')
96 xlabel('DoD')
97 ylabel('Widerstand [Ohm]')
98 grid
99
100 % Temperatur
101 T=50;
102 % formel 2 50°C
103 pa = parameter_r_20deg_1;
104 % Funktion erstellen
```

```
105 resist_R0_50 = @(a,xdata) (pa(1)+ pa(2)*xdata.^pa(3)) + a(1)...
106 * (1-xdata).^a(2)+ a(3)*(T-20) ;
107
108 % Parameter Optimieren
109 n0=[0.0153 0.14 13];
110 parameter_r_50deg_R0 = lsqcurvefit(resist_R0_50,...
111 n0,dod_50deg,r_50deg);
112
113 % Widerstand berechnen
114 r2_1 = resist_R0_50(parameter_r_50deg_R0,dod_50deg);
115
116 % Plot
117 figure(6)
118 subplot(3,1,2)
119 plot(dod_50deg,r_50deg,dod_50deg,r2_1)
120 legend('Messwert.50°C','Formel.50°C','Location','Northwest')
121 xlabel('DoD')
122 ylabel('Widerstand [Ohm]')
123 grid
124
125 % Widerstand Formel bewerten
126 T = -1;
127 resist_R0_50 = @(a,xdata) (pa(1)+ pa(2)*xdata.^pa(3))...
128 + a(1)*(1-xdata).^a(2)+ a(3)*(T-20) ;
129 r_m2_2 = resist_R0_50(parameter_r_50deg_R0,dod_m2deg);
130 figure(6)
131 subplot(3,1,3)
132 plot(dod_m2deg,r_m2deg,dod_m2deg,r_m2_2)
133 legend('Messwert.-2°C','Formel.-2°C','Location','Northwest')
134 xlabel('DoD')
135 ylabel('Widerstand [Ohm]')
136 grid
137 saveas(gcf,'Widerstand der Formel','pdf')
138
139 T = 60;
140 resist_R0_50 = @(a,xdata) (pa(1)+ pa(2)*xdata.^pa(3))...
141 + a(1)*(1-xdata).^a(2)+ a(3)*(T-20) ;
142 r_60 = resist_R0_50(parameter_r_50deg_R0,dod_m2deg);
143 T = 30;
144 resist_R0_50 = @(a,xdata) (pa(1)+ pa(2)*xdata.^pa(3))...
145 + a(1)*(1-xdata).^a(2)+ a(3)*(T-20) ;
146 r_30 = resist_R0_50(parameter_r_50deg_R0,dod_m2deg);
147 T = 10;
148 resist_R0_50 = @(a,xdata) (pa(1)+ pa(2)*xdata.^pa(3))...
```

```

149 + a(1)*(1-xdata).^a(2)+ a(3)*(T-20) ;
150 r_10 = resist_R0_50(parameter_r_50deg_R0,dod_m2deg) ;
151
152 %
153 figure(8)
154 plot(dod_m2deg,r_m2_2,dod_m2deg,r_10,dod_20deg,r_20T,...
155 dod_m2deg,r_30,dod_50deg,r_50T,dod_m2deg,r_60)
156 legend('-2°C','10°C','20°C',...
157 '30°C','50°C','60°C','Location','NorthWest')
158 % title('Resistance versus Temperatur and DoD')
159 xlabel('DoD')
160 ylabel('Widerstand [Ohm]')
161 grid
162 % saveas(gcf,'Widerstand aller Temperaturen','pdf')
163
164
165 %%
166 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Formel Kapazität %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
168 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169 % 20°C
170 C_try_20_1 = 7.3e5;
171 C_try_20_2 = -(q_ocv_table_20deg(11,1)-q_ocv_table_20deg(1,1))'...
172 *3600/(q_ocv_table_20deg(1,2)-q_ocv_table_20deg(11,2));
173
174 % CC-Counting
175 for cc = 1:1:11
176     v_c_try_20_1(cc,1) = q_ocv_table_20deg(1,2) + ...
177     q_ocv_table_20deg(cc,1)*3600/C_try_20_1;
178     v_c_try_20_2(cc,1) = q_ocv_table_20deg(1,2) + ...
179     q_ocv_table_20deg(cc,1)*3600/C_try_20_2;
180
181 end
182
183 % Plot Vergleich mit dem Kondensatorverlauf
184 figure(9)
185 subplot(1,3,1)
186 plot(dod_20deg,q_ocv_table_20deg(:,2),dod_20deg,...
187 v_c_try_20_1,dod_20deg,v_c_try_20_2)
188 legend('Messwert','kleiner Kondensator','großer Kondensator')
189 title('20°C')
190 xlabel('DoD')
191 grid
192 ylabel('OCV [V]')

```

```
193
194 % Plot relative Abweichung
195 figure(10)
196 subplot(1,3,1)
197 mmm_20=(-v_c_try_20_1+q_ocv_table_20deg(:,2))./v_c_try_20_2;
198 plot( dod_20deg,mmm_20)
199 % legend('Veränderungsrate')
200 title('20°C')
201 xlabel('DoD')
202 ylabel('Fehlerrate')
203 grid
204
205 % 50 °C
206 C_try_50_1 = 7.6e5;
207 C_try_50_2 = -(q_ocv_table_50deg(12,1)-q_ocv_table_50deg(1,1))...
208 *3600/(q_ocv_table_50deg(1,2)-q_ocv_table_50deg(12,2));
209 % CC-Counting
210 for cc = 1:1:12
211     v_c_try_50_1(cc,1) = q_ocv_table_50deg(1,2) +...
212     q_ocv_table_50deg(cc,1)*3600/C_try_50_1;
213     v_c_try_50_2(cc,1) = q_ocv_table_50deg(1,2) +...
214     q_ocv_table_50deg(cc,1)*3600/C_try_50_2;
215
216 end
217
218 % Plot Vergleich mit dem Kondensatorverlauf
219 % figure(10)
220 figure(9)
221 subplot(1,3,2)
222 plot(dod_50deg,q_ocv_table_50deg(:,2),dod_50deg,...
223 v_c_try_50_1,dod_50deg,v_c_try_50_2)
224 legend('Messwert','kleiner Kondensator','großer Kondensator')
225 title('50°C')
226 xlabel('DoD')
227 grid
228 ylabel('OCV [V]')
229
230 % Plot relative Abweichung
231 figure(10)
232 subplot(1,3,2)
233 mmm_50=(-v_c_try_50_1+q_ocv_table_50deg(:,2))./v_c_try_50_2;
234 plot( dod_50deg,mmm_50)
235 title('50°C')
236 xlabel('DoD')
```

```
237 grid
238 ylabel('Fehlerrate')
239
240
241 % m2 °C
242 % C_try_m2_1 = 7.8e5;
243 C_try_m2_1 = 5.86e5;
244 C_try_m2_2 = 7.7e5%
245
246 % CC-Counting
247 for cc = 1:1:9
248     v_c_try_m2_1(cc,1) = q_ocv_table_m2deg(1,2) + ...
249     q_ocv_table_m2deg(cc,1)*3600/C_try_m2_1;
250     v_c_try_m2_2(cc,1) = q_ocv_table_m2deg(1,2) + ...
251     q_ocv_table_m2deg(cc,1)*3600/C_try_m2_2;
252
253 end
254
255 % Plot Vergleich mit dem Kondensatorverlauf
256 figure(9)
257 subplot(1,3,3)
258
259 % figure
260 plot(dod_m2deg,q_ocv_table_m2deg(:,2),dod_m2deg,v_c_try_m2_1,...
261 dod_m2deg,v_c_try_m2_2)
262 legend('Messwert','kleiner Kondensator','großer Kondensator')
263 title('-2°C')
264 xlabel('DoD')
265 ylabel('OCV [V]')
266 grid
267 % saveas(gcf,'VergleichderKapazität','pdf')
268
269 figure(10)
270 subplot(1,3,3)
271
272 % Plot relative Abweichung
273 % figure
274 mmm_m2=(-v_c_try_m2_2+q_ocv_table_m2deg(:,2))./v_c_try_m2_2;
275 plot(dod_m2deg,mmm_m2)
276 title('-2°C')
277 xlabel('DoD')
278 ylabel('Fehlerrate')
279 grid
280 % saveas(gcf,'Fehlerrate','pdf')
```

```

281 %%
282 figure(222)
283 hold on
284 plot( dod_m2deg,mmm_m2,'g')
285 plot( dod_20deg,mmm_20,'r')
286 plot( dod_50deg,mmm_50,'b')
287 legend('-2°C','20°C','50°C','Location','NorthWest')
288
289 hold off
290 xlabel('DoD')
291 ylabel('Fehlerrate')
292 grid
293 % saveas(gcf,'Fehlerrate1','pdf')
294
295 % mm= -mmm_50(1:11)+mmm_20;
296 %
297 % n = [0.1 2 0.01 0.1 0.01 0.1];
298 % T = 50;
299 % ocv = @(a,xdata) (T-20)*(a(1)*xdata.^a(2)...
300 % - a(3)*(1-xdata).^a(4) +a(5)*xdata);
301 % opts = optimset('TolFun',1e-10,'TolX',1e-10,'MaxIter',1e10);
302 % p_20 = lsqcurvefit(ocv,n,dod_20deg,mm,[],[],opts);
303 % ocv_20 = ocv(p_20,dod_20deg);
304 % figure(11)
305 % plot(dod_20deg,mm,dod_20deg,ocv_20)
306 % xlabel('DoD')
307 % ylabel('Felerrate')
308 % legend('20°C','50°C','-2°C')
309 % grid
310 %% Fertigstellung
311 % OCV Formel Parameter Optimierung 20°C
312 n = [0.1 2 0.01 0.1 0.01 0.1 0.01 0.1 0.01 0.1];
313 T = 20;
314 ocv = @(a,xdata) a(1)*xdata.*(1-xdata).^a(2) + a(3)*xdata.^a(4)...
315 + (T-20)*(a(1)*xdata.^a(2)-a(3)*xdata.^a(4));
316 opts = optimset('TolFun',1e-10,'TolX',1e-10,'MaxIter',1e10);
317 p_20 = lsqcurvefit(ocv,n,dod_20deg,mmm_20,[],[],opts);
318 ocv_20 = ocv(p_20,dod_20deg);
319 figure(11)
320 subplot(3,1,1)
321 plot(dod_20deg,mmm_20,dod_20deg,ocv_20)
322 xlabel('DoD')
323 ylabel('Fehlerrate')
324 legend('Measwert','LSQ-Wert','Location','SouthEast')

```

```

325 title('20°C')
326 grid
327 %
328 n = [0.1 2 0.01 0.1 0.01 0.1];
329
330 % OCV Formel Parameter Optimierung 50°C
331 T=50;
332 ocv_50T_1 = @(a,xdata) p_20(1)*xdata.*(1-xdata).^(p_20(2))...
333 +p_20(3)*xdata.^p_20(4)+(T-20)*(a(1)*xdata.^a(2)-a(3)*xdata.^a(4));
334 opts = optimset('TolFun',1e-15,'TolX',1e-10,'MaxIter',1e10);
335 p_50_1 = lsqcurvefit(ocv_50T_1,n,dod_50deg,mmm_50,[],[],opts);
336 ocv_50_1 = ocv_50T_1(p_50_1,dod_50deg);
337
338 ParameterDerFormel = [p_20(1:4) p_50_1(1:4)];
339 figure(11)
340 subplot(3,1,2)
341 plot(dod_50deg,mmm_50,dod_50deg,ocv_50_1)%,dod_50deg,ocv_50
342 xlabel('DoD')
343 ylabel('Fehlerrate')
344 legend('Measwert','LSQ-Wert','Location','SouthEast')
345 title('50°C')
346 grid
347
348 % OCV Formel Parameter Optimierung -2°C
349
350 n = [-0.1 2 0.01 0.1 -0.1 2 0.01 0.1 1 1 ];
351 T = -2;
352
353 ocv_m2T = @(xdata) p_20(1)*xdata.*(1-xdata).^(p_20(2))...
354 + p_20(3)*xdata.^p_20(4) + (T-20)*(p_50_1(1)*xdata...
355 .^p_50_1(2)-p_50_1(3)*xdata.^p_50_1(4));
356 ocv_m2T_1 = @(a,xdata) a(1)*xdata.*(1-xdata).^(a(2))...
357 + a(3)*xdata.^a(4)...
358 + (T-20)*(a(5)*xdata.^a(6)-a(7)*xdata.^a(8));
359
360 p_m2 = lsqcurvefit(ocv_m2T_1,n,dod_m2deg,mmm_m2);
361 ocv_m2 = ocv_m2T(dod_m2deg);
362 ocv_m2_1 = ocv_m2T_1(p_m2,dod_m2deg);
363 figure(11)
364 subplot(3,1,3)
365 plot(dod_m2deg,mmm_m2,dod_m2deg,ocv_m2,dod_m2deg,ocv_m2_1)
366 xlabel('DoD')
367 ylabel('Fehlerrate')
368 legend('Measwert','LSQ-Wert-1','LSQ-Wert-2','Location','NorthEast')

```

```
369 title('-2°C')
370 grid
371 % saveas(gcf,'FormelderKapazität','pdf')
372
373
374 %%
375 % % 20°C Gesamtekapazität durchgehende Entladung
376 load('C5_discharge_20deg_stufe.mat')
377 fa = interp1(dod_20deg,ocv_20,cdd_20deg_stufe) ;
378 k=isnan(fa);
379 fa(k)=fa(k(end)+1);
380 time_20=t;
381
382 u_20_1 = interp1(dod_20deg,v_c_try_20_1,cdd_20deg_stufe);
383 u_20_2 = interp1(dod_20deg,v_c_try_20_2,cdd_20deg_stufe);
384
385 u_4 =zeros(1, length(ccd_20deg_stufe));
386 u_4(1)= cvd_20deg_stufe(1);
387 u_5 =zeros(1, length(ccd_20deg_stufe));
388 u_5(1)= cvd_20deg_stufe(1);
389 for k = 2:length(ccd_20deg_stufe)
390     u_4(k)= (cvd_20deg_stufe(1) + cchd_20deg_stufe(k)...
391             *3600/C_try_20_1) *(1+fa(k));
392     % u_5(k) = u_5(k-1) + ccd_20deg_stufe(k)*1/C_try_1*(1+fa(k))*0.8;
393 end
394
395 % u_6 =zeros(1, length(ccd_20deg_stufe));
396 % u_6 = u_5.*(1+fa);
397 %
398 figure(5)
399 subplot(2,1,1)
400 plot(time_20,cvd_20deg_stufe,time_20,u_20_1...
401      ,time_20,u_20_2,time_20,u_4)
402 legend('Messwert','kleiner Kondensator',...
403        'großer Kondensator','Formel')
404 datetick('x','dd/mm')
405 xlabel('Zeit [TT/MM]')
406 ylabel('Spannung [V]')
407 title('Spannung')
408 grid
409
410 subplot(2,1,2)
411 plot(time_20,ccd_20deg_stufe)
412 legend('Messwert')
```

```

413 datetick('x','dd/mm')
414 xlabel('Zeit [TT/MM]')
415 ylabel('Strom [A]')
416 title('Strom')
417 grid
418
419 % title({'\fontsize{14} Terminal Voltage of stage-Discharge...
420 % and Voltage of Capacitor'; 'Discharge with 4.5A in Ambient';
421 % 'Comparsion with Capcitor during the Discharge';...
422 % '\fontsize{10}Terminal Voltage'});
423 % saveas(gcf,'angepassteOCV_20deg','pdf')
424
425 %% 50°C Gesamtkapazität durchgehende Entladung
426 load('C5_int_dis_4A5_1V75_all.mat')
427 time_50deg_1 = (comb_cur_data(:,1)- comb_cur_data(1,1))*24*3600;
428 duration_50deg_1 = round(time_50deg_1(end))-1;
429 lange = 1:1:duration_50deg_1;
430 ccd_50deg = interp1(time_50deg_1,comb_cur_data(:,2),lange);
431 cvd_50deg = interp1(time_50deg_1,comb_volt_data(:,2),lange);
432 ctd_50deg = interp1(time_50deg_1,comb_temp_data(:,2),lange);
433 cchd_50deg = interp1(time_50deg_1,comb_charge_data,lange);
434 time_50deg = interp1(time_50deg_1,comb_cur_data(:,1),lange);
435
436 cdd_50deg_stufe = cchd_50deg/cchd_50deg(end);
437 fa = interp1(dod_50deg,ocv_50_1,cdd_50deg_stufe);
438 fa(1:57)=0;
439
440 u_50_1 = interp1(dod_50deg,v_c_try_50_1,cdd_50deg_stufe);
441 u_50_2 = interp1(dod_50deg,v_c_try_50_2,cdd_50deg_stufe);
442 %
443 u_4=zeros(1, length(ccd_50deg));
444 u_4(1)= cvd_50deg(1);
445 u_5=zeros(1, length(ccd_50deg));
446 u_5(1)= cvd_50deg(1);
447 for k = 5:length(ccd_50deg)
448     u_4(k)= (cvd_50deg(1) + cchd_50deg(k)*3600/...
449     C_try_50_1) *(1+fa(k));
450     % u_5(k) = u_5(k-1) + ccd_50deg_stufe(k)*1/C_try_1*(1+fa(k))*0.8;
451 end
452 u_4(2:4) = u_4(5);
453 % u_6=zeros(1, length(ccd_50deg_stufe));
454 % u_6 = u_5.*(1+fa);
455
456 figure(5)

```

```
457 subplot(2,1,1)
458 plot(time_50deg, cvd_50deg, time_50deg, u_50_1, ...
459 time_50deg, u_50_2, time_50deg, u_4)
460 legend('Messwert', 'kleiner Kondensator', ...
461 'großer Kondensator', 'Formel')
462 datetick('x', 'dd/mm')
463 xlabel('Zeit [TT/MM]')
464 ylabel('Spannung [V]')
465 title('50°C')
466 % title({'\fontsize{14} Terminal Voltage of stage-Discharge...
467 % and Voltage of Capacitor'; 'Discharge with 4.5A in Ambient'; ...
468 % 'Comparsion with Capcitor during the Discharge'; ...
469 % '\fontsize{10}Terminal Voltage'});
470 grid
471
472 subplot(2,1,2)
473 plot(time_50deg, ccd_50deg)
474 legend('Messwert')
475 datetick('x', 'dd/mm')
476 xlabel('Zeit [TT/MM]')
477 ylabel('Strom [A]')
478 title('Strom')
479 grid
480 % saveas(gcf, 'angepassteOCV_50deg', 'pdf')
481
482 %% -2°C gesamtkapazität durchgehende Entladung
483 load('C5_discharge_m2deg_4A5_stage_all_in_all.mat')
484 cdd_m2deg_stufe = cchd_m2deg/cchd_m2deg(end);
485 fa = interp1(dod_m2deg, ocv_m2_1, cdd_m2deg_stufe) ;
486 fa(1:57)=0;
487 % u_m2_1 = interp1(dod_m2deg, v_c_try_m2_1, cdd_m2deg_stufe);
488 % u_m2_2 = interp1(dod_m2deg, v_c_try_m2_2, cdd_m2deg_stufe);
489
490 u_4 = zeros(1, length(ccd_m2deg));
491 u_m2_2 = zeros(1, length(ccd_m2deg));
492 u_m2_1 = zeros(1, length(ccd_m2deg));
493 u_4(1) = cvd_m2deg(1);
494 u_m2_2(1) = cvd_m2deg(1);
495 u_m2_1(1) = cvd_m2deg(1);
496
497 for k = 1:length(ccd_m2deg)
498     u_4(k) = (cvd_m2deg(1) + cchd_m2deg(k)*3600/(C_try_m2_2))...
499     * (1+fa(k));
500 end
```

```

501
502 % C_try_m2_1 = C_try_m2_1 *0.9;
503 % C_try_m2_2 = C_try_m2_2 *0.9;
504 for k = 2:length(ccd_m2deg)
505 %   u_m2_1(k) = u_m2_1(k-1) + ccd_m2deg(k)/(C_try_m2_1*0.68);
506 u_m2_1(k) = (cvd_m2deg(1) + cchd_m2deg(k)*3600/(C_try_m2_1));
507 end
508
509 for k = 2:length(ccd_m2deg)
510   u_m2_2(k) = (cvd_m2deg(1) + cchd_m2deg(k)*3600/(C_try_m2_2));
511 end
512 %
513 % u_m2_1 = interp1(dod_m2deg,v_c_try_m2_1,cdd_m2deg_stufe);
514 % u_m2_2 = interp1(dod_m2deg,v_c_try_m2_2,cdd_m2deg_stufe);
515 % u_m2_1(1:56)=cvd_m2deg(1);
516 % u_m2_2(1:56)=cvd_m2deg(1);
517
518 % u_5(k) = u_5(k-1) + ccd_m2deg_stufe(k)*1/C_try_1*(1+fa(k))*0.8;
519 % u_6=zeros(1, length(ccd_m2deg_stufe));
520 % u_6 = u_5.*(1+fa);
521 %
522 figure(5)
523 subplot(2,1,1)
524 plot(time_m2deg,cvd_m2deg,time_m2deg,u_m2_1,...
525 time_m2deg,u_m2_2,time_m2deg,u_4)
526 legend('Messwert.m2°C','kleiner Kondensator',...
527 'großer Kondensator','angepasstes Element')
528 datetick('x','dd/mm')
529 xlabel('Zeit [TT/MM]')
530 ylabel('Spannung [V]')
531 % title({'\fontsize{14} Terminal Voltage of stage-Discharge and...
532 % Voltage of Capacitor';'Discharge with 4.5A in Ambient';...
533 %'Comparsion with Capcitor during the Discharge';...
534 %'\fontsize{10}Terminal Voltage'});
535 grid
536 subplot(2,1,2)
537 plot(time_m2deg,ccd_m2deg)
538 legend('Messwert')
539 datetick('x','dd/mm')
540 xlabel('Zeit [TT/MM]')
541 ylabel('Strom [A]')
542 title('Strom')
543 grid
544 % saveas(gcf,'angepassteOCV_m2deg','pdf')

```

C.13. Kalman-Filter mit verschiedener Varianz

Listing C.13: Kalman-Filter mit verschiedener Varivanz

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % In diesem Code wird das Kalman-Filter-Modell mit
3  % verschiedenen Varianz betrieben.
4  % und dazugehöriges Ergebnis wird angezeigt.
5
6  clear all;
7  clc;
8  close all;
9  r_3d=[];
10 i_3d =[];
11 abweichung_3d=[];
12 % Varianz Schleife
13 for test = 1:77
14 % load('C5_discharge_50deg.mat')
15 load('C5_discharge_stage.mat')
16 % load('C5_discharge_trasient.mat')
17 % load('C5_discharge_20deg_4A5_20Ah.mat')
18 load('Parameter_Csurface_R.mat')
19
20 % num_samples = length(u_im);
21 % Tc_vec = ones(1,num_samples);
22 % % zeit=t-t(1);
23 % zeit=t;
24 % T=20;
25 % %Künstliche erstellte Rauschen
26 % % ADC Step
27 % ADC_res = 10;
28 % ADC_ref = 2.5;
29 % ADC_step = 2.5/(2^10);
30 % ADC_voltage_divider = 2/3;
31 %
32 % u_im_ADC = zeros(num_samples,1);
33 % noise = random('normal',0,0.0005,10,num_samples);
34 % disp('Berechne verrauschtes ADC-Signal...')
35 % for k = 1:1:num_samples
36 %     if ~mod(k,1000)
37 %         disp([char(8) char(8) char(8) char(8) char(8)...
38 % char(8) char(8) char(8) char(8) char(8) num2str(k)])
39 %     end
40 %     u_im_ADC(k) = mean(round(((u_im(k)*ones(10,1)+noise(:,k))...

```

```
41 % *ADC_voltage_divider)/ADC_step)*ADC_step*3/2);
42 % end
43 % % verrauschte Messwert
44 % u_im = u_im_ADC;
45 %
46 % figure(2)
47 % plot(1:num_samples,u_im,'r',1:num_samples,Vcs_s,'b',...
48 % 1:num_samples,Vcb_s,'black')
49 % legend('Uim(Klemmenspannung)', 'Vcs', 'Vcb')
50 % title('verrauschter Simulationswert der Klemmenspannung')
51 % grid;
52 % saveas(gcf,'verrauschter Simulationswert...
53 % der Klemmenspannung','png')
54
55 % %verschiedene Varianz des Messrauschens
56 % q = [0.001 0.01 0.1 0.001]; % größer, vertrauen mehr auf Messwert
57 % R = 100;
58 % for times=1:5
59 %     if times == 1
60 %         iter = 2;
61 %     else if times == 2
62 %         iter =10;
63 %     else if times == 3
64 %         iter = 30;
65 %     else if times == 4
66 %         iter = 60;
67 %     else
68 %         iter = 120;
69 %     end
70 %     end
71 %     end
72 %     end
73
74 % % Load Messrauschen
75 load(['Varianz_' num2str(test) '.mat'])
76 q = Q;
77 % % Systemrauschen
78 Q=[q(1) 0 0 0;0 q(2) 0 0;0 0 q(3) 0; 0 0 0 q(4)]; % Systemrauschen
79 R = R;
80 % % Abstand der Korrektur
81 iter = I;
82
83 % 20°C DYNAMIK
84 % dod = 1 -(u_im(1) - 1.93) / 0.18; %
```

```

85 % Cbulk = 7.20e5;
86 % % Entladen
87 % Cbulk = 7.3e5;
88 % dod = 1 - (u_im(1) - 1.93)/0.19;
89 % % STUFE
90 Cbulk = 7.28e5;
91 dod = 1 - (u_im(1) - 1.927) / 0.182; %% 20°C STUFE
92 %%% 50°C
93 % u_im = cvd_50deg;
94 % i_im = ccd_50deg;
95 % t = t_50;
96 % Cbulk = 7.6e5;
97 % dod = 1 - (u_im(1) - 1.907) / 0.207; % % 50°C ENTLADEN
98   Rs = parameter_g(1);
99   Csurface = parameter_g(2);
100   % Cbulk = 7.3e5;
101   % Rt = 0;
102   % dod= 1- (u_im(1)-1.95)/.2;
103   % Re = 0.0137 + 0.0121*dod^6.0529;% Formel 2
104   %num_samples =length(u_im);
105   num_samples = length(u_im);
106   Tc_vec = ones(1,num_samples);
107   % zeit=t-t(1);
108   zeit=t;
109   T=20;
110
111 %   Ein paar Ausgaben zur Übersicht...
112 %   disp(['tau_cs = ' num2str(Rs * Csurface)]);
113 %   disp(['Re || Rs = ' num2str(Re*Rs / (Re + Rs))]);
114 %   disp(['Re + Rt = ' num2str(Re + Rt)]);
115
116 % Zeitinvariante Messmatrix
117   C = [0 0 1 0];
118   % C = [0 0 1];
119   % Anfangszustände
120   x=[u_im(1);u_im(1);u_im(1);0]; % Zustandsraum
121   %x=[u_im(1);u_im(1);u_im(1)]; % Zustandsraum
122
123   x1 = x;
124   P = zeros(4);           % Varianz des Schätzfehlers
125   %P = zeros(3);         % Varianz des Schätzfehlers
126
127   % Vektor mit Sampleintervallen berechnen
128   % disp([char(13) 'Berechne Sampleintervalle...']);

```

```
129 % Tc_vec = zeros(num_samples,1);
130 % Tc_vec(1) = 0.25;
131 % for k = 2:1:num_samples
132 %     t_diff = datevec(t(k) - t(k-1));
133 %     Tc_vec(k,1) = t_diff(1,6);
134 % end
135 % disp([char(13) 'Berechnung der Sampleintervalle...
136 % abgeschlossen']);
137
138 % Tc_vec(1) = 0;
139 % for k = 2:1:num_samples
140 %     Tc_vec(k,1) = t(k) - t(k-1);
141 % end
142 % disp([char(13) 'Berechnung der Sampleintervalle...
143 % abgeschlossen']);
144
145 % Matrizen zum speichern festlegen
146 xV = zeros(4,num_samples); % Zustandsvektor aus Schätzung
147 sV = zeros(5,num_samples); % Zustandsvektor aus Korrektur
148 zV = zeros(1,num_samples); % Vektor mit geschätzten Messwerten
149
150 test_p = [];
151 test_k = [];
152 ab = [];
153 dod1=[];
154 r=[];
155
156 %
157 % Vorbereitungen abgeschlossen:
158 disp([char(13) 'Kalmanfilteralgorithmus für Messung 1 mit...
159 ' num2str(num_samples) ' Samples..'']);
160 % Kalman Filter algorithmus
161 for k=1:num_samples
162     %% Phase der Prädiktion
163     % aktuelles Sampleintervall aus Vektor nehmen
164     Tc = Tc_vec(k);
165
166     % BerechnungdesaktuellenWiderstandsmitderangepasstenFormel
167     Re = 0.014 +0.0246*dod^5.9777 -0.0007*(1-dod)^0.0447 ...
168     - 0.0001*(T-20);
169
170     % Zuordnung der Zustandsübergangsmatrix
171     a11 = 1;
172     a12 = 0;
```

```

173     %a11 = 1 - Tc / (Cbulk * (Re + Rs));
174     %a12 = 1 / (Cbulk * (Re + Rs));
175     a13 = 0;
176     a14 = 0;
177
178     a21 = Tc / (Csurface * (Re + Rs));
179     a22 = 1 - Tc / (Csurface * (Re + Rs));
180     a23 = 0;
181     a24 = 0;
182
183     a31 = (-1) * Tc * Rs / (Cbulk * (Re + Rs)^2) + Tc * Re...
184           / (Csurface * (Re + Rs)^2) - Tc * Rs^2 / (Cbulk * Re * ...
185           (Re + Rs)^2) + Tc * Rs / (Csurface * (Re + Rs)^2);
186     a32 = 0;
187     a33 = 1 + Tc * Rs / (Cbulk * Re * (Re + Rs)) - Tc ...
188           / (Csurface * (Re + Rs));
189     a34 = -1*((Re*Rs)/(Re+Rs));
190
191     a41 = 0;
192     a42 = 0;
193     a43 = 0;
194     a44 = 0;
195
196     A = [a11 a12 a13 a14;a21 a22 a23 a24;a31 a32 a33 a34;...
197         a41 a42 a43 a44];
198
199     % Zuordnung der Ausgabematrix
200     b1 = Tc/Cbulk;
201     b2 = Tc * Re / (Csurface * (Re + Rs));
202     b3 = Tc * Re^2 / (Csurface * (Re + Rs)^2) + Tc * Re * Rs...
203           / (Csurface * (Re + Rs)^2) + (Re*Rs/(Re+Rs));
204     b4 = 1;
205
206     B= [b1;b2;b3;b4];
207     % Berechnung der Spannung des Kondensators
208     x(1) = (A(1,:) * x + B(1) * i_im(k)) ;
209
210     % Multiplikation der Spannung des Kondensators
211     % mit der angepassten Formel
212     m = x(1) * (1 + (-0.117 * dod * (1 - dod) ^ (2.348) ...
213           + 0.0212 * dod ^ 0.4096 + (T - 20) * (-0.0156 * dod ^ 1.7468 ...
214           + 0.0161 * dod ^ 1.6306));
215
216     % Berechnung der übrigen Zustandsvariablen

```

```
217     n=[m; x(2:4)];
218     x(2:4) = A(2:4,:) * n + B(2:4) * i_im(k);
219     % Berechnung des aktuellen Ladezustands
220     dod = 1 - (m - 1.927) / 0.182;
221     z1 = x(3);
222     % Berechnung der Fehlerkovarianzmatrix
223     P = A * P * A' + Q;
224     % Abspeichern der Zustandsvariablen
225     sV(:,k) = [x(1); m; x(2:4)];
226     zV(k) = z1;
227
228     %% Phase der Korrektur
229     if (rem(k, iter) == 0)
230         % Übergabe der aktuellen Zustandsvariablen
231         x1 = x;
232         % Berechnung des Kalman Gains
233         K = (P * C') / (C * P * C' + R);
234         % Korrektur der Zustandsvariablen
235         x = x1 + K * (u_im(k) - z1);
236         % Übergabe des korrigierten Zustandsvariablen
237         x1 = x;
238         % Korrektur der Fehlerkovarianzmatrix
239         P = P - K * C * P;
240         % Abpeicherung des korrigierten Zustandsvariablen
241         xV(:,k) = x1;
242     else
243         % Abpeicherung des letzt korrigierten Zustandsvariablen
244         xV(:,k) = x1;
245     end
246
247     end
248     abweichung = (u_im - sV(4,:)) ./ 0.15 * 100;
249     ab = mean(abs(abweichung));
250
251     r_3d = [r_3d; R];
252     i_3d = [i_3d; iter];
253     abweichung_3d = [abweichung_3d; ab];
254     %
255     % Plot
256     % scrsz = get(0, 'ScreenSize');
257     % figure(1)
258     % set(gcf, 'Position', [1 1 1900 1e3]);
259     % plot(zeit, u_im, zeit, sV(4,:), zeit, sV(2,:), zeit, sV(3,:));
260     % legend('Terminal.Voltage.Measure', 'Terminal.Voltage.Kalman', ...
```

```

261 %'Voltage."Capacitor".Kalman','Voltage.Csurface')
262 % title({'\fontsize{14}Comparison of voltages with...
263 % model including RC correct-function']...
264 % ['System noise Q= [' num2str(q(1)),'...
265 % ',num2str(q(2)),' ',num2str(q(3)),'...
266 % ',num2str(q(4)),'']; Measure Noise R = ' num2str(R) ';...
267 % Iteration I = ' num2str(iter) ]...
268 %,['Discharge with constant current 5A in the duration...
269 % about 9 hrs in RT']})
270 % datetick('x','HH')
271 % xlabel('Time in [HH]')
272 % ylabel('Voltage in [V]')
273 % grid
274 %
275 % MaxDiff = max(abs(u_im - sV(4,:)));
276 % TotalDiff = sum(abs(u_im -sV(4,:)));
277 %
278 % save(['C5_discharge_c_r_Q_' num2str(q(1)),'...
279 % '_', num2str(q(2)),'_',...
280 % num2str(q(3)),'_',num2str(q(4)),'_R',...
281 % num2str(R),'_I',num2str(iter),'_mat']...
282 % ,'sV','xV','MaxDiff','TotalDiff')
283 % saveas(gcf,['Foto_C5_discharge_c_r_Q_' num2str(q(1))...
284 % ',_', num2str(q(2)),'_',...
285 % num2str(q(3)),'_',num2str(q(4)),'_R', num2str(R),'_I'...
286 % ,num2str(iter) '.png'],'png')
287 % load('systemwert_discharge.mat')
288 % % scrsz = get(0,'ScreenSize');
289 % hold on
290 % figure(5)
291 % % set(gcf,'Position',[1 1 1900 1e3]);
292 % plot(zeit,u_im,zeit,sV(4,:),zeit,sV(2,:),...
293 % zeit,sV(3,:),zeit,sV_system(4,:));
294 % legend('Terminal.Voltage.Measure',...
295 % 'Terminal.Voltage.Kalman',...
296 %'Voltage."Capacitor".Kalman','Voltage.Csurface','Systemwert')
297 % title({'\fontsize{14}Comparison of voltages with model...
298 % including RC correct-function']...
299 % ['System noise Q= [' num2str(q(1)),'...
300 % ',num2str(q(2)),' ',num2str(q(3)),'...
301 % ',num2str(q(4)),'']; Measure Noise R ...
302 % = ' num2str(R) '; Iteration I = ' num2str(iter) ]...
303 % ,['Discharge with constant current 5A in the duration...
304 % about 9 hrs in RT']})

```

```

305 % datetick('x','HH')
306 % xlabel('Time in [HH]')
307 % ylabel('Voltage in [V]')
308 % grid
309 %
310 % load('systemwert_discharge.mat')
311 % % scrsz = get(0,'ScreenSize');
312 % hold on
313
314 figure(5)
315 subplot(2,1,1)
316 plot(zeit,u_im,zeit,sV(4,:),zeit,sV(2,:));
317 legend('Messwert','KF-Klemmenspannung','KF-Ruhespannung')
318 title({'\fontsize{14}Comparison of voltages with model...
319 including RC correct-function'}...
320 ['System noise Q= [' num2str(q(1))',' ',num2str(q(2)),'...
321 ',num2str(q(3))',' ',num2str(q(4)),'']; Measure Noise R =...
322 ' num2str(R)'; Iteration I = ' num2str(iter) ]...
323 ,['Discharge with constant current 5A in the duration...
324 about 9 hrs in RT']})
325 title('Spannung')
326 datetick('x','dd/mm','keeplimits')
327 xlabel('Zeit [TT/MM]')
328 ylabel('Spannung [V]')
329 grid
330
331 subplot(2,1,2)
332 plot(zeit,abweichung);
333 legend('Abweichung')%,'Voltage.Csurface.Kalman'
334 title({'\fontsize{14}Comparison of voltages with model...
335 including RC correct-function'}...
336 ['System noise Q= [' num2str(q(1))',' ',num2str(q(2)),'...
337 ',num2str(q(3))',' ',num2str(q(4)),'']; Measure Noise R...
338 = ' num2str(R) '; Iteration I = ' num2str(iter) ]...
339 ,['Discharge with constant current 5A in the duration...
340 about 9 hrs in RT']})
341 title('Abweichungrate')
342 datetick('x','dd/mm','keeplimits')
343 xlabel('Zeit [TT/MM]')
344 ylabel('Abweichungrate [%]')
345 grid
346 % Save
347 saveas(gcf,['entldung_20deg_r_c_R' num2str(R)...
348 '_Iter' num2str(iter)],'pdf')

```

```
349     save(['C5_stage_discharge_r_c' num2str(test)...
350         '.mat'], 'Q', 'R', 'iter', 'sV', 'xV', 'Diff')
351     save(['entldung_20deg_r_c_R' num2str(R) '_Iter'...
352         num2str(iter) '.mat'], 'sV', 'abweichung')
353
354
355
356 end
357
358 %% 3d PLOT- Abweichung der durchschnittlichen Fehler
359 hold on
360 as=1;
361 bs=7;
362 za= 7;
363 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
364 as = as+za;
365 bs=bs+za;
366 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
367 as = as+za;
368 bs=bs+za;
369 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
370 as = as+za;
371 bs=bs+za;
372 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
373 as = as+za;
374 bs=bs+za;
375 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
376 as = as+za;
377 bs=bs+za;
378 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
379 as = as+za;
380 bs=bs+za;
381 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
382 as = as+za;
383 bs=bs+za;
384 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
385 as = as+za;
386 bs=bs+za;
387 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
388 as = as+za;
389 bs=bs+za;
390 plot3(r_3d(as:bs), i_3d(as:bs), abweichung_3d(as:bs), '-*')
391 as = as+za;
392 bs=bs+za;
```

```
393 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
394 % grid(gca,'minor')
395 grid
396 hold off
397 xlabel('R','FontSize',16)
398 ylabel('Iteration der Korrektur','FontSize',16)
399 zlabel('Fehlerate [%]','FontSize',16)
400
401 %% Save
402 saveas(gcf,'3d_20deg_dynamik','pdf')
403
404 %% Stem
405 figure(2)
406 stem3(r_3d,i_3d,abweichung_3d,'r')
407 xlabel('Measure Noise','fontsize',14)
408 ylabel('iteration','fontsize',14)
409 zlabel('MaxDiff in [V]','fontsize',14)
410 title('\fontsize{16} Comparson of the Max.Deviation with...
411 different Measure Noise and Iteration of the correct phase')
412 legend('\fontsize{16} R&C in Formel',...
413 '\fontsize{16}R in Formel C = constant')
414
415
416 figure(2)
417
418 stem3(R,iter,MaxDiff)
419 xlabel('Measure Noise','fontsize',12)
420 ylabel('iteration','fontsize',12)
421 zlabel('MaxDiff in [V]','fontsize',12)
422 title('\fontsize{14}Comparson of the Max.Deviation with ...
423 different Measure Noise and Iteration of the correct phase')
424 hold on
425 %
426 figure(3)
427 stem3(R,iter,SumDiff/length(Diff(1,:))*1000)
428 xlabel('Measure Noise','fontsize',12)
429 ylabel('iteration','fontsize',12)
430 zlabel('AvgDiff in [mV]','fontsize',12)
431 title('\fontsize{14}Comparson of the Avg.Deviation with...
432 different Measure Noise and Iteration of the correct phase')
433 hold on;
434
435 % clear
436 load('dynamik_entldung_20deg_r_c_R100_Iter2.mat')
```

```
437 sV_2 = sV;
438 abweichung_2 = abweichung;
439 load('dynamik_entldung_20deg_r_c_R100_Iter10.mat')
440 sV_10 = sV;
441 abweichung_10 = abweichung;
442 load('dynamik_entldung_20deg_r_c_R100_Iter30.mat')
443 sV_30 = sV;
444 abweichung_30 = abweichung;
445 load('dynamik_entldung_20deg_r_c_R100_Iter60.mat')
446 sV_60 = sV;
447 abweichung_60 = abweichung;
448 load('dynamik_entldung_20deg_r_c_R100_Iter120.mat')
449 sV_120 = sV;
450 abweichung_120 = abweichung;
451 figure(222)
452 subplot(2,1,1)
453 plot(zeit,u_im,zeit,sV_2(4,:),'black',zeit,sV_10(4,:),'r',...
454 zeit,sV_30(4,:),'b',zeit,sV_60(4,:),'v',zeit,sV_120(4,:),'g');
455 title({'Absoluter Fehler : 200 mV';'R = 10 ...
456 Q = [0.001 0.01 0.1 0.001]'}))
457 legend('Iteration = 2s','Iteration = 10s','Iteration = 30s',...
458 'Iteration = 60s','Iteration = 120s')
459 datetick('x',15,'keeplimits')
460 % legend('Messwert','Iteration = 2s','Iteration = 600s')
461 xlabel('Zeit [HH:MM]')
462 ylabel('Spannung')
463
464 grid
465 subplot(2,1,2)
466 % plot(zeit,abweichung_2,'black',zeit,abweichung_600,'r');
467 plot(zeit,abweichung_2,'black',zeit,abweichung_10,'r',...
468 zeit,abweichung_30,'b',zeit,abweichung_60,'v'...
469 ,zeit,abweichung_120,'g');
470
471 datetick('x',15,'keeplimits')
472 xlabel('Zeit [HH:MM]')
473 ylabel('Abweichungsrate')
474 grid
475 legend('Iteration = 2s','Iteration = 10s','Iteration = 30s',...
476 'Iteration = 60s','Iteration = 120s')
477 saveas(gcf,'dynamik_entladung_20deg_R10_ITER_Absolute_Fehler'...
478 , 'pdf')
479
480 %%
```

```

481 % zeit=zeit-zeit(1);
482 load('dynamik_entldung_20deg_r_c_R1000_Iter2.mat')
483 sV_2 = sV;
484 abweichung_2 = abweichung;
485 load('dynamik_entldung_20deg_r_c_R1000_Iter60.mat')
486 sV_60 = sV;
487 abweichung_60 = abweichung;
488 load('dynamik_entldung_20deg_r_c_R1000_Iter600.mat')
489 sV_600 = sV;
490 abweichung_600 = abweichung;
491 figure(222)
492 subplot(2,1,1)
493 plot(zeit,u_im,zeit,sV_2(4,:), 'black',zeit,sV_60(4,:)...
494 , 'g',zeit,sV_600(4,:), 'r');
495 title({'Absoluter Fehler: 200 mV'; 'R = 1000...
496   Q = [0.001 0.01 0.1 0.001] '})
497 datetick('x', 'HH:MM', 'keeplimits')
498 legend('Messwert', 'Iteration = 2s', 'Iteration = 60s'...
499 , 'Iteration = 600s')
500 xlabel('Zeit [HH:MM]')
501 ylabel('Spannung')
502 grid
503 subplot(2,1,2)
504 plot(zeit,abweichung_2, 'black',zeit,abweichung_60, 'g'...
505 ,zeit,abweichung_600, 'r');
506 legend('Iteration = 2s', 'Iteration = 60s', 'Iteration = 600s')
507 datetick('x', 15, 'keeplimits')
508 xlabel('Zeit [HH:MM]')
509 ylabel('Abweichungsrate')
510 grid
511 saveas(gcf, 'dynamik_entldaung_20deg_R1000_ITER_Absolute_Fehler'...
512 , 'pdf')

```

C.14. 3d-Plot (Farbe)

Listing C.14: 3d-Plot (Farbe)

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In diesem Code wird die Abweichung bei verschiedenen Einstellung
3 % des Messrauschens bzw. des Abstands der Korrektur dargestellt.
4
5 % load('3d_20deg_dynamik.mat')

```

```
6 x=i_3d;
7 y=r_3d;
8 z=abweichung_3d;
9 F = TriScatteredInterp(x,y,z);
10 [qx,qy] = meshgrid(linspace(min(x),max(x),20)...
11 ,linspace(min(y),max(y),20));
12 qz = F(qx,qy);
13 surf(qx,qy,qz)
14 colormap Jet
15 colorbar
16 xlabel('R','FontSize',16)
17 ylabel('Abstand der Korrektur [s]','FontSize',16)
18 zlabel('Fehlerrate [%]','FontSize',16)
19 grid
20 % saveas(gcf,'1','pdf')
```

C.15. Abweichung des Sensors

Listing C.15: Abweichung des Sensors

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % In diesem Code wird die Abweichung der einzelnen Sensoren
3 % bzw. des gesamten Sensors angezeigt.
4
5 % gemessene Rauschen
6 load('noise_array_2.mat')
7 % mw = mean(sorted_meas_array(:,4).^2);
8 % max = max(sorted_meas_array(:,4));
9 % min = min(sorted_meas_array(:,4));
10 k= find(sorted_meas_array(:,1)==31);
11 a1= sorted_meas_array(k,4);
12 k= find(sorted_meas_array(:,1)==32);
13 a2= sorted_meas_array(k,4);
14 k= find(sorted_meas_array(:,1)==33);
15 a3= sorted_meas_array(k,4);
16 k= find(sorted_meas_array(:,1)==34);
17 a4= sorted_meas_array(k,4);
18 k= find(sorted_meas_array(:,1)==35);
19 a5= sorted_meas_array(k,4);
20 k= find(sorted_meas_array(:,1)==36);
21 a6= sorted_meas_array(k,4);
22
```

```
23 %% Plot - histodiagramm
24 % x-Achse : -2,5 bis 5,9
25 % y-Achse : 0 bis 800
26 xx= -2.5:0.5:5.9
27 figure(1)
28 subplot(2,3,1)
29 hist(a1,xx)
30 ylim([0 800]);
31 grid
32 xlabel('Abweichung [mV]')
33 ylabel('Anzahl der Messpunkte')
34 title('Sensor1')
35 subplot(2,3,2)
36 hist(a2,xx)
37 ylim([0 800]);
38 grid
39 xlabel('Abweichung [mV]')
40 ylabel('Anzahl der Messpunkte')
41 title('Sensor2')
42 subplot(2,3,3)
43 hist(a3,xx)
44 ylim([0 800]);
45 grid
46 xlabel('Abweichung [mV]')
47 ylabel('Anzahl der Messpunkte')
48 title('Sensor3')
49 subplot(2,3,4)
50 hist(a4,xx)
51 ylim([0 800]);
52 grid
53 xlabel('Abweichung [mV]')
54 ylabel('Anzahl der Messpunkte')
55 title('Sensor4')
56 subplot(2,3,5)
57 hist(a5,xx)
58 ylim([0 800]);
59 grid
60 xlabel('Abweichung [mV]')
61 ylabel('Anzahl der Messpunkte')
62 title('Sensor5')
63 subplot(2,3,6)
64 hist(a6,xx)
65 ylim([0 800]);
66 grid
```

```

67 xlabel('Abweichung [mV]')
68 ylabel('Anzahl der Messpunkte')
69 title('Sensor6')
70 % save
71 saveas(gcf,'AbweichungdesSensors','pdf')
72 %%
73 figure(2)
74 hist(sorted_meas_array(:,4),xx)
75 grid
76 xlabel('\fontsize{16}Abweichung [mV]')
77 ylabel('\fontsize{16}Anzahl der Messpunkte')
78 saveas(gcf,'AbweichungdesgesamtenSensors','pdf')

```

C.16. Kalman-Filter mit dem verrauschten Messwert

Listing C.16: Kalman-Filter mit dem verrauschten Fehler

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % In diesem Code wird das Ergebnis des Kalman-Filter
3  % -Modell mit verrauschten Messwert als Eingabe dargestellt.
4
5  clear all;
6  clc;
7  close all;
8  r_3d=[];
9  i_3d =[];
10 abweichung_3d=[];
11
12 % load('C5_discharge_50deg.mat')
13 % load('C5_dynamik_50deg.mat')
14 % load('C5_discharge_trasient.mat')
15 load('C5_stufe_50deg.mat')
16 load('Parameter_Csurface_R.mat')
17
18 t = time_50deg;
19 u_im = cvd_50deg;
20 i_im = ccd_50deg;
21
22 num_samples = length(u_im);
23 a = randperm(num_samples);
24 a1 = round(num_samples*495/10000);
25 u_im(a(1:a1))=u_im(a(1:a1));

```

```
26
27 % Messwert verrauschen
28 a2 = round(num_samples*1.37e3/10000);
29 k=a1+a2;
30 u_im(a(a1:k))=u_im(a(a1:k)); - 0.00117;
31
32 a3 = round(num_samples*1.87e3/10000);
33 k=a1+a2+a3;
34 u_im(a(a1+a2:k))=u_im(a(a1+a2:k)) -0.00035;
35
36 a4 = round(num_samples*1.01e3/10000);
37 k=a1+a2+a3+a4;
38 u_im(a(a1+a2+a3:k))=u_im(a(a1+a2+a3:k)) +0.00047;
39
40 a5 = round(num_samples*17/10000);
41 k=a1+a2+a3+a4+a5;
42 u_im(a(a1+a2+a3+a4:k))=u_im(a(a1+a2+a3+a4:k)) +0.00129;
43
44 a6 = round(num_samples*2.89e3/10000);
45 k1=a1+a2+a3+a4+a5;
46 k=k1+a6;
47 u_im(a(k1:k))=u_im(a(k1:k)) +0.00211;
48
49 a7 = round(num_samples*1.95e3/10000);
50 k1=a1+a2+a3+a4+a5+a6;
51 k=k1+a7;
52 u_im(a(k1:k))=u_im(a(k1:k)) +0.00293;
53
54 a8 = round(num_samples*340/10000);
55 k1=a1+a2+a3+a4+a5+a6+a7;
56 k=k1+a8;
57 u_im(a(k1:k))=u_im(a(k1:k)) +0.00375;
58
59 a9 = round(num_samples*49/10000);
60 k1=a1+a2+a3+a4+a5+a6+a7+a8;
61 k=k1+a9;
62 u_im(a(k1:k))=u_im(a(k1:k)) +0.00457;
63
64 a10 = round(num_samples*1/10000);
65 k1=a1+a2+a3+a4+a5+a6+a7+a8+a9;
66 k=k1+a10;
67 u_im(a(k1:k))=u_im(a(k1:k)) +0.00539;
68
69 % % ADC-Rauschen
```

```

70 % ADC_res = 10;
71 % ADC_ref = 2.5;
72 % ADC_step = 2.5/(2^10);
73 % ADC_voltage_divider = 2/3;
74 %
75 % u_im_ADC = zeros(num_samples,1);
76 % noise = random('normal',0,0.0005,10,num_samples);
77 % disp('Berechne verrauschtes ADC-Signal...')
78 % for k = 1:1:num_samples
79 %     if ~mod(k,1000)
80 %         disp([char(8) char(8) char(8) char(8) char(8)...
81 %             char(8) char(8) char(8) char(8) char(8) num2str(k)])
82 %     end
83 %     u_im_ADC(k) = mean(round(((u_im(k)*ones(10,1)+noise(:,k))...
84 %         *ADC_voltage_divider)/ADC_step)*ADC_step*3/2);
85 % end
86 % u_im = u_im_ADC;
87 %
88 % figure(2)
89 % plot(1:num_samples,u_im,'r',1:num_samples,Vcs_s,'b',...
90 % 1:num_samples,Vcb_s,'black')
91 % legend('Uim(Klemmenspannung)', 'Vcs', 'Vcb')
92 % title('verrauschter Simulationswert der Klemmenspannung')
93 % grid;
94 % saveas(gcf,'verrauschterSimulationswertderKlemmenspannung','png')
95 % % Systemsrauschen
96 q = [0.0009 0.1 0.1 0.001]; % größer, vertrauen mehr auf Messwert
97 % % Messrauschen
98 R = 3.4839;
99 % % Abstand der Korrektur
100 iter = 2;
101 Q=[q(1) 0 0 0;0 q(2) 0 0;0 0 q(3) 0; 0 0 0 q(4)]; % Systemrauschen
102
103
104 % 20°C DYNAMIK
105 % dod = 1 -(u_im(1) - 1.93) / 0.18; %
106 % Cbulk = 7.20e5;
107 % Entladen
108 % Cbulk = 7.3e5;
109 % dod = 1 - (u_im(1) - 1.93)/0.19;
110 % % STUFE
111 % Cbulk = 7.28e5;
112 % dod = 1 -(u_im(1) - 1.927) / 0.182; %% 20°C STUFE
113 % Cbulk=7.5e5; % % %=°C DYNAMIK

```

```

114 % dod = 1 - (u_im(1) - 1.905) / 0.208 %% 50°C DYNAMIK
115 Cbulk = 7.6e5;
116 dod = 1 - (u_im(1) - 1.905) / 0.27; %% 50°C STUFE
117 Rs = parameter_g(1);
118 Csurface = parameter_g(2);
119 % Cbulk = 7.3e5;
120 % Rt = 0;
121 % dod= 1- (u_im(1)-1.95)/.2;
122 % Re = 0.0137 + 0.0121*dod^6.0529;% Formel 2
123 %num_samples =length(u_im);
124 num_samples = length(u_im);
125 Tc_vec = ones(1,num_samples);
126 % zeit=t-t(1);
127 zeit=t;
128 T=50;
129 u_im(1) = u_im(1)-0.01;
130
131 % Ein paar Ausgaben zur Übersicht...
132 % disp(['tau_cs = ' num2str(Rs * Csurface)]);
133 % disp(['Re || Rs = ' num2str(Re*Rs / (Re + Rs))]);
134 % disp(['Re + Rt = ' num2str(Re + Rt)]);
135
136 % Abstand der Korrekturen mit gemessen Spannungswerten
137 %iter = 100; % bei upsampled_by_8
138 % iter = 10;
139
140 % Zeitinvariante Messmatrix
141 C = [0 0 1 0];
142 % C = [0 0 1];
143 % Anfangszustände
144 x=[u_im(1);u_im(1);u_im(1);0]; % Zustandsraum
145 %x=[u_im(1);u_im(1);u_im(1)]; % Zustandsraum
146
147 x1 = x;
148 P = zeros(4); % Varianz des Schätzfehlers
149 %P = zeros(3); % Varianz des Schätzfehlers
150
151 % Vektor mit Sampleintervallen berechnen
152 % disp([char(13) 'Berechne Sampleintervalle...']);
153 % Tc_vec = zeros(num_samples,1);
154 % Tc_vec(1) = 0.25;
155 % for k = 2:1:num_samples
156 %     t_diff = datevec(t(k) - t(k-1));
157 %     Tc_vec(k,1) = t_diff(1,6);

```

```
158 % end
159 % disp([char(13) 'Berechnung der Sampleintervalle...
160 % abgeschlossen']);
161
162 % Tc_vec(1) = 0;
163 % for k = 2:1:num_samples
164 %     Tc_vec(k,1) = t(k) - t(k-1);
165 % end
166 % disp([char(13) 'Berechnung der Sampleintervalle...
167 % abgeschlossen']);
168
169 % Matrizen zum speichern festlegen
170 xV = zeros(4,num_samples); % Zustandsvektor aus Schätzung
171 sV = zeros(5,num_samples); % Zustandsvektor aus Korrektur
172 zV = zeros(1,num_samples); % Vektor mit geschätzten Messwerten
173
174 test_p = [];
175 test_k = [];
176 ab = [];
177 dod1=[];
178 r=[];
179
180 %
181 % Vorbereitungen abgeschlossen:
182 disp([char(13) 'Kalmanfilteralgorithmus für Messung 1...
183 mit ' num2str(num_samples) ' Samples..'']);
184 % Kalman Filter algorithmus
185 for k=1:num_samples
186     %% Phase der Prädiktion
187     % aktuelles Sampleintervall aus Vektor nehmen
188     Tc = Tc_vec(k);
189
190     % Berechnung des aktuellen Widerstands...
191     % mit der angepassten Formel
192     Re = 0.014 +0.0246*dod^5.9777 -0.0007*(1-dod)^0.0447...
193         - 0.0001*(T-20);
194
195     % Zuordnung der Zustandsübergangsmatrix
196     a11 = 1;
197     a12 = 0;
198     %a11 = 1 - Tc / (Cbulk * (Re + Rs));
199     %a12 = 1 / (Cbulk * (Re + Rs));
200     a13 = 0;
201     a14 = 0;
```

```

202
203     a21 = Tc / (Csurface * (Re + Rs));
204     a22 = 1 - Tc / (Csurface * (Re + Rs));
205     a23 = 0;
206     a24 = 0;
207
208     a31 = (-1) * Tc * Rs / (Cbulk * (Re + Rs)^2) + Tc * Re...
209           / (Csurface * (Re + Rs)^2) - Tc * Rs^2 /...
210           (Cbulk * Re * (Re + Rs)^2) + Tc * Rs...
211           / (Csurface * (Re + Rs)^2);
212     a32 = 0;
213     a33 = 1 + Tc * Rs / (Cbulk * Re * (Re + Rs))...
214           - Tc / (Csurface * (Re + Rs));
215     a34 = -1*((Re*Rs)/(Re+Rs));
216
217     a41 = 0;
218     a42 = 0;
219     a43 = 0;
220     a44 = 0;
221
222     A = [a11 a12 a13 a14;a21 a22 a23 a24;a31 a32 a33 a34;...
223         a41 a42 a43 a44];
224
225     % Zuordnung der Ausgabematrix
226     b1 = Tc/Cbulk;
227     b2 = Tc * Re / (Csurface * (Re + Rs));
228     b3 = Tc * Re^2 / (Csurface * (Re + Rs)^2) + Tc * Re * Rs...
229           / (Csurface * (Re + Rs)^2) + (Re*Rs/(Re+Rs));
230     b4 = 1;
231
232     B= [b1;b2;b3;b4];
233     % Berechnung der Spannung des Kondensators
234     x(1) = (A(1,:)*x + B(1)*i_im(k)) ;
235
236     % Multiplikation der Spannung des Kondensators...
237     % mit der angepassten Formel
238     m = x(1) * (1+(-0.117*dod*(1-dod)^(2.348)...
239           + 0.0212*dod^0.4096 + (T-20)*(-0.0156*dod^1.7468...
240           +0.0161*dod^1.6306));
241     % Berechnung der übrigen Zustandsvariablen
242     n=[m; x(2:4)];
243     x(2:4) = A(2:4,:)*n + B(2:4)*i_im(k);
244     % Berechnung des aktuellen Ladezustands
245     dod = 1 - (m- 1.905) / 0.27; %% 50°C STUFE

```

```
246     % Übergabe der aktuell geschätzten Ausgabe
247     z1 = x(3);
248     % Berechnung der Fehlerkovarianzmatrix
249     P = A*P*A'+ Q;
250     % Abspeichern der Zustandsvariablen
251     sV(:,k) = [x(1);m;x(2:4)];
252     zV(k) = z1;
253
254     %% Phase der Korrektur
255     if (rem(k,iter) == 0)
256         % Übergabe der aktuellen Zustandsvariablen
257         x1 = x;
258         % Berechnung des Kalman Gains
259         K = (P*C') / (C*P*C'+R);
260         % Korrektur der Zustandsvariablen
261         x = x1 + K*(u_im(k)-z1);
262         % Übergabe des korrigierten Zustandsvariablen
263         x1 = x;
264         % Korrektur der Fehlerkovarianzmatrix
265         P = P-K*C*P;
266         % Abpeicherung des korrigierten Zustandsvariablen
267         xV(:,k) = x1;
268     else
269         % Abpeicherung des letzt korrigierten Zustandsvariablen
270         xV(:,k) = x1;
271     end
272
273 end
274 % Plot
275 % scrsz = get(0,'ScreenSize');
276 % figure(1)
277 % set(gcf,'Position',[1 1 1900 1e3]);
278 % plot(zeit,u_im,zeit,sV(4,:),zeit,sV(2,:),zeit,sV(3,:));
279 % legend('Terminal.Voltage.Measure',...
280 % 'Terminal.Voltage.Kalman','Voltage."Capacitor".Kalman',...
281 % 'Voltage.Csurface')
282 % title(['\fontsize{14}Comparison of voltages with model...
283 % including RC correct-function']...
284 % ['System noise Q= [' num2str(q(1)),' ',num2str(q(2)),' ...
285 % ',num2str(q(3)),' ',num2str(q(4)),'']; Measure Noise R = ...
286 % ' num2str(R) '; Iteration I = ' num2str(iter) ]...
287 % ,['Discharge with constant current 5A in the duration about...
288 % 9 hrs in RT']})
289 % datetick('x','HH')
```

```

290 % xlabel('Time in [HH]')
291 % ylabel('Voltage in [V]')
292 % grid
293 %
294 % MaxDiff = max(abs(u_im - sV(4,:)));
295 % TotalDiff = sum(abs(u_im -sV(4,:)));
296 %
297 %
298 % % save(['C5_discharge_c_r_Q_' num2str(q(1)),'_'...
299 % , num2str(q(2)),'_' , num2str(q(3)),'_' ,...
300 % num2str(q(4)),'_R',num2str(R),'_I',num2str(iter),'_mat']...
301 % , 'sV', 'xV', 'MaxDiff', 'TotalDiff')
302 % saveas(gcf, ['Foto_C5_discharge_c_r_Q_'...
303 % num2str(q(1)),'_' , num2str(q(2)),'_' ,...
304 % num2str(q(3)),'_' , num2str(q(4)),'_R', num2str(R),'_I'...
305 % , num2str(iter) '.png'],'png')
306
307 %
308 % load('systemwert_discharge.mat')
309 % % scrsz = get(0,'ScreenSize');
310 % hold on
311 % figure(5)
312 % % set(gcf,'Position',[1 1 1900 1e3]);
313 % plot(zeit,u_im,zeit,sV(4,:),zeit,sV(2,:))...
314 % ,zeit,sV(3,:),zeit,sV_system(4,:));
315 % legend('Terminal.Voltage.Measure',...
316 % 'Terminal.Voltage.Kalman',...
317 % 'Voltage."Capacitor".Kalman','Voltage.Csurface','Systemwert')
318 % title({'\fontsize{14}Comparison of voltages with model...
319 % including RC correct-function']...
320 % ['System noise Q= [' num2str(q(1)),' ',num2str(q(2)),'...
321 % ',num2str(q(3)),' ',num2str(q(4)),'']; Measure Noise...
322 % R = ' num2str(R) '; Iteration I = ' num2str(iter) ]...
323 % ,['Discharge with constant current 5A in the duration...
324 % about 9 hrs in RT']})
325 % datetick('x','HH')
326 % xlabel('Time in [HH]')
327 % ylabel('Voltage in [V]')
328 % grid
329 %
330 %
331 %
332 % load('systemwert_discharge.mat')
333 % % scrsz = get(0,'ScreenSize');

```

```

334 % hold on
335
336 % Plot
337 figure(5)
338 subplot(2,1,1)
339 plot(zeit,u_im,zeit,sV(4,:),zeit,sV(2,:));
340 legend('Messwert','KF-Klemmenspannung','KF-Ruhespannung')
341 % title({'\fontsize{14}Comparison of voltages with model...
342 including RC correct-function' }...
343 %['System noise Q= [' num2str(q(1)),'...
344 ',num2str(q(2)),' ',num2str(q(3)),'...
345 ',num2str(q(4)),'']; Measure Noise R = ' num2str(R)...
346 '; Iteration I = ' num2str(iter) ]...
347 %,['Discharge with constant current 5A in the duration...
348 about 9 hrs in RT' ]})
349 title('Spannung')
350 datetick('x','dd/mm','keeplimits')
351 xlabel('Zeit [TT/MM]')
352 ylabel('Spannung [V]')
353 grid
354 abweichung=(u_im-sV(4,:))./0.15*100;
355 ab = mean(abs(abweichung));
356 subplot(2,1,2)
357 plot(zeit,abweichung);
358 legend('Abweichung')%,'Voltage.Csurface.Kalman'
359 % title({'\fontsize{14}Comparison of voltages with...
360 model including RC correct-function' }...
361 %['System noise Q= [' num2str(q(1)),' ',num2str(q(2)),'...
362 ',num2str(q(3)),' ',num2str(q(4)),'']; Measure Noise...
363 R = ' num2str(R) '; Iteration I = ' num2str(iter) ]...
364 %,['Discharge with constant current 5A in the duration...
365 about 9 hrs in RT' ]})
366 title('Abweichungrate')
367 datetick('x','dd/mm','keeplimits')
368 xlabel('Zeit [TT/MM]')
369 ylabel('Abweichungrate [%]')
370 grid
371 % Save
372 saveas(gcf,['entldung_20deg_r_c_R' num2str(R)...
373 '_Iter' num2str(iter)],'pdf')
374 save(['C5_stage_discharge_r_c' num2str(test)...
375 '.mat'], 'Q','R','iter','sV','xV','Diff')
376 save(['entldung_20deg_r_c_R' num2str(R) '_Iter'...
377 num2str(iter) '.mat'],'sV','abweichung')

```

```

378     r_3d = [r_3d; R];
379     i_3d = [i_3d; iter];
380     abweichung_3d =[abweichung_3d; ab];
381 %
382     figure(5)
383     subplot(2,1,1)
384     plot(zeit,u_im,'g',zeit,sV(4,:),'r',zeit,sV(2,:),'b');
385     legend('Messwert','KF-Klemmenspannung','KF-Ruhespannung')
386     % title({'\fontsize{14}Comparison of voltages with...
387     % model including RC correct-function'...]...
388     % ['System noise Q= [' num2str(q(1)),' ',num2str(q(2)),'...
389     % ',num2str(q(3)),' ',num2str(q(4)),'']; Measure Noise R...
390     % = ' num2str(R) '; Iteration I = ' num2str(iter) ]...
391     % ,['Discharge with constant current 5A in the duration...
392     % about 9 hrs in RT']})
393     title('Spannung')
394     datetick('x','dd/mm','keeplimits')
395     xlabel('Zeit [TT/MM]')
396     ylabel('Spannung [V]')
397     grid
398
399     subplot(2,1,2)
400     plot(zeit,i_im)
401     legend('Strom')
402     title('Strom')
403     datetick('x','dd/mm','keeplimits')
404     xlabel('Zeit [TT/MM]')
405     ylabel('Strom [A]')
406     grid
407
408 % saveas(gcf,'verrauschterMesswert_50deg_stufe','pdf')
409 %% 3d PLOT
410 hold on
411 as=1;
412 bs=7;
413 za= 7;
414 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
415 as = as+za;
416 bs=bs+za;
417 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
418 as = as+za;
419 bs=bs+za;
420 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
421 as = as+za;

```

```
422 bs=bs+za;
423 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
424 as = as+za;
425 bs=bs+za;
426 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
427 as = as+za;
428 bs=bs+za;
429 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
430 as = as+za;
431 bs=bs+za;
432 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
433 as = as+za;
434 bs=bs+za;
435 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
436 as = as+za;
437 bs=bs+za;
438 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
439 as = as+za;
440 bs=bs+za;
441 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
442 as = as+za;
443 bs=bs+za;
444 plot3(r_3d(as:bs),i_3d(as:bs),abweichung_3d(as:bs),'-*')
445 % grid(gca,'minor')
446 grid
447 hold off
448 xlabel('R','FontSize',16)
449 ylabel('Iteration der Korrektur','FontSize',16)
450 zlabel('Fehlerate [%]','FontSize',16)
451
452 %%
453 saveas(gcf,'asda','pdf')
454
455 %%
456 figure(2)
457 stem3(r_3d,i_3d,abweichung_3d,'r')
458 xlabel('Measure Noise','fontsize',14)
459 ylabel('iteration','fontsize',14)
460 zlabel('MaxDiff in [V]','fontsize',14)
461 title('\fontsize{16} Comparision of the Max.Deviation with...
462 different Measure Noise and Iteration of the correct phase')
463 legend('\fontsize{16} R&C in Formel','\fontsize{16}R...
464 in Formel C = constant')
465
```

```
466 figure(2)
467
468
469 stem3(R,iter,MaxDiff)
470 xlabel('Measure Noise','fontsize',12)
471 ylabel('iteration','fontsize',12)
472 zlabel('MaxDiff in [V]','fontsize',12)
473 title('\fontsize{14}Comparsion of the Max.Deviation with...
474 different Measure Noise and Iteration of the correct phase')
475 hold on
476 %
477 figure(3)
478 stem3(R,iter,SumDiff/length(Diff(1,:))*1000)
479 xlabel('Measure Noise','fontsize',12)
480 ylabel('iteration','fontsize',12)
481 zlabel('AvgDiff in [mV]','fontsize',12)
482 title('\fontsize{14}Comparsion of the Avg.Deviation with ...
483 different Measure Noise and Iteration of the correct phase')
484 hold on;
485
486
487
488 %%
489 % clear
490 load('dynamik_entldung_20deg_r_c_R100_Iter2.mat')
491 sV_2 = sV;
492 abweichung_2 = abweichung;
493 load('dynamik_entldung_20deg_r_c_R100_Iter10.mat')
494 sV_10 = sV;
495 abweichung_10 = abweichung;
496 load('dynamik_entldung_20deg_r_c_R100_Iter30.mat')
497 sV_30 = sV;
498 abweichung_30 = abweichung;
499 load('dynamik_entldung_20deg_r_c_R100_Iter60.mat')
500 sV_60 = sV;
501 abweichung_60 = abweichung;
502 load('dynamik_entldung_20deg_r_c_R100_Iter120.mat')
503 sV_120 = sV;
504 abweichung_120 = abweichung;
505 figure(222)
506 subplot(2,1,1)
507 plot(zeit,u_im,zeit,sV_2(4,:),'black',zeit,sV_10(4,:),'r',...
508 zeit,sV_30(4,:),'b',zeit,sV_60(4,:),'v',zeit,sV_120(4,:),'g');
509 % title({'Relativer Fehler (Sollwert = Messwert)';...
```

```

510 %'R = 10 Q = [0.001 0.01 0.1 0.001] ');
511 title({'Absoluter Fehler : 200 mV';'R = 10...
512 Q = [0.001 0.01 0.1 0.001] '})
513 legend('Iteration = 2s','Iteration = 10s','Iteration = 30s',...
514 'Iteration = 60s','Iteration = 120s')
515 datetick('x',15,'keeplimits')
516 % legend('Messwert','Iteration = 2s','Iteration = 600s')
517 xlabel('Zeit [HH:MM]')
518 ylabel('Spannung')
519
520 grid
521 subplot(2,1,2)
522 % plot(zeit,abweichung_2,'black',zeit,abweichung_600,'r');
523 plot(zeit,abweichung_2,'black',zeit,abweichung_10,'r',...
524      zeit,abweichung_30,'b',zeit,abweichung_60,'v'...
525      ,zeit,abweichung_120,'g');
526
527 datetick('x',15,'keeplimits')
528 xlabel('Zeit [HH:MM]')
529 ylabel('Abweichungsrate')
530 grid
531 legend('Iteration = 2s','Iteration = 10s','Iteration = 30s',...
532 'Iteration = 60s','Iteration = 120s')
533 saveas(gcf,'dynamik_entladung_20deg_R10_ITER_Absolute_Fehler'...
534         ,'pdf')
535
536 %%
537 % zeit=zeit-zeit(1);
538 load('dynamik_entldung_20deg_r_c_R1000_Iter2.mat')
539 sV_2 = sV;
540 abweichung_2 = abweichung;
541 load('dynamik_entldung_20deg_r_c_R1000_Iter60.mat')
542 sV_60 = sV;
543 abweichung_60 = abweichung;
544 load('dynamik_entldung_20deg_r_c_R1000_Iter600.mat')
545 sV_600 = sV;
546 abweichung_600 = abweichung;
547 figure(222)
548 subplot(2,1,1)
549 plot(zeit,u_im,zeit,sV_2(4,:), 'black',zeit,sV_60(4,:)...
550      ,'g',zeit,sV_600(4,:), 'r');
551 % title({'Relativer Fehler (Sollwert = Messwert)';...
552 %'R = 1000 Q = [0.001 0.01 0.1 0.001] '})
553 title({'Absoluter Fehler: 200 mV';'R = 1000...

```

```
554 | Q = [0.001 0.01 0.1 0.001] '})
555 | datetick('x','HH:MM','keeplimits')
556 | legend('Messwert','Iteration = 2s','Iteration = 60s'...
557 | , 'Iteration = 600s')
558 | xlabel('Zeit [HH:MM]')
559 | ylabel('Spannung')
560 | grid
561 | subplot(2,1,2)
562 |
563 | plot(zeit,abweichung_2,'black',zeit,abweichung_60,'g',...
564 | zeit,abweichung_600,'r');
565 | legend('Iteration = 2s','Iteration = 60s',...
566 | 'Iteration = 600s')
567 | datetick('x',15,'keeplimits')
568 | xlabel('Zeit [HH:MM]')
569 | ylabel('Abweichungsrate')
570 | grid
571 | saveas(gcf,'dynamik_entltdung_20deg_R1000_ITER_Absolute_Fehler'...
572 | , 'pdf')
```

Tabellenverzeichnis

1.1. Verfahren zur Überwachung des Ladezustandes	15
2.1. Überblick über die in Kalman Filter verwendeten Symbole	19
2.2. Vergleich der Modelle	28
3.1. Eingesetzte Messgeräte	35
3.2. Parameter der Formel zur Ruhespannung	69
3.3. Parameter der Formel zum Widerstand	73
3.4. Vergleich der implementierten Modelle	84
B.1. Messung der Ruhespannung bzw. des Ladezustandes 1	119
B.2. Messung der Ruhespannung bzw. des Ladezustandes 2	119
B.3. Messung der Ruhespannung bzw. des Ladezustandes 3	120

Abbildungsverzeichnis

1.1. Konstruktion einer VARTA - Starterbatterie	11
1.2. Batterieüberwachungssystem	14
2.1. Ablauf des Algorithmus	18
2.2. Beispiel des Kalman Filters	21
2.3. Modell 1	25
2.4. Modell 2	26
2.5. Modell 3	26
2.6. Modell 4	27
2.7. Modell 5	27
2.8. Beispiel der Methode kleinsten Quadrate	30
2.9. Beispiel der Methode kleinster Quadrate	33
3.1. Messplatz	35
3.2. Allgemeiner Ablauf des Programms	38
3.3. Modell 4	39
3.4. Das in Simulink erstellte Modell 4	40
3.5. Vergleich des dynamischen Verhaltens des Modells 4 mit dem Messwert	41
3.6. Vergleich des langen dynamischen Verhaltens des Modells 4 mit dem Messwert	42
3.7. Vergleich des Verhaltens der durchgehenden Entladung vom Modell 4 mit dem Messwert	43
3.8. I-U Ladeverfahren	44
3.9. Vergleich des Verhaltens beim Aufladen von Modell 4 mit dem Messwert	45
3.10. Modell 4	47
3.11. Das angepasstes Modell 4 Klasse 1	48
3.12. Wirkung der einzelnen Elemente	49
3.13. Vergleich des dynamischen Verhaltens des Modells mit dem Messwert	50
3.14. Vergleich des langen dynamischen Verhaltens des Modells mit dem Messwert Zyklus1: 1 Stunde; Zyklus 2: 30 Minuten	51
3.15. Vergleich des Verlaufs der kontinuierlichen Entladung des Modells mit dem Messwert	52
3.16. Modell beim Aufladen	53
3.17. Vergleich der Simulation des Modells beim Aufladen mit dem Messwert	54

3.18. erweitertes Modell beim Aufladen	55
3.19. Vergleich der Ausgabe des erweiterten Modells beim Aufladen mit dem Messwert	55
3.20. Modell mit zwei RC-Parallelschaltungen beim Aufladen	56
3.21. Vergleich des Verlaufs des Modells mit dem Messwert	56
3.22. kontinuierliche Entladung bei drei Temperaturen	59
3.23. partielle Entladung bei drei Temperaturen	61
3.24. dynamisches Verhalten beim Entladen bei drei Temperaturen	63
3.25. Ruhespannung in der partiellen Entladung bei 20°C	65
3.26. Darstellung der Ruhespannung bei drei Temperaturen	65
3.27. Spannungsverlauf der Kondensatoren und Klemmenspannung bei 20°C	66
3.28. Vergleich des Verlaufs der Kondensatoren bei unterschiedlichen Temperaturen	67
3.29. Relative Abweichung	68
3.30. Vergleich mit der Ausgabe der Formel	69
3.31. Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei 20°C	70
3.32. Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei 50°C	70
3.33. Vergleich des Verlaufs der Ruhespannung mit dem Ergebnis der Formel bei -2°C	71
3.34. Vergleich des Innenwiderstands	72
3.35. Darstellung des Widerstandes aus der Formel	74
3.36. Darstellung des Widerstandes bei unterschiedlichen Temperaturen aus der Formel des Widerstandes	74
3.37. das angepasste Modell Klasse 2	75
3.38. Ablauf des Benutzer-definierten Blocks	76
3.39. Vergleich mit der kontinuierlichen Entladung bei 20°C	77
3.40. Vergleich mit der kontinuierlichen Entladung bei 50°C	77
3.41. Vergleich mit der partiellen Entladung bei 20°C	78
3.42. Vergleich mit der partiellen Entladung bei 50°C	79
3.43. Vergleich mit dem dynamischen Verhalten bei 20°C	80
3.44. Vergleich mit dem dynamischen Verhalten bei 50°C	80
3.45. Das angepasste Modell 4 Klasse 3	82
4.1. Angepasste Modell 4 Klasse 2	86
4.2. Rauschenverteilung einzelner Sensoren	93
4.3. Rauschenverteilung aller Sensoren	94
4.4. Vergleich der kontinuierlichen Entladung bei 20°C	96
4.5. Vergleich der kontinuierlichen Entladung bei 50°C	96
4.6. Vergleich der partiellen Entladung bei 20°C	97

4.7. Vergleich der partiellen Entladung bei 50°C	98
4.8. Vergleich des dynamischen Verhaltens bei 20°C	99
4.9. Vergleich des dynamischen Verhaltens bei 50°C	100
4.10. Fehlerrate bei der partiellen Entladung 20°C	102
4.11. Fehlerrate bei der partiellen Entladung 50°C	102
4.12. Fehlerrate bei der kontinuierlichen Entladung 20°C	102
4.13. Fehlerrate bei der kontinuierlichen Entladung 50°C	102
4.14. Fehlerrate beim dynamischen Verhalten 20°C	103
4.15. Fehlerrate beim dynamischen Verhalten 50°C	103
4.16. Vergleich mit der verrauschten partiellen Entladung 20°C	103
4.17. Vergleich mit der verrauschten partiellen Entladung 50°C	103
4.18. Vergleich mit der verrauschten kontinuierlichen Entladung 20°C	104
4.19. Vergleich mit der verrauschten kontinuierlichen Entladung 50°C	104
4.20. Vergleich beim stark verrauschten dynamischen Verhalten 20°C	104
4.21. Vergleich beim stark verrauschten dynamischen Verhalten 50°C	104
4.22. Verteilung des künstlichen Rauschens	105
4.23. Vergleich mit der stark verrauschten partiellen Entladung 20°C	106
4.24. Vergleich mit der stark verrauschten partiellen Entladung 50°C	106
4.25. Vergleich mit der stark verrauschten Entladung 20°C	106
4.26. Vergleich mit der stark verrauschten Entladung 50°C	106
4.27. Vergleich beim stark verrauschten dynamischen Verhalten 20°C	107
4.28. Vergleich beim stark verrauschten dynamischen Verhalten 50°C	107
A.1. Modell A1	116
A.2. Modell A2	116
A.3. Modell A3	117
A.4. Modell A4	117
A.5. Modell A5	118

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 24. Juli 2012

Ort, Datum

Unterschrift