

Masterarbeit

**Modellierung von  
Lithium-  
Dünnschichtbatterien für  
die Ablaufplanung im  
Betriebssystem Kratos**

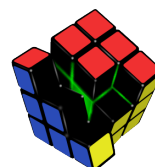
**David Schmidt  
14. September 2016**

Betreuer:

Prof. Dr.-Ing. Olaf Spinczyk

Dipl.-Inf. Markus Buschhoff

Technische Universität Dortmund  
Fakultät für Informatik  
Lehrstuhl 12  
Arbeitsgruppe Eingebettete Systemsoftware  
<http://ess.cs.tu-dortmund.de>





## Zusammenfassung

In eingebetteten Systemen, welche eine Batterie als Energiequelle verwenden, ist die Energie eine begrenzte Ressource. Deshalb ist es von entscheidender Bedeutung, Batterien möglichst effizient auszunutzen, um möglichst viel der in einer Batterie enthaltenen Ladung verwenden zu können. Dies ist durch Berücksichtigung der nichtlinearen Effekte einer Batterie, so wie des Recovery und des Rate Capacity Effekts, möglich. Eine Methode diese Effekte zu berücksichtigen, ist, Entladeprofile, welche besonders bei *Deeply Embedded Systems* maßgeblich durch Hardwareaktivitäten verursacht werden, über eine batteriebewahre Ablaufplanung zu steuern. Hierzu wurde für einen bereits existierenden offline Scheduler ein Batteriemodell (Kinetic Battery Model) für Lithium-Dünnschichtbatterien entwickelt. Das Ergebnis ist ein offline Scheduler, der es durch batteriebewahre Ablaufplanungen ermöglicht, die Betriebszeit sowie die Anzahl von Taskdurchläufen für ein Deeply Embedded Systems gegenüber Ablaufplanungen, welche nicht die nichtlinearen Effekte in einer Batterie explizit berücksichtigen, zu erhöhen.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	2
1.2. Ziele der Arbeit . . . . .	5
1.3. Aufbau der Arbeit . . . . .	6
<b>2. Grundlagen</b>	<b>9</b>
2.1. Eingebettete Systeme . . . . .	9
2.1.1. Energieverbrauch in eingebetteten Systemen . . . . .	10
2.1.2. Energieverbrauch in eingebetteten Betriebssystemen . . . . .	15
2.1.3. Das Betriebssystem KRATOS . . . . .	17
2.2. Batterien . . . . .	19
2.2.1. Prinzipien der Entladung einer Batterie . . . . .	20
2.2.2. Terminologie . . . . .	20
2.2.3. Elektrotechnische Grundlagen . . . . .	22
2.2.4. Nichtlineare Effekte . . . . .	23
2.2.5. Feststoff Lithium-Dünnschichtbatterien . . . . .	29
2.3. Verwendete Hardware . . . . .	31
2.3.1. Messtechnik . . . . .	34
2.3.2. Evaluationsboard . . . . .	37
<b>3. Verwandte Arbeiten</b>	<b>39</b>
3.1. Vorausgegangene Arbeit . . . . .	39
3.1.1. Simulator . . . . .	41
3.1.2. Modellierung des Gesamtsystems . . . . .	42
3.1.3. Scheduler . . . . .	48
3.1.4. Einbindung in Kratos . . . . .	49
3.1.5. Abgrenzung . . . . .	49
3.2. Ähnliche Arbeiten . . . . .	49
<b>4. Batteriemodellierung</b>	<b>55</b>
4.1. Anforderung an ein Batteriemodell . . . . .	55

4.2.	Überblick über Batteriemodelle . . . . .	57
4.2.1.	Elektrochemische Modelle . . . . .	57
4.2.2.	Elektrische Schaltungsmodelle . . . . .	58
4.2.3.	Stochastische Modelle . . . . .	60
4.2.4.	Analytische Modelle . . . . .	64
4.3.	Modellauswahl . . . . .	74
<b>5.</b>	<b>Modellierung der EnFilm-Batterie durch das Kinetic Battery Model</b>	<b>77</b>
5.1.	Anpassungen am Modell . . . . .	77
5.2.	Bestimmung der Parameter des Modells . . . . .	79
5.2.1.	Bestimmung der Parameter des Kapazitätsmodells . . . . .	81
5.2.2.	Bestimmung der Parameter des Spannungsmodells . . . . .	86
5.3.	Güte des Modells . . . . .	93
5.3.1.	Zielsetzung der Validierung . . . . .	93
5.3.2.	Vorstellung des Validierungsverfahrens . . . . .	94
5.3.3.	Analyse der Validierungsergebnisse . . . . .	95
<b>6.</b>	<b>Implementierung des batteriegewahren Schedulers</b>	<b>105</b>
6.1.	Architektur des Scheduler . . . . .	105
6.2.	Integration des Kinetic Battery Model in den offline Scheduler . . .	106
<b>7.</b>	<b>Evaluation</b>	<b>109</b>
7.1.	Zielsetzung der Evaluation . . . . .	109
7.2.	Vorstellung des Evaluationsverfahrens . . . . .	109
7.3.	Analyse der Evaluationsergebnisse . . . . .	116
<b>8.</b>	<b>Fazit und Ausblick</b>	<b>125</b>
8.1.	Fazit . . . . .	125
8.2.	Ausblick . . . . .	127
	<b>Literaturverzeichnis</b>	<b>129</b>
	<b>Abbildungsverzeichnis</b>	<b>137</b>
	<b>Tabellenverzeichnis</b>	<b>141</b>
	<b>Listingverzeichnis</b>	<b>143</b>
<b>A.</b>	<b>Anhang</b>	<b>I</b>

# 1. Einleitung

Der Bereich der Informationstechnologie ist wie kaum ein anderer Bereich unseres Lebens ständigen Änderungen unterworfen. So war bis in die späten 80er Jahre die Informationsverarbeitung mit großen Mainframe-Rechnern verbunden. Danach begann die Ära des Personal Computer (PC), während wir uns heute einer Zeit nähern, die als Post-PC Ära bezeichnet wird. Da Standard-PC immer weniger dominant sind und sein werden. Wir nähern uns der Phase des sogenannten „Ubiquitous Computing“. Dabei werden Prozessoren und Software zunehmend in viel kleineren Systemen verwendet [59]. Charakteristisch ist, dass viele Rechner sich einen Menschen „teilen“ und Dinge des täglichen Lebens miteinander verbunden werden. Diesen Wechsel der Ära prognostizierte auch schon der Visionär Mark Weiser, wie die Abbildung 1.1 verdeutlicht. Weiterhin wird dieses dadurch verdeutlicht, dass im ersten Quartal 2011 laut IDC (International Data Corporation) 371.800.000 Handys verkauft wurden, was 20 % mehr als im ersten Quartal 2010 war. Im Vergleich dazu wuchs der PC-Markt nur noch um 4 %.

Mark Weiser war auch die Person, die den Begriff des Ubiquitous Computing prägte und sagte: „Ubiquitous Computing enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user“ [90].

Die technische Grundlage des Ubiquitous Computing sind die eingebetteten Systeme. Diese können als informationsverarbeitende Systeme, die in ein umgebendes Produkt wie z.B. Flugzeuge, Telekommunikationsgeräte etc. eingebettet sind, definiert werden [59].

Problem vieler eingebetteter Systeme ist, dass diese nicht über eine permanente Stromversorgung verfügen, weil sie z.B. mobil eingesetzt werden. Häufig arbeiten eingebettete Systems mit *Energy Harvesting*, d.h. sie „ernten“ Energie aus einer Umgebung, z.B. Sonnenlicht mit Solarzellen und speichern diese in Energiespeichern, wie Batterien (bzw. Akkumulatoren) oder Kondensatoren, um die Zeit zu überbrücken, in der sie nicht mit neuer Energie versorgt werden können. Weiterhin existieren Systeme, die nur mit Batterien oder Kondensatoren betrieben werden und keine eigene Möglichkeit der Energiegewinnung haben. Es spielt für diese Arten von eingebetteten Systemen eine große Rolle, möglichst effizient mit der vorhandenen Energie umzugehen, weil Nutzbarkeit, Laufzeit und Mobilität der Systeme dadurch beeinflusst werden. Dadurch wird die Laufzeit eines Energiespeichers eines

Systems durch die Kapazität des Speichers und durch den Energieverbrauch des restlichen Systems bestimmt.

Das bedeutet, dass möglichst effizient mit den Energiespeichern umgegangen werden muss. Bisweilen ist es häufig so, dass viele der eingebetteten Systeme nicht des Energiespeichers bzw. derer Eigenschaften gewahr sind und betrachten so gängige Energiespeicher zur Vereinfachung als ideale Bauteile, was aber in der Realität nicht greift. Nun stellt sich die Frage, wie eingebettete Systeme mit einem besseren Bewusstsein über die Energiespeicher, diese effizienter nutzen können. Eine besondere Rolle nehmen dabei die sogenannten *Deeply Embedded Systems* ein, wie z.B. Sensornetzwerke. Bei diesen Systemen dominiert der Energieverbrauch der Peripherie. So kann der Energiebedarf eines Funkchips und der von Sensoren, den Energiebedarf der CPU um ein Vielfaches übersteigen, weil die benötigte Rechenleistung im Verhältnis klein ist und meistens energiesparende Mikroprozessoren ausreichen. Dadurch ist die Verwendung der Peripherie entscheidend für den Energiebedarf dieser Systeme.

Daraus lässt sich die Frage ableiten, wie eingebettete Systeme bzw. die Verwendung der Peripherie durch eingebettete Systeme dahingehend optimiert werden können, dass sie möglichst effizient mit der vorhandenen Energie und damit auch mit den Energiespeichern umgehen können. Genau an dieser Stelle knüpft diese Masterarbeit an. Es sollen über die Modellierung einer Energiequelle, im Speziellen eine Lithium-Dünnschicht-Batterie, für die Ablaufplanung<sup>1</sup> und damit auch für die Steuerung von Hardwareaktivitäten in einem Betriebssystem Möglichkeiten geschaffen werden, die vorhandene Energie in einem Energiespeicher möglichst effizient zu nutzen. Genauer gesagt, soll eine batteriegewahre Ablaufplanung für eingebettete Betriebssysteme geschaffen werden, d.h. solch eine Ablaufplanung ist sich der Batterie mit ihren Eigenschaften bewusst. Also sollen die Kenntnisse der spezifischen Eigenschaften über Energiespeicher, zur effizienteren Nutzung dieser, in das Betriebssystem integriert werden.

## 1.1. Motivation

Das Thema dieser Masterarbeit ist aus verschiedenen Richtungen motiviert. Die eigentliche Motivation zum Thema beruht auf der Tatsache, dass eine Betrachtung von Energiequellen, wie Batterien und Kondensatoren als ideale Bauteile zu kurz greift, da bei genauerer Betrachtung ihr Verhalten nicht linear ist. Diese Energiequellen weisen nichtlineare Effekte auf, die Einfluss darauf haben, welcher Anteil

---

<sup>1</sup>In dieser Arbeit werden die Begriffe Ablaufplanung und Scheduling synonym verwendet, und sind als die Menge der Verzögerungszeiten für eine gegebene Anwendung definiert (s. Kapitel 3).



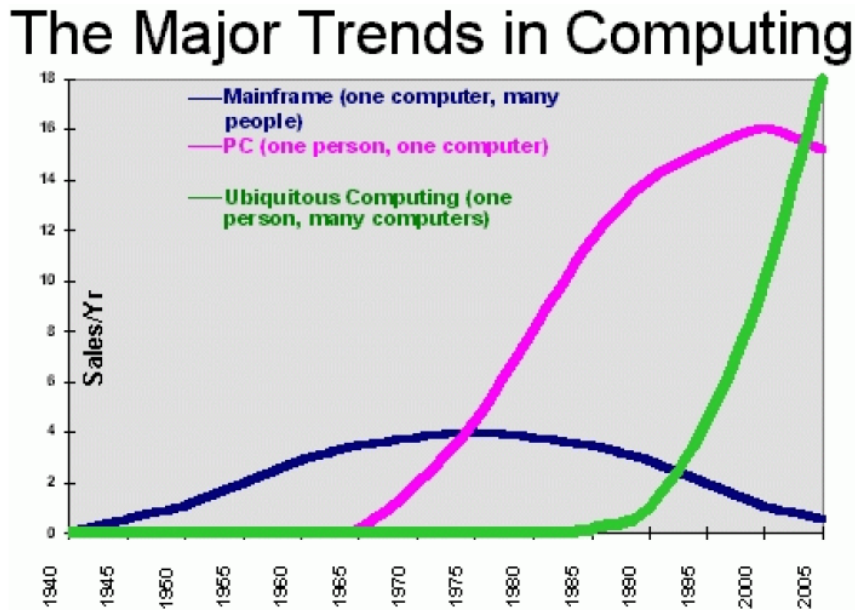


Abbildung 1.1.: Prognose von Mark Weiser über die Trends in der Informationstechnologie [91]

der gespeicherten Energie der Quelle auch wirklich dem Anwender als nutzbare Energie zur Verfügung steht. Das Hauptmotiv ist also, die Betriebszeit und Servicequalität eingebetteter Systeme durch Berücksichtigung dieser nichtlinearen Effekte bei der Planung von Hardwareaktivitäten zu erhöhen. Das bei der Planung von Hardwareaktivitäten zu berücksichtigen, liegt an der in der Einleitung erwähnten Dominanz des Energiebedarfs der Peripherie im Verhältnis zur CPU bei Deeply Embedded Systems.

Die Motivation zur Untersuchung einer Batterie als Energiequelle resultiert aus der vorausgegangenen Diplomarbeit von Michael Hesse mit dem Titel „Berücksichtigung nicht-linearer Energiequelleneffekte im eingebetteten Betriebssystem Kratos“ [34] welcher bereits ein Kondensator als Energiequelle untersucht hatte. Durch die Wahl einer Batterie als Energiequelle soll ein vollständigeres Bild bei der Berücksichtigung von Energiespeichern in der Ablaufplanung geschaffen werden.

Ein weiterer motivierender Aspekt ist, dass viele Deeply Embedded Systems mit Batterien betrieben werden und dass sich Verbesserungen in der Kapazität von Batterien (5-10% jährliche Wachstumsrate) nur langsamer entwickeln, als dies nötig wäre für die wachsende Leistungsanforderung der Systeme, die sie versorgen sollen [46]. Dieses Problem wird auch als *Battery Gap* (deutsch: Batterie-Lücke) bezeichnet und ist in der Abbildung 1.2 veranschaulicht. Zu erkennen ist die Lücke

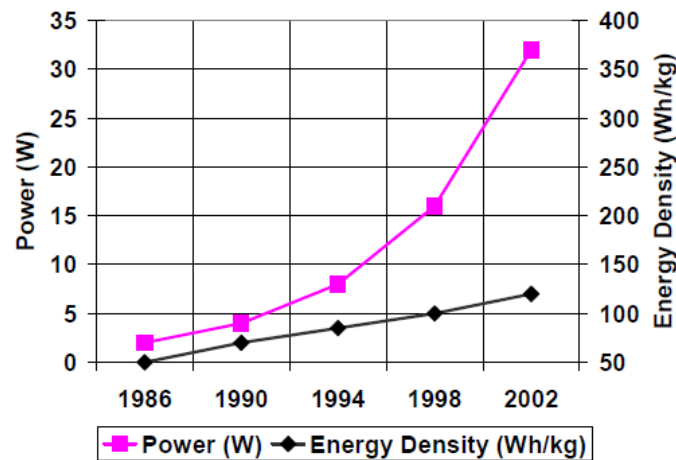


Abbildung 1.2.: Erweiterung der Battery Gap, durch schnell wachsende Leistungsanforderungen und langsame Verbesserungen in der Batterietechnologie [46].

zwischen Trends in der Leistungsaufnahme von Prozessoren [30] und Verbesserungen in der Batterie Kapazität [6] immer größer wird.

Dadurch wird die Batterielaufzeit<sup>2</sup> zu einem primärem Designziel bei der Entwicklung von batteriebetriebenen eingebetteten Systemen, um diese Lücke zu schließen oder zu verkleinern, da die Batterielaufzeit wie schon eingangs erwähnt direkten Einfluss auf die Nutzung des eingebetteten Systems hat. Erste Ansätze zur Reduzierung dieser Lücken waren *Low Power Design Techniken* [8], [65], [66], [1]. Diese Techniken hatten zwar Erfolg bei der Reduktion der Energie, welche aus einer Batterie entnommen wird, und können daher die Batterielaufzeit bis zu einem gewissen Maße erhöhen, es gilt aber, dass eine wirkliche Maximierung der Batterielaufzeit nur dann erreicht werden kann, wenn beide das System und der Energiespeicher verstanden werden. Denn wie eingangs schon erwähnt, ist es unzureichend, eine Batterie als ideales Bauteil zu betrachten. So wurde schon gezeigt, dass die Menge an Energie, welche von einer Batterie zur Verfügung gestellt werden kann, davon abhängt wie die Energie entnommen wird [82, 63, 67, 31, 27, 26, 2, 13, 61, 76, 77, 77, 20, 24, 21, 89, 28, 29]. Dieser zweite Ansatz, also die Fähigkeiten und Grenzen von Batterien zu berücksichtigen, wird auch als *Batterie getriebenes System Design* bezeichnet und ist Bestandteil fällt diese Masterarbeit.

<sup>2</sup>Dabei ist in dieser Arbeit mit Batterielaufzeit, die Zeit gemeint, welche nötig ist um eine Batterie bis zur Erschöpfung zu entladen.

Weiterhin ist die Motivation zur Betrachtung von Lithium-Dünnschichtbatterien in Abstimmung mit dem Fraunhofer-Institut für Materialfluss und Logistik (IML) darin begründet, dass diese Art von Batterie noch sehr unerforscht ist und vielversprechend für den Einsatz in Deeply Embedded Systems sein soll.

Die Motivation zur Modellierung einer Batterie entsteht daher, dass in Deeply Embedded Systems meistens keinerlei Messhardware verbaut ist, weil dies zusätzlichen Hardwareaufwand bedeutet und den Energiebedarf des Systems weiter erhöhen würde.

## 1.2. Ziele der Arbeit

Wie schon erwähnt, schränkt die Batterielaufzeit die Nutzbarkeit, Laufzeit und Mobilität von eingebetteten Systemen ein. Weiterhin wird der Energieverbrauch eines *Deeply Embedded Systems* durch die Benutzung der Peripherie des Systems dominiert. Des Weiteren gibt es die Erkenntnis, dass Energiespeicher wie Batterien nicht als ideale Bauteile betrachtet werden können, da sie eine Reihe von nichtlinearen Effekten aufweisen. So ist z.B. der Zeitpunkt der Energientnahme oder die Höhe des Entladestroms entscheidend. Die Einflüsse dieser Effekte lassen sich aus dem internen Zustand der Batterie ablesen.

Kernziel dieser Masterarbeit ist es zu evaluieren, ob es sich lohnt und wie viel es bringt, einen Energiespeicher, am Beispiel einer Batterie, in der Ablaufplanung von Hardwarezugriffen bzw. Hardwareaktivitäten zu berücksichtigen, um einen Energiespeicher möglichst effizient auszunutzen. Anders formuliert soll die Frage beantwortet werden, ob eine batteriegeheure Ablaufplanung eine eigene Methode dafür ist, Batterien effizienter nutzen zu können.

Deswegen muss untersucht werden, welche nichtlinearen Effekte es bei Batterien gibt und ob die Planung von Hardwarezugriffen einen Einfluss auf die nichtlinearen Effekte einer Batterie hat. Dazu muss eine Möglichkeit geschaffen werden, die Energieverbräuche von Hardwarekomponenten eines *Deeply Embedded Systems*, abhängig vom internen Zustand einer Batterie so zu planen, dass die vorhandene Energie einer Batterie möglichst effizient ausgenutzt wird. Dafür sollen nichtlineare Effekte bei der Ablaufplanung berücksichtigt werden.

In der vorausgegangenen Arbeit von Michael H. [34] wurde dazu schon ein offline Scheduler, welcher einen Kondensator als Energiespeicher berücksichtigt, entwickelt. So ist es ein weiteres Ziel dieser Arbeit, diesen Scheduler für eine batteriegeheure Ablaufplanung anzupassen. Damit der Scheduler eine Batterie mit ihren nichtlinearen Effekten berücksichtigen kann, ist es notwendig ein passendes Modell für die in dieser Arbeit exemplarisch ausgewählte Lithium-Dünnschicht-Batterie zu finden und anzupassen. Anschließend muss dieses Modell in den besagten offline

Scheduler integriert werden. Mit diesem Scheduler soll die Frage beantwortet werden, wie die Energieverbräuche von Hardwarekomponenten geplant werden müssen, um eine Batterie möglichst effizient verwenden zu können.

Schließlich ist es noch nötig, für eine aussagekräftige Evaluation ein Evaluationskonzept auszuarbeiten. So ist z.B. auch der Overhead der Bewertung zu beachten. Mit der Evaluation soll die Frage beantwortet werden, ob es sich lohnt, verglichen mit einer naiven Ablaufplanung, eine batteriegewahre Ablaufplanung durchzuführen. Implizit soll damit auch geklärt werden, ob sich der von Michael Hesse verwendete Ansatz zur batteriegewahren Ablaufplanung eignet.

### 1.3. Aufbau der Arbeit

Diese Ausarbeitung ist folgendermaßen strukturiert. Kapitel 1 führt in die Thematik dieser Masterarbeit ein und stellt die Motivation zur Bearbeitung des Themas vor. Weiterhin werden die angestrebten Ziele dieser Arbeit formuliert.

Im anschließenden Kapitel 2 werden die benötigten Grundlagen geschaffen, die zum Verständnis und zur Nachvollziehbarkeit der einzelnen Entwurfs- und Implementierungsentscheidungen notwendig sind. So wird in diesem Kapitel noch einmal genauer der Kontext der eingebetteten Systeme eingeschränkt, die Grundlage dieser Arbeit sind. Außerdem werden die benötigten Grundlagen für eingebettete Betriebssysteme vorgestellt und beschrieben, was nötig ist bzw. welche Informationen gebraucht werden, damit mit ihnen eine batteriegewahre Ablaufplanung möglich ist. Im Zusammenhang wird auch das eingebettete Betriebssystem Kratos präsentiert. Nachfolgend wird in das Gebiet der Batterien eingeführt und deren nichtlineare Effekte erläutert. Außerdem wird geschildert wie diese vorteilhaft in der Ablaufplanung genutzt werden kann. Weiterhin wird die in dieser Arbeit verwendete Lithium-Dünnschichtbatterie vorgestellt. Der letzte Abschnitt dieses Kapitels beschäftigt sich mit der in dieser Arbeit verwendeten Messhardware und ein Evaluationsboard wird vorgestellt.

Das Kapitel 3 geht zuerst in besonderem Maße auf die vorangegangene Diplomarbeit von Michael Hesse ein, da diese Basis der vorliegenden Arbeit ist. Danach werden weitere verwandte Forschungsarbeiten vorgestellt und Einflüsse aus diesen Arbeiten herausgestellt, aber auch Unterschiede zum Ansatz dieser Arbeit hervorgehoben.

Im nachfolgenden Kapitel 4 wird zuerst geschildert, welche Kriterien bei der Auswahl eines geeigneten Batteriemodells für diese Arbeit entscheidend sind. Daran anschließend wird ein Überblick über die gängigsten Batteriemodelle gegeben. Abschließend wird die Auswahl des für diese Arbeit verwendeten Modells begründet.

Kapitel 5 erläutert die Umsetzung des in dieser Masterarbeit eingesetzten *Kinetic*

*Battery Model* für die exemplarisch verwendete EnFilm-Batterie. Zu Beginn wird auf die Anpassungen am Modell eingegangen, welche nötig waren, damit das Modell für die Zwecke dieser Arbeit verwendet werden kann. Daran anschließend wird erläutert wie die Parameter bestimmt werden, welche notwendig sind, damit das Modell eine spezifische Batterie modelliert. Das Kapitel endet mit einer Analyse der Güte des Modells bezüglich der exemplarisch verwendeten EnFilm-Batterie.

Anschließend wird in Kapitel 6 die Implementierung des batteriegewahren Schedulers unter Verwendung der vorangegangenen Diplomarbeit von Michael Hesse als Basis dargestellt.

Im vorletzten Kapitel 7 wird die Evaluation des umgesetzten Konzepts eines batteriegewahren Schedulers und deren Ergebnisse veranschaulicht. Hierzu wird zuerst das Evaluationskonzept vorgestellt und die Ziele der Evaluation aufgezeigt. Zum Schluss werden die Ergebnisse der Evaluation analysiert. Es wird untersucht, was eine batteriegewahre Ablaufplanung gegenüber einer naiven Ablaufplanung an Vorteilen bietet bzw. welcher Overhead mit der Lösung einhergeht.

Die Ausarbeitung schließt mit einem Fazit und einem Ausblick.



## 2. Grundlagen

Dieses Kapitel umfasst die Grundlagen, die zum Verständnis des Themas dieser Abschlussarbeit notwendig sind.

Zu Beginn dieses Kapitels werden eingebettete Systeme und deren Betriebssysteme erläutert. Genauer gesagt wird beschrieben, welche Anforderungen bzw. Rahmenbedingungen ein eingebettetes System erfüllen muss, damit es für eine batteriegewahre Ablaufplanung geeignet ist. Weiterhin wird beschrieben, wo Energie in eingebetteten Systemen verbraucht wird und wie diese Verbrauchsinformationen in ein eingebettetes System integriert werden können. In diesem Zusammenhang wird auch das in dieser Arbeit exemplarisch verwendete eingebettete Betriebssystem Kratos vorgestellt.

Anschließend werden die nötigen Grundlagen über Batterien und deren nichtlineare Effekte dargestellt. Weiterhin wird erläutert, wie diese Effekte zum Vorteil in der Ablaufplanung genutzt werden können. Danach wird die verwendete Lithium-Dünnschichtbatterie EnFilm erklärt.

Abschließend wird in diesem Kapitel die eingesetzte Hardware vorgestellt. Dazu wird zuerst die verwendete Messhardware erläutert und daran anschließend das verwendete Evaluationsboard.

### 2.1. Eingebettete Systeme

Wie auch schon in der Einleitung beschrieben, können eingebettete Systeme als informationsverarbeitende Systeme, die in ein umgebendes Produkt wie z.B. Flugzeuge, Telekommunikationsgeräte etc. eingebettet sind, definiert werden [59].

Um Energiespeicher, wie eine Batterie, in der Ablaufplanung zu berücksichtigen, ist es notwendig, dass sich der Energieverbrauch eines eingebetteten Systems im Voraus gut planen lässt. Daher sind Systeme, deren Hardwareaktivitäten stark von Benutzerinteraktionen, also vom Zufall abhängig sind, nicht geeignet für eine batteriegewahre Ablaufplanung, da sich bei dieser Art von Systemen der Energiebedarf nur schlecht im Voraus planen lässt. Diese Planung ist aber Voraussetzung, um die nichtlinearen Effekte zum Vorteil einer effizienteren Nutzung der Batterie berücksichtigen zu können.

Ein weiteres Kriterium, das eingebettete Systeme für eine batteriegewahre Ablaufplanung erfüllen müssen, ist, dass deren Aufgaben nicht in Echtzeit abgearbeitet werden müssen. Da es bei Systemen mit Echtzeitanforderungen keinen Spielraum für eine batteriegewahre Ablaufplanung gibt.

Generell müssen eingebettete Systeme folgende Anforderungen erfüllen, damit sich eine batteriegewahre Ablaufplanung überhaupt realisieren lässt. [34]:

- Die Zeitpunkte der Bearbeitung von Aufgaben unterliegt keinen Echtzeitanforderungen.
- Die Aufgaben des Systems sind gut vorhersagbar.
- Es gibt längere Phasen niedrigen Energieverbrauchs.
- Das System führt die Aufgaben zyklisch aus.

Das Feld der *Deeply Embedded System* ist dabei eine besondere Art von eingebetteten Systemen. Hierzu zählen z.B. Sensorknoten und ubiquitäre Systeme. Wie auch schon erwähnt, ist es charakteristisch für diese Art von eingebetteten Systemen, dass ihr Energieverbrauch durch die Peripherie wie Funk und Sensoren dominiert wird. Die weitere Einschränkung in dieser Arbeit auf diese Art von Systemen liegt zum einen darin begründet, dass sich die Energieverbräuche der Peripherie leicht über die Ablaufplanung steuern lassen und diese meist den größten Teil am Gesamtverbrauch des Systems ausmachen. Zum anderen erfüllen sie oft die oben genannten Anforderungen.

### 2.1.1. Energieverbrauch in eingebetteten Systemen

Um bei Digitalschaltungen und eingebetteten Systemen die Leistungsaufnahme verstehen zu können, werden im folgenden Abschnitt die Hauptursachen für die Entstehung der Verlustleistung dieser Systeme erklärt. Generell gilt, dass die „verbrauchte“ Energie in Wärme umgewandelt und abgestrahlt wird [78].

Heutzutage dominiert bei Digitalschaltungen die CMOS<sup>1</sup> Halbleitertechnologie. Abbildung 2.1 zeigt einen CMOS-Inverter am Beispiel für eine CMOS-Schaltung. Weiterhin veranschaulicht die Abbildung 2.2, dass sowohl ein P-Kanal Transistor als auch ein N-Kanal Transistor verwendet wird. Dies ist typisch für die CMOS-Technologie. Ferner ist in Abbildung 2.2 zu erkennen, dass die beiden Transistoren komplementär arbeiten. Das bedeutet, dass der P-Kanal Transistor bei *GND* (Ground) an *D* (Drain) durchschaltet und sperrt bei  $V_{CC}$ . Der N-Kanal Transistor sperrt bei *GND* an *G* (Gate) und schaltet durch bei  $V_{CC}$ .

---

<sup>1</sup>Complementary Metal Oxide Semiconductor



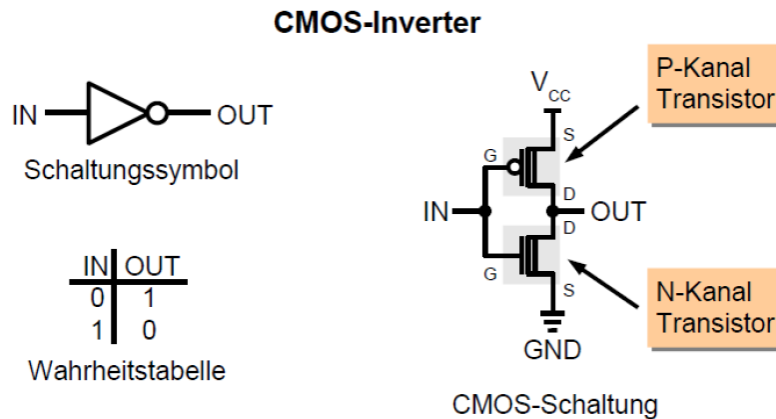


Abbildung 2.1.: Beispiel einer Schaltung in CMOS-Technologie [78].

Im Allgemeinen gilt, dass die CMOS-Technologie sehr stromsparend arbeitet, weil Ströme im Prinzip nur beim Schalten fließen. Dennoch treten in der Realität noch weitere Ströme auf, welche im Folgenden beschrieben werden [88]. Zuvor werden noch die wichtigen physikalischen Größen, Energie und Leistung definiert.

#### 2.1.1.1. Leistung

Als Leistung wird die pro Zeiteinheit verrichtete Arbeit bezeichnet. Sie wird mit dem Buchstaben  $P$  (von Power) gekennzeichnet. Dabei wird die Leistung  $P$  aus dem Produkt der Spannung  $U$  und dem Strom  $I$  berechnet:

$$P = U \cdot I. \quad (2.1)$$

Als Einheit für die Leistung ergibt sich:

$$[P] = [U] \cdot [I] = \text{VA}. \quad (2.2)$$

Die Einheit VA wird mit Watt (W) bezeichnet.  $P$  hat also die Einheit 1W (Watt) = 1V (Volt) · 1A (Ampere) = 1 Joule/s =  $1 \frac{\text{kgm}^2}{\text{s}^3}$  [64].

#### 2.1.1.2. Energie

Aus dem Integral des Produktes der elektrischen Spannung  $u(t)$  und Stromstärke  $i(t)$  über die Zeit  $t$  berechnet sich die elektrische Energie  $E$ , die auch als elektrische Arbeit  $W$  bezeichnet wird (entnommen aus [43]):

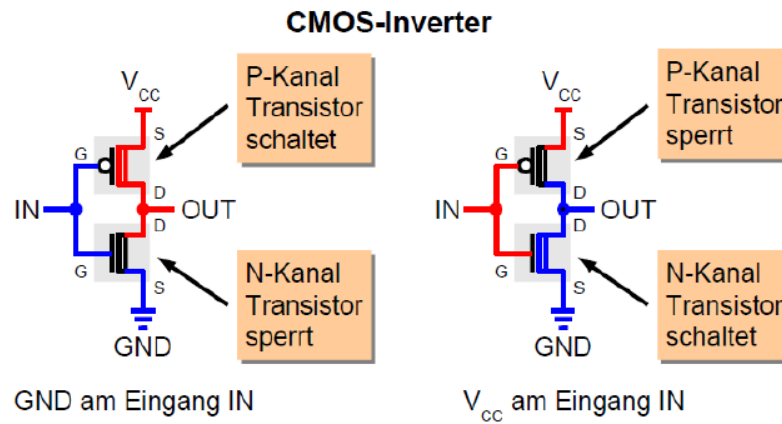


Abbildung 2.2.: Funktionsweise des CMOS-Inverter [78].

$$E = \int_{t_0}^{t_1} u(t) \cdot i(t) dt. \tag{2.3}$$

E hat die Einheit 1J (Joule) = 1Ws = 1VAs. Ist die elektrische Spannung  $U$  und die Stromstärke  $I$  konstant ergibt sich die Energie mit  $T = t_1 - t_0$  wie folgt:

$$E = U \cdot I \cdot T. \tag{2.4}$$

Durch Einsetzen der Formel 2.1 in die Gleichung 2.4 ergibt sich:

$$E = P \cdot T. \tag{2.5}$$

### 2.1.1.3. Schaltstrom

Entscheidend für den Energieverbrauch einer CMOS-Schaltung ist mit 70 % bis 90 % Anteil am Gesamtenergieverbrauch der Schaltstrom (engl. Switching Current). Der Schaltstrom entsteht durch Laden und Entladen von Lastkapazitäten am Ausgang einer Zelle (siehe auch Abbildung 2.3) [78]. Dabei tritt Schaltstrom nur bei Zustandsänderungen an einem Transistor auf, da dieser zusammen mit der Leitung einen Kondensator darstellt, der geladen und entladen werden muss. Die Leistungsaufnahme  $P_{sw}$  einer Zelle lässt sich wie folgt bestimmen:

$$P_{sw} = \frac{V_{dd}^2}{2} \sum_{nets(i)} (C_{Load_i} \cdot TR_i). \tag{2.6}$$

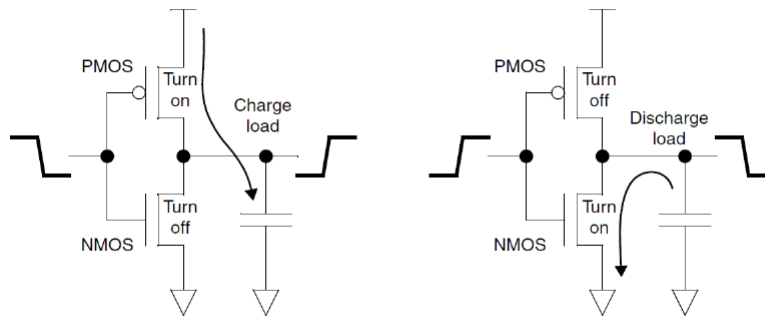


Abbildung 2.3.: Auftreten von Schaltstrom [81].

Dabei entspricht  $V_{dd}$  der Versorgungsspannung,  $C_{Load_i}$  den Lastkapazitäten einer Zelle und  $TR_i$  der Wechselrate beim Laden und Entladen der Kapazität.

#### 2.1.1.4. Kurzschlussstrom

Wenn komplementäre Transistoren nicht exakt gleichzeitig ihren Zustand ändern, tritt der Kurzschlussstrom (engl. Short Circuit Current) auf. Auch der Kurzschlussstrom tritt nur bei Zustandsänderungen auf, wobei die Spannung an G nahe  $V_{CC}/2$  ist (siehe Abbildung 2.4). Am Gesamtenergieverbrauch macht der Kurzschlussstrom zwischen 10 % bis 30 % aus [78]:

$$P_{sc} = \sum_{cell(j)} f(C_{Load_j}, TransTime_{input}) \cdot TR_j. \quad (2.7)$$

Mit  $C_{Load_j}$  für die Lastkapazität einer Zelle.  $TransTime_{input}$  entspricht der Schaltzeit eines Transistors und  $TR_j$  der Wechselrate beim Laden und Entladen der Kapazität.

#### 2.1.1.5. Leckstrom

Ein weiterer Strom ist der Leckstrom (engl. Leakage Current), welcher aber nur ca. 1 % des Gesamtenergieverbrauchs ausmacht. Der Leckstrom ist ein unerwünschter Strom innerhalb der Transistoren, welcher permanent auftritt. In der Abbildung 2.5 ist der Effekt zusehen [78]. Er lässt sich wie folgt ermitteln:

$$P_{lk} = \sum_{cells(i)} P_{CellLeakage_i}. \quad (2.8)$$

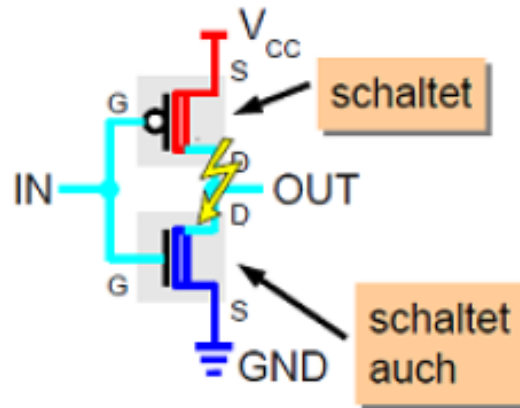


Abbildung 2.4.: Auftreten eines Kurzschlussstroms [78].

$P_{CellLeakage_i}$  entspricht dem Energieverbrauch einer Zelle in einem inaktiven Zustand.

#### 2.1.1.6. Leistung einer CMOS-Schaltung

Die Leistung einer CMOS-Schaltung lässt sich im Allgemeinen durch die folgende Formel abschätzen:

$$P \approx C \cdot U^2 \cdot f, \quad (2.9)$$

wobei  $C$  der Kapazität entspricht,  $U$  der Spannung und  $f$  die Schaltfrequenz ist. Der entscheidende Faktor ist dabei die Versorgungsspannung, da diese quadratisch in die Leistung eingeht [78].

Daher sind Halbleiterhersteller bemüht, die minimal erforderliche Betriebsspannung möglichst gering zu halten. Das Problem daran ist, dass dies die maximale Taktrate der Schaltung begrenzt. Denn mit einer geringeren Betriebsspannung benötigt die Gatekapazität eine längere Ladezeit, bevor der Transistor vollständig durchschaltet oder sperrt [23].

Des Weiteren spielt auch die Frequenz (Schaltvorgänge, dynamische Vorgänge) eine Rolle. So skalieren der Schaltstrom und der Kurzschlussstrom proportional zur Umschalthäufigkeit bzw. der Taktfrequenz im System. Deswegen werden in Mikrocontrollern und anderen Digitalerschaltungen nicht benötigte Komponenten durch Abschalten ihres Arbeitstaktes deaktiviert. Der Energieverbrauch dieser Komponenten reduziert sich dann auf die Leckströme [23].

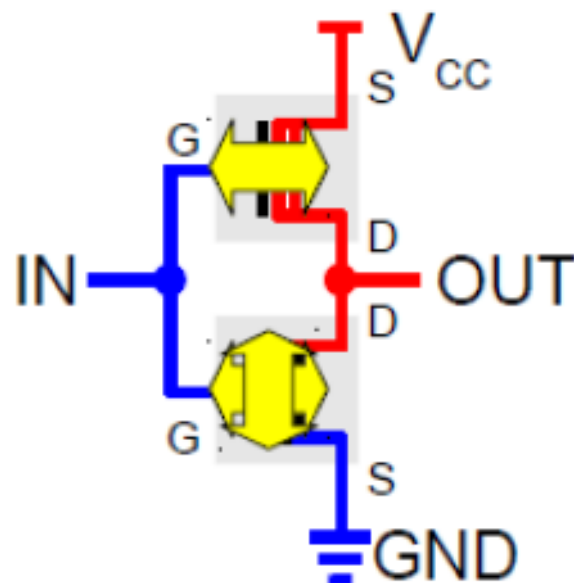


Abbildung 2.5.: Auftreten von Leckströmen [78].

### 2.1.1.7. Weitere Verbraucher

Zu den oben genannten Effekten kommen noch für eingebettete Systeme typische Verbraucher hinzu, welcher aber meist den größten Energiebedarf im gesamten System haben. Deshalb sind sie von entscheidender Bedeutung für diese Arbeit. So sind dies z.B. Licht- und Temperatursensoren. Diese verändern ihren Widerstand und können nur mit einem Stromfluss durch den Sensor eine messbare Größe liefern. Weiterhin benötigen Funkmodule Energie zum Versorgen von Verstärkern sowie zum Aussenden der elektromagnetischen Wellen [23].

## 2.1.2. Energieverbrauch in eingebetteten Betriebssystemen

Als Zwischenschicht zwischen den Anwendungen und der Hardware wird auch bei eingebetteten Systemen ein Betriebssystem verwendet. So hat auch hier das Betriebssystem die Aufgabe, die Hardware anzusteuern und die Tasks der Anwendungen zu schedulen. Ein Beispiel Betriebssystem für eingebettete Systeme ist Kratos, welches im nächsten Abschnitt vorgestellt wird. Nachdem im vorherigen Abschnitt die Ursachen für Energieverbräuche in einem eingebetteten System beschrieben wurden, stellt sich jetzt die Frage, wie die Verbrauchsinformationen in ein Be-

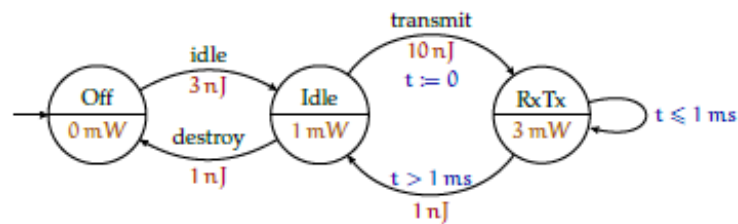


Abbildung 2.6.: PTA einer fiktiven Kommunikationsschnittstelle. Die Übergänge und Zustände tragen ihre jeweiligen Kosten. Durch Eintreten eines Ereignisses oder durch Erfüllen einer zeitlichen Bedingung können Zustandsübergänge ausgelöst werden [23].

triebssystem gelangen bzw. wie sie bereitgestellt werden können, damit diese zum Vorteil genutzt werden können. Denn damit eine batteriegewahre Ablaufplanung überhaupt möglich ist, reicht es nicht aus nur zu wissen, wann eine bestimmte Hardwareaktivität erfolgen soll, sondern auch welche Leistungsaufnahme damit einhergeht [34].

Eine Möglichkeit diese Informationen bereitzustellen ist durch die von Robert Falkenberg in seiner Arbeit „Entwurf eines energiegewahren Treibermodells für eingebettete Betriebssysteme“ für das Betriebssystem Kratos entwickelten DFA<sup>2</sup>-Treiber [23] gegeben. Dabei bilden DFA-Treiber einzelne Hardwarekomponenten wie die Peripherie auf Priced Timed Automata (PTA) ab (vergleiche Abbildung 2.6) und stellen so den Zusammenhang zwischen Peripheriezugriffen und dem damit verbundenen Verbrauch her [34]. Diese Automaten beschreiben die einzelnen Zustände und Zustandsübergänge der Hardwarekomponenten. Weiterhin wird für Zustände die Leistungsaufnahme der Zustände festgelegt und in den Zuständen abgelegt. Des Weiteren enthält ein Automat für die Zustandsübergänge den Energiebedarf der Übergänge [34]. Zur Verwendung der DFA-Treiber wird allerdings eine konstante Versorgungsspannung vorausgesetzt.

So ist es mit dieser Treiberschicht möglich, Informationen über den Energieverbrauch einer Aktion einer Hardwarekomponente zu bekommen. Abbildung 2.7 zeigt den Ansatz der DFA-Treiber im Vergleich zu klassischen Treibern.

Die vorangegangene Arbeit von Michael Hesse hat den Ansatz der automatenbasierten Hardwaretreiber aufgegriffen und da auch für eine batteriegewahre Ablaufplanung das Wissen über Leistung und Energieverbräuche essentiell ist, wird auch in dieser Arbeit der Ansatz in einer abgewandelten Form verwendet. Abgeändert wurde der Ansatz, weil in dieser Arbeit nicht von einer konstanten Versorgungs-

<sup>2</sup>deterministischer endlicher Automat, engl. Deterministic Finite Automaton (DFA)

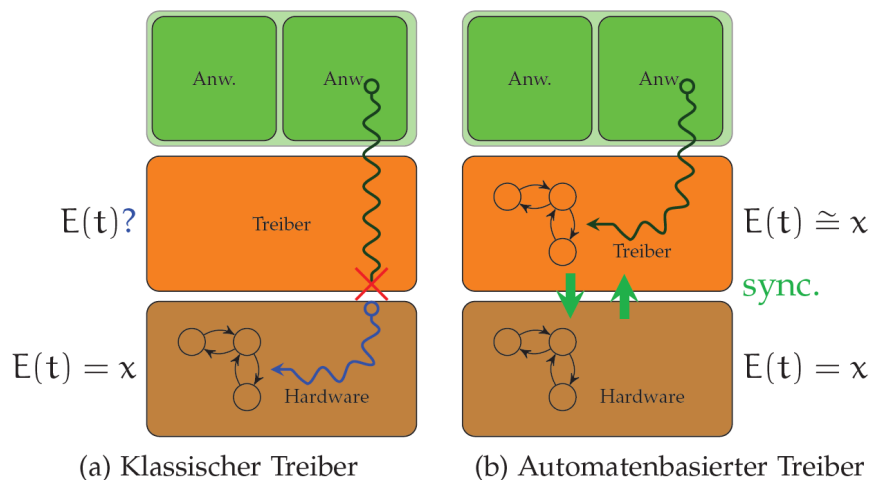


Abbildung 2.7.: Links im Bild ist ein klassischer Treiberansatz, welcher keine Informationen über den Energieverbrauch bereitstellt. Rechts im Bild ist der DFA-Treiberansatz, welcher Rückschlüsse über den Leistungsbedarf des aktuellen Hardwarezustandes liefern kann [23].

spannung ausgegangen wurde. Weitere Details dazu werden im Verlauf der Arbeit gegeben.

### 2.1.3. Das Betriebssystem KRATOS

Als beispielhaftes Betriebssystem zur Umsetzung einer batteriegefahrenen Ablaufplanung für ein eingebettetes System wurde Kratos verwendet. Das Betriebssystem Kratos ist ein Betriebssystem für eingebettete Systeme [7]. Kratos entstand dabei als eine featureorientierte Erweiterung aus AOSTuBS<sup>3</sup>. AOSTuBS ist ein für Lehr- und Forschungszwecke entworfenes Betriebssystem. Das Betriebssystem ist in der Programmiersprache AspectC++ implementiert worden. Dadurch ist es möglich, aspektorientiert Erweiterungen hinzuzufügen, ohne dass Veränderungen am Quelltext des Gesamtsystems vorgenommen werden müssen.

Kratos ist strukturell gesehen in Schichten aufgebaut (Abbildung 2.8), wobei die Schichten sich aus mehreren Klassen zusammensetzen.

Die Unterbrechung von einer aktuellen Routine und die Aufnahme bzw. Wiederaufnahme eines anderen Kontexts ermöglicht die unmittelbar auf der Hardware aufbauende Schicht der Kontrollflussabstraktion. Dabei läuft dies folgendermaßen ab: Die aktuellen Registerwerte einschließlich des Stack Pointers (SP) werden

<sup>3</sup>Aspektorientiertes Studenten Betriebssystem

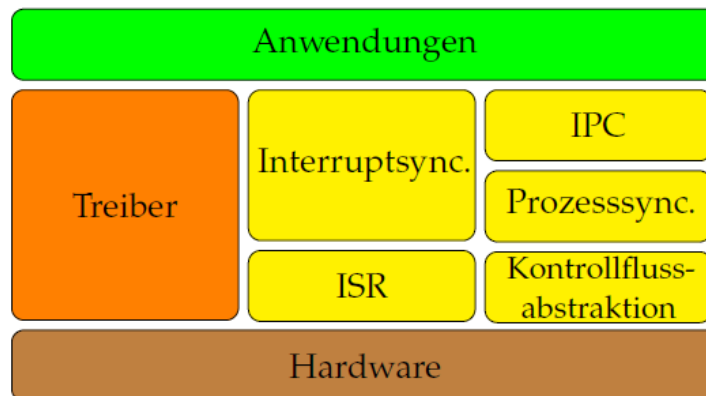


Abbildung 2.8.: Schichtenstruktur von Kratos [23].

auf den Stack der aktiven Routine gespeichert und die Werte aus dem Zielkontext wiederhergestellt. Über dieser Schicht befindet sich ein preemptiver zeitscheiben-basierter Scheduler, der die angemeldeten Prozesse synchronisiert. Dabei wird er in Abständen von  $10ms$  durch eine Unterbrechung nach Ablauf eines Timers aufgerufen und führt anschließend einen Wechsel auf den Kontext des nächsten Prozesses durch, welcher bereit ist. Auf der Schicht des Schedulers befindet sich die IPC-Schicht. Sie verwaltet wartende bzw. schlafende Prozesse. Weiterhin stellt diese Schicht Wartemechanismen wie Semaphoren (`wait()`- und `signal()`-Methoden) und Wecker (`Buzzer`) mit `sleep()`-Methoden zur Verfügung.

Eine weitere Schicht, welche auf der Hardware aufbaut, ist die Interruptbehandlungs-routinen-Schicht. Damit bietet Kratos Datenstrukturen und Methoden zum Einhängen von Interruptbehandlungs-routinen an. Eine geordnete Abarbeitung von Unterbrechungen stellt die darüber befindliche Schicht (Interrupt-synchronisation) zur Verfügung. Dies ist nötig, da auch Unterbrechungen auftreten können, wenn eine Interruptbehandlungs-routine noch nicht abgeschlossen ist. Dazu werden in Kratos Interruptbehandlungs-routinen in zwei Teile unterteilt. Das sind zum einen der Prolog, der unmittelbar mit gesperrten Unterbrechungen abzuarbeiten ist, und zum anderen der eventuell verzögerte Epilog.

Des Weiteren baut auch die Schicht der Treiber direkt auf der Hardware auf und zieht dabei eine direkte Verbindung zur Ebene der Anwendungen. Treiber können dabei direkt auf der Hardware arbeiten oder auf anderen Treibern aufbauen.

Die oberste Schicht ist die der Anwendungen, welche der Scheduler verwaltet. Diese Anwendungen erfüllen die eigentliche Aufgabe des eingebetteten Systems und werden als Klassen implementiert [23].



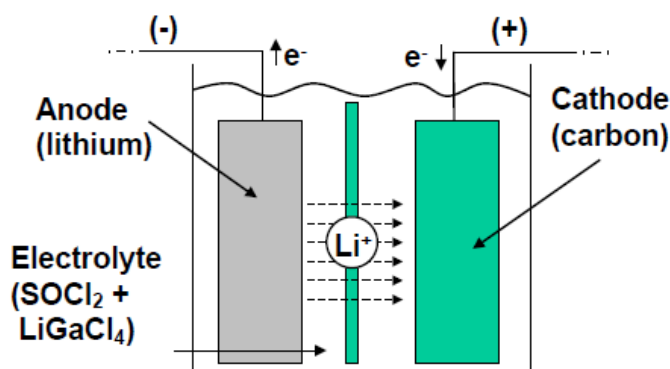


Abbildung 2.9.: Grundaufbau einer Batterie am Beispiel Lithium/Thionylchlorid Batterie [46].

## 2.2. Batterien

Im Allgemeinen ist eine Batterie eine Anordnung von galvanischen Zellen, die eine Gesamtstromquelle ergeben. Dabei sind galvanische Zellen elektrolytische Elemente, die elektrische Energie durch elektrochemische Vorgänge erzeugen. Der Aufbau einer Batterie kann dabei aus Primär- wie auch aus Sekundärzellen erfolgen. Bei Sekundärzellen ist die stromliefernde Reaktion reversibel, d.h. sie können wieder aufgeladen werden und deshalb auch als Akkumulator bezeichnet werden. Primärzellen lassen sich dagegen nur einmal entladen [39]. Im weiteren Verlauf dieser Arbeit ist mit der Bezeichnung Batterie immer ein Akkumulator gemeint.

Also ist eine Batterie eine Vorrichtung, welche chemische Energie mittels elektrochemischer Oxidation-Reduktion (Redo) Reaktion in elektrische Energie umwandelt und mit der Umkehrung des Prozesses wiedergeladen werden kann[48]. Eine Batterie besteht im Wesentlichen aus drei Hauptkomponenten:

1. Anode oder negative Elektrode,
2. Kathode oder positive Elektrode,
3. Elektrolyt.

Diese drei Komponenten bilden den Grundaufbau einer Batterie und sind in der Abbildung 2.9 dargestellt.

### 2.2.1. Prinzipien der Entladung einer Batterie

Wie oben schon erwähnt, besteht eine Batterie aus einer Anode, einer Kathode und einem Elektrolyt. Das Elektrolyt trennt die beiden Elektroden voneinander und stellt einen Mechanismus zum Transport von Ladung zwischen ihnen zur Verfügung. Die nachfolgende Erläuterung des Prozesses der Entladung einer Batterie wird am Beispiel einer Lithium/Thionylchlorid Batterie beschrieben.

Während der Entladung (einer Batterie) findet eine Oxidation an der Anode ( $Li$ ) statt, welche dazu führt, dass Elektronen erzeugt werden. Diese Elektronen fließen durch den externen Stromkreis. Zeitgleich bewegen sich positiv geladene Ionen ( $Li^+$ ) mit Hilfe der Diffusion durch das Elektrolyt ( $SOCl_2$ ) und von der Anode zur Kathode. In der Kathode treten Reduktionsreaktionen an freien Reaktionsstellen auf, dabei werden negativ geladene Ionen ( $Cl^-$ ) generiert. Diese Ionen vereinigen sich mit den positiv geladenen Ionen ( $Li^+$ ) zu einer unlöslichen Verbindung ( $LiCl$ ), welche sich in der Kathode ablagert. Stellen an denen diese Verbindung deponiert ist, werden inaktiv und können somit nicht mehr verwendet werden. Im Laufe des Entladeprozesses werden immer mehr und mehr Stellen unverfügbar, bis der Zustand der kompletten Entladung eingetreten ist [46].

### 2.2.2. Terminologie

Zum Verständnis von Batterie bzw. deren Beschreibungen und Datenblättern werden im Folgenden einige wichtige Begriffe erläutert [83].

Die nachfolgenden Variablen werden verwendet, um den aktuellen Zustand einer Batterie zu beschreiben:

- State of Charge (SoC) (dt. Ladezustand): Gibt die aktuelle Kapazität einer Batterie als Prozentsatz von der maximalen Kapazität an.
- Depth of Discharge (DoD) (dt. Entladezustand): Der Prozentsatz der Kapazität einer Batterie, welcher entnommen wurde, gemessen an der maximalen Kapazität.
- Terminal Voltage (dt. Klemmspannung): Die Spannung zwischen den Klemmen einer Batterie, wenn eine Last angeschlossen ist. Die Klemmspannung variiert mit dem SoC und dem Entlade-/Ladestrom. Einheit: Volt
- Open-Circuit Voltage ( $V_{oc}$ ) (dt. Leerlaufspannung): Die Spannung zwischen den Klemmen einer Batterie, wenn keine Last angeschlossen ist. Die Leerlaufspannung hängt von dem Ladezustand der Batterie ab und steigt mit dem SoC. Einheit: Volt

Folgenden Begriffe werden im Allgemeinen zur Beschreibung einer Batterie verwendet:

- Nominal Voltage (dt. Nennspannung): Ist die Referenzspannung der Batterie, manchmal auch als die „normale“ Spannung der Batterie bezeichnet. Einheit: Volt
- Cut-Off Voltage ( $V_{\text{cut}}$ ) (dt. Abschaltungsspannung): Ist die kleinste erlaubte Spannung. Im Allgemeinen wird mit diesem Wert der Zustand definiert, in dem die Batterie leer ist. Einheit: Volt
- Kapazität oder Nominale Kapazität: Die coulometrisch Kapazität, also die gesamten Amperestunden, die verfügbar sind, wenn die Batterie mit einem bestimmten Entladestrom von 100 % SoC bis zur Cut-Off Spannung entladen wird. Dabei wird die Kapazität wie folgt berechnet:  $Q = T * I$  mit  $T$  für Zeit (Einheit: Stunden h),  $Q$  für Ladung (Einheit: Amperestunden Ah) und  $I$  für Entladestrom (Einheit: Ampere A)
- Leistung: Die Leistung einer Batterie ist die Menge an elektrischer Energie, die pro Zeiteinheit entnommen werden kann. Ihre Einheit ist in der Regel Watt (W) und wird wie folgt berechnet:  $P = U * I$  mit  $P$  für elektrische Leistung,  $U$  für elektrische Spannung und der elektrischen Stromstärke  $I$ .
- Energieinhalt: Selten wird die in einer Batterie gespeicherten Energie (elektrische Arbeit) angegeben. Typisch ist die Angabe der Energie pro Masse oder pro Volumen. Dabei wird die elektrische Arbeit bei gegebener Spannung  $U$  wie folgt berechnet:  $W = U * Q$  mit  $W$  für die elektrische Arbeit (Einheit: Wattstunde Wh) und  $Q$  für die Ladung (Einheit: Amperestunden Ah).
- C-Rates (dt. C-Rate): Zum Beschreiben von Batterien wird oft der Entladestrom als C-Rate ausgedrückt, um gegen die Batteriekapazität zu normalisieren, da diese zwischen Batterien oft sehr unterschiedlich ist. Eine C-Rate ist ein Maß, der Rate mit einer Batterie relativ zu ihrer maximalen Kapazität entladen wird. Eine 1C Rate bedeutet, dass der Entladestrom die Batterie in 1 Stunde vollständig entlädt. Für eine Batterie mit 100 Ah ist das Äquivalent zu einem Entladestrom von 100 A.
- Maximum Continuous Discharge Current (dt. maximaler kontinuierlicher Entladestrom): Ist der maximale Strom, mit dem die Batterie entladen werden kann, ohne Schaden zu nehmen (Einheit: Ampere).
- Cycle Life (dt. Lebensdauer): Anzahl der Entlade-/Ladezyklen die die Batterie durchhält, bis sich spezifische Performance-Kriterien nicht mehr erfüllen.

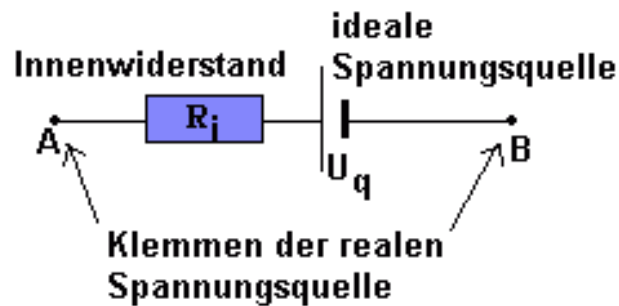


Abbildung 2.10.: Ersatzschaltbild einer Batterie [47].

- **Selbstentladung:** Batterien unterliegen bei der Lagerung einer Selbstentladung, dabei hängt die Geschwindigkeit der Selbstentladung unter anderem vom Batterietyp und der Temperatur ab. Je niedriger die Lagertemperatur, desto geringer ist die Selbstentladung.

### 2.2.3. Elektrotechnische Grundlagen

Eine wichtige Eigenschaft einer Batterie ist, dass diese wie jede reale Spannungsquelle einen Innenwiderstand besitzt. In der Batterie ist er in chemisch-galvanischen Widerständen begründet, denn der Ladungstransport mittels Ionenstrom durch das Elektrolyt und die dabei ablaufenden chemischen Reaktionen verbrauchen Energie. Wird also eine reale Spannungsquelle durch einen Widerstand belastet, sinkt die zwischen den Polen (Klemmen) auftretende Spannung ab. Sie ist also im belasteten Zustand kleiner als im unbelasteten Zustand. Zur Veranschaulichung wird eine reale Batterie durch ein Ersatzschaltbild als eine Serienschaltung aus einer idealen Spannungsquelle mit einer Quellenspannung (Urspannung)  $U_q$  und einem Widerstand (Innenwiderstand  $R_i$ ) dargestellt. Dies ist in Abbildung 2.10 zu sehen.

Wird also die Spannungsquelle durch einen Lastwiderstand belastet, liegt am Lastwiderstand  $R_L$  nicht die ganze Quellenspannung an den Klemmen A und B an, weil ein Teil im Innenwiderstand  $R_i$  der Batterie verloren geht (siehe auch Abbildung 2.11). So ergibt sich die Klemmenspannung  $U_{KL}$  durch:

$$U_{KL} = U_q - I * R_i \quad (2.10)$$

Die im unbelasteten Zustand zwischen den Klemmen liegende Spannung  $U_q$  wird als Quellenspannung oder Leerlaufspannung bezeichnet. Weiterhin sind die Grö-

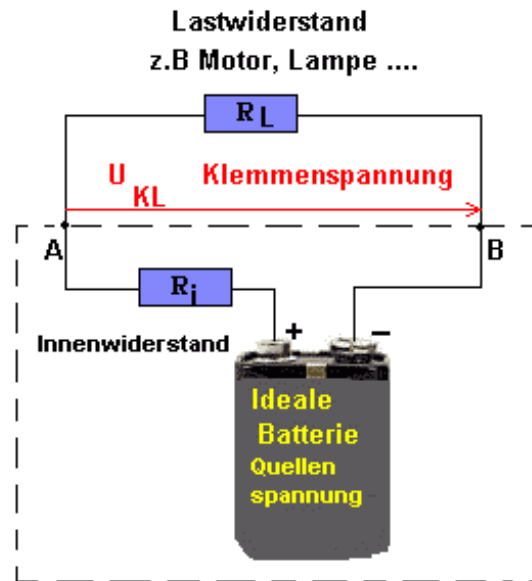


Abbildung 2.11.: Schematische Darstellung einer realen Batterie mit einem Lastwiderstand  $R_L$ . Dabei stellt die gestrichelte Umrandung die reale Batterie dar [47].

ßen  $U_q$  und  $R_i$  im Allgemeinen nahezu unabhängig von der Höhe des fließenden Belastungsstroms  $I$  [32].

Generell kann noch gesagt werden, dass der interne Widerstand für das Laden und Entladen verschieden und auch vom SoC einer Batterie abhängig ist. Dabei gilt: Wenn der interne Widerstand steigt, sinkt die Effizienz der Batterie. Weiterhin wird die thermische Stabilität reduziert, umso mehr Ladeenergie in Wärme umgewandelt wird [83].

#### 2.2.4. Nichtlineare Effekte

Die zwei wichtigsten Eigenschaften einer Batterie aus der Sichtweise des Nutzers, sind die Spannung und Kapazität. So hat eine ideale Batterie eine konstante Spannung während des gesamten Entladeprozesses und fällt sofort ab, wenn die Batterie ganz entladen ist. Weiterhin hat sie eine konstante Kapazität, egal wie groß der Entladestrom ist. Diese beiden Sachverhalte sind in der Abbildung 2.12 noch einmal graphisch zu sehen.

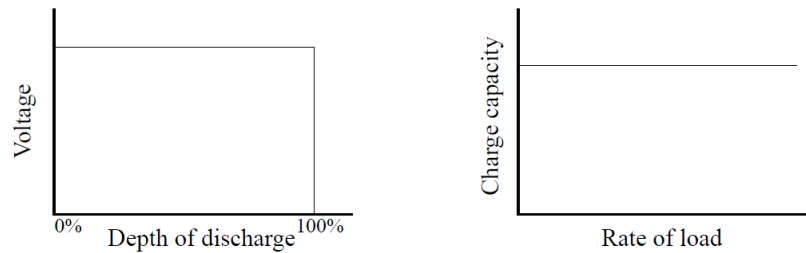


Abbildung 2.12.: Eigenschaften einer idealen Batterie: Konstante Spannung und Kapazität [58].

Während bei der idealen Batterie Spannung und Kapazität konstant sind, ist es in der Realität nicht so, d.h. es ist nicht möglich, die ganze Energie, welche in einer geladenen Batterie entladen wird, zu nutzen. Die Verluste sind meist in elektrochemischen Prozessen begründet und weisen ein nichtlineares Verhalten auf.

So zeigt die Abbildung 2.13 die Spannung einer Batterie als eine Funktion über die Entladezeit für zwei verschiedene Lasten. Die Last der Kurve 1 ist kleiner als die der Kurve 2, dabei ist die Spannung während der Entladung für die Kurve 2 kleiner als für die Kurve 1. Die niedrige Spannung der Kurve 2 lässt sich mit dem in Abschnitt 2.2.3 beschriebenen Innenwiderstand und weiteren Polarisationsseffekten von aktivem Material an beiden Elektroden begründen. Weiterhin fällt die Spannung während des Entladens durch den Anstieg des internen Widerstandes und Veränderungen im aktiven Material, wobei noch weiteren Faktoren eine Rolle spielen [48].

Die Einflüsse auf die Kapazität werden im Folgenden erklärt und sind Basis für die batteriegewahre Ablaufplanung.

Hierzu aber erst einmal noch ein paar Definitionen. Eine Batterie wird über ihre Leerlaufspannung ( $V_{oc}$ ) und ihre Cut-Off Spannung ( $V_{cut}$ ) charakterisiert. Die Kapazität einer Batterie lässt sich auf drei Wegen ausdrücken [46]:

- Theoretische Kapazität: Sie basiert auf der Menge der Energie, welche in der Batterie gespeichert ist, und ist eine obere Schranke für die Energie, die theoretisch aus der Batterie extrahiert werden kann.
- Standardkapazität: Dies ist die Energie, welche aus der Batterie unter Standardlastbedingungen entnommen werden kann. Die genauen Bedingungen dazu definiert der Hersteller der Batterie. Hierzu legt er z.B. die Last und Temperatur fest.

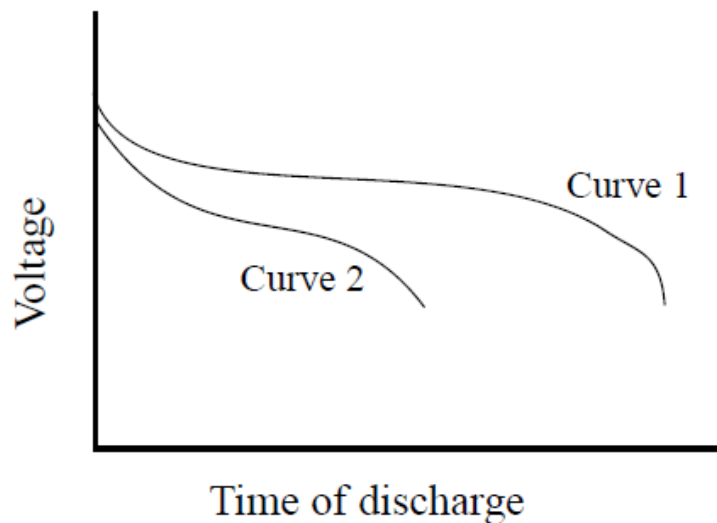


Abbildung 2.13.: Die Spannung einer Batterie als eine Funktion über der Entladezeit für zwei verschiedene Lasten [48].

- Tatsächliche Kapazität: Liefert die Menge an Energie, welche die Batterie unter einer gegebenen Last liefert.

Generell kann also zwischen der theoretisch vorhandenen Energie und der tatsächlich verfügbaren Energie unterschieden werden. Dabei kann die tatsächliche Kapazität zwar die Standardkapazität übersteigen, nicht aber die theoretische Kapazität. Im weiteren Verlauf dieser Arbeit ist immer die tatsächliche Kapazität gemeint. Die tatsächliche Kapazität lässt sich durch eine Vielzahl von Effekten beeinflussen, wobei im Folgenden zwei vorgestellt werden. Diese Effekte lassen sich in gewissen Grenzen über das Lastprofil steuern (welches z.B. über die Ablaufplanung gut beeinflussen lässt) und können so zu einer effizienteren Nutzung einer Batterie verwendet werden. So kann jetzt ein System als batterieeffizientes System bezeichnet werden, wenn die Entladeprofil-Eigenschaften des Systems zu einer Verbesserung der tatsächlichen Kapazität führen.

Das ist zum einen der *Rate Capacity* Effekt, welcher sich durch eine Abhängigkeit zwischen der tatsächlichen Kapazität und der Höhe des Entladestroms äußert und zum anderen ist es der *Recovery* Effekt, welcher sich in einer Erholung der Ladung während der Ruhephasen bzw. Phasen mit geringeren Lasten darstellt. Beide Effekte können die Kapazität und die Laufzeit einer Batterie beeinflussen.

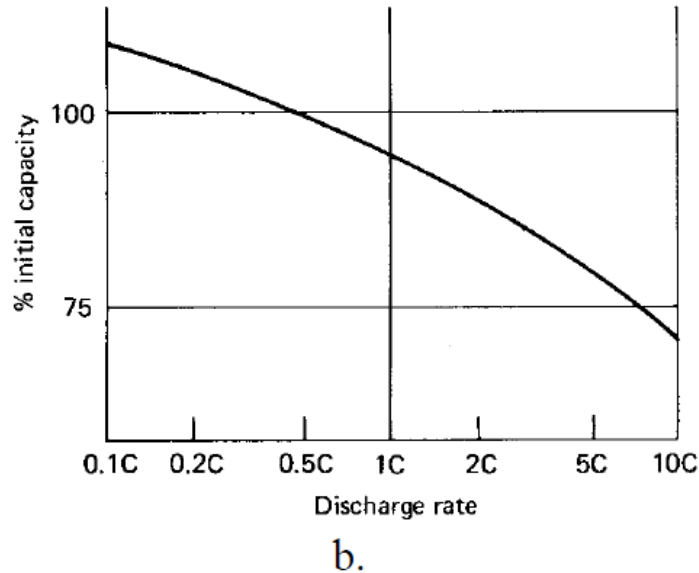


Abbildung 2.14.: Der Rate Capacity Effekt [48].

#### 2.2.4.1. Rate Capacity Effekt

Der Rate Capacity Effekt beschreibt einen Kapazitätsverlust mit steigendem Entladestrom. Abbildung 2.14 zeigt Kapazitätsverluste mit steigendem Laststrom für einen typischen Nickel-Cadmium-Akkumulator (NiCd) [48]. Zu sehen ist ein Kapazitätsverlust von 40% über einen Bereich von Entladezeiten von 10 (0,1 C-Rate) Stunden bis 0.1 (10 C-Rate) Stunden.

Begründet liegt dieser Effekt in elektrochemischen Reaktionen. Denn die Laufzeit einer Batterie hängt von der Verfügbar- und Erreichbarkeit von aktiven Reaktionsstellen in der Kathode ab. So ist die Verteilung von inaktiven Reaktionsstellen im Material der Kathode, während Phasen mit niedrigem Entladestrom mehr oder wenig gleichmäßig sind. Ist der Entladestrom hoch, treten Reduktionen nur an der äußeren Oberfläche der Kathode auf. Dadurch wird die Oberfläche der Kathode mit einer unlöslichen Verbindung ummantelt, welche den Zugriff zu vielen aktiven internen Reaktionsstellen verhindert. Als Ergebnis wird die Batterie als entladen angesehen, obwohl noch viele aktive Reaktionsstellen in der Kathode unbenutzt sind. Dies verringert die Kapazität der Batterie. Also ist der Effekt dieses Phänomens eine Abhängigkeit zwischen der Kapazität einer Batterie und der Rate mit der die Batterie entladen wird [20]. Um diese Effekte zum Vorteil auszunutzen, also



zur effizienten Nutzung der Batterie, sollte der Entladestrom also möglichst klein gehalten werden. In Bezug auf ein *Deeply Embedded System* sollte z.B. möglichst wenig Hardware parallel arbeiten.

#### 2.2.4.2. Recovery Effekt

Der Recovery Effekt äußert sich dadurch, dass eine Reduktion der Last an der Batterie für einen Zeitraum in einem Anstieg der Kapazität der Batterie bewirkt. Dieses Verhalten ist in der Abbildung 2.15 veranschaulicht. Während die Last reduziert wird, steigt die Spannung an und die Gesamtzeit der Entladung erhöht sich. So kann eine Entladung mit intermittierender Last mehr Kapazität aus der Batterie entnehmen, wie eine Entladung mit konstanter Last. Der Grad, bis zu dem sich die Batterie erholen kann, hängt von dem Entladestrom und der Länge der Zeit für die die Last reduziert ist, ab. Generell gilt, dass die Erholung größer nach höheren Entladeströmen ist. Weiterhin hängt der Effekt auch noch von weiteren Faktoren wie der Batteriekonstruktion, der Temperatur oder der Cut-Off Spannung ab.

Auch eine Ursache dieser Effekte ist in einem elektrochemischen Phänomen zu finden. So ist auch die Verfügbarkeit von geladenen Ionen ( $Li^+$ ) ein Faktor, der festlegt, welche Menge an Energie die Batterie liefern kann [24]. Wird kein Strom aus der Batterie entnommen, ist die Konzentration von positiv geladenen Ionen ( $Li^+$ ) gleichmäßig an der Elektroden-Elektrolyt-Schnittstelle verteilt. Wenn Strom aus der Batterie entnommen wird, werden positiv geladene Ionen an der Kathoden-Elektrolyt-Schnittstelle konsumiert und durch neue Ionen, welche von der Anode mittels Diffusion durch das Elektrolyt kommen, ersetzt. Ist aber die Stromentnahme aus der Batterie stark genug, kann der Diffusionsprozess nicht mehr mit dem Konsum von Ionen an der Kathode schritthalten. Als Ergebnis sinkt die Konzentration von positiv geladenen Ionen nahe der Kathode und steigt nahe der Anode. Dies führt zu einem Abfall der Batteriespannung [46]. Besteht für die Batterie die Möglichkeit einer Pause oder einer Reduktion der Last, kann vereinfacht gesagt werden, dass der Diffusionsprozess wieder nach kommt und eine Ladungserholung an der Kathode stattfindet. Dies resultiert in einer Steigerung der Kapazität und Laufzeit der Batterie.

Auch dieser Effekt lässt sich zum Vorteil in einem Deeply Emdebbed System nutzen, um eine effizientere Nutzung einer Batterie zu gewährleisten. So sollte nach einer hohen Belastung der Batterie immer eine Pause mit geringerer Belastung eingefügt werden.

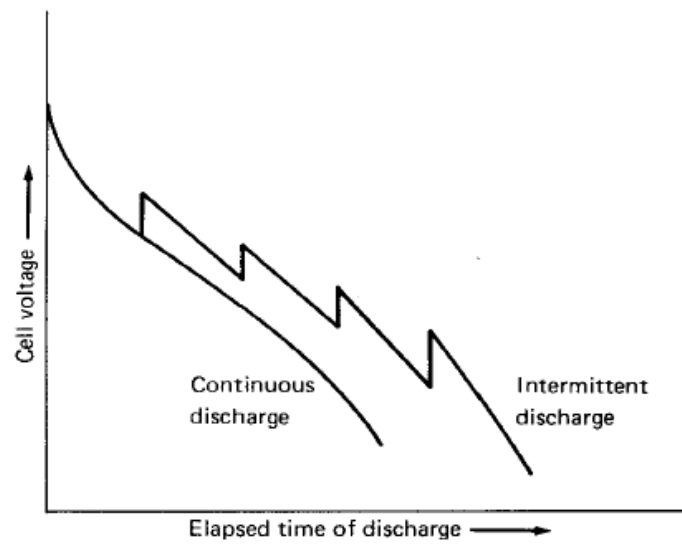


Abbildung 2.15.: Der Recovery Effekt bei einer Batterie. Die untere Kurve zeigt den zeitlichen Verlauf der Klemmspannung bei einem konstanter Entladestrom. Die obere Kurve zeigt den Verlauf der Klemmspannung bei einem intermittierenden Entladestrom, wobei die Pausen zur besseren Darstellung nicht in der Kurve dargestellt sind [48].

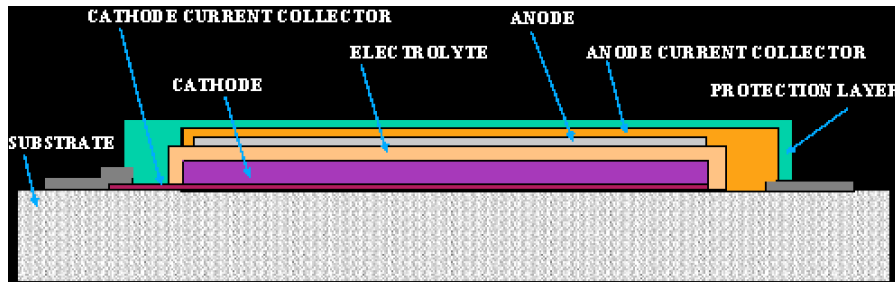


Abbildung 2.16.: Schematischer Querschnitt einer Dünnschichtbatterie [22].

### 2.2.5. Feststoff Lithium-Dünnschichtbatterien

Wie in der Einleitung bereits erwähnt, beschäftigt sich diese Arbeit nicht mit Batterien im Allgemeinen, sondern konzentriert sich auf den Bereich der Feststoff Lithium-Dünnschichtbatterie. Dies liegt zum einen daran, dass verschiedene Batterien aufgrund ihrer Konstruktion sehr unterschiedlich sein können und es so nicht möglich ist, allgemein gültige Aussagen zu treffen. Zwar gelten viele Eigenschaften für alle Batterien, aber sie unterscheiden sich dennoch im Detail.

Ein viel wichtigerer Aspekt zur Entscheidung der Untersuchung einer Lithium-Dünnschichtbatterie ist, dass diese Technologie relativ neu und aufgrund ihrer flachen, kleinen und sicheren Bauform vielversprechend für das Gebiet der *Deeply Embedded Systems* ist.

Der Entschluss zur Betrachtung der EnFilm EFL700A39 von ST als Exemplar einer Feststoff Lithium-Dünnschichtbatterien entstand in Zusammenarbeiten mit dem Fraunhofer-Institut für Materialfluss und Logistik (IML) und ist darin begründet, dass diese Art von Batterien noch sehr unerforscht ist. Im Folgenden wird kurz in die Technologie, die hinter der Lithium-Dünnschichtbatterie steckt, eingeführt und anschließend wird die in dieser Arbeit verwendete EnFilm EFL700A39 vorgestellt.

#### 2.2.5.1. Technologie

Der Name Feststoff Lithium-Dünnschichtbatterie beinhaltet gleich mehrere Eigenschaften, welche diesen Typ von Batterie aus- und auch besonders machen. Dabei steckt hinter dem Begriff Dünnschichtbatterie eine Fertigungsart, mit der die Batterie sehr flach produziert werden kann. Vereinfacht dargestellt werden die Komponenten der Batterie als dünner Film (kleiner als  $5\ \mu\text{m}$ ) auf einem Substrat abgelagert. Der typische Aufbau einer Dünnschichtbatterie ist in der Abbildung 2.16 zu sehen. Zum genaueren Verständnis des Verfahrens sei auf [22] verwiesen.

Der Begriff Lithium steht bei dieser Art von Batterien für das Material der Anode (der Batterie). Denn im Gegensatz zu „normalen“ Lithium-Ionen Batterien

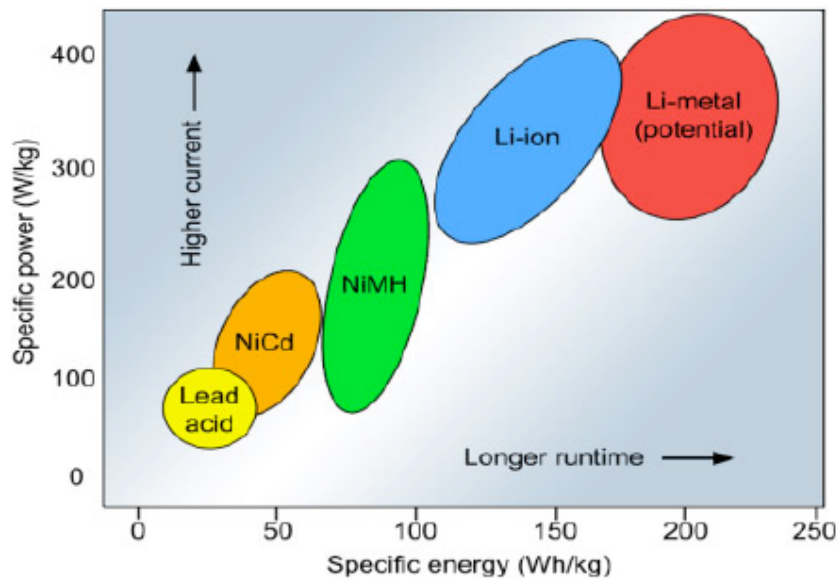


Abbildung 2.17.: Vergleich der verschiedenen existierenden Batterietypen [40].

besteht die Anode bei Lithium-Dünnschichtbatterien aus reinem Lithium-Metall und nicht aus einer Lithiumverbindung. Der Vorteil von Batterien auf Basis von reinem Lithium ist, dass diese eine höhere spezifische Leistung und Energie haben (siehe auch Abbildung 2.17).

Möglich ist die Verwendung von reinem Lithium als Anodenmaterial nur wegen einer weiteren Besonderheit von Feststoff Lithium-Dünnschichtbatterien, nämlich das verwendete Elektrolyt. Der Begriff Feststoff spielt dabei auf die Konsistenz des Elektrolyts an. Diese ist nämlich fest und nicht wie üblich bei Batterien flüssig, denn nur so ist es möglich, reines Lithium in Batterien, wegen der hohen Reaktivität von reinem Lithium gegenüber Wasser, zu verwenden. Meist wird als Elektrolytmaterial das am Oak Ridge National Labs entwickelte LiPON Feststoff Elektrolyt, welches ein amorphes Glas ist, verbaut [40].

Vorteile dieser Art von Batterien ist eine höhere Sicherheit durch das Feststoffelektrolyt, eine höhere Leistung und längere Lebensdauer. Nachteile bestehen in den höheren Fertigungskosten und auch in der größeren Temperaturempfindlichkeit [40].

### 2.2.5.2. EnFilm

Wie schon eingangs erwähnt, wurde die EnFilm EFL700A39 von ST exemplarisch als Feststoff Dünnschicht für diese Arbeit ausgewählt. Ihre genaue Bezeichnung lautet: EnFilm - Rechargeable Solid State Lithium Thin Film Battery. Abbildung 2.18 zeigt die EnFilm und verdeutlicht auch die Dimensionen der Batterie. Aufgebaut ist die EnFilm aus folgenden Materialien (siehe auch Abbildung 2.19):

- Die Kathode der EnFilm besteht aus  $\text{LiCoO}_2$ .
- Das Elektrolyt besteht aus LiPON Keramik.
- Die Anode ist aus reinem Lithium.

Sie zeichnet sich durch einen geringeren Kapazitätsverlust von 20 % über 10 Jahre, eine lange Lebenszeit und durch ihre Bauform aus. Weiterhin besteht aufgrund des Feststoff-Elektrolyts nicht die Gefahr von Feuer oder einer Explosion. Dabei verbindet sie die Vorteile eines Kondensators und einer Batterie durch eine flache Entladekurve (vgl. Abbildung 2.20) und einen schnellen Aufladevorgang (vgl. Abbildung 2.21). Geladen wird die Batterie mit einer konstanten Spannungsquelle von 4.2 V [79]. Geeignet ist die EnFilm für einen großen Bereich von Anwendungen, so z.B. für [79]:

- Internet der Dinge (Industrie 4.0),
- Sensornetzwerke,
- Smart Cards,
- RF ID Tags,
- Energiespeicher für Energy Harvesting.

In der Tabelle 2.1 sind alle wichtigen Eigenschaften der EnFilm EFL700A39 zusammengefasst [79]. Für weitere Informationen sei auf das Datenblatt verwiesen [79].

## 2.3. Verwendete Hardware

In den nächsten beiden Abschnitten wird zum einen das verwendete Messgerät namens U2722A USB Modular Source Measure Unit von Keysight vorgestellt, welches zur Kalibrierung und Validierung des Batteriemodells und bei der Evaluation des umgesetzten Gesamtkonzepts seine Verwendung fand. Anschließend wird das auch bei der Evaluation des Gesamtkonzepts verwendete Evaluationsboard MSP430FR5969 LaunchPad beschrieben.



Abbildung 2.18.: Die EnFilm EFL700A39 von ST [80].

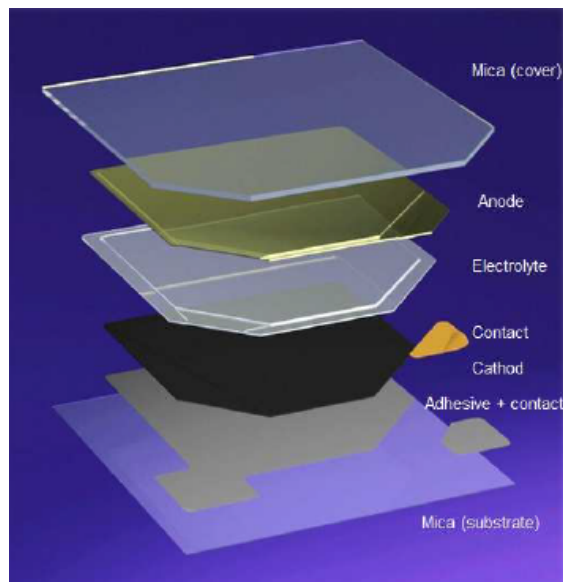


Abbildung 2.19.: Die Bestandteile der Enfilm [80].

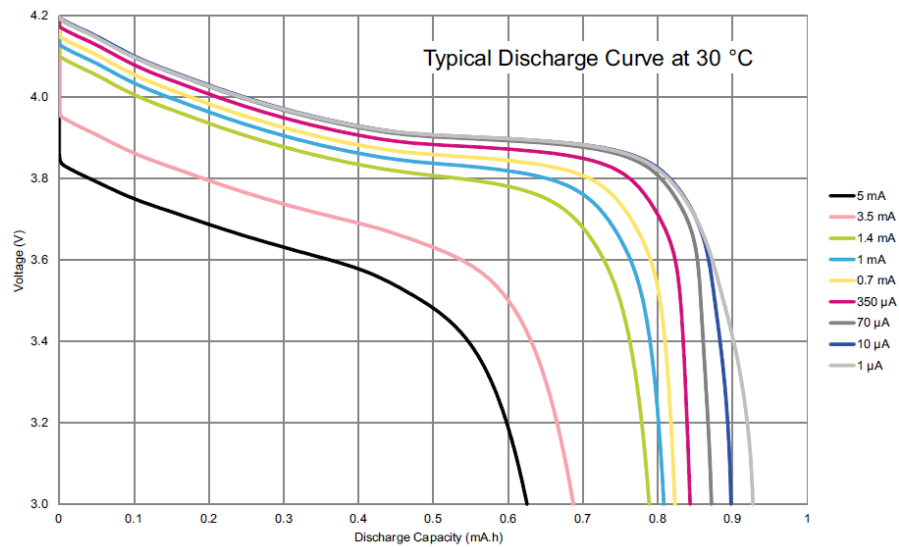


Abbildung 2.20.: Typische Entladekurven der EnFilm [79].

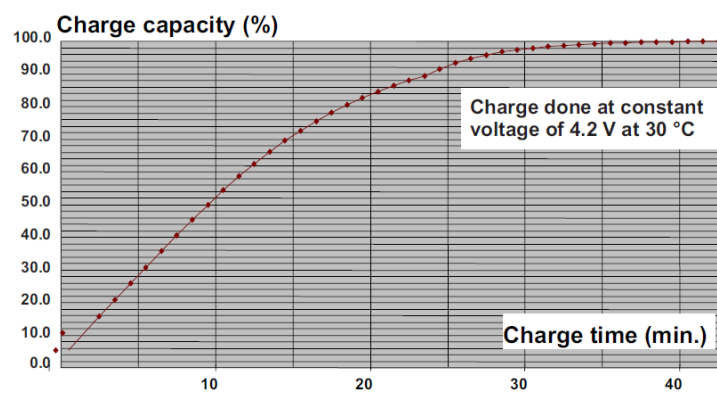


Abbildung 2.21.: Typische Ladekurve der Enfilm [79].

Parameter	EnFilm EFL700A39
Dimension	25.7 x 25.7 mm
Dicke	220 $\mu\text{m}$
Kapazität	700 $\mu\text{Ah}$
Nennspannung ( $V_{\text{nonminimal}}$ )	3.9 V
Betriebsspannung ( $V_{\text{op}}$ )	3.0 bis 4.2 V
Interner Widerstand ( $R_{\text{int}}$ )	100 ohm
Maximaler kontinuierlicher Entladestrom ( $I_c$ )	5 mA
Lebensdauer ( $C_{\text{lfe}}$ ) <sup>1</sup>	4000 Zyklen
Selbstentladung ( $S_{\text{Disch}}$ )	3 % Lagungsverlust pro Jahr
Betriebstemperatur ( $T_{\text{op}}$ )	-20 bis 60 °C

Tabelle 2.1.: Wichtige Merkmale der EnFilm EFL700A39 [79].

### 2.3.1. Messtechnik

Für die verschiedenen Messungen, die im Laufe dieser Arbeit an der Batterie und zur Evaluation nötig waren, wurde das Messgerät U2722A USB Modular Source Measure Unit von Keysight eingesetzt (siehe Abbildung 2.22). Vorteil dieses Messgerätes ist, dass eine Source Measure Unit im Vergleich z.B. zu einem normalen Amperemeter eine Gleichstromquelle ist, welche sowohl Strom und Spannung vorgeben, als auch beides gleichzeitig messen kann. Abbildung 2.23 zeigt dazu einen beispielhaften Aufbau einer Source Measure Unit mit den Leitungen *SMU Force* und *SMU Sens* zum Vorgeben und Messen von Strom und Spannung.

Eine weitere wichtige Eigenschaft der U2722A USB Modular Source Measure Unit ist, dass es ein 4-Quadranten Messgerät ist. Dadurch ist es mit dem Messgerät möglich in 4-Quadranten ( $\pm 20\text{V}$ ,  $\pm 120\text{mA}$ ) zu operieren (vgl. Abbildung 2.24). Dabei stehen bei der für diese Arbeit eingesetzten Variante des Messgeräts 3 Kanäle jeweils zum Messen und Vorgeben von Spannung und Strom zur Verfügung [85]. Weiterhin verfügt das Messgerät über eine hohe Messempfindlichkeit von 100 pA mit 16-Bit Auflösung und einer Basisgenauigkeit von 0.1 %. Ebenso ist das Messgerät fähig, Strommessungen bis in den nA Bereich durchzuführen [85]. Für weitere Angaben zu den Genauigkeiten in den verschiedenen Messbereichen sei auf das Datenblatt [85] verwiesen.

Ferner ist es mit dem Gerät möglich, eine Kelvin (4-Kabel) Messung durchzuführen. Damit wird ausgeschlossen, dass eine Messung durch den Widerstand der Kabel, welche zur Messungen verwendet werden, beeinflusst wird. Dazu werden bei einer Kelvin (4-Kabel) Spannungsmessung die Effekte des Widerstands der Kabel, durch Separierung der Kabel, welche den Strom vorgeben, von denen, die die Spannung messen, eliminiert (siehe auch Abbildung 2.25) [84].





Abbildung 2.22.: U2722A USB Modular Source Measure Unit [87].

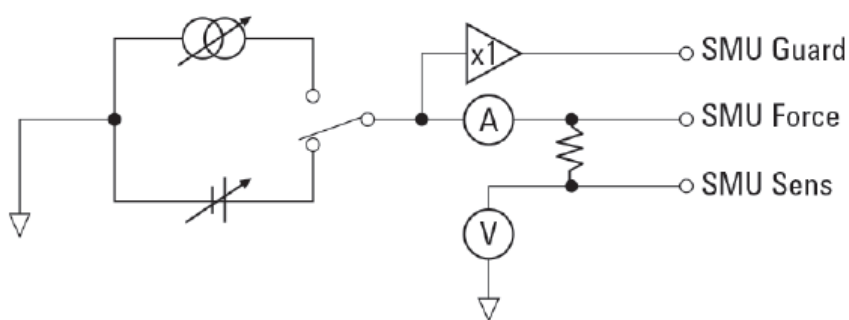


Abbildung 2.23.: Aufbau einer Source Measure Unit [86].

Die U2722A USB Modular Source Measure Unit wird per USB 2.0 an einen PC angeschlossen und kann zum einen mit Software vom Hersteller Keysight gesteuert werden, oder aber es kann eine selbst entwickelte Software mit verschiedenen Application Development Environment wie Keysight VEE, NI LabView and Microsoft Visual Studio erstellt werden [87].

In dieser Arbeit reichte es, sich auf den Keysight Measurement Manager (AMM) zu beschränken. Diese Software bietet eine graphische Benutzeroberfläche (vgl. Abbildung 2.26), mit deren Hilfe es möglich ist, das Messgerät zu konfigurieren, Messungen durchzuführen und die Messwerte aufzuzeichnen. Die aufgezeichneten Messwerte können beispielsweise als CSV exportiert werden.

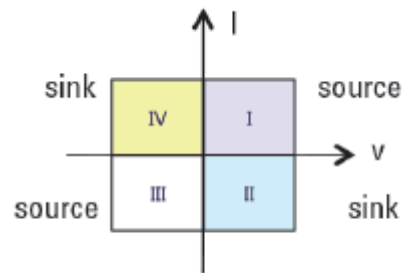


Abbildung 2.24.: 4-Quadranten Operationen [86].

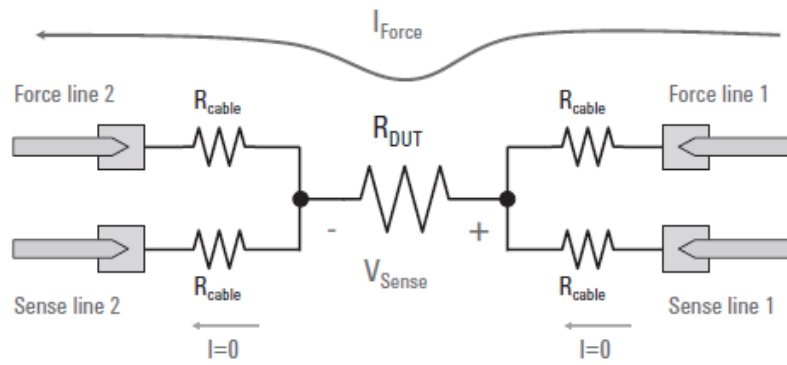


Abbildung 2.25.: Kelvin (4-Kabel) Messung [84].

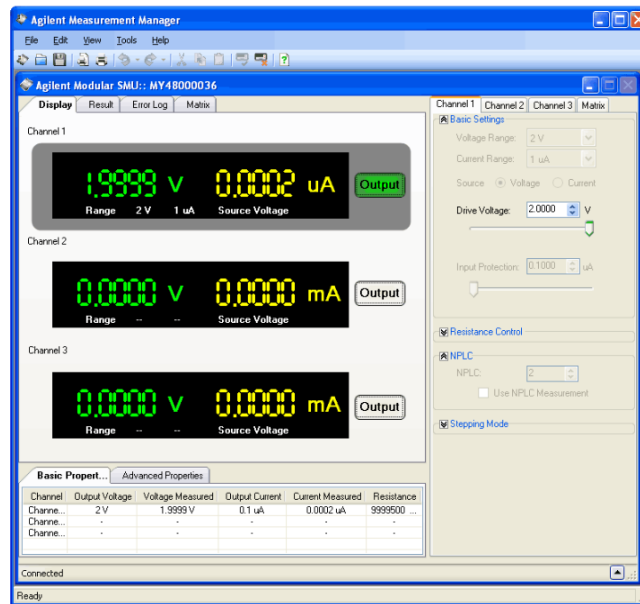


Abbildung 2.26.: Keysight Measurement Manager (AMM) [87].

### 2.3.2. Evaluationsboard

Zur Evaluation von einer batteriegewahren Ablaufplanung auf einem realen System war das MSP430FR5969 LaunchPad von Texas Instruments zur Ansteuerung der Peripherie mit dem in Abschnitt 2.1.3 bereits erwähnten Betriebssystem Kratos angedacht. Da aber zum Zeitpunkt der Evaluation dieser Arbeit keine funktionsfähige Batterie mehr zur Verfügung stand, konnte die Evaluation nur simulativ durchgeführt werden und nicht mit dem MSP430FR5969 LaunchPad. Trotzdem wird der angedachte Aufbau des Evaluationsystems im Kapitel 7 Evaluationsboard und deshalb wird im Folgenden das Evaluationsboard erläutert.

Das MSP430FR5969 LaunchPad ist ein Mikrocontroller Development Board mit dem MSP430FR5969 Mikrocontroller (siehe Abbildung 2.27). Es dient zur Erprobung und Evaluation von Schaltungen mit MSP430. Der MSP430FR5969 Mikrocontroller ist ein Ultra-Low-Power 16-Bit RISC Architektur Mikrocontroller mit 16 MHz und zeichnet sich weiter durch den eingebetteten FRAM<sup>4</sup> aus, welcher ein nicht flüchtiger Speicher ist, bekannt für seine geringe Leistungsaufnahme und hohe Schreibgeschwindigkeit [38].

Die Anschlüsse des Mikrocontrollers auf der Platine sind dabei größtenteils zu zwei Pfostenleisten geführt, um möglichst einfach zusätzliche Peripherie verwenden

<sup>4</sup>Ferroelectric Random Access Memory



Abbildung 2.27.: Das MSP430FR5969 LaunchPad. Zu sehen ist unten zum einen die Hauptplatine mit dem MSP430FR5969 Mikrocontroller und oben die integrierte Programmierschaltung zum Aufspielen der Firmware und Debuggen [38].

zu können. Ferner stehen zwei Taster und zwei LED's frei zur Verwendung, und ein Kondensator zur Spannungsversorgung kann optional hinzugeschaltet werden. Des Weiteren befindet sich auf der Platine ein Programmiermodul, mit dessen Hilfe der Mikrocontroller per USB programmiert werden kann. Auch ist darüber das Debuggen und Aufspielen von Firmware möglich [23].

## 3. Verwandte Arbeiten

Die Berücksichtigung von Energiequellen im Speziellen von Batterien und deren effizienteren Nutzung im Design von eingebetteten Systemen, ist mit dem Gebiet des batteriegetriebenen System Design Gegenstand der aktuellen Forschung. Dieses Kapitel gibt einen Überblick über die verschiedenen ähnlichen Ansätze, stellt Gemeinsamkeiten und Abgrenzungen und zu dieser Masterarbeit dar, wobei einer der Ansätze auch Grundlage dieser Arbeit ist. Deshalb stellt dieses Kapitel zuerst die vorausgegangene Arbeit von Michael Hesse aus dem Jahre 2015 vor, da diese Arbeit Basis dieser Untersuchung ist, und geht besonders auf die Teile ein, die auch für diese Arbeit relevant sind. Anschließend werden weitere ähnliche Arbeiten vorgestellt, welche mit verschiedenen Ansätzen versuchen, Batterien effizienter auszunutzen.

### 3.1. Vorausgegangene Arbeit

Die Idee zu dieser Arbeit kam, wie eingangs schon erwähnt, durch die Diplomarbeit „Berücksichtigung nicht-linearer Energiequelleneffekte im eingebetteten Betriebssystem Kratos“ [34] von Michael Hesse. Die Zielsetzung dieser Diplomarbeit ist es, durch die Planung von Zugriffen auf Hardwarekomponenten unter Berücksichtigung des internen Zustands bzw. der nicht linearen Effekte eines Kondensators, diesen möglichst effizient auszunutzen. Das Ergebnis ist ein offline Scheduler, welcher die Betriebszeit und Servicequalität eines eingebetteten Systems optimiert. Auf diesen offline Scheduler soll im Folgenden zum besseren Verständnis etwas genauer eingegangen werden.

Grundlage der Arbeit von Michael Hesse ist die Annahme, dass die Zeitpunkte, zu denen Energie verbraucht wird, einen Einfluss auf den inneren Zustand einer Energiequelle haben und dass der innere Zustand wiederum bestimmt, wie viel Energie eine Quelle zu einem Zeitpunkt abgeben kann und welcher Anteil der gespeicherten Energie verloren geht. Es ist also der Zeitpunkt entscheidend, zu dem Energie aus der Quelle entnommen wird.

Ein Anlass zu weiteren Untersuchungen ist die Annahme, dass bei eingebetteten Systemen im Speziellen bei *Deeply Embedded System*, wie auch schon beschrieben, der Energieverbrauch maßgeblich durch die Hardwareaktivitäten bestimmt

wird, welche durch die Anwendungssoftware über Treiberfunktion initiiert werden. Weiterhin könnte das Betriebssystem über ein Scheduling festlegen, zu welchem Zeitpunkt die Hardware angesteuert wird. Insgesamt werden also folgende Punkte beachtet [34]:

- Durch den internen Zustand einer Energiequelle wird der optimale Zeitpunkt für einen Energieverbrauch bestimmt.
- Jeder einzelne Energieverbrauch beeinflusst den Zustand der Energiequelle.
- Durch nichtlineare Effekte oder parasitäre Leckwiderstände kann sich jederzeit der Zustand der Energiequelle von alleine ändern.
- Die Nutzung der nichtlinearen Energiequelleneffekte können widersprüchliche Anforderungen an das Scheduling stellen.

Zur Lösung dieses Problems wurde der besagte offline Scheduler entworfen, welcher unter Berücksichtigung der angeschlossenen Energiequelle die richtigen bzw. optimalen Zeitpunkte für die durch die Anwendungssoftware verursachten Aufrufe von Hardwaretreibern bestimmen sollte. Dazu wurde der Begriff des Scheduling in der Diplomarbeit als die Menge der Verzögerungszeiten für eine gegebene Anwendung definiert. Denn wie schon erwähnt lassen sich viele nichtlineare Energiequelleneffekte durch die Wahl des Zeitpunktes einer Belastung kontrollieren. Weiterhin sind die Belastungen durch Aufrufe von Hardwaretreiberfunktionen des Betriebssystems zu bestimmen, so dass eine Verzögerung dieser Aufrufe auch den Zeitpunkt der Belastung einer Energiequelle bestimmt [34]. Zur Verzögerung werden die entsprechenden Tasks vorübergehend angehalten. Die genannten Verzögerungen können dabei aber auch an jeder anderen beliebigen Stelle als nur vor Treiberaufrufen verwendet werden.

Ferner ist es möglich, zeitliche Rahmenbedingungen für die Verzögerungen bzw. Pausen, welche für die einzelnen Treiberaufrufe eingehalten werden sollen, vorzugeben. Diese ist meist durch die angesteuerte Hardware bestimmt (z.B. Bustiming). So kann für die Pausendauer ein zeitlicher Rahmen von Mikrosekunden bis Tagen vorgegeben werden.

Des Weiteren wird in der Diplomarbeit im Zusammenhang mit dem Scheduler der Begriff der Servicequalität eingeführt. Dabei ist die Servicequalität unter einem gegebenen Energievorrat eine Abwägung zwischen der Betriebszeit und Häufigkeit, mit der die Anwendung einzelne Aufgaben durchführt [34]. Diese Servicequalität lässt sich mit Hilfe des Schedulers für ein gewünschtes Scheduling festlegen.

Der Grund für ein offline Scheduler liegt in der Annahme begründet, dass der Rechenaufwand für ein optimales Scheduling sehr hoch ist, denn zum Planen der

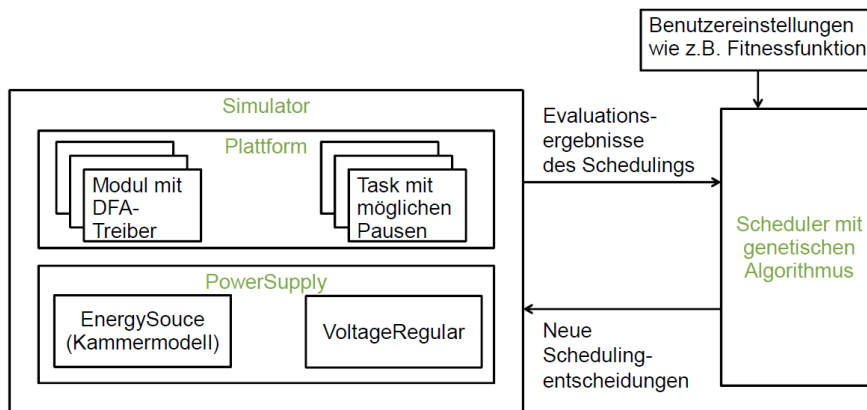


Abbildung 3.1.: Schematische Darstellung des offline Schedulers mit seinen Bestandteilen zur Ermittlung einer energiequellengewahren Ablaufplanung.

Energieverbräuche muss der Zustand der Energiequelle zu jedem Zeitpunkt bekannt sein, ebenso alle zukünftig durch die Software und anderweitig verursachten Hardwareaktivitäten bekannt sein müssen [34]. Weiterhin wurde angenommen, dass ein online Scheduler aufgrund seines Ressourcenbedarfs und Rechenaufwands mehr Energie verbrauchen würde, als durch ein energiequellengewahres Scheduling gespart werden kann. Auch sprach gegen ein online Scheduler die Annahme, dass hierfür Greedy-Algorithmen verwendet werden, welche nur die nächsten Energieverbräuche betrachten und so, im Vergleich mit dem Traveling Salesman Problem, nicht von einer guten Teillösung auf eine gute Gesamtlösung geschlossen werden kann [34]. Er kommt zu dem Schluss, dass sich in der Theorie und Praxis erkennen lässt, dass sich eine Energiequelle, wie ein Kondensator, unter Berücksichtigung der nichtlinearen Effekte, effizienter ausnutzen lässt. Der Gesamtaufbau des offline Schedulers ist in der Abbildung 3.1 zu sehen. Der offline Scheduler besteht dabei im Wesentlichen aus einem Simulator und dem eigentlichen Scheduler. Im Folgenden werden die einzelnen Komponenten und deren Funktionsweise genauer erläutert. Dabei sind die folgenden Abschnitte angelehnt an [34].

### 3.1.1. Simulator

Der Simulator wird benötigt, um ermitteln zu können, wie effizient ein bestimmtes Scheduling eine Energiequelle verwendet. Dabei bildet der Simulator die Energiequelle, die einzelnen Hardwarekomponenten und die Abarbeitung der Anwendungs-

software nach. Der Simulator beschränkt sich bei der Simulation auf den Anteil, der für den Energieverbrauch entscheidend ist, dies bedeutet, er simuliert die Zeitpunkte, zu denen sich der Energieverbrauch des Systems verändert sowie die Auswirkungen der Veränderung des Energieverbrauchs auf die Quelle. So ist er in der Lage zu berechnen, wie viele Aktionen über welchen Zeitraum die Anwendungssoftware durchführen kann. Der Simulator wurde von Grund auf neu entwickelt. Es wurde nicht auf bestehende Frameworks zurückgegriffen, weil diese einen anderen Schwerpunkt haben oder deren Simulation zu umfangreich ausfällt. So konnte auch der Aufwand zur Modellierung der Hard- und Software reduziert werden.

### 3.1.2. Modellierung des Gesamtsystems

Wie in Abbildung 3.1 zu sehen ist, bekommt der Simulator als Eingabe das zu simulierende Gesamtsystem. Dazu wird jetzt im Folgenden beschrieben, wie sich dieses System zusammensetzt und wie es modelliert wird.

Das Gesamtsystem ist hierarchisch aufgebaut. Auf der obersten Ebene ist die Plattform und die Stromversorgung definiert. In der Plattform enthalten sind die Module und Tasks. Die Module modellieren die verwendeten Hardwarekomponenten und die Tasks spiegeln die vom Betriebssystem zu steuernden Tasks wieder. Für jeden Task wird über die Reihenfolge in der Systemdefinition eine Priorität und Gewichtung vergeben, wobei die Gewichtung zur Steuerung und Bewertung von Scheduling genutzt werden kann. Die Stromversorgung teilt sich wiederum in Modelle für einen Spannungsregler und für eine Energiequelle auf. Bei Annahme einer konstanten Spannungsquelle kann auch die Definition eines Spannungsreglers entfallen. Andernfalls ist eine Definition nötig, weil die Module von einer konstanten Versorgungsspannung ausgehen. Die einzelnen Bestandteile werden in XML definiert, wobei folgendes zu beachten ist:

- Alle Angaben der Zeit erfolgen in Mikrosekunden.
- Angaben wie Spannungen und Energie werden in ihrer jeweiligen Einheit angegeben.

In Listing 3.1 ist exemplarisch eine Definition eines Gesamtsystems angegeben [34].

Listing 3.1: Definition eines Systems in XML.

```
<systemdefinition>
  <platform>
    <name>SimpleWSN</name>
```



```

    <shortdescription>Ein einfacher
      Sensorknoten</shortdescription>
    <modules>
      <path>sensor.xml</path>
      <path>display.xml</path>
      <path>wifi.xml</path>
    </modules>
    <tasks>
      <path weight=" 0.6 ">measuring.xml</path>
      <path weight=" 0.3 ">send.xml</path>
      <path weight=" 0.1 ">show.xml</path>
    </tasks>
  </platform>

  <powersupply>
    <voltage regulator type=" tablebasedvoltage regulator "
      path=" MCP1640voltageRegulator.xml ">
      <outputvoltage>3.3</outputvoltage>
      <source type=" supercap " path=" redcap.xml "/>
    </voltage regulator>
  </powersupply>
</systemdefinition>

```

### 3.1.2.1. Energiequelle

Da die Modellierung der Energiequelle in dieser Arbeit von Grund auf neu gestaltet wird, wird hier nur zur Vollständigkeit die Modellierung von Michael Hesse kurz angerissen und für weitere Informationen auf [34] verwiesen. Als Energiequelle ist in Arbeit von Michael Hesse ein Kondensator modelliert worden, wobei die modellierenden nicht nichtlinearen Effekte der Selbstentladung und der Erholung berücksichtigt werden. Gewählt wurde ein Kondensator, weil die genannten Effekte in ähnlicher Form auch in anderen Energiequellen anzutreffen sind und weil die Effekte in einem zeitlichen Rahmen von Minuten bis Stunden ablaufen, der es gestattet, Messungen zu Evaluationen durchzuführen. Weiterhin wurde noch ein Spannungsregler modelliert, da für die Verwendung von DFA-Treibern eine konstante Spannungsversorgung notwendig ist. Zur Modellierung eines Kondensators wurde von Michael H. das Kammermodell nach Dow [19] verwendet. Dies modelliert die Effekte der Selbstentladung durch einen Leckwiderstand und die dielektrische Relaxation (Erholung) mit Hilfe von mehreren parallelen RC-Kreisen. Abbildung 3.2 zeigt das verwendete Kammermodell. Für den Spannungsregler wurde aufgrund des fehlenden internen Zustands ein einfaches tabellengestütztes Modell verwendet. Sowohl der Kondensator als auch der Spannungsregler werden auch wieder mit

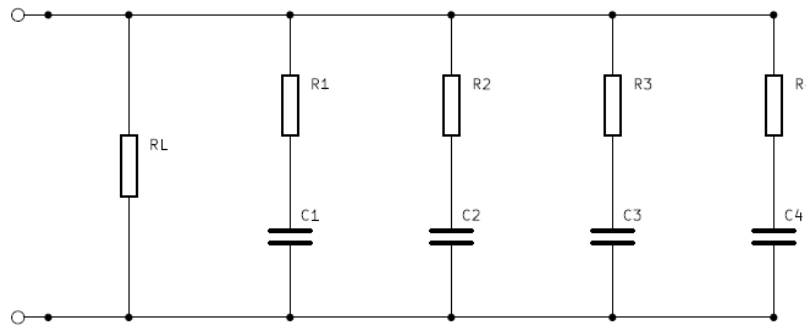


Abbildung 3.2.: Das Kammermodell zur Modellierung eines Kondensator [34].

Hilfe einer Definition in XML modelliert.

### 3.1.2.2. Module

Module beschreiben wie schon gesagt, die einzelnen Hardwarekomponenten des zu simulierenden Systems. Die Beschreibung der Hardware beschränkt sich auf den Energieverbrauch bzw. den Leistungsbedarf der Hardware, da nur diese für das Scheduling relevant sind. Dazu hat Michael Hesse die in den Grundlagen vorgestellten DFA-Treiber bzw. Priced-Time-Automata von Robert Falkenberg verwendet. Diese bilden die Zustände der Hardwarekomponenten auf Zustände der Automaten ab und beinhalten die benötigte Leistung für die Zustände und die benötigte Energie für Zustandsübergänge. Zustandsübergänge werden durch Timeouts oder über Tasks direkt ausgelöst.

Auch Module werden wieder als XML dem Simulator zur Verfügung gestellt. Hierzu zeigt Listing 3.2 exemplarisch die Definition in XML für den Automaten in der Abbildung 2.6 aus Kapitel 2 [34]. Wie in den Grundlagen zu den DFA-Treibern beschrieben, werden auch die Module in dieser Arbeit aufgrund des Verzichts auf einen Spannungsregler nicht in der Form für diese Arbeit weiter verwendet, worauf noch in Kapitel 6 eingegangen wird.

Listing 3.2: Definition eines Moduls in XML.

```
<module>
  <name>CommunicationModule</name>
  <initialstate>Off</initialstate>

  <states>
    <state>
      <name>Off</name>
```

```

        <power>0</power>
    </state>
    <state>
        <name>Idle</name>
        <power>1e-3</power>
    </state>
    <state>
        <name>RxTx</name>
        <power>3e-3</power>
    </state>
</states>

<transitions>
    <transition>
        <name>idle</name>
        <srcstate>Off</srcstate>
        <deststate>Idle</deststate>
        <energy>3e-9</energy>
    </transition>
    <transition>
        <name>destroy</name>
        <srcstate>Off</srcstate>
        <deststate>Idle</deststate>
        <energy>1e-9</energy>
    </transition>
    <transition>
        <name>transmit</name>
        <srcstate>Idle</srcstate>
        <deststate>RxTx</deststate>
        <energy>10e-9</energy>
    </transition>
    <transition>
        <name>timeoutTransition</name>
        <srcstate>RxTx</srcstate>
        <deststate>Idle</deststate>
        <energy>1e-9</energy>
        <timeout>1000</timeout>
    </transition>
</transitions>
</module>

```

### 3.1.2.3. Tasks

Softwaretasks, die von einem Betriebssystem gescheduled werden, werden durch Tasks modelliert. Auch hier wird nicht der ganze Task modelliert, sondern es wer-

den nur die Programmteile beschrieben, welche Treiberaufrufe beinhalten, aufgrund der Annahme, dass Treiberaufrufe maßgeblich für den Energieverbrauch verantwortlich sind. Sollte aber noch andere Hardware den Energieverbrauch beeinflussen (z.B. Wechsel der MCU vom Aktiv- in den Idle-Modus), kann diese Hardware auch als Modul definiert werden. Aufgrund dessen, dass ein statisches Scheduling generiert werden soll, müssen Treiberaufrufe deterministisch sein. Daraus folgt, dass es keine bedingten Treiberaufrufe geben darf und Schleifen nur mit einer bekannten Anzahl von Durchläufen erlaubt sind. Der Aufbau eines Tasks wird jetzt im Folgenden anhand eines Beispiels erklärt. Ein Task soll vier Mal am Tag jeweils 100 Byte über das Kommunikationsmodul aus Abschnitt 3.1.2.2 versenden. Hierbei ist zu beachten, dass häufig zeitliche Einschränkungen zur Ansteuerung von Hardware eingehalten werden müssen. Diese zeitlichen Einschränkungen werden dem Scheduler durch spezielle Ausdrücke direkt an den Stellen, wo eine Pause eingefügt werden soll, mitgeteilt. Diese Pausen sollten so großzügig wie nur möglich sein, damit der Scheduler einen ausreichenden Spielraum für einen optimalen Wert bezüglich eines energiequellengewahren Scheduling hat. Das Listing 3.3 zeigt in Pseudocode den Beispiel Task mit den erlaubten Rahmen für die Pausen [34].

Listing 3.3: Task mit Angabe von Pausen.

```

Wecke Kommunikationsmodul auf
for i = 1 to 100
  for j = 1 to 8
    sende Bit
    Schlafe 1 – 10 ms
  endfor
  Schlafe 10 – 2000 ms
endfor
Schicke Kommunikationsmodul in den Schlafmodus
Schlafe 71/2 – 81/2 Stunden

```

Zur Definition eines Tasks in XML stellt Michael Hesse eine Reihe von Anweisungen (Actions) zur Verfügung:

- `<changestate module="x" transition="y" \>`  
Führt bei Modul x die Transition y durch. Weiterhin wird der ggf. damit verbundene Energieverbrauch und die Änderung in der Leistung mit dem Energiemodell verrechnet.
- `<delay duration="x"\>`  
Mit dieser Anweisung wird der Task für x Mikrosekunden angehalten.
- `<pause minduration="x" maxduration="y" label="text"\>`  
Angabe einer Pause für den Task mit den Grenzen (von x bis y) innerhalb

deren der Scheduler den genauen Wert für die Pausen festlegen kann. Mit Hilfe des Labels können später in der realen Anwendung die Pausen an die richtige Stelle im Quellcode eingefügt werden.

- `<loop iteration="x"\>` und `<endloop\>`  
Anweisungen zum Öffnen einer Schleife mit x Durchläufen und Schließen der selbigen. Dabei gilt, dass für Pausen innerhalb der Schleife der Scheduler nur einen Wert für alle Schleifendurchläufe bestimmt. Für individuelle Pausen pro Durchlauf muss die Schleife abgerollt werden.
- `<lock module="x"\>` und `<unlock module="x"\>`  
Anweisungen für den exklusiven Zugriff eines Tasks auf Module und deren Freigabe. Alle Task die, auf ein gelocktes Modul zugreifen, werden suspendiert, bis das Modul wieder freigegeben wird.

Mittels dieser Anweisungen lässt sich z.B. der Task aus Listing 3.3 in XML modellieren. Die zugehörige Definition in XML ist in Listing 3.4 zu finden [34].

Listing 3.4: Definition eines Tasks in XML.

```

<task>
  <name>sendTask</name>
  <program>
    <changestate module="CommunicationModule" transition="idle" />
    <loop iterations="100" />
      <loop iterations="8" />
        <changestate module="CommunicationModule"
          transition="transmit" />
        <pause minduration="1000" maxduration="10000"
          label="pauseAfterTransmit" />
      <endloop />
      <pause minduration="10e3" maxduration="2000e3"
        label="pauseAfterSendbyte" />
    <endloop />
    <changestate module="CommunicationModule"
      transition="destroy" />
    <pause minduration="27000e6" maxduration="30600e6"
      label="sleepTillNextCycle" />
  </program>
</task>

```

### 3.1.3. Scheduler

Alleine mit dem Simulator und einem Scheduling lässt sich allerdings nicht die Güte des Scheduling in Bezug auf einen effizienten Nutzer einer Energiequelle bestimmen. Dies ist erst im Vergleich zu anderen Scheduling möglich, denn der Simulator dient nur dazu, das Ergebnis eines Scheduling in Bezug auf die Betriebszeit und Servicequalität bestimmen zu können. Zur Erzeugung verschiedener Scheduling, d.h. zur Auswahl unterschiedlicher Pausen in den Grenzen der Vorgaben für die Pausen der jeweiligen Verzögerungen für einen Treiberaufruf, ist ein genetischer Algorithmus gewählt worden. Begründet wurde die Wahl eines genetischen Algorithmus auf Grund der Größe des Lösungsraumes für die oben definierte Art des Scheduling. Dieser Lösungsraum bekommt durch jede Pause eine weitere Dimension dazu. Mit Hilfe des genetischen Algorithmus soll sich einem optimalen Scheduling genähert werden. Dabei soll die Dauer der Verzögerung vom Scheduler automatisch bestimmt werden, wobei er den optimalen Zeitpunkt bezüglich des aktuellen Zustands der Energiequelle auswählt. Über verschiedene Fitnessfunktionen ist es möglich, die aus dem Scheduling resultierende Betriebszeit und Servicequalität der Anwendung zu steuern. Das heißt: Mit Hilfe der Fitnessfunktionen ist es möglich, dem Scheduler ein Ziel vorzugeben. Die verschiedenen implementierten Fitnessfunktionen werden im Folgenden vorgestellt:

- **LIFETIME**: Dient zur Maximierung der Betriebszeit.
- **WEIGHTEDTASKCOUNT**: Diese Fitnessfunktion maximiert die gewichtete Summe der Durchläufe.
- **PARETO**: Addiert nach der Bestimmung der Betriebszeit und der gewichteten Summe der einzelnen Taskdurchläufe mit einer Gewichtung  $G$  die beiden zuvor bestimmten Werte auf:  $Fitnesswert = G * lifetime + (1 - G) * taskcount$ .
- **TARGETLIFETIME**: Maximiert die gewichtete Summe der Taskdurchläufe unter Berücksichtigung einer vorgegebenen Betriebszeit.
- **TARGETWEIGHTEDTASKCOUNT**: Maximiert die Betriebszeit unter Beachtung einer vorgegebenen Summe von gewichteten Taskdurchläufen.

Die Wahl der Fitnessfunktion und weitere Einstellungen werden dem offline Scheduler per Textdatei übergeben. Für die Implementierungsdetails, die weiteren Einstellungen und genaue Syntax der Eingabe des offline Scheduler bzw. des genetischen Algorithmus wie z.B. der Größe der Population, der Anzahl der Generationen, die Kreuzungs- und die Mutationsrate wird auf die Diplomarbeit von Michael H. verwiesen [34].

### 3.1.4. Einbindung in Kratos

Damit ein ermitteltes statisches Scheduling in eine Anwendung übernommen werden kann, wurde in der Arbeit von Michael Hesse die Lösung vorgeschlagen, die ermittelten Pausen direkt in der Anwendung an die gewünschten Stellen zu definieren. Hierzu wird, wie auch in den Grundlagen zu Kratos in Abschnitt 2.1.3 beschrieben, die Möglichkeit genutzt, dass sich ein Thread über den Aufruf der Betriebssystemfunktion `Guarded_Buzzer.sleep(int ms)` selbstständig suspendieren lassen kann. Damit die jeweiligen Pausen im Quellcode der Anwendung auch die richtige Pausenlänge erhalten, wurde der Weg gewählt, die Sleep-Funktion mit einem Macro als Parameter aufzurufen `sleep(LABEL)`. Das Label muss dabei mit dem Label der Pausen in der Taskdefinition übereinstimmen. Der offline Scheduler gibt bei erfolgreicher Suche nach einem Scheduling eine Header-Datei aus, in der den Labels die ermittelten Pausenlängen für den Funktionsaufruf zugewiesen werden. Vorteil der Lösung ist, dass es auch für andere Betriebssysteme als Kratos verwendet werden kann und dass der Overhead gering ist.

### 3.1.5. Abgrenzung

Diese Masterarbeit grenzt sich im Allgemeinen von der Diplomarbeit von Michael H. dadurch ab, dass anstelle eines Kondensators bzw. Supercaps eine Batterie als Energiequelle verwendet wird. Dadurch sind es auch andere nichtlineare Effekte, welche beim Scheduling beachtet werden müssen.

Des Weiteren wird auch kein Spannungsregler verwendet, wodurch auch die Art der Verwendung der DFA-Treiber angepasst werden muss. Die Gründe hierzu werden im Verlauf der Arbeit gegeben.

Ferner geht Michael Hesse nur davon aus, dass der Zeitpunkt, zu dem Energie entnommen wird, entscheidend ist. In dieser Arbeit wird zusätzlich zum Zeitpunkt auch der Einfluss der Höhe des Entladestroms untersucht, was z.B. bei der Entscheidung für eine parallele Ausführung von Hardware von Bedeutung ist. Weiterhin geht er davon aus, dass die Nutzung der nichtlinearen Effekte widersprüchliche Anforderungen an das Scheduling stellen kann. Dies ist aber bei den Recovery und Rate Capacity Effekt nicht der Fall.

## 3.2. Ähnliche Arbeiten

An dieser Stelle wird ein Überblick über weitere ähnliche Arbeiten gegeben. Es gibt viele Arbeiten die sich mit dem Konstruktion von batterieeffizienten Systemen beschäftigen. So gibt es Techniken, welche auf der Optimierung des Systems

selbst basieren. Dazu gehören batteriegetriebene Strategien für die Skalierung der Frequenz (Frequency Scaling) und Spannung (Voltage Scaling), zum Power Management und Task Scheduling. Weiterhin gibt es Techniken, Multi-Batteriesysteme, welche Batterie-Scheduling beschreiben. Ferner existieren Verfahren für batterieeffiziente Kommunikation und Routing Protokoll. Des Weiteren gibt es Verfahren, welche auf rekonfigurierbaren Architekturen basieren oder speziell für Sensornetze entwickelt wurden. Auch gibt es Methoden, welche schon beim Hardware Design nichtlineare Effekte beachten, um nur einiges zu nennen. Alle diese Methoden werden im Folgenden genauer beschrieben und auf Gemeinsamkeiten und Unterschiede zu dieser Arbeit eingegangen.

In [58] wird ein Ansatz zum Frequency Scaling vorgestellt. Zusammenfassend verwenden Frequency Scaling Ansätze, Informationen von Batteriemodellen, welche nichtlineare Effekte berücksichtigen, zum Verändern der Taktrate von Systemkomponenten zur Laufzeit. Da die Ansätze auch Workload-Maße (Run- und Idle-Zeiten), Modelle von der Systemleistung und Performance verwenden, können diese Ansätze sicherstellen, dass eine Batterie ohne große Performance-Einbuße effizient genutzt wird. Ansätze zum Skalieren der Spannung werden in [63] und in [49] beschrieben. Der Ansatz in [63] verwendet eine analytische Technik zum Finden der Versorgungsspannung mit dem besten Trade-off zwischen der Batteriekapazität und der Performance. Dabei versucht der Ansatz nicht die Form des Profils des Entladestroms zu verändern, sondern ist es mit der Technik möglich, wenn das Profil des Entladestroms bekannt ist, eine konstante Versorgungsspannung zum Optimieren der Batterielaufzeit zu finden. Hingegen verändert der Ansatz in [49] die Form des Profils des Entladestroms. Er basiert auf der Reallokation von Leerlaufzeiten zum Ermöglichen von Voltage Scaling. Hierzu wird ein Scheduling einer Reihe von globalen und lokalen Transformationen zum Formen des Entladestromprofils unterzogen, mit dem Wissen über den Rate Capacity Effekt und zum Ermöglichen der Reallokation von der Leerlaufzeit. Die Leerlaufzeit, zugehörig zu den verschiedenen Tasks, wird dann zum Voltage Scaling, mit dem Ziel das Entladestromprofil abzufachen, verwendet. Mit diesem Ansatz wird eine Erhöhung der Batterielaufzeit von 76 % erreicht. Beide vorgestellten Ansätze sind statische Ansätze. Eine weitere Methode kommt aus dem Bereich des dynamischen Power Management, genauer gesagt ist es eine batteriegetriebene Power Management Technik auf Systemebene [3]. Die Technik basiert auf der Kontrolle des Operationsstatus des Systems abhängig vom Ladezustand der Batterie. Eine Fallstudie wurde mit einem digitalen Audio-Rekorder gemacht. Dabei nutzt die Technik den Rate Capacity Effekt auf folgende Weise aus: So lange die Spannung der Batterie nicht unter ein bestimmtes Level fällt, arbeitet der Rekorder mit hoher Audioqualität, was mit einem hohen Entladestrom einhergeht. Fällt die Spannung unter das Level, wird die Audioqualität reduziert und somit auch die Höhe des Entladestroms. Wobei der gewählte



Ansatz ein reaktiver Ansatz ist, weil die batteriegetriebene Strategie erst dann zum Zuge kommt, wenn die Spannung eine bestimmte Schwelle unterschritten hat. Die bis hier vorgestellten Arbeiten grenzen sich insofern von dieser Masterarbeit ab, als dass sie nicht durch eine Ablaufplanung, sondern über Voltage und Frequency Scaling oder durch Power Management versuchen, nichtlineare Effekte zu nutzen. Aber auch diese Verfahren verwenden zum Teil Batteriemodelle, die nichtlinearen Effekte berücksichtigen.

Im dem Bereich des Task Scheduling, in welchem sich auch diese Arbeit bewegt, gibt es eine Reihe von Arbeiten. So wurde z.B. in [49] ebenfalls ein Algorithmus zum batteriegefahrenen statischen Scheduling für Echtzeit-Systeme, mit dem Ziel das Entladeprofil zu optimieren, entwickelt. Hierzu wird zuerst ein statisches Schedule erzeugt, welches anschließend in zwei Schritten zum Optimieren des Entladeprofils transformiert wird. Die erste Transformation reduziert Spitzen im Stromverbrauch und erhöht die Flexibilität im Schedule. Der zweite Schritt beinhaltet lokale Transformationen, um eine in [63] vorgestellte Kostenfunktion zu minimieren. Es wurde so eine Erhöhung der Batterielaufzeit von 29 % erreicht. Ähnlich sind die Methoden in [73] und [16]. Auch sie nutzen Informationen aus Batteriemodellen, um ein Scheduling zu erzeugen. Hauptsächlich versuchen die Techniken das Entladeprofil eines Systems so zuzuschneiden, dass dieses der optimalen Entladerate der Batterie entspricht. In [35] wird ein batteriegefahrter Echtzeit-Task-Scheduling Algorithmus vorgestellt für Wireless Sensor Netzwerke. Dieser Algorithmus berücksichtigt den Recovery Effekt und besteht aus drei Phasen und wurde mit dem Ziel entworfen, die Batterielaufzeit zu maximieren sowie alle Zeitschranken der Tasks einzuhalten. Ein Ansatz für Wireless Mesh Netzwerke ist in [50] zu finden. Die Arbeit [45] stellt auch wieder einen Scheduling Algorithmus für ein statisches Scheduling mit Echtzeitanforderungen zur Optimierung der Batterielaufzeit vor. Hierbei werden neue Aspekte des Recovery Effektes berücksichtigt, was zu Verbesserungen der Batterielaufzeit von bis zu 29.41 % führt. Mit dieser Arbeit haben alle diese Ansätze gemeinsam, dass sie unter Berücksichtigung nichtlinearer Effekte beim Scheduling von Tasks versuchen, die Batterielaufzeit zu erhöhen. Dazu verwenden auch die meisten Ansätze Batteriemodelle und sind auch statisch. Der größte Unterschied zu dieser Arbeit ist, dass die vorgestellten Ansätze immer für Echtzeitsysteme entworfen wurden und so werden Möglichkeiten zum Ausnutzen der nichtlinearen Effekte eingeschränkt. Weiterhin berücksichtigen die meisten Ansätze nur den Recovery Effekt und nicht auch den Rate Capacity Effekt. Ferner verwendet keine der Arbeiten einen genetischen Algorithmus zum Finden eines Scheduling und sie beschränken sich auch nicht auf *Deeply Embedded System*, d.h. sie versuchen nicht über die Steuerung der Hardwareaktivitäten der Peripherie die nichtlinearen Effekte auszunutzen.

Eine weitere Möglichkeit, nichtlineare Effekte zu nutzen, besteht bei Systemen,

welche mehrere Batterien verwenden. Dabei wird unter Berücksichtigung der nicht-linearen Effekte unter allen Batterien eine ausgewählt, welche den aktuellen Strombedarf liefern soll. Diese Art von Methodik wird auch als Batterie-Scheduling bezeichnet. Dabei können die Ansätze zum Batterie-Scheduling in drei Klassen eingeteilt werden. So gibt es statisches Batterie-Scheduling [3, 4, 10], welches keine Laufzeitinformationen verwendet. Weiterhin existieren dynamische Ansätze, welche auf Basis der Klemmspannung der Batterie Batterie-Scheduling betreiben [3, 4, 10]. Zuletzt gibt es einen dynamischen Ansatz, der auf der Basis des Entladestroms Batterie-Scheduling durchführt [93]. Zu diesem Kontext passt auch die Arbeit [51]. Hier wird eine intelligente Batterie vorgestellt bzw. ein Array von Batterien, welche die Logik zum effizienten Nutzen einer Batterie von dem System in die Batterie selbst verlagert. Hierbei wird der Status jeder Zelle im Array in Echtzeit verwaltet und unter Beachtung der nicht linearen Effekte eine gerade günstige Batterie im Array für eine angeforderte Last verwendet. Dieser Ansatz verspricht eine Verbesserung von 22 % in Bezug auf die Betriebszeit der Batterien und verspricht weitere Verbesserungen in Kombination mit einer batteriegefahrenen Ablaufplanung. Diese Ansätze unterscheiden sich alleine schon dadurch von dieser Arbeit, dass sie mehrere Batterien benötigen und dass sie nicht ein Scheduling auf Tasksebene betreiben.

Ebenso gibt es Verfahren für batterieeffizientes Routing und Traffic Shaping. Die Idee ist, bei Kommunikationen und Routing-Protokollen nichtlineare Effekte zu beachten. In [11] wird ein Algorithmus zum Traffic Shaping vorgestellt, welcher das Entladen einer Batterie unterbricht, bis die Batterie sich zu einem gewissen Maße erholt hat, falls ein bestimmter Schwellenwert unterschritten wird. Hierzu passt auch die Arbeit [14]. Diese zeigt Wege auf, wie die Energieeffizienz von Kommunikationsprotokollen über die Ausnutzung des Charge Recovery Effektes verbessert werden können. So wird dargelegt, dass die Eigenschaft der stoßweisen Übertragung von Daten vieler Kommunikationsgeräte gut zur Ausnutzung des Recovery Effektes passt. Ein Verfahren für batterieeffizientes Routing wird in [15] vorgestellt. Das batterieeffiziente Routing Protokoll basiert dabei zur Auswahl einer Route auf einer Metrik, welche pro Knoten unter anderem den Zustand der Batterie (SoC) und Strafen durch den Rate Capacity für Knoten, welche eine hohe Sendeleistung brauchen berücksichtigt, um die Route mit einem geringen Einfluss auf die Batteriekapazität der beteiligten Knoten zu ermitteln. Inaktive Knoten haben die Möglichkeit sich zu erholen (Recovery Effekt). Verfahren zum batterieeffizienten Routing und Traffic Shaping sind vom Ansatz her schon grundlegend anders als diese Masterarbeit, da sie sich speziell auf Kommunikation und Routing Protokolle beziehen.

Weiterhin gibt es noch andere Bereiche, in denen nichtlineare Effekte von Batterien berücksichtigt werden, wobei sie sich von dieser Arbeit alleine schon dadurch abgrenzen, dass sie entweder andere Architekturen verwenden oder nicht

auf Scheduling von Tasks beruhen. So etwa in konfigurierbaren Architekturen. Die Arbeit [94] stellt einen Ansatz für Grobgranular rekonfigurierbare Architekturen (CGRA)<sup>1</sup> vor, welche mit Hilfe eines Optimierungsmodells, das nichtlineare Effekte beachtet, die Batteriebetriebszeit verlängert. Auf Basis dieses Modells wird eine Task-Abbildung und ein Batterie-Scheduling geplant, sowie eine Verbesserung der Batterielaufzeit von 26.22 % erzielt. Für FPGAs<sup>2</sup> wird in [44] eine Heuristik, mit deren Hilfe ein batteriebewusstes Scheduling unter Berücksichtigung verschiedener Implementierungen für einen Task im FPGA erzeugt werden.

In [33] wird ein Algorithmus zur Abschätzung der Batterielaufzeit von Lithium-Ionen-Zellen zur Simulationslaufzeit, welcher nichtlineare Effekte berücksichtigt, für die Integration in Netzwerk-Simulationsumgebungen für mobile Systeme vorgestellt.

Weiterhin werden in der Forschungsarbeit [63] nichtlineare Effekte schon bei dem Design von Hardware im Speziellen bei der Bewertung von Low-Power-Design Techniken beachtet bzw. wird es angeregt, dies zu tun. So wird gezeigt, dass die maximale Batterielaufzeit dann erreicht wird, wenn die Varianz der Verteilung des Entladestroms minimiert ist.

Weitere Arbeiten untersuchen den Einfluss und den Nutzen von nichtlinearen Effekten bei Batterien im Kontext von Wireless Sensor Netzwerken [62, 9]. In diesen Kontext passt auch die Arbeit [18], welche ein Framework zum batteriebewussten Sensor Management vorstellt.

Ebenso wird in [61] ein Framework zum Schätzen der Batterielaufzeit unter Berücksichtigung nichtlinearer Effekte für eingebettete Systeme vorgestellt.

Es gibt noch viele weitere verwandte Arbeiten, deren Vorstellung aber den Rahmen dieser Arbeit sprengen würde.

---

<sup>1</sup>Coarse-Grained Reconfigurable Architecture

<sup>2</sup>Field Programmable Gate Arrays



## 4. Batteriemodellierung

In diesem Kapitel geht es darum, ein geeignetes Batteriemodell zur Modellierung einer Lithium-Dünnschichtbatterie für die Ablaufplanung zu finden. Denn der offline Scheduler benötigt bei der Suche nach einer batteriegefahrenen Ablaufplanung logischerweise Kenntnisse über die verwendete Batterie bzw. über deren Zustand. Diese Kenntnisse werden dem Scheduler über ein Batteriemodell zu Verfügung gestellt. Dabei sind Batteriemodelle in der Lage, verschiedene Eigenschaften von realen Batterien abzubilden und werden dazu verwendet, Vorhersagen über die realen Batterien zu tätigen. So ist es, wie in den Zielen dieser Arbeit beschrieben, eines der Kernziele, ein Batteriemodell zu finden und ggf. anzupassen, welches in den offline Scheduler integriert werden kann. Da es aber den Rahmen dieser Arbeit sprengen würde, ein eigenes Modell zu entwickeln, wurde sich darauf beschränkt, ein Modell aus der Literatur zu verwenden. Dort finden sich eine Fülle von Batteriemodellen bzw. Batteriemodellklassen, welche für die unterschiedlichsten Anwendungen entwickelt worden sind. Daher ist es nötig, sich erst einmal einen Überblick über die gängigsten Modelle zu verschaffen. Bevor aber ein geeignetes Batteriemodell identifiziert werden kann, ist es aber zuvor erforderlich, sich über die Anforderungen, die das Modell der Batterie erfüllen muss, sicher zu sein. So beginnt das Kapitel mit der Vorstellung der Anforderungen an das benötigte Batteriemodell, damit es für die Bestimmung einer batteriegefahrenen Ablaufplanung mit Hilfe des offline Schedulers geeignet ist. Anschließend wird der besagte Überblick über die Batteriemodelle mit ihren jeweiligen Vor- und Nachteilen gegeben. Zum Abschluss des Kapitels wird die Entscheidung, für das in dieser Arbeit eingesetzte Modell begründet.

### 4.1. Anforderung an ein Batteriemodell

Wie schon eingangs erwähnt, wird für diese Arbeit ein Batteriemodell zur Modellierung von Lithium-Dünnschichtbatterien für die Bestimmung einer batteriegefahrenen Ablaufplanung unter Verwendung eines offline Schedulers benötigt. Damit ein Batteriemodell zu dieser Aufgabe passt, muss dieses eine Reihe von Anforderungen erfüllen. Im Allgemeinen muss es vorhersagen können, wann eine Batterie bei einem gegebenen Entladeprofil, unter Berücksichtigung der nichtlinearen Effekte,

erschöpft ist. Im Speziellen ist eine passende Schnittstelle nötig, damit das Modell in dem offline Scheduler verwendet werden kann. Details dazu werden in Kapitel 6 vorgestellt. In diesem Zusammenhang muss noch erwähnt werden, dass ein Batteriemodell, welches nur den Entladevorgang einer Batterie simuliert und nicht auch den Ladevorgang, ausreichend für diese Arbeit ist, da zur Ermittlung einer batterie-gewahren Ablaufplanung nur Prognosen über die Entladung einer Batterie nötig sind. Dabei muss das Modell zu jedem Zeitpunkt die aktuelle Klemmspannung bzw. verbleibende Ladung einer Batterie zur Verfügung stellen bzw. voraussagen können. Weiterhin ist es wichtig, dass das Batteriemodell den passenden Typ von Laststrom unterstützt. Hier muss das Modell nicht nur einen konstanten, sondern auch einen variablen Laststrom erlauben, weil in den meisten realen Systemen nicht immer konstant eine Last anliegt. So hat z.B. eine Funkübertragung andere Lasten als ein Sensor. Des Weiteren muss es auch möglich sein, beschreiben zu können, wie lange eine Last anliegt, damit es möglich ist, ein typisches Entladeprofil eines Deeply Embedded Systems darstellen zu können.

Entscheidend ist auch, dass das Modell die beiden nichtlinearen Effekte (Recovery und Rate Capacity Effekt), welche in dieser Arbeit berücksichtigt werden, abdeckt. Um die Komplexität des Modells in diesem Zusammenhang nicht unnötig zu erhöhen, ist es wichtig, dass nicht noch weitere Effekte wie Temperatureffekte berücksichtigt werden. Vor allem aber muss das Modell über eine ausreichende Genauigkeit in der Vorhersage der realen Batterie verfügen. In diesem Zusammenhang ist es also entscheidend, dass das Modell überhaupt zu der in dieser Arbeit exemplarisch als Lithium-Dünnschichtbatterie eingesetzten, EnFilm-Batterie passt. So ist es von Bedeutung, wie die beiden nichtlinearen Effekte im Modell im Verhältnis zur realen Batterie modelliert sind. Eine weitere Anforderung ist, dass der Aufwand der Berechnung für die Simulation nicht zu hoch ist, d.h. nicht mehrere Stunden oder sogar Tage dauern kann. Diese Zeiten wären unpraktikabel, zumal auch der eingesetzte genetische Algorithmus (je nach Einstellung für z.B. der Anzahl von Generationen oder die Populationsgröße) an sich schon zu sehr hohen Laufzeiten führen kann. Ferner sollte auch der Konfigurationsaufwand nicht zu groß sein. So sollte z.B. die Anzahl der Parameter, welche zur Verwendung des Modells von der realen Batterie benötigt werden, überschaubar sein. In diesem Zusammenhang ist es auch entscheidend, dass die Werte dieser Parameter leicht zu beschaffen sind und nicht tiefgehendes Wissen über die Batterie nötig ist, welche unter Umständen nur der Hersteller hat.

Am Rande besteht noch die Anforderung, dass das Batteriemodell möglichst intuitiv ist, wodurch es einfacher wird, das Modell zu verstehen und somit auch die Abläufe in einer Batterie. Dies wird bei der späteren Evaluation dieser Arbeit hilfreich sein.

## 4.2. Überblick über Batteriemodelle

Für eine konstante Last und eine ideale Batterie kann die Batterielaufzeit  $T$  mittels Division der Ladung  $Q$  durch den Entladestrom berechnet werden:

$$T = \frac{Q}{I}. \quad (4.1)$$

Wie schon in den Grundlagen beschrieben, gilt diese Formel (Modell) aufgrund der nichtlinearen Effekte nicht bei realen Batterien. Aus diesem Grund wurden eine Vielzahl von Batteriemodellen entwickelt, welche unter Berücksichtigung nichtlinearer Effekte versuchen, die Realität genauer abzubilden. Nachfolgend wird ein Überblick über die gängigen Modelle in der Literatur bzw. Forschung gegeben. Dieser Überblick dient dazu, potenziell mögliche Modelle das für diese Arbeit benötigte Batteriemodell vorzustellen und die Entscheidung für das verwendete Batteriemodell zu begründen. Dabei lassen sich die Modelle in analytische, stochastische und elektrochemische Modelle und elektrische Schaltungsmodelle einteilen.

### 4.2.1. Elektrochemische Modelle

Die Modelle, die zur Klasse der elektrochemischen Modellen gehören, basieren auf den elektrochemischen, thermodynamischen Prozessen und der physikalischen Konstruktion einer Batterie. Die Modelle aus diesen Klassen gehören zu den akkuratesten Batteriemodellen, weil diese Modelle Batterien sehr ausführlich beschreiben.

Das wohl bekannteste Batteriemodell aus diesem Bereich ist das von Doyle, Fuller und Newman entwickelte elektrochemische Modell für Lithium und Lithium-Ionen Zellen [20, 25, 24], darum wird es hier exemplarisch vorgestellt.

Das Modell besteht dabei aus sechs verbundenen bzw. gekoppelten nichtlinearen Differentialgleichungen. Die Lösung dieser Gleichungen gibt die Spannung und den Strom als Funktionen über die Zeit an. Weiterhin gibt die Lösung die Potentiale in dem Elektrolyt und der Elektrode-Phasen, die Salzkonzentration, die Reaktionsrate und die Stromdichte in dem Elektrolyt als eine Funktion über die Zeit und Position in der Zelle an.

Zu dem beschriebenen Batteriemodell gibt es ein frei verfügbares Fortran-Programm namens *Dualfoil* [60], welches dieses Modell verwendet um Lithium-Ionen-Batterien zu simulieren. Dabei berechnet dieses Programm für ein vorgegebenes Lastprofil (eine Sequenz von konstanten Stromschritten) wie sich alle genannten Batterieeigenschaften über die Zeit verändern. Mit Hilfe der Ausgabedaten ist es möglich, die Batterielaufzeit zu erhalten. Dazu wird einfach der Zeitpunkt ausgelesen, zu dem die Zellspannung unter die Cut-Off-Spannung fällt [42].

Um das Modell zu konfigurieren ist es neben der Eingabe des Lastprofils nötig, über 50 batterieabhängige Parameter zu setzen, wobei viele Werte der Parameter, wenn überhaupt nur, durch den Hersteller der Batterie zu bekommen sind. So ist es z.B. nötig, die Dicke der Elektroden und die initiale Salzkonzentration des Elektrolyts anzugeben [42]. Weiterhin sind komplexe, numerische Techniken nötig, um die Gleichung dieses Modells zu lösen, was dazu führt, dass die Berechnung für ein Lastprofil mehrere Stunden bis Tage dauern kann [72].

Dennoch ist die Genauigkeit dieses Modells sehr hoch. Deshalb wird Dualfoil oft dazu verwendet, andere Batteriemodelle zu bewerten, ohne eine reale Batterie als Vergleich zu benutzen [42].

Das Modell von Doyle, Fuller und Newman ist aber nicht das einzige Modell dieser Klasse, so gibt es noch weitere Modelle [77, 21, 89, 28, 29]. Diese unterscheiden sich größtenteils dadurch, dass sie für andere Batterietypen entwickelt wurden.

Zusammenfassend gesagt, können elektrochemische Modelle wegen ihres hohen Detailgrades ein sehr genaues Abbild der Realität schaffen, was aber meist mit einem hohen Konfigurationsaufwand und sehr langen Berechnungszeiten einhergeht. Häufig werden diese Modelle deshalb im Bereich der Batterieentwicklung verwendet. Des Weiteren sind die Modelle in der Lage, viele Entladeeffekte unter variabler Last zu berücksichtigen. Hierzu zählen der Rate Capacity und Recovery Effekt sowie auch thermische Effekte.

### 4.2.2. Elektrische Schaltungsmodelle

Diese Klasse von Modellen benutzt eine äquivalente elektrische Schaltung zum Modellieren der Batterieentladung. Die erste Anwendung für ein Schaltungsmodell zum Modellieren einer Batterie wurde von Hageman für die Simulation von Nickel-Cadmium, Alkali Batterien und Bleiakkumulatoren vorgestellt [31]. Zur Simulation wurden einfache PSpice<sup>1</sup> Schaltungen verwendet, wobei der Kern aller Modelle für die verschiedenen Typen derselbe ist. Der Kern besteht hierzu aus folgenden Teilen (siehe auch Abbildung 4.1) [42]:

- Ein Kondensator repräsentiert die Kapazität der Batterie,
- ein Entladeratennormalisierer bestimmt den Kapazitätsverlust bei hohen Entladeströmen,
- eine Schaltung zum Entladen der Kapazität der Batterie,
- eine Spannung vs. State of Charge (SoC) Lookup-Tabelle,

---

<sup>1</sup>Personal Simulation Program with Integrated Circuit Emphasis



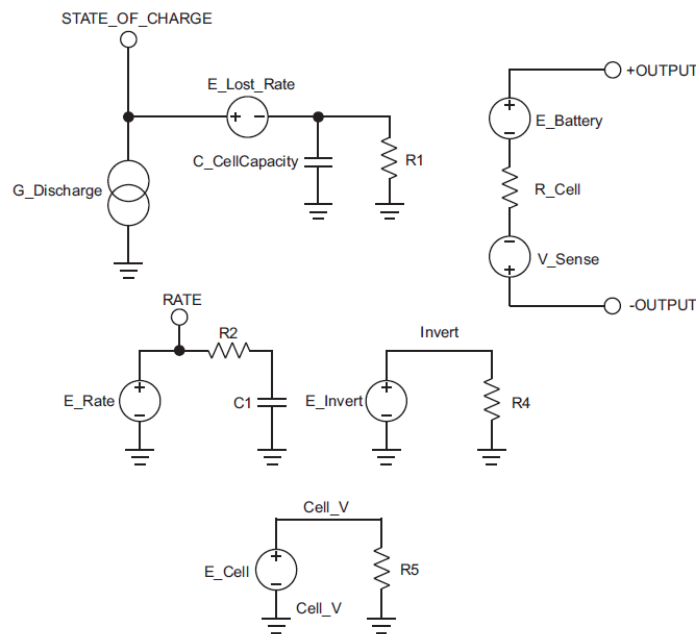


Abbildung 4.1.: Basisschaltungen zum Modellieren einer beliebigen Zelle

- ein Widerstand, welchen den Widerstand der Batterie repräsentiert.

Zum Anpassen an einem spezifischem Batterietyp sind nur wenige Änderungen nötig.

Auch in dieser Klasse gibt es eine Menge von weiteren Modellen. So verwendet [26] ähnliche Techniken zum Modellieren von Ni-MH<sup>2</sup> Batterien. Des Weiteren wird in [27] ein Ansatz zum Modellieren von Lithium-Ionen-Batterien beschrieben. In [2] wird ein zeitdiskreter Ansatz für Lithium-Ionen-Batterien auf Basis der Approximation des zeitkontinuierlichen Schaltungsmodells aus [27] vorgestellt. Ferner wird in [5] ein Ansatz für ein elektrisches Schaltungsmodell einer Nickel-Cadmium-Batterie durch Gruppierung der Gleichungen beschrieben, welche die Prozesse in der Batterie beschreiben.

Elektrische Schaltungsmodelle unterstützen eine variable Last. Weiter modellieren sie thermische Effekte und den Rate Capacity Effekt, wobei keiner dieser Modelle den Recovery Effekt abbilden kann. Vorteil dieser Modelle ist, dass sie in existierenden Schaltungssimulatoren einfach simuliert werden können. Ferner eignen sie sich auch gut dazu, ein komplettes System in zeitdiskreten oder zeitkontinuierlichen Umgebungen zu simulieren, falls Modelle für die anderen Systemkom-

<sup>2</sup>Nickel-Metallhydrid

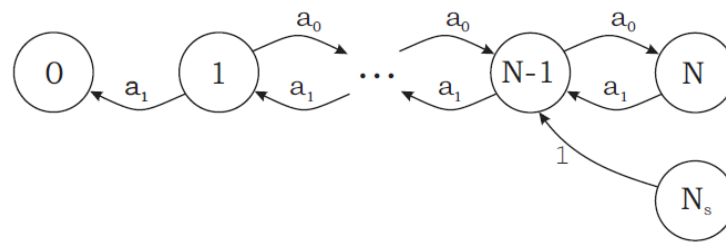


Abbildung 4.2.: Das Basis Markow-Ketten-Batteriemodell von Chiasserini und Rao [14].

ponenten vorhanden sind. Diese Art von Batteriemodellen ist zwar einfacher als die elektrochemischen Modelle und daher auch weniger berechnungsintensiv, aber sie haben auch einen hohen Konfigurationsaufwand. Zwar ist die Anzahl der Parameter in den Modellen nicht so hoch, aber die in einigen Modellen verwendeten Look-Up-Tabellen erfordern einen hohen Aufwand zum Konfigurieren. Weiterhin sind diese Modelle mit einem Fehler ca. 10 % auch weniger akkurat.

### 4.2.3. Stochastische Modelle

Stochastische Modelle beschreiben die Entladung und den Recovery Effekt als stochastische Prozesse und ebenso die Batterie, ähnlich wie die analytischen Modelle auf so einem abstrakteren Level.

#### 4.2.3.1. Chiasserini und Rao

Die ersten stochastischen Modelle wurden zwischen 1999 und 2001 von Chiasserini und Rao in einer Reihe von Arbeiten veröffentlicht [14, 12, 11, 10]. Diese Modelle wurden auf der Basis von zeitdiskreten Markow-Ketten entwickelt. So wurden in [14] zwei Modelle für eine Batterie in einem mobilen Kommunikationsgerät zur Übertragung von Paketen vorgestellt. Das erste Modell ist in der Abbildung 4.2 zu sehen.

Es beschreibt eine Batterie über eine zeitdiskrete Markow-Kette mit  $N + 1$  Zuständen, dabei entsprechen die Nummern der Zustände der Anzahl an Ladungseinheiten, die in der Batterie verfügbar sind. Weiterhin entspricht eine Ladungseinheit der Menge an Energie, welche nötig ist, ein Paket zu übertragen, und  $N$  ist die Anzahl an Ladungseinheiten, welche basierend auf der kontinuierlichen Nutzung direkt verfügbar sind. Des Weiteren wird in diesem Modell bei der Ausführung in jedem Zeitschritt entweder mit Wahrscheinlichkeit  $a_1 = q$  eine Ladungseinheit

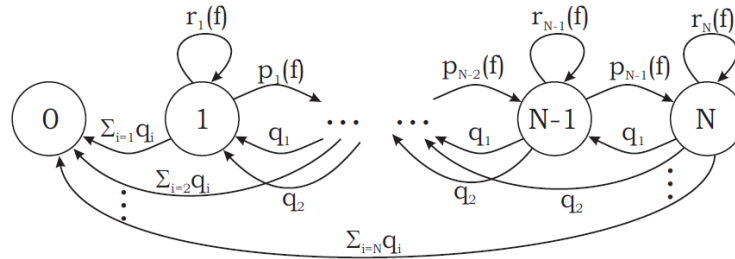


Abbildung 4.3.: Das erweiterte Markow-Ketten Batteriemodell von Chiasserini und Rao [10].

verbraucht oder mit Wahrscheinlichkeit  $a_0 = 1 - q$  eine Ladungseinheit wiederhergestellt (Recovery). Sodann gilt, dass die Batterie leer ist, wenn der Zustand  $0$  erreicht ist oder wenn das Maximum von  $T$  Ladungseinheiten verbraucht wurde, dabei entspricht  $T$  der theoretischen Kapazität der Batterie ( $T > N$ ). Dieses Modell ist zu vereinfacht, da es z.B. nicht berücksichtigen kann, dass sich der Entladestrom in den meisten Systemen verändert. Das zweite Modell versucht dies zu verbessern [42].

Mit dem zweiten Modell aus [14] wurde das erste Modell um zwei neue Aspekte erweitert. Zum einen kann in jedem Zeitschritt mehr als eine Ladungseinheit verbraucht werden und zum anderen wurde eine nicht Null Wahrscheinlichkeit eingeführt im selben Zustand zu bleiben, was bedeutet, dass nichts verbraucht wird und kein Recovery zu dem Zeitpunkt stattfindet. Der erste Punkt wurde eingebaut, um einen stoßweisen Verbrauch von Energie darstellen zu können [42].

Auch das zweite Modell berücksichtigt nicht, wie das erste, dass die Rate des Recovery während der Entladung nicht konstant ist.

Mit den Modellen in [12, 11, 10] wurden weitere Erweiterungen eingeführt. So ist die Wahrscheinlichkeit zum Recovery abhängig vom Zustand gemacht worden. Dadurch wurde folgendes Verhalten eingebracht: Ist weniger Ladung verfügbar, ist die Wahrscheinlichkeit zum Wiederherstellen einer Ladungseinheit kleiner. Ferner wurde eine Phasenabhängigkeit eingeführt. Die Phasennummer ( $f$ ) ist dabei eine Funktion über die Anzahl an Ladungseinheiten, welche konsumiert wurden, und es gilt: Wenn mehr Ladungseinheiten verbraucht wurden, steigt die Phasennummer und das führt dazu, dass sich die Wahrscheinlichkeit zum Recovery verringert [42]. Das Zustandsdiagramm des Modells mit allen Erweiterungen ist in Abbildung 4.3 zu sehen, dabei steht  $q_i$  für die Wahrscheinlichkeit, dass  $i$  Ladungseinheiten in einem Zeitfenster angefordert werden.

Ferner kann die Batterie in Ruhephasen entweder eine Ladungseinheit wieder-

herstellen, oder sie bleibt mit Wahrscheinlichkeit  $r_i(f)$  im selben Zustand, wobei sich die Wahrscheinlichkeit zum Recovery im Zustand  $j$  und Phase  $f$  wie folgt berechnet [42]:

$$q_j(f) = q_0 e^{(N-j)g_N - g_C(f)}, \quad (4.2)$$

wobei  $g_N$  und  $g_C(f)$  abhängig vom Recovery-Verhalten der Batterie sind. Auch dieses Modell wurde in [10] mit dem Dualfoil-Simulationsprogramm verglichen, mit den Ergebnissen dass es durchschnittlich 1% und maximal 4% abweicht, bei einer gepulsten Entladung.

Nachteil dieser Modelle sind, ein festes Lastprofil zu modellieren, da es nicht möglich ist, die Reihenfolge in welcher die Transitionen genommen werden, zu steuern. Außerdem berücksichtigt das Modell auch nur den Recovery Effekt und nicht auch den Rate Capacity Effekt. Um den Rate Capacity Effekt zu berücksichtigen haben, Debashis Panigrahi et al. in [61] eine Look-up-Tabelle zum Modell hinzugefügt. Dies führte zu einem Modell, das schnell ist und ähnlich gute Vorhersagen wie das Dualfoil-Simulationsprogramm erreicht. Aber das Problem, dass es nicht möglich ist, ein festes Lastprofil zu definieren, besteht weiterhin.

#### 4.2.3.2. Stochastisch modifiziertes KiBaM

Auch auf Basis des Kinetic Battery Modells (s. Abschnitt 4.2.4.3) gibt es ein stochastisches Batteriemodell, welches hier zur Vollständigkeit vorgestellt wird. Es wurde 2005 von Rao et al. in [74] vorgeschlagen. Da das stochastische *KiBaM* dazu entwickelt wurde, Ni-MH Batterien zu modellieren und nicht für Bleiakkumulatoren wie das originale *KiBaM* (s. Abschnitt 4.2.4), sind laut [74] einige Veränderungen am Modell nötig gewesen. Dazu wurde in dem Term, welcher den Fluss der Ladung vom Behälter mit der gebundenen Ladung zum Behälter mit der frei verfügbaren Ladung entspricht, ein weiterer Faktor  $h_2$  hinzugefügt und verändert zu (s. Formel 4.14) [42]:

$$\begin{cases} \frac{dy_1}{dt} = -I + k_s h_2 (h_2 - h_1), \\ \frac{dy_2}{dt} = -k_s h_2 (h_2 - h_1). \end{cases} \quad (4.3)$$

Dieser Faktor sorgt dafür, dass der Recovery Effekt langsamer ist, wenn weniger Ladung in der Batterie übrig ist. Des Weiteren wurde im stochastischen Teil des Modells die Wahrscheinlichkeit dafür hinzugefügt, dass kein Recovery während der Ruhephasen eintritt. Insgesamt wird die Batterie durch einen zeitdiskreten transienten Markow-Prozess beschrieben. Pro Zustand in der Markow-Kette gibt es drei Parameter  $(i, j, t)$ , wobei  $i$  und  $j$  die diskretisierten Level des Behälters mit der frei verfügbaren Ladung bzw. des Behälters mit der gebundenen Ladung

sind. Weiterhin ist  $t$  die Länge der aktuellen Ruhephase, d.h. es ist die Anzahl an Zeitschritten, seitdem das letzte Mal Ladung aus der Batterie entnommene wurde [42]. Die Transitionen des Prozesses lassen sich dabei wie folgt zusammenfassen [42]:

$$(i, j, t) \rightarrow \begin{cases} (i + Q, j - Q, t + 1) \\ (i, j, t + 1) \\ (i - I + J, j - J, 0), \end{cases} \quad (4.4)$$

dabei entsprechen die ersten beiden Transitionen Zeitpunkte, in denen der Strom Null ist. So stellt die Batterie mit Wahrscheinlichkeit  $p_r$   $Q$  Ladungseinheiten wieder her und mit Wahrscheinlichkeit  $p_{nr}$  findet kein Recovery statt. Die beiden Wahrscheinlichkeiten hängen dabei von  $t$  ab. Die letzte Transition entspricht den Zeitpunkten, zu denen Strom aus der Batterie gezogen wird. Hier werden mit Wahrscheinlichkeit  $q_I$ ,  $I$  Ladungseinheiten aus dem Behälter mit der frei verfügbaren Ladung genommen und  $J$  Ladungseinheiten vom Behälter mit der gebundene Ladung zum Behälter mit der frei verfügbaren Ladung übertragen [42].

Auch hier besteht wieder das Problem, dass das Lastprofil nicht gesteuert werden kann. Zwar werden die  $q_I$  durch das Lastprofil definiert, weil aber  $q_I$  für alle Zustände gleich ist und so ist es nicht möglich, zu steuern in welcher Reihenfolge der Strom aus der Batterie gezogen wird. Vorteil ist, dass das Modell aufgrund der Basis des *KiBaMs* sowohl den Recovery Effekt als auch den Rate Capacity Effekt abdeckt.

Ferner wird in [74] das Modell mit Experimenten an einer Ni-MH Batterie verglichen, und es wurde gezeigt, dass das Modell die Batterielaufzeit und die entnommene Ladung der Batterie mit einem maximalen Fehler von 2.65 % vorhersagen kann.

Generell liegt das Problem bei den stochastischen Modellen aufgrund der nicht-deterministischen Lastprofile darin, dass ein Lastprofil nicht gesteuert werden kann. Weiterhin werden auch nicht immer beide nichtlinearen Effekte in den Modellen beachtet. Auch unterstützen nicht alle Modelle eine variable Last. Ferner ist der Aufwand für die Berechnung der Modelle meist mittelmäßig im Vergleich zu anderen Modellen bzw. können die Modelle (Zustandsmenge) sehr groß werden. Hierzu sind dann z.B. numerische Verfahren nötig um die Modelle zu analysieren. Auch nicht zu vernachlässigen ist der Konfigurationsaufwand z.B. für die Look-up-Tabelle der Variante aus [61].

#### 4.2.4. Analytische Modelle

Die analytischen Modelle beschreiben eine Batterie auf einem höheren Abstraktionslevel als die bisher vorgestellten Modelle. Dazu verwenden sie meist wenige analytische Ausdrücke, was diese Batteriemodelle auch einfacher macht. Im Folgenden werden die prominentesten Vertreter dieser Klasse vorgestellt.

##### 4.2.4.1. Peukert's Law

Peukert's Law ist das einfachste Modell zur Vorhersage der Batterielaufzeit, welches nichtlineare Effekte berücksichtigt [70]. Dieses Modell modelliert den nichtlinearen Zusammenhang zwischen der Batterielaufzeit und der Rate der Entladung der Batterie. Also wird nur der Rate Capacity Effekt und nicht auch der Recovery Effekt beachtet. So lässt sich mit Hilfe von Peukert's Law die Batterielaufzeit ( $L$ ) wie folgt berechnen [42]:

$$L = \frac{a}{I^b}, \quad (4.5)$$

dabei ist  $I$  der Entladestrom, und  $a$  und  $b$  sind Konstanten. Die Werte der Konstanten sind aus Experimenten zu beziehen. Die Konstante  $a$  spiegelt elektrochemische Eigenschaften, die physikalische Konstruktion und die Betriebsumgebung der Batterie wieder. Daneben spiegelt die Konstante  $b$  den Rate Capacity Effekt wieder. Im Idealfall ist  $a$  gleich der Kapazität der Batterie und  $b$  ist gleich eins. In der Realität hat  $a$  aber einen Wert nahe der Kapazität der Batterie und  $b$  ist größer als eins [42]. Laut [58] liegt der Wert von  $b$  zwischen 1.2 und 1.7. Das Problem dieses Modells ist, dass es zwar für durchgehend konstanten Strom eine gute Vorhersage liefert, aber nicht für die variable bzw. unterbrochene Last [42]. Als Lösung hierzu geben Rakhmatov and Vrudhula in [70] eine erweiterte Version der Formel 4.5 an. Dazu wird  $I$  aus Formel 4.5 ersetzt, durch den durchschnittlichen Strom bis zu  $t = L$ . Dann gilt für ein stückweises konstantes Entladeprofil, mit  $t_k$  als die Punkte in der Zeit, wo sich der Strom ändert (s. Abbildung 4.4) [42]:

$$L = \frac{a}{\left[ \frac{\sum_{k=1}^n I_k(t_k - t_{k-1})}{L} \right]^b}. \quad (4.6)$$

Problem dieser Version ist, dass diese zu einfach ist, weil auch sie den Recovery Effekt nicht beachtet. Dies bestätigten auch die Untersuchungen der Autoren Rakhmatov and Vrudhula in [70, 69, 68], wo sie die erweiterte Peukert's Formel mit dem Dualfoil-Simulationsprogramm verglichen haben. Die Ergebnisse des Dualfoil-Simulationsprogramms dienten dabei als Referenzwerte. Hier zeigten die Ergebnisse des Vergleiches für eine konstant kontinuierliche Last, dass die erweiterte Peukert's

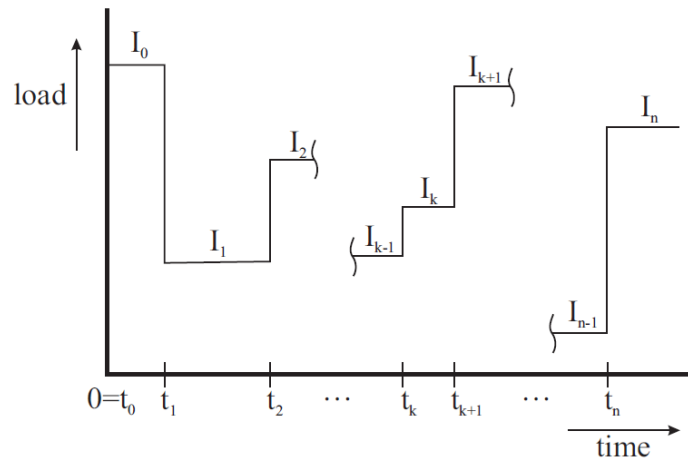


Abbildung 4.4.: Beispiel eines stückweise konstanten Entladeprofils [42].

Formel einen durchschnittlichen Fehler von 14 % und maximal einen Fehler von 43 % hat. Es zeigte sich, dass die Formel für kleine Lasten gut arbeitet, aber nicht für große Lasten. Für variable und unterbrochene Belastungen ist die Formel noch viel schlechter, was auf die fehlende Beachtung des Recovery Effekts zurückzuführen ist.

#### 4.2.4.2. Rakhmatov and Vrudhula's Diffusionsmodell

Ein weiteres analytisches Modell wurde von Rakhmatov und Vrudhula, welches auf der Diffusion der Ionen in dem Elektrolyt basiert, entwickelt [70, 69, 68]. Das Modell beschreibt die Konzentrationsentwicklung des aktiven Materials im Elektrolyt, um die Batterielaufzeit für eine gegebene Last vorherzusagen. Dabei wird im Modell angenommen, dass die Prozesse an beiden Elektroden gleich sind. Daher ist die Batterie symmetrisch in Bezug auf die Elektroden, und wovon nur eine Elektrode betrachtet wird[41].

Abbildung 4.5 zeigt zur Veranschaulichung eine vereinfachte Sicht auf die Batterieoperationen nach dem Diffusionsmodell. In Abbildung 4.5a ist für eine vollgeladene Batterie zu sehen, dass die Konzentration aktiven Materials konstant über die volle Breite ( $w$ ) des Elektrolyts liegt. Wird aber die Batterie belastet, resultieren die elektrochemischen Reaktionen in einer Reduktion der Konzentration vom aktivem Material nahe der Elektrode (s. Abbildung 4.5b). Dabei entsteht ein Gefälle durch das Elektrolyt, welches dazu führt, dass das aktive Material sich hin zur Elektrode ausbreitet. Wird jetzt die Last von der Batterie entfernt, erholt sich die

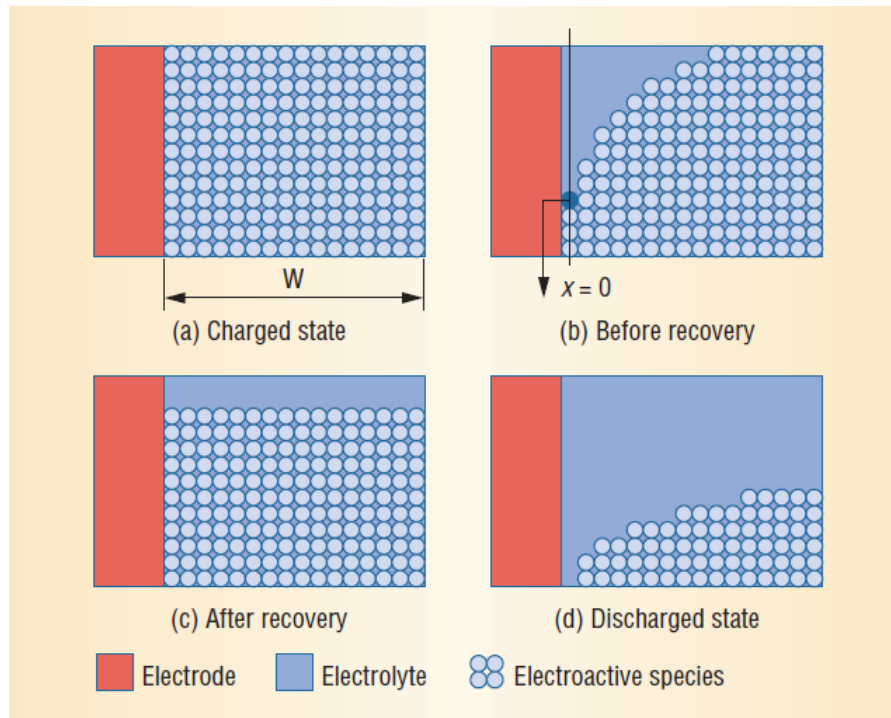


Abbildung 4.5.: Physikalisches Bild des Modells nach Rakhmatov und Vrudhula [72].

Konzentration des aktiven Materials an der Elektrode aufgrund der Diffusion. Es gilt aber, dass die Konzentration nicht mehr so hoch sein kann wie die der vollen Batterie (s. Abbildung 4.5c). Fällt die Konzentration an der Elektrode unter eine gewisse Schwelle, können die chemischen Reaktionen nicht mehr aufrecht erhalten werden und die Batterie gilt als leer (s. Abbildung 4.5d).

Die Konzentration des aktiven Materials zum Zeitpunkt  $t$  und der Abstand  $x \in [0, w]$  von der Elektrode ist bezeichnet mit  $C(x, t)$ , und für eine volle Batterie ist die Konzentration konstant über die Länge des Elektrolyts:  $C(x, 0) = C^*, x \in [0, w]$ . Die Batterie wird als leer betrachtet, wenn  $C(0, t)$  unter die Schwelle  $C_{cutoff}$  fällt [41]. In dem Modell wird die Entwicklung der Konzentration mit Fick's Gesetz beschrieben [70]:

$$\begin{cases} -J(x, t) = D \frac{\partial C(x, t)}{\partial x}, \\ \frac{\partial C(x, t)}{\partial t} = D \frac{\partial^2 C(x, t)}{\partial x^2}, \end{cases} \quad (4.7)$$

mit  $j(x, t)$  für den Fluss des aktiven Materials zum Zeitpunkt  $t$  und Abstand  $x$  von der Elektrode, und  $D$  für Diffusionskonstante. Dabei ist der Fluss an der



Elektrodenoberfläche ( $x = 0$ ) proportional zum Strom  $[i(t)]$  und der Fluss auf der anderen Seite des Diffusionsbereichs ( $x = w$ ) gleich null [41]. Dies führt zu den folgenden Grenzbedingungen [41]:

$$\begin{cases} D \frac{\partial C(x,t)}{\partial x} \Big|_{x=0} = \frac{i(t)}{\nu F A}, \\ D \frac{\partial C(x,t)}{\partial x} \Big|_{x=w} = 0, \end{cases} \quad (4.8)$$

mit  $A$  für die Elektrodenoberfläche,  $F$  für die Faraday-Konstante und  $\nu$  für die Anzahl an Elektronen, welche in den elektrochemischen Reaktionen an der Elektrodenoberfläche beteiligt sind.

Mittels Laplace Transformation ist es möglich, eine analytische Lösung aus den Differentialgleichungen 4.7 und Grenzbedingungen 4.8 zu erhalten. Aus dieser Lösung ist es möglich, folgenden Ausdruck, welcher sich auf die Last, die Batterie-laufzeit ( $L$ ) und die Batterieparameter bezieht, zu erhalten [42]:

$$\alpha = \int_0^L \frac{i(t)}{\sqrt{L-\tau}} d\tau + 2 \sum_{m=1}^{\infty} \int_0^L \frac{i(t)}{\sqrt{L-\tau}} e^{-\frac{\beta^2 m^2}{L-\tau}} d\tau, \quad (4.9)$$

mit  $\alpha = \nu F A \sqrt{\pi D C^*} \rho(L)$ ,  $\beta = \frac{w}{\sqrt{D}}$  und  $C^*$  für die Konzentration zu  $t = 0$ , und  $\rho(l) = 1 - \frac{C(0,L)}{C^*}$ . Für den Spezialfall eines konstanten Entladestroms ( $I$ ) kann der Ausdruck für  $\alpha$  vereinfacht werden zu [42]:

$$\alpha = 2I\sqrt{L} + 2 \sum_{m=1}^{\infty} \left( \sqrt{L} e^{-\frac{\beta^2 m^2}{L}} - \beta m \sqrt{\pi} \phi \left( \frac{\beta m}{\sqrt{L}} \right) \right), \quad (4.10)$$

mit  $\phi(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$ . Eine gute Approximation für  $\alpha$  ist durch die ersten zehn Terme der unendlichen Summe gegeben, und zusammen mit einer Approximation für  $\phi$  gilt [42]:

$$\alpha = 2I\sqrt{L} \left( 1 + 2 \sum_{m=1}^{10} \left( e^{-\frac{\beta^2 m^2}{L}} - \frac{\pi e^{-\frac{\beta^2 m^2}{L}}}{\pi - 1 + \sqrt{1 + \pi \frac{L}{\beta^2 m^2}}} \right) \right). \quad (4.11)$$

Um mit Hilfe von 4.11 die Batterieauslaufzeit für einen gegebenen Entladestrom zu berechnen, ist es nötig, die Werte für  $\alpha$  und  $\beta$  aus experimentellen Daten zu schätzen. Weiterhin lässt sich aus der Lösung der Laplace Transformation ein Ausdruck finden für den Ladungsverlust der Batterie [71]:

$$\sigma(t) = \underbrace{\int_0^t i(\tau) d\tau}_{l(t)} + \underbrace{\int_0^t i(\tau) \left( 2 \sum_{m=1}^{\infty} e^{-\beta^2 m^2 (t-\tau)} \right) d\tau}_{u(t)}, \quad (4.12)$$

mit  $\beta = \pi\sqrt{d}/w$ . Dabei kann der Ladungsverlust in zwei Teile zerlegt werden:  $[l(t)]$  der Ladungsverlust bezüglich der Last an der Batterie und die unverfügbare Ladung  $[u(t)]$ . Die zweite Ladung ist die Ladung, welche ungenutzt in der Batterie bleibt. Weiterhin ist die Batterie leer, wenn der Ladungsverlust (Formel 4.12) gleich der Kapazität der Batterie ist [41].

Rakhmatov und Vrudhula haben in [70, 69, 68] wie zuvor mit der erweiterten Peukert's Formel ihr Modell mit dem Dualfoil-Simulationsprogramm für Batterien verglichen. Für eine konstant kontinuierliche Last hat das Modell einen durchschnittlichen Fehler von 3% und maximal von 6 % bei der Vorhersage von Batterielaufzeiten. Für variable und unterbrochene Belastungen hat das Modell einen maximalen Fehler von 2.7 % und im Durchschnitt weniger als 1 % verglichen mit dem Dualfoil-Simulationsprogramm [42].

#### 4.2.4.3. Kinetic Battery Model

Das Kinetic Battery Model (*KiBaM*) wurde von Manwell und McGowan entwickelt [54, 53, 52] und bündelt phänomenologische und physikalische Effekte. Dabei bezieht sich der Begriff „Kinetic“ aus dem Namen des Modells darauf, dass das Modell einen chemischen kinetischen Prozess als Basis verwendet. Dieses Modell gilt als sehr intuitives Batteriemodell. Es berechnet zum einen die Klemmspannung als eine Funktion des SoC und der Lade-/Entladerate als auch die Veränderungen der Kapazität durch die Lade-/Entladerate. Es ist also möglich, mit dem Modell sowohl Lade- als auch Entladevorgänge zu simulieren. Die Basis des *KiBaM* ist die Annahme, dass ein Teil der Ladung in einer Batterie frei verfügbar und der Rest der Ladung gebunden ist.

Im Kinetic Battery Model wird die Batterie als eine Spannungsquelle in Reihe mit einem Widerstand  $R_0$  betrachtet (s. Abbildung 4.6).

Dieser Widerstand ist der Innenwiderstand einer Batterie (s. auch Kapitel 2 Abschnitt 2.2.3), und es wird angenommen das  $R_0$  konstant ist. Des Weiteren gilt, dass die interne Spannung  $E$  (Quellenspannung) sich mit dem Zustand der Batterie (SoC) verändert und die Klemmspannung  $V$  durch folgende Formel gegeben ist [53]:

$$V = E - IR_0 \quad (4.13)$$

Im Folgenden werden jetzt die zwei Hauptbestandteile des Batteriemodells weiter erläutert. Das ist zum einen das Kapazitätsmodell und zum anderen das Spannungsmodell, wobei letzteres auf dem ersten aufbaut.

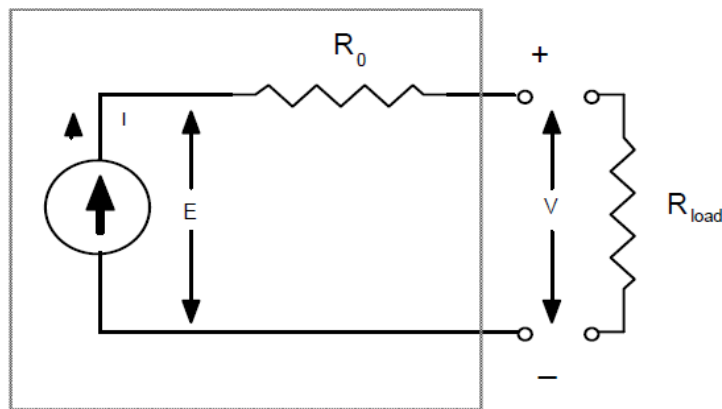


Abbildung 4.6.: Schematische Darstellung eines allgemeinen Batteriesystems [56].

### Kapazitätsmodell

Um die Kapazitätseffekte abbilden zu können, ist die Spannungsquelle modelliert als zwei Behälter, welche durch einen Leitwert (oder auch Ventil) voneinander getrennt sind. Ein Behälter enthält die Ladung, die sofort der Last zur Verfügung steht (Available Charge). Der andere Behälter enthält die Ladung, welche chemisch gebunden ist (Bound Charge). Dabei entspricht der Leitwert zwischen den beiden Behältern der Rate des chemischen Reaktions-/Diffusionsprozesses (Erster Ordnung), wodurch die gebundene Ladung wieder verfügbar wird. Weiterhin gilt, dass die Rate, mit der die gebundene Ladung wieder verfügbar wird, proportional zu der Differenz in der Höhe der beiden Behälter ist [53]. Abbildung 4.7 zeigt dazu eine Illustration des Modells.

Die Behälter sind gleich tief, haben aber eine unterschiedliche Breite, was den verschiedenen Volumen entspricht. So ist die Breite des Behälters mit der verfügbaren Ladung  $c$ , und die Breite des Behälters mit der gebundenen Ladung  $1 - c$ . Das gemeinsame Volumen der beiden Behälter wird mit  $q_{max}$  bezeichnet. Sind beide Behälter voll, ist der „Kopf“ der Behälter  $h_{max}$  gleich  $q_{max}$ . Weiterhin gilt, dass das Ventil zwischen den beiden Behältern einen festen Wert  $k'$  hat. Es gilt also, dass die Rate, zu der Ladung zwischen den Behältern fließt, zum einen vom Höhenunterschied der Behälter und zum anderen vom Parameter  $k'$  abhängt [53].

Des Weiteren wird zwischen der Last und der Batterie ein Stromregler eingesetzt, damit pro Zeitschritt, der gerade von Interesse ist, der Strom  $I$  konstant ist. Die Annahme eines konstanten Stroms pro Zeitschritt ist nötig, damit eine Simulation über Zeitschritte möglich ist [53]. Zum genauen Verständnis des Modells wird jetzt

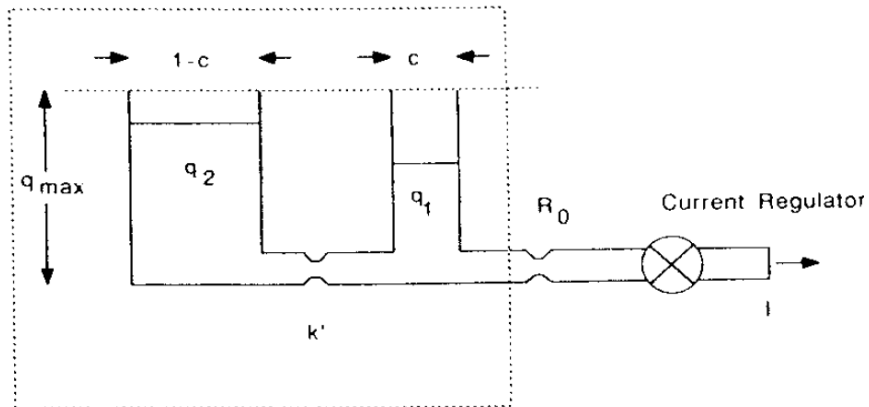


Abbildung 4.7.: Das Kinetic Battery Model modelliert eine Batterie als Behälter [53].

die analytische Basis dieses Batteriemodells beschrieben. Die Veränderung der Ladung in beiden Behältern ist mit der Erhaltung der Energie, angewandt auf die Batterie, durch das folgende System von Differentialgleichungen gegeben [53]:

$$\begin{cases} \frac{dq_1}{dt} = -I - k'(h_1 - h_2), \\ \frac{dq_2}{dt} = k'(h_1 - h_2), \end{cases} \quad (4.14)$$

mit  $q_1$  für die verfügbare Ladung,  $q_2$  für die gebundene Ladung und  $q_{max}$  für die maximale Ladung. Als initiale Bedingung gilt  $q_1 = c * q_{max}$  und  $q_2 = (1 - c) * q_{max}$ .

Weiterhin gilt, dass die Höhe des jeweiligen Behälters sich wie folgt ermitteln lässt [53]:

$$h_1 = \frac{q_1}{c} \quad (4.15)$$

$$h_2 = \frac{q_2}{1 - c} \quad (4.16)$$

Des Weiteren wird zur Vereinfachung eine neue Ratenkonstante  $k$  definiert [53]:

$$k = \frac{k'}{c(1 - c)} \quad (4.17)$$

Womit sich das System von Differentialgleichungen umformen lässt zu [53]:

$$\begin{cases} \frac{dq_1}{dt} = -I - k(1 - c)q_1 + kcq_2, \\ \frac{dq_2}{dt} = +k(1 - c)q_1 - kcq_2, \end{cases} \quad (4.18)$$

Das System von Differentialgleichungen aus 4.18 lässt sich mittels Laplace Transformation lösen [53]:

$$\begin{cases} q_1 = q_{1,0}e^{-kt} + \frac{(q_0kc-I)(1-e^{-kt})}{k} - \frac{Ic(kt-1+e^{-kt})}{k}, \\ q_2 = q_{2,0}e^{-kt} + q_0(1-c)(1-e^{-kt}) - \frac{I(1-c)(kt-1+e^{-kt})}{k}, \end{cases} \quad (4.19)$$

mit  $q_{1,0}$  und  $q_{2,0}$  für die Menge an verfügbarer bzw. gebundener Ladung zur Beginn der Berechnung und für  $q_0$  gilt  $q_0 = q_{1,0} + q_{2,0}$ .

Mit Hilfe beider Formeln ist es möglich, das Verhalten einer Batterie bezüglich der Kapazität für einen stückweise konstanten Strom zu simulieren, wobei die Zeitschritte beliebig sein können. Nach Berechnung eines Schrittes mit einem Strom  $I$  über eine Zeit  $t$  sind in  $q_1$  und  $q_2$  die neuen Füllstände der Behälter, welche dann wieder als initiale Bedingungen für den nächsten Schritt verwendet werden müssen. Wenn also ein Strom  $I$  gezogen wird, reduziert sich die Ladung im Behälter mit der verfügbaren Ladung ( $q_1$ ), und dadurch wächst der Höhenunterschied der Behälter. Wird der Strom reduziert bzw. ist Null, fließt Ladung vom Behälter der gebundenen Ladung zu dem mit der verfügbaren Ladung, bis  $h_1$  und  $h_2$  wieder gleich sind. Also wird in einer Ruhephase mehr Ladung verfügbar und die Batterie hält länger als bei einem kontinuierlichen Strom. Über diesen Weg berücksichtigt das Modell den Recovery Effekt. Der Rate Capacity wird auch vom Modell abgedeckt, denn für höhere Entladeströme wird der Behälter der verfügbaren Ladung schneller erschöpft und weniger Zeit ist dafür, dass gebundene Ladung zur verfügbaren Ladung fließen kann, verfügbar. Ist der Behälter der verfügbaren Ladung erschöpft, gilt die Batterie als erschöpft. Je schneller das passiert, desto mehr Ladung bleibt unbenutzt im Behälter der gebundenen Ladung. Weiterhin ist zu beachten, dass das Kapazitätsmodell eine spezifische Batterie über die drei Konstanten  $c$ ,  $k$  und  $q_{max}$  charakterisiert.

## Spannungsmodell

Neben dem Kapazitätsmodell besteht *KiBaM* auch aus einem Spannungsmodell, mit dem die Effekte des Ladens und Entladens auf die Spannung modellieren werden können, wobei im Folgenden nur auf die Spannungsmodellierung für die Entladung eingegangen wird. Für die Modellierung der Spannung beim Laden wird auf [53] verwiesen. Das Spannungsmodell basiert auf dem Battery Energy Storage Test („BEST“) Modell von Hyman et al [37, 36] und den Kapazitätsschätzungen aus dem Kapazitätsteil des *KiBaM*.

Dabei berücksichtigt das Spannungsmodell, dass die Klemmspannung einer Batterie abhängt von [56]:

- dem Ladezustand der Batterie (SoC)
- dem Innenwiderstand der Batterie
- der Höhe des Entladestroms

Wie eingangs bereits erwähnt, wird die Batterie im Modell als eine Spannungsquelle in Reihe mit einem Widerstand modelliert, und die Klemmspannung ist über die Formel 4.13 gegeben. Dabei ist interne Spannung  $E$  in der Formel 4.13 von dem Ladezustand abhängig. Die ursprüngliche Variante für die interne Spannung  $E$  sieht folgende Formel vor [54]:

$$E = E_{min} + (E_{0,d} - E_{min}) \frac{q_1}{q_{1,max}}, \quad (4.20)$$

mit  $E_{min}$  für die minimal erlaubte interne Entladespannung („Die Batterie ist erschöpft“) und  $E_{0,d}$  für die maximale interne Entladespannung („Die Batterie ist geladen“) und  $q_{1,max}$  für die maximale Ladung in  $q_1$ . Problem dieses Modells ist, dass es nicht in der Lage ist den starken Anstieg der Spannung am Ende des Ladens und den starken Abfall beim Entladen, wenn die Batterie fast erschöpft ist, darstellen kann. Um dies zu verbessern, wurde in [53] folgende Formel für  $E$  vorgeschlagen:

$$E = E_0 + AX + CX/(D - X) \quad (4.21)$$

mit

- $E$  = Interne Spannung,
- $E_0$  = Interne Spannung der vollgeladene Batterie,
- $A$  = Initiale lineare Abweichung der internen Spannung mit den SoC,
- $C, D$  = Spiegeln den Abfall der Batteriespannung wieder, wenn die Batterie zunehmend entladen wird,
- $X$  = Normalisierte aus der Batterie entnommene Ladung,
- $R_0$  = Innenwiderstand.

Für  $X$  existieren eine Reihe von Formeln in der Literatur. In der ersten Veröffentlichung [53] ist  $X$  wie folgt definiert:

$$X = q_{out} \frac{q_{max}}{q_0(I)}, \quad (4.22)$$

mit  $q_{out}$  für die aktuell entnommene Ladung und  $q_0(I)$  für die unter dem Strom  $I$  maximal entnehmbare Ladung aus der Batterie bis diese erschöpft ist.

In [56] ist  $X$  wie folgt definiert:

$$X = (q_{max} - q)/q_0(I), \quad (4.23)$$

mit  $q = q_1 + q_2$  für die gesamte aktuelle Ladung, also die Summe der beiden Behälter zum aktuellen Zeitpunkt. Eine weitere Art der Formel 4.23 existiert in [55], welche anstelle von  $q_{max}$   $q_0(I)$  nimmt.

Zur Modellierung einer bestimmten Batterie mit dem *KibaM* sind also zum einen die drei Parameter  $k$ ,  $c$  und  $q_{max}$  für das Kapazitätsmodell nötig und zum anderen die Parameter  $E_0$ ,  $A$ ,  $C$ ,  $D$  und  $R_0$ . Dabei können alle Parameter mit einer nichtlinearen Kurvenanpassung bzw. nichtlinearen kleinste Quadrat-Methode (engl. Non-Linear Least Square Curve Fitting) aus Testdaten der realen Batterie bestimmt werden oder aber auch bis auf die Parameter  $E_0$ ,  $A$ ,  $C$ ,  $D$  auch aus dem Datenblatt des Herstellers der Batterie.

Zur Simulation eines Zeitschrittes hingegen sind als Eingabe der Entladestrom  $I$ , die Länge des Zeitschrittes der Entladung  $t$  und die aktuellen Füllstände der Behälter  $q_1$  und  $q_2$  nötig. Als Ausgabe liefert das Modell die aktuelle Klemmspannung  $V$  und die neuen Füllstände der Behälter  $q_1$  und  $q_2$  nach dem Zeitschritt  $t$  mit dem Entladestrom  $I$ .

Vorteile des Kinetic Battery Modells ist, dass es beide für diese Arbeit wichtigen Effekte berücksichtigt. Es ist ein intuitives Modell, wodurch es einfacher ist, die Abläufe in der Batterie nachzuvollziehen. Ebenso unterstützt das Modell die Simulation mit beliebigem stückweise konstanten Strom in Bezug auf den Kapazitätsteil des Modells. Außerdem ist aufgrund der wenigen Parameter der Konfigurationsaufwand relativ gering, und auch der Berechnungsaufwand hält sich in Grenzen, weil sich die Berechnung für den Kapazitätsteil auf die zwei Formeln 4.19 reduziert und für den Spannungsteil des Modells auf die Formel für  $V$ ,  $E$  und  $X$ . Weiterhin ist nach [42] die Genauigkeit des Modells sehr hoch. Ebenso bringt das Modell eine gute Basis für das Spannungsmodell mit. Nachteil des Spannungsmodells hingegen ist, dass es nicht ohne weitere Anpassungen für ein beliebiges Entladeprofil verwendet werden kann, weil es nur für eine konstante Entladung gedacht ist (s. Formeln 4.22, 4.23). Weiterhin sind deshalb auch nicht die nichtlinearen Effekte im Spannungsmodell abgebildet, insbesondere nicht der Recovery Effekt.

#### 4.2.4.4. Weitere analytische Modelle

Weitere Modelle wurden entwickelt, mit deren Hilfe die Kapazität mit Funktionen über den Entladestrom, Temperatur und mehrere Konstanten ausgedrückt wird. Dazu wurde eine Vielzahl von Techniken verwendet, um die exakte Form der Funktion zu finden. So wurde z.B. in [82] eine Regressionsanalyse vorgestellt mit der die Koeffizienten, in einem Ausdruck, der auf dem Weibull Fehlermodell basiert, geschätzt werden können. Dieses Modell unterstützt konstante Lasten, den Rate Capacity und thermische Effekte. Das Modell aus [63] ist ein variables Last Modell, welches mit Hilfe empirischer Gleichungen den Rate Capacity Effekt berücksichtigt. Es kann dazu verwendet werden, Variationen in der Kapazität für Entladeprofile, welche verschiedenen Wahrscheinlichkeitsdichtefunktionen folgen, vorherzusagen. Weiterhin wird in [67] noch ein Modell vorgestellt, welches die Gesetze der chemischen Kinetik verwendet, um mathematische Ausdrücke abzuleiten. Es verwendet dabei statische Verfahren, um die Parameter des Modells zu schätzen und unterstützt variable Lasten. Peng Rong and Pedram stellen in [75] ein Batteriemodell für die Schätzung der verbleibenden Kapazität für konstante Lasten vor. Sie leiten unter anderem einen Ausdruck für die Klemmspannung ab und berücksichtigen auch Temperatureffekte. Weiterhin sind für die Konfiguration des Modells mehr als 15 verschiedene Parameter zu bestimmen.

Generell gilt, dass analytische Modelle sehr flexibel sind und einfach für eine spezifische Batterie konfiguriert werden können, und die Berechnung der Modelle ist meist effizient, weil sie die Auswertung relativ einfacher analytischer Ausdrücke ermöglicht. Ferner unterstützen einige der Modelle den Rate Capacity Effekt und Recovery Effekt. Ebenso gibt es auch hier Modelle, die variable Lasten unterstützen.

### 4.3. Modellauswahl

Da es den Rahmen dieser Arbeit gesprengt hätte, mehrere Modelle praktisch zu bewerten, und um anschließend eine Entscheidung für ein Modell zu treffen, wurde deshalb der Weg gewählt die Entscheidung für ein Modell erst einmal auf rein theoretischer Basis zu treffen und anschließend diese Entscheidung praktisch zu validieren, ob das gefundene Modell zur EnFilm-Batterie passt. Die entsprechende Validierung ist im nächsten Kapitel 5 beschrieben. Im Folgenden wird die Entscheidung für das in dieser Arbeit verwendete Modell erläutert. Wie zu Beginn des Kapitels beschrieben, muss das gesuchte Batteriemodell zur Modellierung von Lithium-Dünnschichtbatterien für die Bestimmung einer batteriegefahren Ablaufplanung unter Verwendung eines offline Schedulers gewisse Anforderungen erfüllen. Generell



muss das Modell in der Lage sein, dem offline Scheduler zu jedem Zeitpunkt Informationen über den Zustand der Batterie in Bezug auf die aktuelle Klemmspannung bzw. die verbleibende Ladung der Batterie zur Verfügung zu stellen. Vereinfacht gesagt, wird einfach Modell gesucht, welches die beiden nichtlinearen Effekte, d.h. den Recovery und Rate Capacity Effekt, berücksichtigt.

Dazu wurden im Vorfeld gängige Batteriemodelle aus der Literatur vorgestellt, klassifiziert und bewertet, damit es im Folgenden möglich ist, die Entscheidung für das in dieser Arbeit verwendete Batteriemodell zu begründen.

Obwohl die elektrochemischen Modelle die genauesten Modelle sind und deshalb wie das Dualfoil-Simulationsprogramm oft dazu verwendet werden, als „Realität“ um andere Batteriemodelle hinsichtlich ihrer Genauigkeit zu bewerten, sind die Modelle zu komplex für die Zwecke dieser Arbeit. So ist ein sehr tiefes Wissen über die zu modellierende Batterie nötig, um die Parameter zu bestimmen, von denen es im Falle von Dualfoil mehr als 50 Stück gibt. Weiterhin ist der Aufwand für die Berechnung der Modelle meist sehr hoch, was zu Ausführungszeiten von Stunden oder sogar Tagen führen kann.

Auch die elektrischen Schaltungsmodelle sind für die Belange dieser Arbeit zu komplex. Zwar haben sie einen geringeren Berechnungsaufwand, dafür ist aber der Aufwand der Konfiguration der Modelle zu hoch. Weiterhin unterstützt auch keiner der hier vorgestellten Modelle den Recovery Effekt oder modelliert thermische Effekte, welche in dieser Arbeit nicht betrachtet werden.

Die stochastischen Modelle weisen das Problem auf, dass es nicht möglich ist, das Lastprofil zu steuern und sie sind deshalb ungeeignet für diese Arbeit, weil es nötig ist, dass ein bestimmtes Entladeprofil vorgegeben werden kann. Weiterhin können diese Modelle auch sehr groß in Bezug auf die Zustandsmenge werden.

Die Klasse der analytischen Modelle eignet sich dagegen gut für die Zwecke dieser Arbeit, weil sie sehr flexibel sind und einfach für eine spezifische Batterie konfiguriert werden können. Des Weiteren ist die Berechnung der Modelle meist effizient, weil sich die Simulation auf die Auswertung relativ einfach analytischer Ausdrücke reduziert. Wobei aber beide Varianten der Peukert's Formel zu einfach sind, weil sie nicht den Recovery Effekt berücksichtigt. Dies führt dazu, dass die Batterieaufladungszeiten für Entlademuster mit Ruhephasen unterschätzt werden. Das *KiBaM* und das Diffusionsmodell von Rakhmatov und Vrudhula berücksichtigen beide den Recovery und Rate Capacity Effekt. Weiterhin verwenden beide Modelle ein System von zwei Differentialgleichungen, um eine Batterie für ein beliebiges stückweise konstantes Lastprofil zu simulieren. Für das *KiBaM* spricht, dass es einfacher zu handhaben ist. Weiterhin wird in [41] gezeigt, dass *KiBaM* eine Approximation erster Ordnung vom Diffusionsmodell ist, und dass beide Modelle eine ähnliche Güte haben. Auch dies spricht dafür, das einfachere *KiBaM* zu verwenden. Des Weiteren beschreibt das *KiBaM* die Batterie in einer intuitiveren Art und Weise

als das Diffusionsmodell, denn die Differentialgleichungen des *KiBaM* geben einen guten Einblick, wie das Batterieverhalten modelliert ist.

Generell erfüllt das *KiBaM* alle in Abschnitt 4.1 beschriebenen Anforderungen. Dies hat dazu geführt, dass das *KiBaM* für diese Arbeit verwendet wurde. Aber wie schon beschrieben, müssen noch Anpassungen am Spannungsteil des Modells vorgenommen und das Modell für die, in dieser Arbeit verwendete EnFilm-Batterie, angepasst werden, d.h. es müssen die batteriespezifischen Parameter des *KiBaM* aus dem Datenblatt der EnFilm oder aus Experimenten mit ihr gewonnen werden. Für das Bestimmen der Parameter gibt es schon fertige Software, was auch für die Wahl des Modells gesprochen hat.

Die Anpassungen, das Ermitteln der Parameter und die Validierung des Modells bezüglich der EnFilm-Batterie werden im anschließenden Kapitel 5 beschrieben.

# 5. Modellierung der EnFilm-Batterie durch das Kinetic Battery Model

Wie bereits im vorherigen Kapitel 4 beschrieben, wurde sich in dieser Arbeit für das *KiBaM* entschieden. Es ist es aber nicht einfach möglich, das Modell so ohne weiteres in den offline Scheduler zu integrieren. So wurde bereits erwähnt, dass Anpassungen am Spannungsteil des *KiBaM* nötig sind. Weiterhin ist es nötig, die batteriespezifischen Parameter für den Kapazitätsteil sowie auch für den Spannungsteil des Modells bezüglich der EnFilm-Batterie zu bestimmen. Schlussendlich gilt es zu zeigen, wie nahe das Modellverhalten an der Realität bzw. an der Enfilm-Batterie ist. Dazu werden im Folgenden zuerst die nötigen Anpassungen am Spannungsteil beschrieben. Daran anschließend wird die Bestimmung der Parameter beschrieben, wobei zuerst auf die Parameter des Kapazitätsteil des Modells eingegangen wird. Anknüpfend daran folgt die Bestimmung der Parameter des Spannungsteils, da der Kapazitätsteil des Modells die Basis des Spannungsteils darstellt. Dieses Kapitel schließt mit der Validierung des *KiBaM* bezüglich der Enfilm-Batterie. In diesem Zusammenhang wird die Zielsetzung der Validierung und auch die eingesetzten Validierungsverfahren beschrieben.

## 5.1. Anpassungen am Modell

Wie schon bei der Vorstellung des *KiBaM* beschrieben, ist der Spannungsteil des Modells für die Annahme entwickelt worden, dass bis zur Erschöpfung der Batterie ein konstanter Strom vorliegt. Deshalb ist es nicht ohne weitere Anpassungen möglich, den Spannungsteil für ein beliebiges Entladeprofil zu verwenden, was aber für diese Arbeit nötig ist. Der Grund dafür ist in der Formel 4.22 aus Kapitel 4 Abschnitt 4.2.4.3 bzw. auch in den anderen, in der Literatur zu finden, Formeln für die normalisierte entnommene Ladung ( $X$ ) zu finden. Es wird immer für die Normalisierung, mit  $q_0(I)$  für die unter dem Strom  $I$  maximal entnehmbare Ladung, davon ausgegangen, dass eine Entladung mit einem konstanten Strom  $I$  bis zur Erschöpfung der Batterie erfolgt. Weiterhin sind auch nicht die nichtlinearen Effekte im Spannungsmodell abgebildet, insbesondere nicht der Recovery Effekt. Das ist im Falle der Formel 4.22 darin begründet, dass über  $q_{out}$  nur die aktuell

entnommene Ladung betrachtet wird bzw. für die beiden anderen Formeln aus der Literatur [56, 55] darin begründet, dass sie an Stelle von  $q_{out}$   $q$  mit  $q = q_1 + q_2$  (aktuell insgesamt enthaltene Ladung in beiden Behältern) verwenden. So werden in allen drei Varianten die Behälter des *KiBaM* zusammenbetrachtet. Es wird also nicht berücksichtigt, dass die gebundene Ladung nicht zur freien Verfügung steht und die nichtlinearen Effekte, insbesondere der Recovery Effekte einen Einfluss auf die Spannung haben, da die Spannung sich im Falle einer Ruhephase nach einer Belastung erholt (s. Kapitel 2 Abschnitt 2.2.4). Weiterhin fällt die Spannung wegen des Rate Capacity Effekts im Falle einer höheren Belastung schneller ab als bei einer niedrigen. Mit der Annahme, dass die gebundene Ladung nicht zur freien Verfügung steht und somit keinen Einfluss bzw. einen zu vernachlässigenden Einfluss auf Spannung hat, wurde geschlussfolgert, dass die frei verfügbare Ladung einen maßgeblichen Einfluss auf die Spannung haben muss. Zusammen mit dem Einfluss der nichtlinearen Effekte und in Anlehnung an die Formel 4.20 aus Kapitel 4 Abschnitt 4.2.4.3 wurde für diese Arbeit deshalb folgende Formel für  $X$  entwickelt:

$$X = (q_{1max} - q_1)/q_{1max}, \quad (5.1)$$

mit  $q_1$  für die Ladung im Behälter der frei verfügbaren Ladung und  $q_{1max}$  für die maximale Ladung im Behälter für die verfügbare Ladung. Weiterhin berechnet sich  $q_{1max}$  wie folgt:

$$q_{1max} = cq_{max}, \quad (5.2)$$

mit  $c$  für Breite des Behälters mit der verfügbaren Ladung und  $q_{max}$  für das gemeinsame Volumen der beiden Behälter. Mit der Formel 5.1 werden jetzt im Spannungsteil des *KiBaM* die beiden nichtlinearen Effekte berücksichtigt. So hängt die Berechnung von  $X$  nicht mehr von der aktuell entnommenen ( $q_{out}$ ) oder der aktuell insgesamt enthaltenen Ladung ( $q$ ) ab, sondern von  $q_1$ . Damit haben die beiden nichtlinearen Effekte Einfluss auf die Spannung, weil für den Recovery Effekt gilt, dass in Ruhephasen die Ladungsmenge  $q_1$  und somit auch die Spannung ansteigt. Für den Rate Capacity Effekt gilt, dass die Ladung  $q_1$  schneller erschöpft ist, bei höheren Entladeströmen und somit auch die Spannung schnell abfällt als bei niedrigen Entladeströmen, weil weniger Zeit bleibt für die gebundene Ladung zum Behälter der frei verfügbaren Ladung zu fließen. Weiterhin besteht bei der neuen Formel für  $X$  auch nicht mehr die Annahme, dass die Batterie bis zu Erschöpfung mit einem konstanten Strom entladen wird, da die Formel nicht mehr von einem konstanten Entladestrom ( $I$ ) abhängt. Eine Validierung dieser Anpassung wird im Abschnitt 5.3 gegeben.

## 5.2. Bestimmung der Parameter des Modells

Nachdem im vorherigen Abschnitt beschrieben wurde, wie das Modell für die Zwecke dieser Arbeit angepasst wurde, wird jetzt vorgestellt, wie die batteriespezifischen Parameter für das Spannungs- und Kapazitätsmodell (s. Kapitel 4 Abschnitt 4.2.4.3) am Beispiel der Enfilm-Batterie bestimmt werden. Für die Bestimmung der Parameter gibt es von der Universität Massachusetts Amherst eine Software mit GUI namens *KiBaM Parameter Finder*<sup>1</sup>. Mit Hilfe dieser Software ist es möglich aus Messdaten einer realen Batterie die Parameter für das Spannungs- und Kapazitätsmodell zu bestimmen. Schon die ersten Ergebnisse der Simulation, mit konstanten Entladeströmen, des Kapazitätsmodells mit den Parametern aus dem *KiBaM Parameter Finder* waren aber im Vergleich zur realen Batterie bezüglich der Abweichungen in den ermittelten entnehmbaren Ladungen bei konstanter Entladung nicht zufriedenstellend. Dies liegt zum größten Teil daran, dass es bei der Software keine Möglichkeit gibt, die gewünschte Genauigkeit bzw. den Fehler, den das Modell haben darf einzustellen. Dazu kam das Problem, dass es für größere Datenmengen nicht möglich war, überhaupt Parameter zu bestimmen. Dies ist aber für ein vollständiges Abbild der Realität, besonders für das Kapazitätsmodell, entscheidend. Darauf wird im Folgenden noch genauer eingegangen. Da es keine Dokumentation zur Implementierung des *KiBaM Parameter Finder* gibt, war es nicht möglich, gesicherte Gründe für diese Probleme zu finden. Ein generelles Problem ist aber ein Qualitätsproblem der Daten, welches zur Bestimmung der Parameter benötigt werden (s.u.). Aufgrund dieser Probleme war es notwendig für die Bestimmung der Parameter des Kapazitätsmodells eine Alternative zu verwenden. Aufgrund dessen wurde auf die *Kinetic Battery Model Library* von Julius Susanto aus dem Jahre 2014<sup>2</sup> zum Schätzen der Parameter des Kapazitätsteils zurückgegriffen, welche nicht die genannten Probleme aufweist. Dabei enthält die *Kinetic Battery Model Library* nicht die Möglichkeit des bestimmens der Parameter für das Spannungsmodell. Außer des Algorithmus zum Bestimmen der Parameter für das Kapazitätsmodell enthält die Software auch eine Implementierung des Kapazitätsmodells. Diese wurde erweitert um das Spannungsmodell für die Simulation des Modells zum Zwecke der Validierung verwendet. Für die Bestimmung der Parameter des Spannungsmodells musste deshalb trotzdem auf den *KiBaM Parameter Finder* zurückgegriffen werden, wobei hier die Ergebnisse recht akzeptabel waren. Auf die Algorithmen zum Bestimmen der Parameter wird im Laufe des Kapitels genauer eingegangen.

Sowohl die Software für die Bestimmung der Parameter des Kapazitätsmodells

---

<sup>1</sup><https://www.umass.edu/windenergy/research/topics/tools/software/kibam>

<sup>2</sup><https://github.com/susantoj/kinetic-battery>

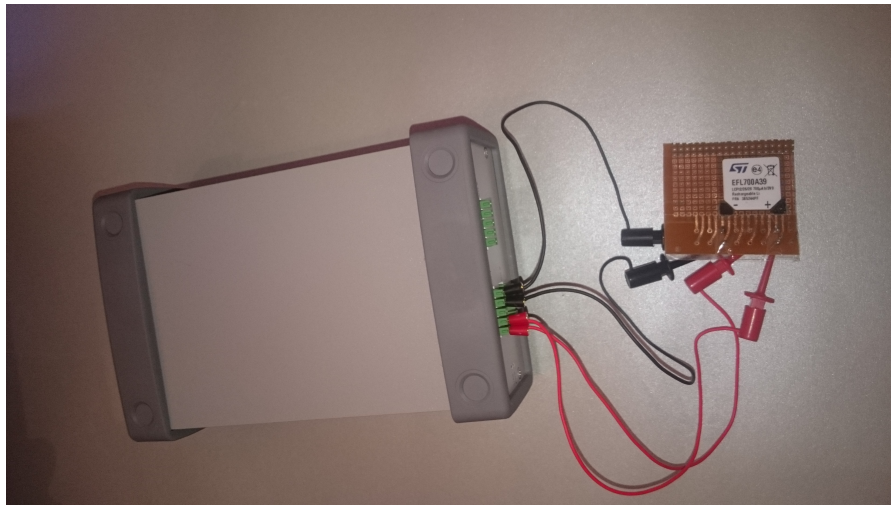


Abbildung 5.1.: Der Messaufbau mit dem Messgeräte (links) und der Enfilm-Batterie auf der Platine (rechts).

als auch die Software für die Bestimmung der Parameter des Spannungsmodells benötigen zur Ermittlung der Parameter reale Daten der zu modellierenden Batterie. Hierzu wurde zuerst versucht, die jeweils benötigten Daten durch Vermessen der EnFilm-Batterie zu bestimmen. Dazu wurde die Batterie zum leichteren Anschließen auf eine Platine mit Pins zum Anschluss des Messgerätes gelötet. Dies war nötig, weil die Batterie, wie in den Grundlagen beschrieben, sehr dünn ist und es sonst nicht die möglich war, das Messgerät anzuschließen. Das verwendete Messgerät (U2722A USB Modular Source Measure Unit von Keysight) wurde auch bereits in den Grundlagen vorgestellt und der komplette Versuchsaufbau ist in Abbildung 5.1 zusehen.

Mit Hilfe dieses Versuchsaufbaus wurde versucht, die Messkurven für konstante Ströme aus dem Datenblatt der Enfilm zu ermitteln (s. Abbildung 2.20 in den Grundlagen). Dabei wurde die Batterie durch eine vom Messgerät vorgegebene Last (jeweilige Last aus dem Datenblatt) entladen bis die Cut-Off-Spannung von 3.0 Volt erreicht war. Weiterhin wurde gleichzeitig die Spannung gemessen (für die Genauigkeiten des Messgerätes in den einzelnen Messbereichen wird auf das Datenblatt des Messgerätes verwiesen [85]). Während der Messungen ist aufgefallen, dass die Messwerte (Spannung) sehr stark von der Temperatur abhängig sind. Die Messung auf dem Datenblatt wurden dabei alle bei 30 °C durchgeführt. So führten Temperaturen von unter 30 °C dazu, dass die Spannung während einer Messung insgesamt niedriger war als die auf dem Datenblatt. Außerdem war auch die entnommene Ladung geringer. Da es während einer Messung, welche auch

mal 90 Stunden oder sogar über 900 Stunden dauern kann, ohne Einfluss auf die Temperatur zu Änderungen dieser kommen kann und für diese Arbeit keine Möglichkeit bestand die Temperatur während einer Messung, geschweige denn für alle Messungen, konstant zu halten, wurde sich in dieser Arbeit dazu entschieden die für die Bestimmung der Parameter erforderlichen Daten aus dem Datenblatt der EnFilm-Batterie zu entnehmen. (s. 2.20 in den Grundlagen). Eine Verwendung der Daten aus Messungen würde aufgrund der schlechten Datenqualität dazu führen, dass die Parameterbestimmung erschwert bzw. unmöglich wird. Der Vorteil ist, dass die Messdaten aus dem Datenblatt alle unter den gleichen Temperaturbedingungen von 30 °C ermittelt wurden. Der Nachteil hingegen ist, dass das Ablesen der Daten aus der Graphik zwangsweise zu Fehlern führt (niedrige Datenqualität), was aber mangels einer Alternative in Kauf genommen werden musste. Dadurch ist die Aussagekraft einer späteren Validierung mit diesen Daten auch eingeschränkt.

Die Simulation des *KiBaM*, welche zur Validierung des Modells benötigt wurde, ist im Wesentlichen die Berechnung der beiden Behälterfüllstände und die anschließende Berechnung der Spannung. Dabei ist ein Simulationsschritt die Neuberechnung der Behälterfüllstände ( $q_1, q_2$ ) und die anschließende Berechnung der Spannung ( $V$ ) für einen Zeitschritt  $t$  mit einem Entladestrom  $I$  und den vorherigen Behälterfüllständen ( $q_1, q_2$ ). Pro Simulationsschritt werden die ermittelten Werte für die entnommene Kapazität und für die Klemmspannung ( $V$ ) in ihrer jeweiligen Einheit protokolliert. Ein solcher Simulationsschritt wird solange wiederholt bis der Behälter mit der verfügbaren Ladung ( $q_1$ ) erschöpft ist bzw. der Behälter für den nächsten Simulationsschritt nicht mehr ausreichend gefüllt ist. Bei der Simulation zur Validierung des Modells wurde, um eine akzeptable Ausführungszeit zu erreichen, die Zeit  $t$  für einen Simulationsschritt auf eine Sekunde festgelegt.

Im Nachfolgenden wird genauer auf die Bestimmung der Parameter eingegangen.

### 5.2.1. Bestimmung der Parameter des Kapazitätsmodells

Die Software zum Schätzen der Parameter des Kapazitätsmodells verwendet die Methode des *Non-Linear Least Square Curve Fitting* für die Bestimmung der Parameter. Dabei basiert die verwendete Methode auf der Marquardt-Technik [57]. Diese Technik verwendet zum einen die Funktionsform einer Gleichung und die partiellen Ableitungen in Bezug auf die gesuchten Parameter. Weiterhin wird eine Minimierungsmethode auf Basis von Matrizen in einem iterativen Prozess verwendet, um die Parameter zu finden, welche die minimale quadratische Differenz zwischen den Eingabedaten (Reale Daten) und dem Modell ergeben. Generell gilt, dass der Algorithmus mit einer hohen Wahrscheinlichkeit auch bei schlechten Startbedingungen konvergiert. Die Konvergenz ist allerdings nicht garantiert. Die mathematischen Grundlagen zur Bestimmung der Parameter  $c, k, q_{max}$  sind dabei in [53]

Entladestrom (A)	Batterielaufzeit (h)
0.005	0.126
0.0035	0.1942
0.0014	0.5607
0.001	0.81
0.0007	1.1857
0.00035	2.4143
0.00007	12.4286
0.00001	90
0.000001	930

Tabelle 5.1.: Die zur Enfilm-Batterie ermittelte Batterielaufzeit der entsprechenden Entladeströme.

zu finden.

Als Eingabe bekommt der Algorithmus zuerst Daten über die reale Batterie in Form eines Vektors von Paaren aus Entladestrom und Batterielaufzeit. Ein Paar (Tupel) sagt also jeweils aus, wie lange es dauert die Batterie mit einem gegebenen konstanten Entladestroms zu entladen. Die benötigten Daten für den Algorithmus wurden aus den abgelesenen Messwerten aus dem Datenbatt mit folgender Formel berechnet (s. auch Kapitel 2 Abschnitt 2.2.2):

$$T = \frac{Q}{I} \quad (5.3)$$

mit  $T$  für die Batterielaufzeit,  $Q$  für die entnommene Ladung aus der Batterie und  $I$  für den Entladestrom. In Tabelle 5.1 sind die ermittelten Werte für die EnFilm-Batterie gelistet. Als Voraussetzung zum Ermitteln der Parameter müssen die Daten, wie sie im Diagramm 2.20 in den Grundlagen enthalten sind, vorliegen.

Weitere Eingaben für den Algorithmus sind die maximale Anzahl von Iterationen des Algorithmus, der erlaubte Fehler (Least Squares Error) und einem Vektor mit geratenen Werten für  $k, c, p_{max}$ . Der Algorithmus terminiert entweder, wenn die maximale Anzahl von Iterationen erreicht ist und keine Lösung gefunden wurde (Ergebnis: keine Konvergenz gefunden) oder auch wenn eine Lösung gefunden wurde innerhalb der erlaubten Anzahl von Iterationen (Ergebnis: Konvergenz gefunden).

Das Problem ist, dass in der Literatur [53, 56, 55] und auch in der Dokumentation der Software zur Bestimmung der Parameter nur beschrieben ist, dass mindestens drei der oben beschriebenen Tupel aus Entladestrom und Batterielaufzeit angegeben werden müssen. Aufgrund dessen, dass nicht genauer spezifiziert wird, welche Wertetupel welcher Kurven angegeben werden sollen und es mindestens drei sein



sollen, wurde in dieser Arbeit angenommen, dass es besser ist, je mehr Paare angegeben werden um ein Modell mit einer hohen Güte zu bekommen.

Für diese Annahme spricht, dass die ermittelten Werte für  $k$ ,  $c$ ,  $q_{max}$  je nach dem welche Wertetupel verwendet werden, sehr stark schwanken können. Am Beispiel von  $q_{max}$  ist zu beobachten, dass Wertetupel aus Kurven, die weit rechts im Diagramm 2.20 liegen logischerweise zu einem höheren  $q_{max}$  führen als Wertetupel, die weiter links im Diagramm liegen. Daher ist es wichtig, mit den Kurven bzw. Wertepaaren einen möglichst großen Bereich innerhalb der für eine Batterie erlaubten Entladeströme abzudecken, wobei es meist für den Entladestrom nur eine obere Schranke gibt. Als untere Schranke sollte der kleinste Entladestrom gewählt werden, der noch durch das Modell abgedeckt werden soll. Soll ferner durch das Modell die maximal zu erreichende Kapazität berücksichtigt werden, ist es wichtig, dass Daten über Kurven für sehr kleine (theoretisch unendliche kleine) Entladeströme vorliegen. In dieser Arbeit musste sich, aufgrund Probleme mit den Messungen, auf die Daten im Datenblatt beschränkt werden. Das Schwanken der Parameter, dass insbesondere durch das Weglassen von Kurven zur Bestimmung der Parameter die ganz rechts oder links liegen, bedingt ist, führt dazu dass diese nicht mehr durch das Modell gut abgedeckt werden können, was logisch ist, da für diese Bereiche keine Daten vorliegen. Generell gilt jedoch, dass es einfacher ist, ein genaues Modell für Kurven zu finden, die nahe beieinander liegen, was aber nur gilt, wenn das Modell auch nur für den, durch die Kurven abgedeckten Bereich verwendet wird. Weiterhin ist eine hohe Datenqualität auch entscheidend für das Finden eines guten Modells.

Da es sein kann, dass der Algorithmus zum Finden der Parameter nicht mit einen allen Wertetupeln eine Lösung, mit einer hinreichend kleinen durchschnittlichen prozentualen Abweichung vom den realen Daten, aufgrund z.B. einer niedrigen Datenqualität, findet. Wie es auch bei den Daten aus der Tabelle 5.1 der Fall war, wurde eine Heuristik zum Finden passender Parameter für ein gutes Modell entwickelt. Dabei wurde in dieser Arbeit eine durchschnittliche prozentuale Abweichung von 2 %, welcher als Richtwert bei der Validierung in der Heuristik dient, als ausreichend angenommen, um den Rahmen der Arbeit zu wahren. Weiterhin bezieht sich in dieser Arbeit der Fehler auch nur auf eine Validierung mit konstanten Entladeströmen, denn aufgrund der Probleme mit den Messungen an der realen Batterie nicht genügend Daten für die Validierung mit variablen Entladeströmen vorhanden waren. Als Voraussetzung für diese Heuristik sollten die Kurven aus den denen Wertetupel ermittelt wurden, aber möglichst nahe beieinander liegen (insbesondere in Bezug auf die entnommene Ladung). Dadurch liegen dann auch die entsprechenden Wertetupel nahe beieinander. Da es im Laufe der Heuristik nötig sein kann, dass Wertetupel, welche unter anderem aus den äußersten Kurven aller Kurven resultieren, zum Erhöhen der Datenqualität entfernt werden müssen. Dies führt dazu, wenn die Kurven nicht nahe beieinander liegen, dass große Teile der

Realität durch das Modell nicht mehr gut abgedeckt werden können (z.B. bei den Beobachtungen zu  $q_{max}$ ). Liegen die Kurven nach der Voraussetzung nahe beieinander, geht nur ein kleiner Teil der Realität pro Wertepaar, welches entfernt wird, verloren. Dabei gilt, je enger die Kurven, desto besser. Die Heuristik wurde auf Grundlage der obigen Annahme und Voraussetzungen wie folgt definiert:

1. Rufe den Algorithmus für die Bestimmung der Parameter mit allen gegebenen Wertetupeln (bzw. mit allen noch übriggebliebenen Wertetupeln), einen Square Error von 0.1 (bzw. mit dem verringerten Square Error), einer maximalen Anzahl von 1000000 Iterationen und einem geratenen Vektor mit den Werten für  $(k, c, q_{max})$  auf.
  - Terminiert der Algorithmus innerhalb der maximalen Anzahl von 1000000 Iterationen, hat er eine Lösung mit einem kleineren Square Error als den angegebenen gefunden. Mache weiter mit Schritt 2.
  - Terminiert er nicht, hat er keine Lösungen innerhalb der maximalen Anzahl von 1000000 Iterationen gefunden. Starte mit Schritt 1. von Beginn an, aber entferne zuvor ein beliebiges Wertetupel. Ist dabei die minimale Anzahl von drei Wertetupel unterschritten, breche die Heuristik ab. Führt das entfernte Wertetupel nicht dazu, dass der Algorithmus eine Lösung findet, füge dieses Wertetupel wieder hinzu.
2. Validiere das Modell, welches mit den gefundenen Werten instanziiert wurde, durch die Simulation (wie oben beschrieben) für alle bis jetzt nicht für die Bestimmung der Parameter verwendeten Entladeströmen aus Wertetupeln und vergleiche die Simulationsdaten mit den entsprechenden real vorhandenen Daten. Ist die durchschnittliche prozentuale Abweichung kleiner gleich 2 % (bzw. kleiner gleich einer beliebigen anderen festgelegten durchschnittlichen prozentualen Abweichung), breche hier die Heuristik ab. Ist die durchschnittliche prozentuale Abweichung größer als 2 % (bzw. größer als einer beliebigen anderen festgelegten durchschnittlichen prozentualen Abweichung), gehe zu Schritt 1. und verringere den Square Error um den Faktor 0.1.

Findet die Heuristik keine Lösung, besteht die Möglichkeit, über eine Erhöhung der Datenqualität und einer erneuter Ausführung der Heuristik eine Lösung zu finden. Es kann aber auch sein, dass das Modell die Realität zu dieser Genauigkeit nicht abbilden kann und die vorgegebene durchschnittliche prozentuale Abweichung zum Finden einer Lösung erhöht werden muss. Diese Heuristik hat nicht die beste Lösung zum Ziel, sondern soll nur als Ansatz zum Finden eines Modells mit einer vorgegebenen durchschnittlichen prozentualen Abweichung dienen. Die Einschränkung auf 1000000 Iterationen soll, dabei die Ausführungszeit des Algorithmus auf

Parameter	ermittelter Wert
$q_{max}$ (Ah)	0.000871045131
$k$	11.1587886
$c$	0.594952992

Tabelle 5.2.: Die ermittelten Werte der Parameter des Kapazitätsmodells für die Enfilm-Batterie.

einen praktikablen Wert halten. Im 2. Schritt der Heuristik ist es zur Kreuzvalidierung nötig bzw. sinnvoll, keine der Wertetupel zu nehmen, mit welchen eine Lösung gefunden wurde. Da in dieser Arbeit, aufgrund der oben genannten Probleme, aber nicht genügend Daten für eine vernünftige Kreuzvalidierung zur Verfügung standen, wurden trotzdem alle Daten einschließlich derer, die zur Bestimmung der Parameter verwendet wurden, auch zur Validierung benutzt. Zwar ist dadurch die Aussagekraft der Validierung beschränkt, sie gibt jedoch zumindest einen Anhaltspunkt bzw. eine Richtung für die Güte des Modells vor. Weiterhin spricht auch dafür, dass sich in dieser Arbeit darauf beschränkt wurde, die Entladeströme bzw. den Entladestrom in nächster Nähe der Daten, welche zur Bestimmung der Parameter geführt haben, zu verwenden. Weiterhin wird bei der Simulation für die Validierung und auch bei dem späteren Einsatz des Modells im Scheduler nicht die Batterielaufzeit  $T$  aus den Wertetupeln verwendet, sondern, wie eingangs bereits beschrieben, solange ausgeführt bis  $q_1$  leer ist. Was die Aussagekraft der Validierung wieder erhöht, weil nur die Entladeströme identisch zu denen sind, welche bei der Parameterbestimmung verwendet wurden, und nicht aber auch die Batterielaufzeiten. So kann sein, dass  $q_1$  nach einer Batterielaufzeit  $T$  nicht erschöpft ist und so die Simulation zu anderen Werten kommt. So werden zur Validierung nicht genau dieselben Daten verwendet, welche auch zur Bestimmung der Parameter verwendet werden.

Mit Hilfe dieser Heuristik wurden für die EnFilm-Batterie die Parameter für das Kapazitätsmodell bestimmt. Die ermittelten Werte sind in Tabelle 5.2 zu sehen. Dabei wurden die Parameter mit Hilfe der Wertepaare in der Tabelle 5.3 bestimmt.

Zu beachten ist, dass der Wert für  $q_{max}$  nicht der maximalen Kapazität des Entladestroms von  $1 \mu A$  entspricht, weil diese sowie auch die Kurve des Entladestroms von  $10 \mu A$  nicht in den Daten, welche zur Bestimmung dieser Parameter verwendet wurden, nicht enthalten waren (s. auch bei der Diskussion oben über  $q_{max}$ ).

Die Validierung zum Kapazitätsmodell bzw. zum *KiBaM* insgesamt wird in Abschnitt 5.3 vorgestellt. Abschließend ist zu sagen, dass es keinen „perfekten“ Algorithmus für eine nichtlineare Anpassung gibt und vieles das Ergebnis beeinflussen kann [17].

Entladestrom (A)	Batterielaufzeit (h)
0.005	0.126
0.0035	0.1942
0.0014	0.5607
0.001	0.81
0.0007	1.1857
0.00035	2.4143
0.00007	12.4286

Tabelle 5.3.: Die Wertepaare, welche zur Bestimmung der Parameter verwendet wurden.

### 5.2.2. Bestimmung der Parameter des Spannungsmodells

Wie eingangs beschrieben wird für die Ermittlung der Parameter des Spannungsmodells trotz der genannten Probleme der *KiBaM Parameter Finder* der Universität Massachusetts Amherst verwendet. Genau wie der Algorithmus zur Bestimmung der Parameter des Kapazitätsmodells ist die Basis des *KiBaM Parameter Finder* eine Methode des *Non-Linear Least Square Curve Fitting*, welche wiederum auf der, im vorherigen Abschnitt beschriebenen, Marquardt-Technik [57] basiert. Abbildung 5.2 zeigt die GUI des *KiBaM Parameter Finder*. Wie auch schon beschrieben, ist es mit dem *KiBaM* möglich Entlade- und Ladeprozesse zu simulieren, deshalb ist es mit *KiBaM Parameter Finder* möglich, Parameter für beide Fälle zu bestimmen. Im Folgenden wird sich aber nur auf die Parameter für den Entladeprozess konzentriert.

Auch hier sind mindestens vier Eingabedatenmengen in einer Datei nötig, damit die Software überhaupt Parameter ermitteln kann. Dabei ist mit vier Eingabedatenmengen gemeint, dass für vier verschiedene Entladeströme jeweils eine Menge von Wertetupeln mit Klemmspannung ( $V$ ) vs. entnommene Ladung (Ah) vorliegen. Wobei aber bei den Daten der EnFilm-Batterie (s. Abbildung 2.20) alle Werte um einen Faktor von 1000 skaliert werden müssen, damit der *KiBaM Parameter Finder* richtig arbeitet. Dies hat aber keinen Einfluss auf die ermittelten Parameter, weil die Form der Kurven der Daten nicht durch eine Skalierung verändert wird.<sup>3</sup> Der Grund dafür ist, dass die Software für Batterie mit einer größeren Kapazität entwickelt wurde. Die Tabelle 5.4 gibt ein Beispiel einer Eingabedatenmenge für einen Entladestrom von 0.005 mA an, wobei die Daten auch schon skaliert wurden. Dabei wurden die Daten aus dem Datenblatt an signifikanten Stellen abgelesen und inter-

<sup>3</sup>Nur der ermittelte Werte für den Innenwiderstand  $R_0$  muss entspricht dem Faktor 0.001 wieder angepasst werden.

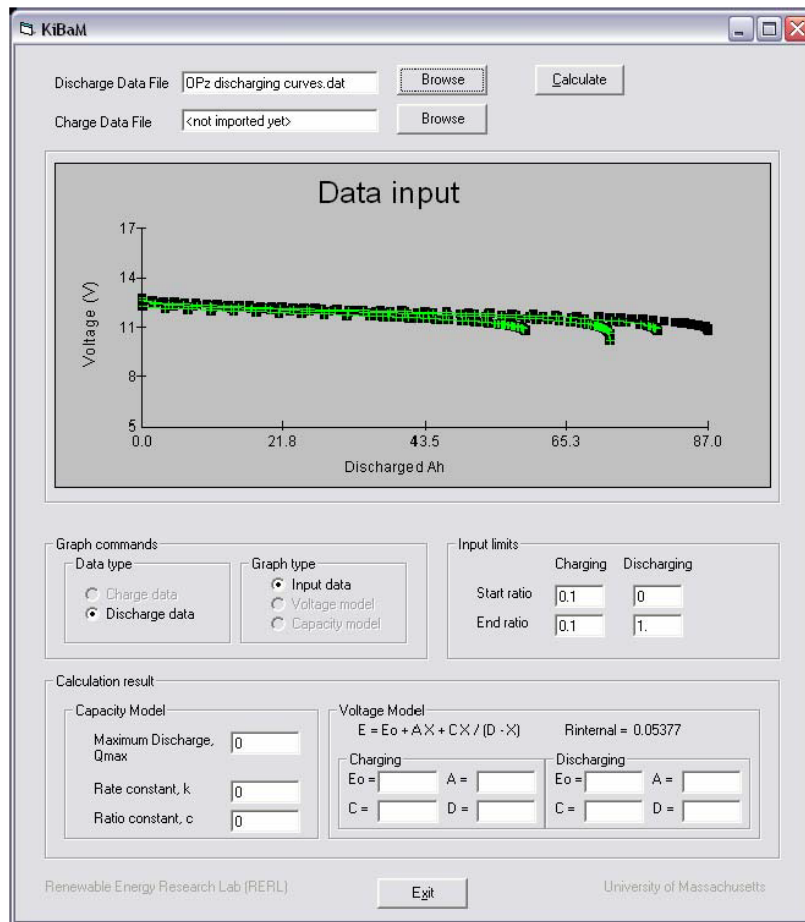


Abbildung 5.2.: GUI des *KiBaM Parameter Finder* [92]. Zu sehen ist oben die Möglichkeit, Daten über den Entlade-/Ladeprozess einer Batterie einzugeben. In der oberen Mitte ist die graphische Darstellung der Eingabedaten bzw. der Modelle (Spannungs- und Kapazitätsmodell) möglich. In der unteren Mitte links kann zwischen den Eingabedaten und verschiedenen Typen von Graphen gewechselt werden. In der unteren Mitte rechts kann die betrachtete Menge der Eingabedaten begrenzt werden und ganz unten werden die ermittelten Parameter angezeigt.

poliert. Wie auch der andere Algorithmus, versucht sich die Software den Kurven anzunähern.

Zur Bestimmung der Parameter für das Spannungsmodell durchläuft die Software im Wesentlichen folgende Schritte:

1. Elimination des initialen Spannungsabfalls in jeder Menge von Punkten.
2. Extrapolation der Spannungskurven bis zum Startpunkt der Entladung (Zeitpunkt Null) zum Finden der Spannung einer vollen Batterie für einen gegebenen Entladestrom ( $V_0(I)$ ).
3. Anwendung einer linearen Anpassung an  $V_0(I)/I$  um  $R_0$  (Innenwiderstand) zu finden.
4. Ergänzung von  $IR_0$  zu jedem Spannungswert um die normalisierte interne Spannung  $E$  zu finden.
5. Justierung der Achse der entnommenen Ladung der Batterie durch Multiplikation mit der maximalen Kapazität und durch Division mit der Kapazität zu dem Entladestrom.
6. Ausführung nichtlineare Anpassung an  $E$  vs.  $X$  um die Parameter  $E_0$ ,  $A$ ,  $C$  und  $D$  zu finden.

Auch hier ist wieder die mathematische Grundlage des Verfahrens in [53] zu finden. Weiterhin sind in [55, 92] Beschreibungen für die Bedingung des *KiBaM Parameter Finder* zu finden. Problem dieser Software ist, wie oben beschrieben, dass es nicht möglich ist, mit den für diese Arbeit vorhandenen Daten für eine größere Menge an Kurven eine Parameterbestimmung durchzuführen, weil der *KiBaM Parameter Finder* eine Konvergenz findet. So war es z.B. nicht möglich, für neun Kurven oder alle Kombinationen aus acht Kurven (ohne die Reihenfolge zu beachten) aus dem Datenblatt Parameter zu bestimmen. Dies ist wahrscheinlich auf eine schlechte Datenqualität zurückzuführen. Weiterhin besteht das Problem, dass die Software nur über die GUI gesteuert werden kann und es so nicht möglich ist, wie bei der Heuristik zum Kapazitätsmodell, alle Kombination von Kurven und Anzahl von Kurven schnell durchzutesten. Alles per Hand durchzuführen, wäre nicht praktikabel. Zusätzlich zu diesem Aufwand kommt der Aufwand, gefundene Lösungen zu vergleichen, um die Beste zu finden hinzu.

So wurde sich in dieser Arbeit dazu entschieden, mit der minimalen Anzahl von vier Eingabedatenmengen (entspricht vier Kurven) eine Lösung zu finden, da es wahrscheinlicher ist, dass der Algorithmus für eine geringe Menge an Kurven eine

Entladestrom (A)	Entnommene Ladung (Ah)	Klemmspannung (V)
5	0.0000	3.8300
5	0.0157	3.8200
5	0.0313	3.8100
5	0.0470	3.8000
5	0.0647	3.7833
5	0.0823	3.7667
5	0.1000	3.7500
5	0.1333	3.7300
5	0.1667	3.7100
5	0.2000	3.6900
5	0.2333	3.6700
5	0.2667	3.6500
5	0.3000	3.6300
5	0.3208	3.6200
5	0.3417	3.6100
5	0.3625	3.6000
5	0.3750	3.5900
5	0.3875	3.5800
5	0.4000	3.5700
5	0.4333	3.5367
5	0.4667	3.5033
5	0.5000	3.4700
5	0.5167	3.4467
5	0.5333	3.4233
5	0.5500	3.4000
5	0.5667	3.3333
5	0.5833	3.2667
5	0.6000	3.2000
5	0.6083	3.1333
5	0.6167	3.0667
5	0.6250	3.0000

Tabelle 5.4.: Beispiel einer Eingabedatenmenge für den *KiBaM Parameter Finder*.

1. Kurve	2. Kurve	3. Kurve	4. Kurve
5 mA	3.5 mA	1.4 mA	1 mA
5 mA	3.5 mA	1.4 mA	0.7 mA
5 mA	3.5 mA	1.4 mA	350 $\mu$ A
5 mA	3.5 mA	1.4 mA	70 $\mu$ A
5 mA	3.5 mA	1.4 mA	10 $\mu$ A
5 mA	3.5 mA	1.4 mA	1 $\mu$ A
3.5 mA	1.4 mA	1 mA	0.7 mA
3.5 mA	1.4 mA	1 mA	350 $\mu$ A
3.5 mA	1.4 mA	1 mA	70 $\mu$ A
3.5 mA	1.4 mA	1 mA	10 $\mu$ A
3.5 mA	1.4 mA	1 mA	1 $\mu$ A
3.5 mA	1.4 mA	1 mA	5 mA
1.4 mA	1 mA	0.7 mA	350 $\mu$ A
...	...	...	...

Tabelle 5.5.: Ein Ausschnitt aus den verwendeten Kombinationen zur Bestimmung der Parameter des Spannungsmodells.

Lösung findet. Dies liegt daran, dass potenziell weniger Daten von schlechter Qualität enthalten sein können. Aber auch hier wurden nicht alle 126 Kombinationen durchgetestet, weil es immer noch zu aufwändig wäre, alles von Hand durchzuspielen und zu validieren. Deshalb wurde ein Schema angewendet, bei welchem sich eine Kombination von der nächsten nur einen Wert unterscheidet. Hintergrund dieses Verfahrens ist, dass durch das Austauschen jeweils nur einer Kurve ein besseres Verständnis über die Datenqualität der einzelnen Datenmengen (Kurven) gewonnen werden kann und weiterhin eine immer noch gute Abdeckung des Arbeitsbereiches der Batterie (bezogen auf die erlaubten Entladeströme) gegeben ist. Mit diesen Methoden sind nur 36 Durchläufe nötig. Tabelle 5.5 gibt dazu einen Ausschnitt aus der Menge aller Kombinationen an, wobei jede Zeile einer Kombination entspricht und ein Wert in einer Zeile stellvertretend für die Kurve mit diesem Entladestrom steht.

Nach dem Durchspielen aller dieser Kombinationen hatte sich gezeigt, dass sich unter der Menge der Kombinationen die Lösung, welche möglichst viele der verschiedenen Kurvenformen der Entladekurven enthält, als gut erweist. Denn analog zur Bereichsabdeckung bei dem Kapazitätsmodell ist es wichtig, dass alle Kurvenformen beachtet werden. Nur so ist es möglich, dass das Spannungsmodell in dem gesamten Arbeitsbereich, bezogen auf die erlaubten Ströme, ein gutes Abbild der Realität hat. So ist eine Instanz des Modells mit einer Lösung, welche nur Kurven



enthält, die sehr nahe beieinander liegen und sich vom Verlauf sehr ähnlich sind, in diesem Bereich sehr akkurat, aber nicht in den nicht abgedeckten anderen Bereichen. So zeigt die Abbildung 5.3 die Kurve für einen Entladestrom von 3.5 mA aus dem Datenblatt und die entsprechende Kurve aus dem Modell (im Folgenden mit Modell 1 bezeichnet). Dieses Modell wurde dabei mit den Parametern, welche mit folgender Kombination von Kurven für die Entladeströme 5 mA, 350  $\mu$ A, 70  $\mu$ A und 10  $\mu$ A gefunden wurden, instanziiert. Abbildung 5.4 zeigt auch die Kurve für einen Entladestrom von 3.5 mA aus dem Datenblatt und die entsprechende Kurve aus dem Modell (im Folgenden mit Modell 2 bezeichnet), wobei dieses Modell hingegen mit den Parametern, welche mit folgender Kombination von Kurven für die Entladeströme 1 mA 0.7 mA 350  $\mu$ A 70  $\mu$ A gefunden wurden, instanziiert wurde.<sup>4</sup>

Zu sehen ist im Vergleich der beiden Bilder, dass die Kurve des Modells aus Abbildung 5.3 viel näher an der Kurve aus dem Datenblatt ist, als die Kurve des anderen Modells, welche in Abbildung 5.4 zu sehen ist. Dies bestätigt auch das oben Beschriebene, dass möglichst alle Kurvenformen abgedeckt werden müssen. Das ist mit der Kombination 5 mA, 350  $\mu$ A, 70  $\mu$ A und 10  $\mu$ A eher gegeben als mit der anderen vorgestellten Kombination, denn die andere deckt z.B. wegen der fehlenden Kurve für 5mA nicht die großen Entladeströme ab. So zeigt das Modell 1 nach der Validierung durch Vergleich mit den Kurven, welche nicht zur Bestimmung der Parameter verwendet worden sind, mit den entsprechenden Kurven im Datenblatt, eine durchschnittliche prozentuale Abweichung von 1.46 % im Vergleich dazu zeigt das Modell 2 eine durchschnittliche prozentuale Abweichung von 3.03 %.

Auf Grundlage dieser Erkenntnis wurden ausgehend von den Datenmengen, mit deren Hilfe die Parameter für das Modell 1 bestimmt wurden, weitere Untersuchungen durchgeführt. Dabei wurden die äußersten Kurven für 5 mA bzw. 10  $\mu$ A Entladestrom festgehalten und für dazwischen liegenden Entladeströme alle Kombination untersucht. Dabei konnte die durchschnittliche prozentuale Abweichung um 0.06 % im Vergleich zu Modell 1 verringert werden. Aufgrund der Ungenauigkeiten bezüglich der Vergleichsbasis der abgelesenen Daten aus dem Datenblatt, ist die Verbesserung fast zu vernachlässigen. Nichts desto trotz wurde diese Lösung, welche aus den Entladekurven für 5 mA, 0.7 mA, 350  $\mu$ A und 10  $\mu$ A bestimmt wurde, für das Spannungsmodell der Enfilm-Batterie verwendet. Dazu enthält Tabelle 5.6 die ermittelten Werte der Parameter für das Spannungsmodell. Aber generell ist auch hier wieder entscheidend, welche Teile der Realität das Modell bezüglich der erlaubten Ladung abdecken soll. Denn, wie schon beschrieben, ist es einfacher, für einen kleinen Bereich der Realität Parameter zu bestimmen, welche meist be-

---

<sup>4</sup>Wobei aus Gründen der Vergleichbarkeit die Kurven aus dem Modell nur bis zu dem Punkt gezeichnet wurden, bei dem die Batterie für diesen Entladestrom laut Datenblatt erschöpft bzw. die Batterie laut Modell erschöpft ist. Die Abweichungen im Kapazitätsmodell zur Realität werden im Verlauf dieses Kapitels noch behandelt.

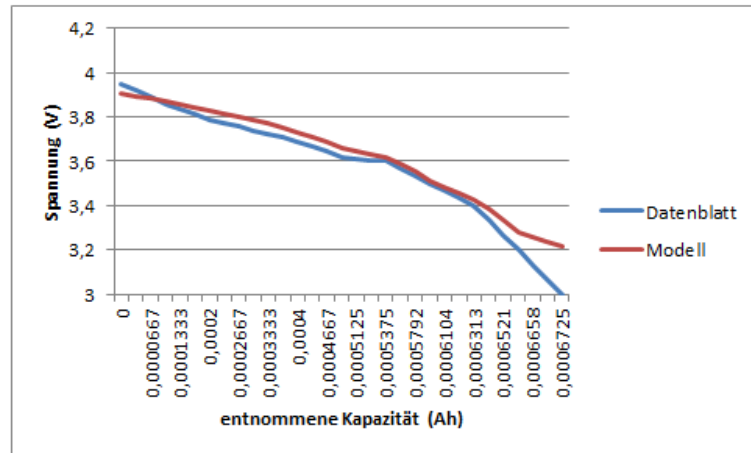


Abbildung 5.3.: Entladekurven entsprechend des Entladestroms 3.5mA aus dem Datenblatt und Modell 1.

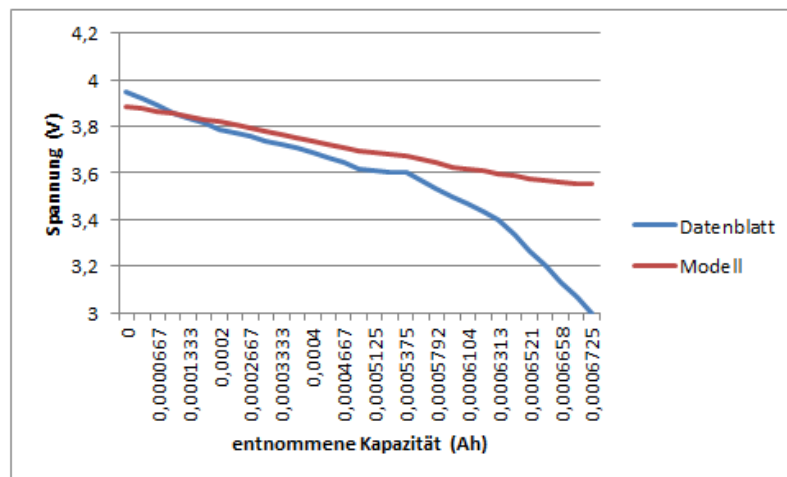


Abbildung 5.4.: Entladekurven entsprechend des Entladestroms 3.5mA aus dem Datenblatt und Modell 2.

Parameter	ermittelter Wert
$E_0$	4.1193
$A$	-0.1638
$C$	-0.0497
$D$	1.0809
$R_0$	77.82

Tabelle 5.6.: Die ermittelten Werte der Parameter des Spannungsmodells für die Enfilm-Batterie.

zogen auf diesen Teil der Realität auch zu einem genaueren Modell führen. Da aber in dieser Arbeit möglichst ein großer Teil der Realität abgedeckt werden soll, wurde die oben genannte Lösung gewählt. Im Folgenden wird auf die Güte des so bestimmten Modells für die Enfilm-Batterie eingegangen, wo auch verdeutlicht wird, wie die, in diesem Abschnitt aufgeführten, durchschnittlichen prozentualen Abweichungen zustande gekommen sind.

### 5.3. Güte des Modells

Im diesem Abschnitt soll die Güte des *KiBaM* vorgestellt werden, genauer gesagt die Güte des für die Enfilm-Batterie bestimmten *KiBaM*. Dies ist für die abschließende Evaluation der batteriegefahrenen Ablaufplanung nötig, da nur durch Kenntnisse über die Güte des Batteriemodells auch schlussendlich Aussagen über die Güte des Konzepts zur Ermittlung einer batteriegefahrenen Ablaufplanung getätigt werden können. Der Grund dafür ist, dass das *KiBaM* in den offline Scheduler zur Bestimmung einer batteriegefahrenen Ablaufplanung integriert ist und der Scheduler Informationen über den Zustand der Batterie für Schedulingentscheidungen aus dem Modell nimmt. Die Güte eines ermittelten Scheduling im Verhältnis zur Realität wird maßgeblich durch die Güte des Batteriemodells bestimmt. Deshalb werden im Folgenden die Ergebnisse der Validierung des Kapazitäts- und Spannungsmodells und damit die Güte des *KiBaMs* beschrieben.

#### 5.3.1. Zielsetzung der Validierung

Die Zielsetzung der Validierung im Allgemeinen ist die Bestimmung der Güte des *KiBaM*, im Speziellen die Bestimmung der Güte des Kapazitäts- und Spannungsmodells. Ziel ist es also, sowohl für konstante als auch variable Entladeströme das Modell mit der Realität zu vergleichen. In diesem Zusammenhang soll auch gezeigt werden, wie gut die nichtlinearen Effekte (Recovery und Rate Capacity Effekt) der

EnFilm-Batterie durch das Modell dargestellt werden. Hierbei soll der Einfluss auf die Kapazität wie auch auf die Spannung untersucht werden. Damit soll auch gezeigt werden, wie valide die in Abschnitt 5.1 vorgestellte Anpassung der Formel für die normalisierte entnommene Ladung  $X$  ist. Insgesamt soll also gezeigt werden, wo die Grenzen des Modells bezüglich der Realität liegen.

### 5.3.2. Vorstellung des Validierungsverfahrens

Die Validierung des *KiBaM* lässt sich in zwei Teile zerlegen. So wurden Untersuchungen für konstante Entladeströme durchgeführt. Dabei wird zuerst das Kapazitätsmodell, weil es die Basis des Spannungsmodells bildet, validiert. Anschließend wird das Spannungsmodell für konstante Entladeströme validiert. Die Untersuchungen zu konstanten Entladeströmen sind auch deshalb nötig, weil es so möglich ist, das *KiBaM* bezüglich des Rate Capacity Effekte zu überprüfen. Nach der Untersuchung mit konstanten Entladeströmen werden Untersuchungen zu variablen Entladeströmen vorgestellt. Damit ist es zum einen möglich, die Güte unter variablen Entladeströmen zu zeigen und zum anderen, den Recovery Effekt zu validieren. Die Vergleichsdaten kommen dabei für die konstanten Entladeströme aus dem Datenblatt der Enfilm-Batterie [79], denn wie eingangs beschrieben, wurden die batteriespezifischen Parameter für das Modell auch mit den Daten aus dem Datenblatt ermittelt. Dies lag an den beschriebenen Problemen des Temperatureinflusses, das zu einer schlechten Datenqualität führt und so die Bestimmung der Parameter weiter erschwert hätte. Begründet ist also diese Entscheidung dadurch, dass Vergleichsdaten, welche unter konstanten Bedingungen erstellt wurden, so gegeben sind und es im Rahmen dieser Arbeit keine vernünftigen Möglichkeiten gab, solche Bedingungen zu schaffen. Wie schon im Abschnitt 5.2.1 ist es bei der Validierung nicht sinnvoll, dieselben Daten zu verwenden (Kreuzvalidierung), welche auch zur Bestimmung der Parameter verwendet wurden. Da es aber im Falle des Kapazitätsmodells aufgrund der geringen Menge an Daten nicht anders möglich war, musste dieser Weg dennoch gewählt werden (s. Diskussion oben Abschnitt 5.2.1).

Neben den Vergleichsdaten für konstante Entladeströme werden Daten zum Validieren der Güte des Modells unter variablen Entladeströmen benötigt. Da es hierzu im Datenblatt keine Daten bzw. keine brauchbaren Daten gab, musste hier trotz der beschriebenen Probleme auf eine Vermessung der realen EnFilm-Batterie zurückgegriffen werden. Dazu wurde der in Abschnitt 5.2 beschriebene Versuchsaufbau verwendet.

Die Daten aus dem Modell wurden dabei, wie oben im Verlauf des Kapitels bereits erläutert, durch Simulation des Modells bestimmt. Für die konstanten Entladeströme wird im Falle der Validierung das Kapazitätsmodell bis zur Erschöpfung

von  $q_1$  ausgeführt und im Falle des Spannungsmodell bis zu der Kapazität, welche auch von der realen Enfilm-Batterie erreicht wird oder bis zu dem Punkt, an dem die Batterie laut Modell früher als in der Realität erschöpft ist. Diese Weg wird zur Übersichtlichkeit gewählt, so dass die Abweichungen in der Kapazität separat dargestellt werden können. Für die variablen Entladeströme wird die Simulation eines vorgegebenen Entladeprofils bis zur Erschöpfung von  $q_1$  ausgeführt. Als Maß zum Vergleich wurde in dieser Arbeit die prozentuale Abweichung gewählt, welche sich wie folgt berechnet:

$$\text{Prozentuale Abweichung} = \frac{|\text{Datenblattwert} - \text{Modellwert}|}{\text{Datenblattwert}} * 100. \quad (5.4)$$

Im Falle einer durchschnittlichen prozentualen Abweichung wurde der Durchschnitt über die jeweils betrachteten prozentualen Abweichungen gebildet.

Insgesamt soll also das Modell bezüglich der Spannung und der entnommenen Ladung unter verschiedenen Lastsituationen untersucht werden.

### 5.3.3. Analyse der Validierungsergebnisse

Im Folgenden werden die Ergebnisse der Analyse der Validierung des *KiBaM*, welches für die Enfilm-Batterie angepasst wurde, vorgestellt. Wie in Abschnitt 5.3.2 werden zuerst die Ergebnisse des Kapazitätsmodells für konstante Entladeströme gezeigt, um den Rate Capacity Effekt im Kapazitätsmodell zu validieren. Dazu zeigt Abbildung 5.5, die mit dem Kapazitätsmodell erzielten Kapazitäten bzw. entnommenen Ladungen im Vergleich zu den realen durch die EnFilm-Batterie gegeben Kapazitäten für die einzelnen konstanten Entladeströme aus dem Datenblatt. Die durchschnittliche prozentuale Abweichung im Vergleich zur Realität beträgt 1.69 %, wobei auch die Daten, welche zur Parameterbestimmung verwendet wurden, bei der Validierung mit berücksichtigt wurden. Ohne diese Daten mit einzubeziehen, ist die durchschnittliche prozentuale Abweichung 4.56 %, wobei hier nur noch die beiden letzten Kurven mit den Entladeströmen von  $10 \mu\text{A}$  und  $1 \mu\text{A}$  zu Validierung zur Verfügung standen. Weiterhin ist zu beachten, dass hier so große Abweichungen vorhanden sind, weil, wie oben schon beschrieben, die beiden Entladeströme, welche die größte entnommene Ladung erzielten, nicht in der Menge der Daten waren, welche die Parameter bestimmt haben. So konnte der Wert von  $q_{max}$  nur unterhalb dieser Werte liegen. Weiterhin beträgt der maximale Fehler 5.83 %, dieser ist auch für die Kurve von  $1 \mu\text{A}$  Entladeströme ermittelt worden. Generell überschätzt das Modell bis auf die beiden Kurven mit den Entladeströmen von  $10 \mu\text{A}$  und  $1 \mu\text{A}$  die Realität. Insgesamt ist aber der Rate Capacity Effekt

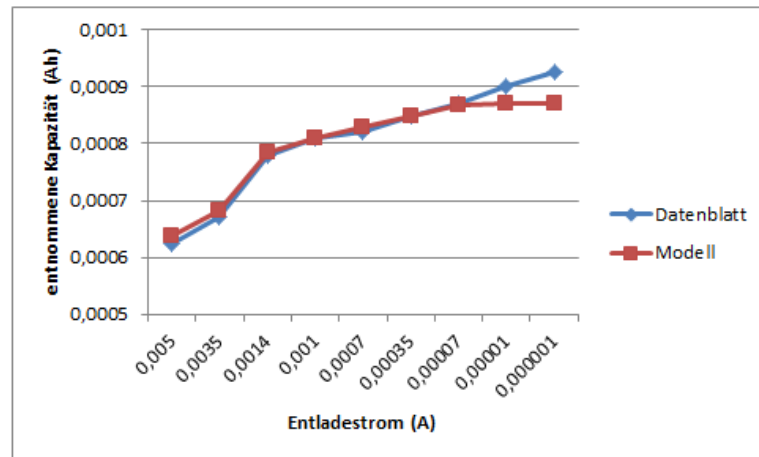


Abbildung 5.5.: Vergleich der entnehmbaren Ladung für verschiedene Entladeströme zwischen dem *KiBaM* und der EnFilm-Batterie.

mit einer durchschnittlichen prozentualen Abweichung von 1.69 % recht gut im Kapazitätsmodell dargestellt.

Bei dem Spannungsmodell beträgt die durchschnittliche prozentuale Abweichung 1.4 %, wobei hier nur die Kurven zur Validierung eingeflossen sind, welche nicht zur Parameterbestimmung verwendet wurden. Dazu zeigen die Abbildungen 5.6, 5.7, 5.8, 5.9, 5.10 jeweils den Verlauf der Spannung für die verschiedenen Entladeströme im Vergleich zu den entsprechenden Spannungsverläufen für die reale EnFilm-Batterie. Zu sehen ist, dass besonders zu Beginn und am Ende einer Entladung die Abweichungen größer ausfallen, als im mittleren Teil des Entladeprozesses. Hier wurden Abweichungen von bis zu 9.5 % verzeichnet, was auch der maximale Fehler unter allen Kurven ist. Wahrscheinlich liegt es daran, dass das *KiBaM* eigentlich für anderen Batterietypen, wie Bleiakkumulatoren, entworfen wurde und diese meist einen flacheren Spannungsverlauf haben [42]. Zeitlich gesehen sind der Anfang und das Ende der Entladung im Verhältnis zum restlichen Teil aber zu vernachlässigen. Dieser Mittelteil wird auch für die meisten Entladeströme gut durch das *KiBaM* dargestellt. Weiterhin kann also gesagt werden, dass auch das Spannungsmodell und somit die angepasste Formel zur Bestimmung von  $X$  den Rate Capacity Effekte recht akkurat darstellen.

Nachdem die Untersuchungen mit konstanten Entladeströmen gezeigt wurden, werden jetzt die Untersuchungen zu variablen Entladeströmen vorgestellt.

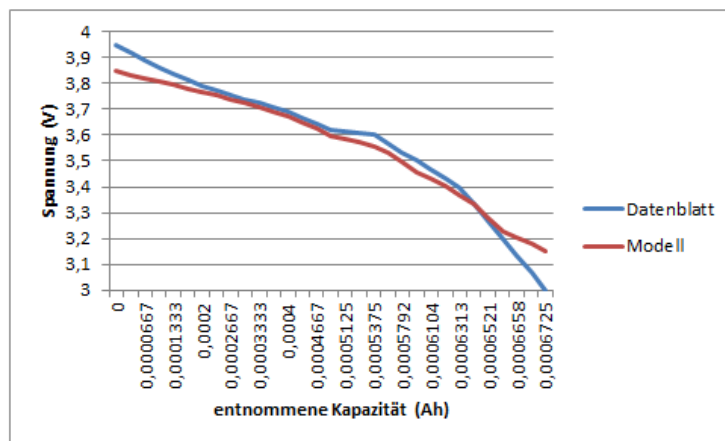


Abbildung 5.6.: Vergleich des Spannungsverlaufs für den Entladestrom von 3.5 mA zwischen dem *KiBaM* und der EnFilm-Batterie.

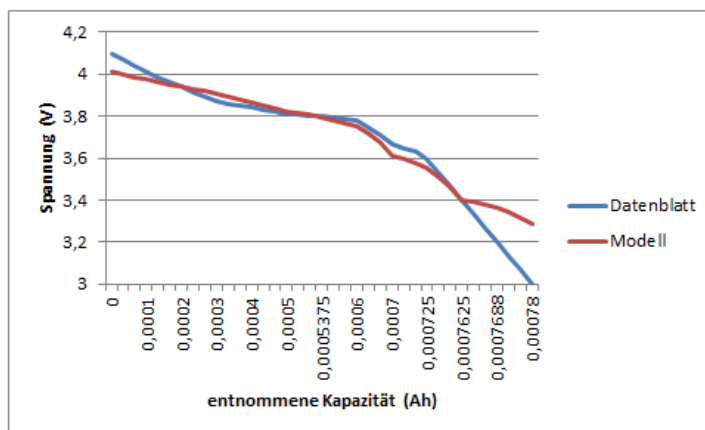


Abbildung 5.7.: Vergleich des Spannungsverlaufs für den Entladestrom von 1.4 mA zwischen dem *KiBaM* und der EnFilm-Batterie.

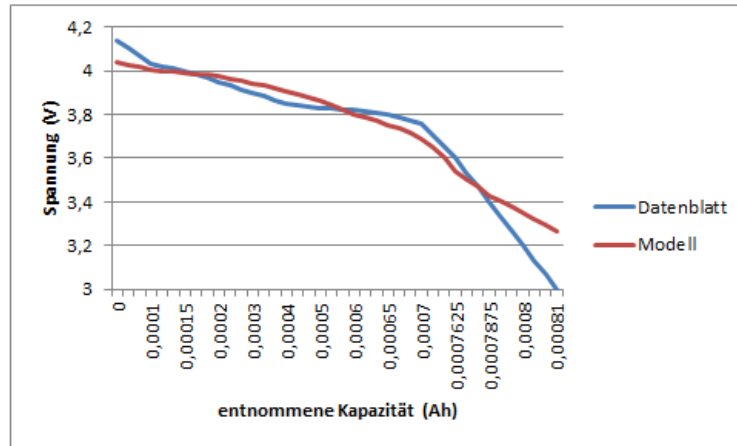


Abbildung 5.8.: Vergleich des Spannungsverlaufs für den Entladestrom von 1 mA zwischen dem *KiBaM* und der EnFilm-Batterie.

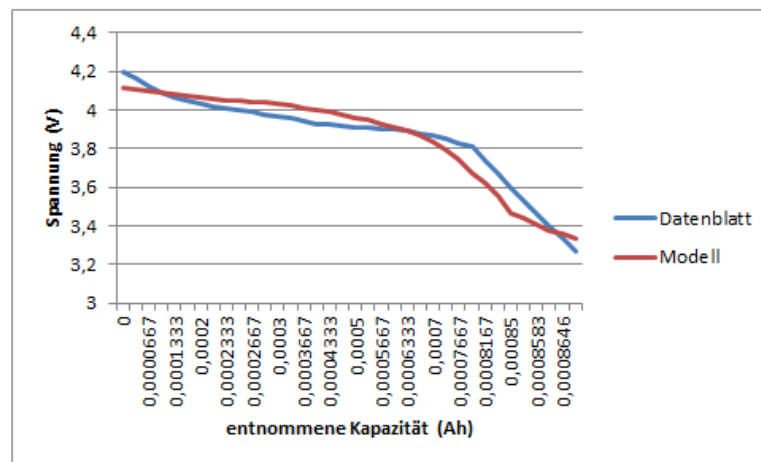


Abbildung 5.9.: Vergleich des Spannungsverlaufs für den Entladestrom von 70 µA zwischen dem *KiBaM* und der EnFilm-Batterie.



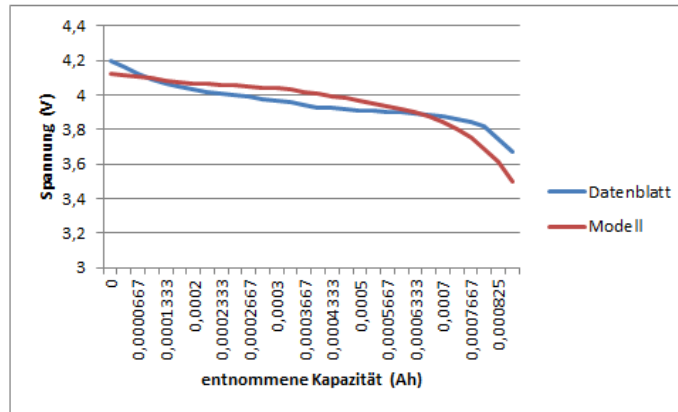


Abbildung 5.10.: Vergleich des Spannungsverlaufs für den Entladestrom von  $1 \mu\text{A}$  zwischen dem *KiBaM* und der EnFilm-Batterie.

Außer der beschriebenen Probleme mit der Temperatur während der Messungen, war das Problem, dass die Batterie sehr empfindlich ist und viele Batterien deshalb im Laufe der Arbeit nicht mehr funktionsfähig waren. Die genaue Ursache hierfür konnte im Laufe der Arbeit nicht in Erfahrung gebracht werden. Angenommen wurde eine falsche Lagerung bei einem zu niedrigen SoC der Batterie. Aufgrund dieser Probleme war es auch im Rahmen der Arbeit nicht möglich, viele Untersuchungen zu variablen Entladeströmen vorzunehmen. Deshalb wurde sich auf zwei Untersuchungen beschränkt. Dabei wurden alle Messungen bei ca.  $25 \text{ }^\circ\text{C}$  durchgeführt. In der ersten Untersuchung wurde die reale Batterie immer im Wechsel bis zur Erschöpfung für eine Minute mit  $5 \text{ mA}$  anschließend für drei Minuten mit  $1 \text{ mA}$  entladen. Dasselbe Muster wurde auch für die Simulation des Modells verwendet.

In der Abbildung 5.11 ist der Spannungsverlauf der Enfilm-Batterie und in der Abbildung 5.12 der Spannungsverlauf, welche mit dem *KibaM* ermittelt wurde, zu sehen. Zu erkennen ist, dass der Spannungsverlauf sowohl des Modells als auch von der Batterie sehr ähnlich verlaufen. Auffällig ist nur, dass die Kurve der Enfilm-Batterie konkav, also nach innen gewölbt ist, hingegen die Kurve vom Modell konvex, also nach außen gewölbt ist. Mit Hilfe des Modells wurde für dieses Muster eine entnommene Ladung von  $0.75 \text{ mAh}$  und mit der Messung an der realen Batterie wurde eine Ladung von  $0.73 \text{ mAh}$  ermittelte. Dies entspricht einer Abweichung von  $2.7 \%$ . Für die Spannung wurde eine durchschnittliche prozentuale Abweichung von  $2.4 \%$  ermittelt.

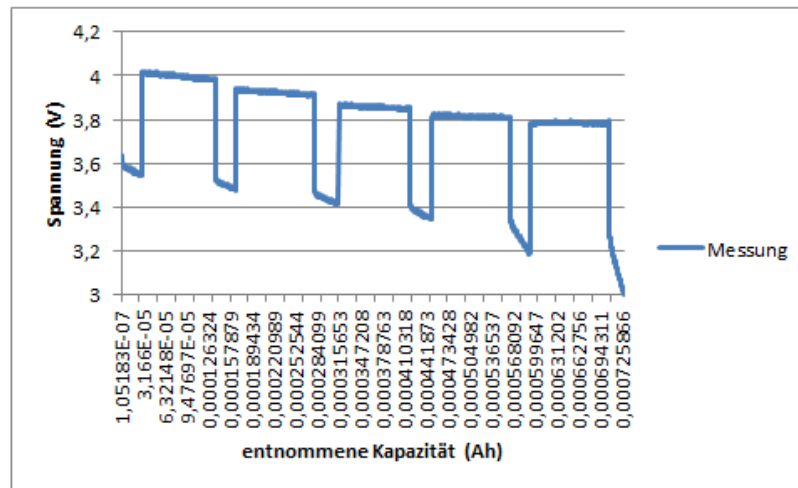


Abbildung 5.11.: Spannungsverlauf der EnFilm-Batterie bei einer Minute mit 5 mA Entladestrom und drei Minuten mit 1 mA Entladestrom im Wechsel bis die Batterie erschöpft ist.

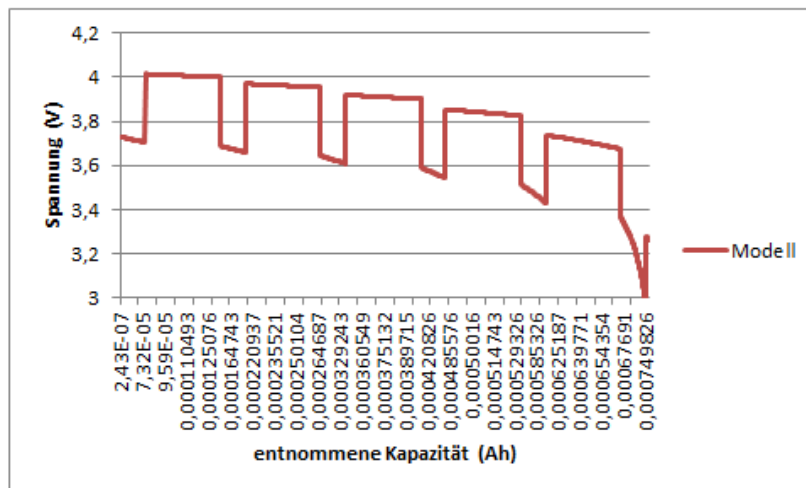


Abbildung 5.12.: Spannungsverlauf des Modells bei einer Minute mit 5 mA Entladestrom und drei Minuten mit 1 mA Entladestrom im Wechsel bis die Batterie erschöpft ist.

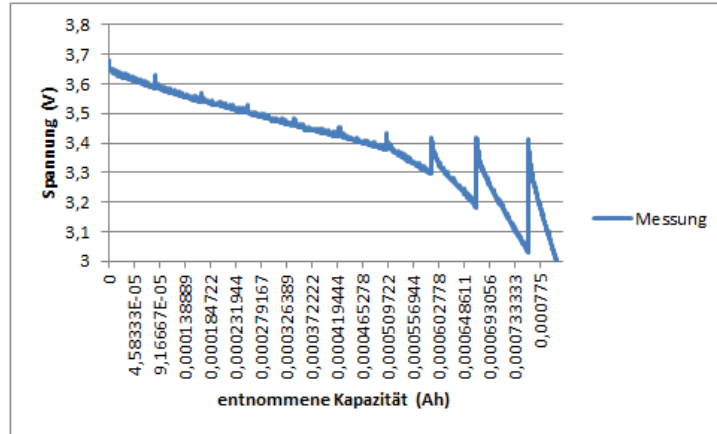


Abbildung 5.13.: Spannungsverlauf der EnFilm-Batterie bei einer Minute mit 5 mA Entladestrom und 1 Stunde Pause im Wechsel bis die Batterie erschöpft ist.

Als weitere Untersuchung wurde die Batterie im Wechsel für eine Minute bei 5 mA entladen und anschließend eine Pause von einer Stunde gemacht. Auch dieses Muster wurde wieder mit dem Modell simuliert. Dieses Muster wurde gewählt, um den Recovery Effekt des Modells mit der realen Batterie zu vergleichen. Dazu ist in der Abbildung 5.13 der Spannungsverlauf für die Enfilm-Batterie und für das Modell der entsprechende Verlauf in der Abbildung 5.14 dargestellt. Dabei sind die Pausen in beiden Spannungsverläufen nicht dargestellt.

Zu sehen ist, dass der Recovery Effekt in Bezug auf die Spannung an der Enfilm-Batterie deutlich ausgeprägter ist. Dies ist aber vor allem darin begründet, dass der Spannungsabfall bevor die Pause einsetzt deutlich größer ist, was sich wahrscheinlich auf den Temperatureinfluss auf die Spannung zurückführen lässt. So erholt sich die Enfilm-Batterie bezogen auf die Spannung maximal um 11 % und im Modell ist die maximale Erholung 3 %. Dabei fand die maximale Erholung in beiden Fällen nach der letzten Pausen statt.

Generell gilt aber, dass auf den reinen Verlauf der Kurven bezogen, der Effekt im Modell ähnlich zur Realität dargestellt wird. So ist der Recovery Effekt in der Realität zu Beginn der Entladung weniger ausgeprägt als am Ende, das spiegelt auch das Modell wieder. Was aber auffällig ist, ist, dass der Verlauf der Kurven

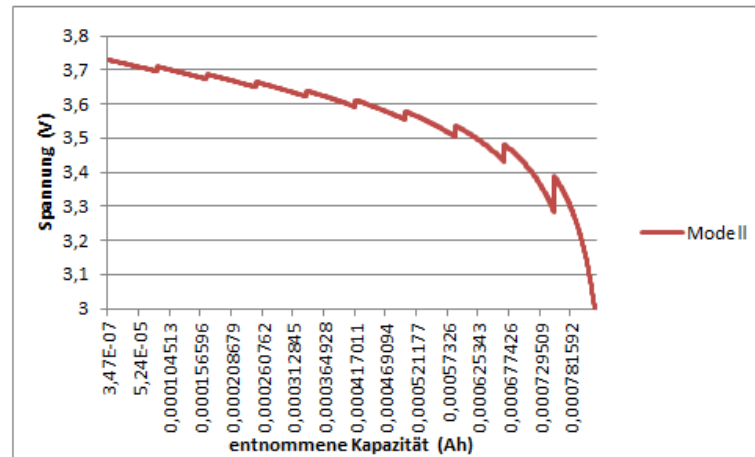


Abbildung 5.14.: Spannungsverlauf des Modells bei einer Minute mit 5 mA Entladestrom und 1 Stunde Pause im Wechsel bis die *KiBaM* erschöpft ist.

nach den Pausen unterschiedlich ist. So verlaufen die Kurve im Modell konvex und die Kurve der EnFilm-Batterie konkav. Weiterhin ist, bezogen auf die Spannung, nach einer Pause, die durchschnittliche prozentuale Abweichung 3 %. Es zeigt sich also, dass die Anpassung der Formel für die Bestimmung von  $X$  dazu geführt hat, dass das Spannungsmodell auch in Bezug auf den Recovery Effekt diesen recht akkurat darstellen kann.

Für die entnommene Ladung unter diesem Entladeprofil wurde für die Enfilm-Batterie ein Wert von 0.805 mA ermittelt und das Modell hingegen sagt einen Wert von 0.0825 mAh voraus. Dies entspricht einer Abweichung von 2.5 %. Dieser Wert bestätigt auch die in der ersten Untersuchung ermittelte Abweichung von 2.7 %.

Im Allgemeinen wäre es nötig, noch weitere Untersuchungen durchführen, um eine aussagekräftige Basis zur Bestimmung der Güte des Modells zu haben. Aber wie schon erwähnt, war dies aufgrund der Probleme mit der Enfilm-Batterie nicht möglich. Dadurch ist es nicht möglich, genaue Maße für das für die EnFilm-Batterie angepasste *KiBaM* zu geben. Es zeichnet sich aber ab, dass das Modell akkurater für konstante als für variable Entladeströme ist. Was weiterhin gesagt werden kann, ist, dass die Heuristiken, welche zur Bestimmung der Parameter sowohl für das Spannungs- als auch für das Kapazitätsmodell verwendet, eine gute Basis bilden um ein akkurates *KiBaM* einer spezifischen Batterie zu erhalten. In diesem Zusam-

menhang ist aber eine gute Datenqualität Grundvoraussetzung. Des Weiteren war es, aber auch nicht das Ziel, das beste Modell, sondern ein gutes Modell zu finden, um den Rahmen dieser Arbeit nicht zu sprengen.

Weiterhin ist zu sehen, dass es wichtig ist, bei der Bestimmung der Parameter darauf zu achten, dass der Bereich von Entladeströmen berücksichtigt wird, mit dem später auch das Modell arbeiten soll. In [41] wird geschildert, dass schon kleinere Veränderungen in den Parametern dazu führen können, dass ermittelte Batterielaufzeiten für ein und denselben Entladestrom stark auseinander laufen können. Dies wurde aber auch schon in Abschnitt 5.2.1 diskutiert. Wichtig ist auch zu bedenken, dass das Kapazitätsmodell die Basis für das Spannungsmodell bildet und so die Genauigkeit des Spannungsmodells maßgeblich beeinflusst wird. Generell müssten auch noch weitere Untersuchungen bezüglich des Recovery Effekts durchgeführt werden, denn so ist es z.B. nicht klar, wie lange der Recovery Effekt nach einem bestimmten Entladestrom andauert und ob er sofort nach Beginn einer Ruhephase (bzw. Phase mit niedrigen Entladestrom) einsetzt. Es könnte also sein, dass z.B. der Fluss zwischen den beiden Behältern noch angepasst werden muss, um das Modell weiter zu verbessern, aber hierfür fehlten im Rahmen dieser Arbeit die Mittel. So wurden in [74] beispielsweise genauere Untersuchungen für eine Ni-MH Batterie durchgeführt und das *KiBaM* entsprechend angepasst. Weiterhin gilt, dass das *KiBaM* keine Temperatur berücksichtigt und die Batterie, welche mit dem Modell modelliert werden soll, stark temperaturabhängiges Verhalten aufweist. Deswegen ist es im grunde genommen nötig, für verschiedene Temperaturen jeweils die Parameter des *KiBaM* zu bestimmen. So gibt es dann für die verschiedenen Temperaturen verschiedene *KiBaM* Ausprägungen. Insgesamt kann aber gesagt werden, dass zumindest auf der Basis der Validierung in dieser Arbeit das Modell recht akkurat ist. Es ist gut dazu geeignet, dem offline Scheduling die richtigen Informationen zumindest darüber zu geben, unter welchen Bedingungen mehr bzw. weniger Ladung aus der Batterie entnommen werden und damit ist der Scheduler zumindest im Verhältnis gesehen in der Lage, zu entscheiden, welche Scheduling besser bezüglich einer batteriegefahren Ablaufplanung als andere sind. Wobei bei den hier gezeigten Untersuchungen zu sehen ist, dass die Vorhersagen nicht nur von der Richtung (Qualität) her stimmten, sondern auch bezüglich der quantitativen Aussagen relativ nahe an der Realität sind.



## 6. Implementierung des batteriegewahren Schedulers

In diesem Kapitel wird die Implementierung des batteriegewahren offline Schedulers beschrieben. Der Scheduler wird dabei nicht von Grund auf neu entwickelt, sondern auf Basis des im Kapitel 3 beschriebenen offline Schedulers von Michael H. Damit aber der offline Scheduler für diese Arbeit verwendet werden kann, sind Änderungen notwendig gewesen. Michael Hesse hatte für seine Arbeit einen Kondensator im Simulator modelliert. So musste zum einen das *KiBaM* in den Simulator eingebaut werden, damit es möglich ist, eine Batterie als Energiequelle zur Bestimmung einer batteriegewahren Ablaufplanung zu berücksichtigen. Zum anderen wurde in dieser Arbeit darauf verzichtet, einen Spannungsregler, wie es Michael Hesse gemacht hatte, zu simulieren. Begründet ist dies dadurch, dass so die Komplexität der Simulation verringert wird. Durch den Verzicht auf einen Spannungsregler ist es aber nicht mehr möglich, den in Kapitel 2 beschriebenen Ansatz der automatenbasierten Hardwaretreiber zu verwenden wie es ursprünglich gedacht war. Dies liegt daran, dass bei diesem Ansatz von einer konstanten Versorgungsspannung ausgegangen wird und das ist durch den Verzicht auf den Spannungsregler nicht mehr der Fall. Im Zusammenhang mit dem automatenbasierten Hardwaretreiber wurde auch darauf verzichtet, den Energieverbrauch der Zustandsübergänge zu betrachten, welche z.B. in einer Peripherie beim Wechsel von einem zum anderen Zustand benötigt wird. Es wurde darauf verzichtet, weil diese Energieverbräuche im Verhältnis zu den restlichen Verbräuchen relativ gering ausfallen. Im Folgenden wird zuerst auf die Architektur des Schedulers bzw. auf den Teil, der für diese Arbeit von Bedeutung ist, eingegangen. Anschließend wird beschrieben wie das *KiBaM* in den offline Scheduler integriert wird und welche Anpassungen bezüglich des Ansatzes der automatenbasierten Hardwaretreiber nötig waren. Ebenso wird beschrieben, welche Annahmen damit verbunden sind.

### 6.1. Architektur des Scheduler

Der grobe Aufbau des offline Schedulers wurde schon in Kapitel 3 beschrieben (s. auch Abbildung 3.1 ), wobei im Folgenden genauer erläutert werden soll, in

welchen Teilen die Änderungen am offline Scheduler durchgeführt worden sind. Der offline Scheduler wurde mit Java implementiert und ist dabei hierarchisch aufgebaut worden. Die bedeutendste Klasse ist die Klasse *Scheduler*. Diese Klasse verwendet genetische Algorithmen, um neue Schedulingentscheidungen zu treffen, welche mit Hilfe des *Simulators* bezüglich eines energiequellengewahren Scheduling bewertet werden. Als Framework für die genetischen Algorithmen wurde *JGAP*<sup>1</sup> verwendet. Der Simulator beinhaltet eine *Plattform* und eine *PowerSupply*. Die *PowerSupply* beinhaltet wiederum eine *EnergySource*. Die *Plattform* beinhaltet die *Moduls* und *Tasks* (s. auch Kapitel 3 Abschnitt 3.1.2.2 und 3.1.2.3). Weiterhin organisiert die *Plattform* die zeitlich korrekte Abarbeitung der *Actions* eines Tasks.

Ändert sich durch eine Aktion in einem Task der Leistungsbedarf, schickt die *Plattform* ein *EnergyConsumptionChangedEvent* an die *PowerSupply*, welche wiederum das Event weiter an die *EnergySource* schickt. Diese verarbeitet das Event, indem der neue Zustand der Energiequelle berechnet wird, dabei wird der Zeitraum seit dem letzten Event bis zu dem Eintreffen eines neuen Events als Zeitraum genommen, über den ein konstanter Leistungsbedarf vorlag. Unter Berücksichtigung dieses Leistungsbedarfs und des Zeitraums wird der neue Zustand der Energiequelle ermittelt. Des Weiteren berechnet die *Plattform* beim Versenden des Events den neuen Leistungsbedarf, da dieser sich mit Eintreten des Events geändert hat. Die Verrechnung des Leistungsbedarfs mit der Energiequelle findet bei Eintritt eines kommenden Events statt. So ist der Zeitraum, bekannt über den die sich der Leistungsbedarf nicht verändert hat. Die eingangs beschriebenen Änderungen betreffen zum einen die *EnergySource* und auch die *Plattform*. So wurde die *EnergySource* dahingehend verändert, dass sie als Energiequelle eine Batterie in Form des *Ki-BaM* modelliert und nicht einen Kondensator. Des Weiteren musste die *Plattform* so angepasst werden, dass nicht mehr beim Auslösen eines Events die Energie für eine Zustandsänderung berücksichtigt wird und anstelle eines Leistungsbedarfs ein berechneter Widerstand zur *EnergySource* geschickt wird. Im folgenden Abschnitt werden die vorgenommenen Änderungen genauer erläutert und begründet.

## 6.2. Integration des Kinetic Battery Model in den offline Scheduler

Wie schon erwähnt, war es aufgrund des Verzichts auf einen Spannungsregler nötig, die beschriebenen DFA-Treiber, welche Hardwarekomponenten als Priced-Time-Automatons definieren, anders zu verwenden als sie ursprünglich gedacht waren. So ist es jetzt erforderlich, bei der Definition eines Moduls anstelle einer Leistungs-

---

<sup>1</sup>Java Genetic Algorithm and Programming



aufnahme für einen Zustand einen Widerstand in Ohm anzugeben, welche die Last dieses Zustandes repräsentiert (vgl. Listing 3.2 und Listing 7.2). Dabei wurde angenommen, dass ein Modul, welches, z.B. eine Peripherie darstellt durch eine Menge von Widerständen, die parallel zur Batterie geschaltet werden, und den einzelnen Zustände der Peripherie entsprechend, abstrahiert werden kann. Im Falle von mehreren Modulen ist es also eine Schaltung mit so vielen Widerständen wie Zuständen in den Modulen, welche einzeln, je nachdem in welchen Zustand sich die Module befinden, parallel zur Batterie geschaltet werden können. Weiterhin ist es nicht mehr möglich, einen Energiebedarf für einen Zustandsübergang anzugeben. Durch diese Änderung war es nötig, in der *Plattform* dafür zu sorgen, dass nicht mehr wie oben beschrieben, nach Eintreten eines Events den neuen Leistungsbedarf durch Aufaddieren aller Leistungen der Zustände in denen sich die Module aktuell befinden, zu berechnen, sondern durch die Berechnung des Gesamtwiderstands aller Widerstände, welche aktive Zustände repräsentieren. Da diese parallel geschaltet sind, geschieht dies nach folgender Formel:

$$\frac{1}{R_{ges}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots, \quad (6.1)$$

mit  $R_{ges}$  für den Gesamtwiderstand und mit  $R_1$ ,  $R_2$  und  $R_3$  für die Widerstände der Zustände, welche aktiv sind. Durch dieses Vorgehen werden die Verbräuche der Tasks also addiert, was aber nicht ganz der Realität entspricht. Weiterhin wird der Sonderfall, dass nur ein Zustand aktiv ist, separat behandelt.

Eine weitere Änderung betraf die *EnergySoucre*. Diese wurde dahingehend verändert, dass sie nicht mehr einen Kondensator sondern mit Hilfe des *KiBaM* eine Batterie modelliert. Dabei modelliert sie nicht irgendeine Batterie, sondern mit den beschriebenen Anpassungen und ermittelten Parametern aus Kapitel 5 die EnFilm-Batterie. Hier wurde nicht auf die Implementierung aus der *Kinetic Battery Model Library* zurückgegriffen, welche zur Validierung verwendet wurde, da diese in Python geschrieben ist. So wurde das Kapazitäts- und Spannungsmodell entsprechend den Formeln aus Kapitel 4 und Abschnitt 4.2.4.3 jeweils in einer Methode implementiert, wobei das Spannungsmodell mit der angepassten Formel für normalisierte entnommene Ladung  $X$  implementiert wurde. Dabei ruft die Methode des Kapazitätsmodells die Methode des Spannungsmodells auf. Weiterhin wurden die Parameter für das Kapazitäts- und Spannungsmodell als globale Variablen zur Verfügung gestellt. Dieser Weg wurde deshalb gewählt, weil selten Änderungen an Parametern nötig sind. Damit ist es jetzt möglich, für einen gegebenen Entladestrom  $I$  und eine Entladezeit  $t$  die neuen Zustände der Behälter für die gebundene bzw. frei verfügbare Ladung zu berechnen und die entsprechend veränderte Spannung, wie schon im vorherigen Kapitel beschrieben wurde. Eine dritte Methode (*stateConsumption*) steuert nach Eintritt eines Events mit Hilfe der beiden Me-

thoden für das Kapazitäts- und Spannungsmodell die Abrechnungen der Last für den Zeitraum des Eintritts des vorherigen Events bis zum Zeitpunkt des Events, welches den Aufruf der Methode *stateConsumption* ausgelöst hat. Da aber durch die Veränderungen in den automatenbasierten Hardwaretreibern und in den *Plattformen* nur ein Gesamtwiderstand für den entsprechenden Zeitraum bekannt ist und die Schnittstelle des Kapazitätsmodells einen Entladestrom benötigt, ist es in der Methode *stateConsumption* nötig, über die Formel:

$$I = \frac{U}{R}, \quad (6.2)$$

mit  $I$  für Stromstärke (Entladestrom),  $U$  für die Spannung und  $R$  für den Widerstand (Gesamtwiderstand) den Entladestrom zu berechnen. Unter der Annahme, dass die Spannung über den gesamten Zeitraum konstant wäre, für den der berechnete Gesamtwiderstand anliegt, wäre auch der Entladestrom konstant. Da dies bei einer realen Batterie nicht der Fall ist, würde es zu großen Ungenauigkeiten in den Berechnungen der Batterielaufzeit kommen. Daher wurde der Weg gewählt, den Zeitraum, über den der berechnete Gesamtwiderstands anliegt, in kleine gleich große Zeitschritte zu zerlegen, um die Genauigkeit zu erhöhen. Dabei kann hier ein beliebiger Wert, je nach gewünschter Genauigkeit, gewählt werden. So wird jetzt für jede dieser Zeitspannen ein neuer Entladestrom auf der Grundlage der Spannung des vorherigen Zeitschritts ermittelt. Anschließend wird das Kapazitätsmodell mit diesem Entladestrom und einem Zeitschritt ausgeführt und damit auch das Spannungsmodell. Dadurch werden die neuen Behälterfüllstände und die neue Spannung berechnet. Dies wird solange durchgeführt bis alle Zeitschritte des Zeitraums durchlaufen wurden oder die Batterie laut *KiBaM* erschöpft ist. Ist die Batterie laut *KiBaM* nicht erschöpft, wird das gleich nach Eintreten eines neuen Events durchgeführt bis schlussendlich die Batterie erschöpft ist, also der Behälter der frei verfügbaren Ladung leer ist. Ist die Batterie laut Modell erschöpft, wird dies dem Scheduler über eine Ausnahme mitgeteilt. Neben diesen Methoden musste noch die Möglichkeit geschaffen werden, das Modell mit den entsprechenden Füllständen der Behälter zu initialisieren bzw. zu reseten, damit der Scheduler nach einer Simulation das Batteriemodell zurücksetzen kann, um ein neues Scheduling simulieren zu können.

Neben dieser Implementierung von *KiBaM* gibt es noch eine Implementierung, welche genauso aufgebaut ist. Sie lässt sich direkt zur Simulation unabhängig vom *Scheduler* verwenden und wird im folgenden Kapitel zur Simulation von Referenzsystemen dienen.

# 7. Evaluation

In diesem Kapitel wird die Bestimmung von batteriegefahren Ablaufplanungen mit Hilfe des beschriebenen offline Schedulers und somit indirekt auch die Modellierung von Lithium-Dünnschichtbatterien für die Ablaufplanung bewertet. Dementsprechend wird nachfolgend erst die Zielsetzung der Evaluation beschrieben. Hier wird dargestellt, was mit der Evaluation erreicht werden soll. Anschließend wird das Evaluationsverfahren beschrieben, mit dem die Bestimmung von batteriegefahren Ablaufplanungen unter Zuhilfenahme des beschriebenen offline Schedulers bewertet werden soll. Im letzten Abschnitt werden die Ergebnisse der Evaluation sowohl vorgestellt als auch analysiert.

## 7.1. Zielsetzung der Evaluation

Im Allgemeinen ist Ziel der Evaluation, die Frage zu beantworten, ob sich eine batteriegefähre Ablaufplanung, welche über den offline Scheduler bestimmt wurde, gegenüber einer naiven Ablaufplanung überhaupt lohnt? Dabei ist eine naive Ablaufplanung von Tasks, eine Planung, bei der die Batterie als ideales Bauteil betrachtet wird und deren Eigenschaften, wie der nichtlinearen Effekte, außen vorge lassen werden. In diesen Zusammenhang soll auch gezeigt werden, ob der gewählte Ansatz des offline Schedulers geeignet ist, um eine batteriegefähre Ablaufplanung zu bestimmen. Weiterhin soll geklärt werden, ob sich das gewählte Batteriemodell für diese Zwecke eignet. Ferner sollen auch die Grenzen des offline Schedulers aufgezeigt werden.

## 7.2. Vorstellung des Evaluationsverfahrens

Wie schon bei der Zielsetzung der Evaluation erwähnt, soll untersucht werden, ob sich eine batteriegefähre Ablaufplanung gegenüber einer naiven Ablaufplanung rentiert. Um dies zu zeigen, wurden batteriegefähre Ablaufplanungen, welche der offline Scheduler erzeugt hat, mit naiven Ablaufplanungen verglichen. Um dies zu bewerkstelligen, wurden verschiedene Systeme definiert, auf denen Tasks geschuldet werden. Weiterhin ist es für den offline Scheduler notwendig, dass die Systeme in

Parameter	Wert
Populationsgröße	250
Anzahl Generationen	100
Kreuzungsrate	35 %
Mutationsrate	15 %

Tabelle 7.1.: Verwendete Einstellungen des offline Schedulers in der Evaluation.

der Form wie sie in Kapitel 3 beschrieben sind, als XML-Definition modelliert werden müssen. Deshalb wird diese Art der Definition auch in der Evaluation dazu benutzt, um die Systeme zu beschreiben. Des Weiteren ist es notwendig, naive Ablaufplanungen zu definieren. Generell wurde sich dazu entschieden, eine Evaluation anhand einfacher Systeme durchzuführen, um die gewonnenen Ergebnisse besser deuten zu können.

Für den offline Scheduler wurden für alle in dieser Evaluation ermittelten Ablaufplanungen die Einstellungen verwendet, welche in der Tabelle 7.1 abgebildet sind. Weiterhin wurde zur Ermittlung alles Scheduling der offline Scheduler 10-mal ausgeführt und das beste Ergebnis dieser 10 Durchläufe genommen. Ferner wurden alle naiven Ablaufplanungen für diese Evaluation mit dem *KiBaM* simuliert. Die Maße, mit denen die Ablaufplanungen verglichen wurden, sind die Betriebszeit und die Anzahl von durchgeführten Aufgaben. Insgesamt wurde also ein Vergleich auf Simulationsergebnissen durchgeführt.

Ein erstes System, mit dem experimentiert wurde, besteht nur aus einem Task und einem Modul. Es ist deshalb ein einfaches System gewählt worden, um einen besseren Überblick über die Abläufe im System zu haben. Dadurch ist es bei der späteren Auswertung einfacher, Ergebnisse zu deuten. Die dazugehörige Systemdefinition ist in Listing 7.1 dargestellt.

Listing 7.1: XML-Definition des Systems mit einem Modul und einem Task.

```
<systemdefinition>
  <platform>
    <name>Enfilm_system</name>
    <modules>
      <path>Transmit_Module.xml</path>
    </modules>
    <tasks>
      <path weight="1">transmit_Task.xml</path>
    </tasks>
  </platform>
  <powersupply>
```

```

        <source type=" battery " />
    </powersupply>
</systemdefinition>

```

Das Modul des Systems besteht aus zwei Zuständen. Einem mit einem sehr hohen Widerstand und einem mit einem geringen Widerstand. So stellt der Zustand mit dem hohen Widerstand ein Idle-Zustand<sup>1</sup> dar und der andere Widerstand den Zustand, in dem sich die Hardware des Systems befindet, wenn ein Funk-Chip eine Übertragung durchführt. Für dieses Modul ist in Listing 7.2 die entsprechende Definition in XML dargestellt.

Listing 7.2: XML-Definition des Funk-Moduls mit den zwei Zuständen Idle und Transmit.

```

<module>

  <name>Transmit_Module</name>
  <initialstate>IDLE</initialstate>

  <states>
    <state>
      <name>IDLE</name>
      <resistance>210000</resistance>
    </state>
    <state>
      <name>TRANSMIT</name>
      <resistance>840</resistance>
    </state>
  </states>

  <transitions>
    <transition>
      <name>transmitOn</name>
      <srcstate>IDLE</srcstate>
      <deststate>TRANSMIT</deststate>
    </transition>
    <transition>
      <name>idle</name>
      <srcstate>TRANSMIT</srcstate>
      <deststate>IDLE</deststate>
    </transition>
  </transitions>

</module>

```

<sup>1</sup>Angenommen wird, dass dieser Zustand den Idle des Gesamtsystem darstellen soll.

Der Task des Testsystems ist dabei so definiert, dass er für eine Sekunde eine Übertragung durchführt, also wechselt die Hardware in den Zustand *Transmit*. Nach dieser Sekunde veranlasst der Task, dass die Hardware wieder in den Zustand *Idle* zurückkehrt<sup>2</sup>. Danach legt sich der Task schlafen. Wie lange er sich schlafen legt, hängt von der spezifischen Ablaufplanung ab. Dies wird zum einen durch den offline Scheduler und zum anderen durch eine naive Ablaufplanung bestimmt. Im Task vorgegeben ist, dass dieser mindestens einmal am Tag laufen bzw. übertragen muss. Diese Einschränkung dient nur dem offline Scheduler und wurde so großzügig gewählt, damit der offline Scheduler genügend Freiheiten für die Ermittlung einer batteriegefahrenen Ablaufplanung hat. In Listing 7.3 wird der beschriebene Task auch wieder als XML definiert.<sup>3</sup>

Listing 7.3: XML-Definition des Funk-Task.

```
<task>
  <name>Transmit_Task</name>
  <program>
    <pause minduration="0" maxduration="86400000000"
      label="pause_vor_transmitOn" />
    <changestate module="transmit_Module"
      transition="transmitOn" />
    <delay duration="1000000" />
    <changestate module="transmit_Module" transition="idle" />
  </program>
</task>
```

Ein weiteres System, welches zur Evaluation verwendet wurde, ergänzt das erste System um ein Modul, welches einen Sensor darstellen soll, und einen weiteren Task, welcher den Sensor abfragt. In Listing 7.4 ist die zugehörige Definition in XML zusehen.

Listing 7.4: XML-Definition des System mit zwei Modulen und zwei Tasks.

```
<systemdefinition>
  <platform>
    <name>Enfilm_system2</name>
    <modules>
      <path>Transmit_Module.xml</path>
      <path>Measure_Module.xml</path>
    </modules>
  </platform>
</systemdefinition>
```

<sup>2</sup>Dieser Fall wird im Folgendem dadurch bezeichnet, dass der Task sich in einer Idle-Phase befindet.

<sup>3</sup>Alle Angaben sind dabei in Mikrosekunden.

```

    </modules>
    <tasks>
      <path weight="1">Transmit_Task.xml</path>
      <path weight="1">Measure_Task.xml</path>
    </tasks>
  </platform>

  <powersupply>
    <source type="battery" />
  </powersupply>
</systemdefinition>

```

Das zusätzliche Modul besteht dabei auch wieder aus zwei Zuständen. Einem Zustand, der ein Messen eines Sensors entspricht und einem Zustand, der dem abgeschalteten Sensor entspricht. Dazu enthält Listing 7.5 eine entsprechende Definition in XML, wobei der Widerstand von Null einem unendlich hohen Widerstand entspricht, um so den abgeschalteten Sensor darzustellen.

Listing 7.5: XML-Definition des Mess-Moduls mit den zwei Zuständen Off und Measure.

```

<module>

  <name>Measure_Module</name>
  <initialstate>OFF</initialstate>

  <states>
    <state>
      <name>OFF</name>
      <resistance>0</resistance>
    </state>
    <state>
      <name>MEASURE</name>
      <resistance>4200</resistance>
    </state>
  </states>

  <transitions>
    <transition>
      <name>measureOn</name>
      <srcstate>OFF</srcstate>
      <deststate>MEASURE</deststate>
    </transition>
    <transition>
      <name>off</name>
      <srcstate>MEASURE</srcstate>
      <deststate>OFF</deststate>
    </transition>
  </transitions>

```

```

        </transition>
    </transitions>
</module>

```

Der zusätzliche Task ist dabei analog zum ersten Task aus Listing 7.3. Nur dass er das Mess-Modul ansteuert und ist in Listing 7.6 beschrieben.

Listing 7.6: XML-Definition des Mess-Tasks.

```

<task>

    <name>Transmit_Task</name>

    <program>
        <pause minduration="0" maxduration="8640000000"
            label="pause_vor_measureOn" />
        <changestate module="Measure_Module" transition="measureOn" />
        <delay duration="1000000" />
        <changestate module="Measure_Module" transition="off" />
    </program>

</task>

```

Damit naive Ablaufplanungen angeben und auch batteriegewahre Ablaufplanung bestimmt werden können, ist es notwendig, eine Zielsetzung zu definieren, welche die Ablaufplanungen verfolgen sollen. Dadurch ist es erst möglich, die Ablaufplanung unter der gegebenen Zielsetzung zu vergleichen. Hierzu werden verschiedene Zielsetzungen, welche immer mehr Komplexität in die Ablaufplanung bringen, definiert. Diese werden im Folgenden mit einer entsprechenden naiven Ablaufplanung angegeben.

Für die erste Stufe der Untersuchungen wurde die Zielsetzung definiert, dass die Durchläufe des Tasks aus Listing 7.3 maximiert werden sollen. Eine denkbare naive Ablaufplanung wäre es, den Tasks ohne Pausen hintereinander auszuführen. Für den offline Scheduler wurde die Fitnessfunktion *Weightedtaskcount* verwendet, welche die Anzahl der Tasksdurchläufe maximiert.

In einer zweiten Stufe wurde die Zielsetzung dahingehend verändert, dass das System eine Stunde am „Leben“ sein und dabei die Anzahl der Durchläufe des Tasks maximiert werden sollen. Diese Zielsetzung entspricht dabei auch mehr einem realen Anwendungsfall als die erste Zielsetzung, da oft bekannt ist, wie lange ein System laufen soll. Hierzu wurde für den offline Scheduler auf die Fitnessfunktion *Targetlifetime* zurückgegriffen um eine entsprechende Ablaufplanung zur ermitteln. Diese Fitnessfunktion erlaubt es, eine anvisierte Betriebszeit vorzugeben und dabei die Anzahl von Tasksdurchläufen innerhalb der Zeit zu maximieren. Eine



denkbare naive Ablaufplanung wäre eine Gleichverteilung der Taskausführungen über eine Stunde. Der Zeitraum von einer Stunde wurde aufgrund der geringen Kapazität der Batterie im Verhältnis zu den Lasten, den die Tasks aus Listing 7.3 verursacht, gewählt. Für die theoretisch maximale Anzahl von Durchläufen unter Berücksichtigung der Kosten für den Idle-Zustand des Tasks wurde folgenden Formel angenommen:

$$N = \frac{E_{batt} - E_{IdleProTag}}{E_{Transmit} - E_{IdleProTransmit}} \quad (7.1)$$

mit

- $E_{batt}$  entspricht der Energie der Batterie,
- $E_{IdleProTag}$  entspricht dem Energieverbrauch von einer Stunde im Zustand Idle,
- $E_{Transmit}$  entspricht dem Energieverbrauch einer Transmit-Operation (Zustand Transmit),
- $E_{IdleProTransmit}$  entspricht den Kosten des Idle-Zustands die durch  $E_{IdleProTag}$  pro Transmit-Operation zu viel abgezogen wurden.<sup>4</sup>

Dabei wurde für die Bestimmung von  $E_{batt}$  eine Spannung von 4.2 V und eine Ladung von 1,1 mA angenommen, da dies der größte aus dem Datenblatt zu findende Wert war. Weiterhin wurde für die restlichen Kosten eine Spannung von 3.0 V angenommen. Weiterhin muss  $N$  im Falle einer ungeraden Zahl abgerundet werden, weil nur ganze Transmit-Operationen durchgeführt werden können. Mit dieser Formel ist eine Obergrenze für die maximale Anzahl von Durchläufen eines Tasks unter Beachtung der Ladung einer Batterie gegeben. Die Gleichverteilung der Tasks über eine Ausführung von einer Stunden ist dann mit folgender Formel gegeben:

$$T_{Idle} = \left( \frac{1h}{N} \right) - t_{Transmit}, \quad (7.2)$$

mit  $t_{Transmit}$  für die Zeit die eine Transmit-Operationen braucht.  $T_{Idle}$  beschreibt dabei die Länge einer Idle-Phase.

Für eine letzte Untersuchung wurde das zweite System aus Listing 7.4 verwendet. Als Zielfunktion wurde die vorherige Zielfunktion verwendet, nur das jetzt beide Tasks berücksichtigt werden sollen. Also soll auch für dieses System eine Ablaufplanung gefunden werden, die das System eine Stunden am „Leben“ hält. Dabei soll

---

<sup>4</sup>Dies ist nötig, weil immer nur ein Zustand zur selben Zeit aktiv sein kann.

Transmit (s)	Idle (s)
Kontinuierlich	0
0.001	0.001
1	1
1	2
1	4

Tabelle 7.2.: Die für verschiedene Frequenzen verwendete Zeiten.

die Anzahl Durchläufe beider Tasks maximiert werden. Zum Vergleich mit einer ermittelten Ablaufplanung des offline Schedulers wurde eine naive Ablaufplanung analog zu der vorherigen angenommen. Nur, dass die Tasks dabei parallel ausgeführt wurden. Diese Annahme entspricht nicht ganz der Realität, weil Tasks nicht exakt parallel ausgeführt werden können. Darüber hinaus lässt sich der Energieverbrauch der beiden Tasks nicht durch Addition der einzelnen Verbräuche berechnen. Dies wurde nur zur Vereinfachung angenommen. Auch hierbei verwendet der offline Scheduler wieder die Fitnessfunktion *Targetlifetime*, um ein entsprechendes Scheduling zu finden.

Bevor aber ein Vergleich von der naiven Ablaufplanung mit der batteriegewahren Ablaufplanung durchgeführt wird, soll untersucht werden, welchen ein Einfluss verschiedene Frequenzen auf die Belastung einer Batterie und verschiedene Längen der Pausen (bzw. Idle-Phasen) auf die, aus einer Batterie entnehmbare Ladung haben. Damit soll der Einfluss der Verteilung von Tasks in einer Ablaufplanung auf die Batterielaufzeit verdeutlicht werden. Dazu werden verschiedene Szenarien mit dem *KiBaM* simuliert bis die Batterie laut *KiBaM* erschöpft ist. Diese Szenarien bestehen darin, dass *KiBaM* mit zwei unterschiedlichen Lasten im Wechsel belastet werden. Dabei wurden zur besseren Vergleichbarkeit mit anderen durchgeführten Untersuchungen die Widerstände aus dem Modul, welches in Listing 7.2 definiert ist, verwendet. Die werden entsprechend dem Zustand, den sie repräsentieren, bezeichnet. In Tabelle 7.2 sind die entsprechenden Zeiträume, in denen sich in den entsprechenden Zuständen aufgehalten wird, in Sekunden angegeben.

### 7.3. Analyse der Evaluationsergebnisse

Im Folgenden sollen die Ergebnisse der Evaluation vorgestellt und analysiert werden. Wie im vorherigen Abschnitt beschrieben, werden zuerst die Ergebnisse der Untersuchungen bezüglich des Einflusses der Frequenz der Belastung einer Batterie und der Einfluss von verschiedenen langen Idle-Phasen niedrigerer Belastung auf die entnehmbare Ladung aus einer Batterie untersucht. Die Ergebnisse lassen

Measure (s)	Idle (s)	Entladezeit (h)	Entnommene Ladung (mAh)
Kontinuierlich	0	0.15	0.66
0.001	0.001	0.31	0.73
1	1	0.33	0.75
1	2	0.53	0.79
1	4	0.91	0.82

Tabelle 7.3.: Die für verschiedene Frequenzen ermittelten Entladezeiten und entnommene Ladungen.

klar erkennen, dass sowohl die Frequenz als auch die Länge der Idle-Phase einen maßgeblichen Einfluss auf die entnommene Ladung aus der Batterie haben. In der Tabelle 7.3 sind die mit *KiBaM* ermittelten Werte für die untersuchten Entladeprofile dargestellt. Es ist zu sehen, dass sowohl eine geringere Frequenz als auch eine Vergrößerung der Idle-Phasen dafür sorgt, dass sich die entnommene Ladung erhöht. Hierzu wurde in den Untersuchungen eine Steigerung der entnommenen Ladung von 24 % ermittelt. So zeigt sich also, dass eine Abhängigkeit zwischen der entnommenen Ladung und der Verteilung von Idle-Zeiten existiert. Generell gilt, dass sich bei einer Verringerung der Frequenz und einer Vergrößerung der Idle-Dauer die entnehmbare Ladung vergrößert. Dies zeigen auch Rao et al. in einer ähnlichen Untersuchung [74]. Die Ergebnisse deuten also daraufhin, dass es auch in der Ablaufplanung Pläne gibt, die die Batterie aufgrund der Frequenz mit der die Batterie belastet wird und auch der Länge der Idle-Phasen effizienter ausnutzen.

Nachfolgend werden die Ergebnisse für den Vergleich der naiven und batterie-gewahren Ablaufplanung mit der Zielsetzung, dass die Durchläufe der Tasks aus Listing 7.3 maximiert werden, dargelegt. Dabei ist der naive Ablaufplan, wie oben beschrieben. Der batterie-gewahre Ablaufplan mit dem offline Scheduler und der Fitnessfunktion *Weightedtaskcount* bestimmt. Weiterhin wurde dazu das System aus 7.1 verwendet. Die naive Ablaufplanung erreicht dabei 556 Durchläufe für den Task. Dagegen erreicht die batterie-gewahre Ablaufplanung, die der offline Scheduler ermittelt hat, 665 Durchläufe für den Task. Das ist eine Steigerung von 19 % in der Anzahl der Taskdurchläufe. Diese Steigerung ist größtenteils auf den Recovery Effekt zurückzuführen, von dem der naive Ablaufplan nicht profitiert, da er keine Idle-Phasen zwischen den einzelnen Durchläufen hat<sup>5</sup> Weiterhin führt eine permanent anliegende hohe Last schneller dazu, dass die Batterie erschöpft ist und nicht die komplette, in der Batterie enthaltene Ladung, entnommen werden

<sup>5</sup>Diese Annahme gilt nur wenn die durch den Task veranlassten Zustandswechsel vom dem Zustand Transmit zum Zustand Idle und wieder zurück in den Zustand Transmit in Null Zeit geschehen, was aber nicht wirklich realistisch.

kann (Rate Capacity Effekt). Es liegt, wie schon erwähnt, daran, dass die chemisch gebundene Ladung nicht mit der Geschwindigkeit als frei verfügbare Ladung zur Verfügung steht wie Ladung entnommen wird. So kommt der naive Plan auf eine Ausführungszeit von neun Minuten und die batteriegewahre Ablaufplanung auf 94 Minuten, da diese jeweils pro Taskdurchlauf ca. acht Sekunden im Idle-Zustand ist. Obwohl diese Pause relativ klein ist, führt sie laut *KiBaM* zu einer Erholung der Batterie. Das ist wahrscheinlich darauf zurückzuführen, dass die EnFilm-Batterie, welche *KiBaM* abbildet, die Vorteile eines Kondensators und einer Batterie vereint. Weiterhin beträgt die entnommene Ladung aus der Batterie für den naiven Ansatz ca. 0.66 mAh und für die batteriegewahre Ablaufplanung ca. 0.87 mAh. Die batteriegewahre Ablaufplanung hat also so gut wie die maximale Kapazität laut *KiBaM* aus der Batterie entnehmen können. Auch hier spiegeln sich die zu Beginn beschriebenen Einflüsse der Frequenz und der Länge der Pausendauer auf die entnehmbare Ladung wieder. Die erste Untersuchung zeigt also, dass es mit Hilfe des offline Schedulers unter der gegebenen Zielfunktion möglich ist, eine effizientere Ablaufplanung, in Bezug auf eine Batterie, als es durch eine naive Ablaufplanung der Fall ist, zu finden.

Aber auch die Ergebnisse der zweiten Untersuchung mit der Zielsetzung, dass das System eine Stunde am „Leben“ sein soll und dabei die Anzahl der Durchläufe des Tasks maximiert werden sollen, zeigen ähnlich gute Ergebnisse für den offline Scheduler im Verhältnis zum naiven Vorgehen. Für die naive Ablaufplanung wurde mit der obigen Formel 7.1 ein Wert von 1544 für die theoretisch maximale Anzahl von Durchläufen und damit über die Formel 7.2 eine Länge von 1.3 Sekunden für die Idle-Phase ermittelt. Nach der Simulation dieser Ablaufplanung mit dem *KiBaM* wurde eine Betriebszeit von 24 Minuten und eine Anzahl von 619 Durchläufen des Tasks berechnet. Damit hat der naive Ablaufplan das Ziel von 60 Minuten Betriebszeit um 60 % verfehlt. Mit Hilfe der ermittelten batteriegewahren Ablaufplanung des offline Schedulers wurde das Ziel von 60 Minuten Betriebszeit erreicht, d.h. das System konnte 60 Minuten in Betrieb gehalten werden bevor die Batterie erschöpft war. Weiterhin wurde mit dieser Ablaufplanung eine Anzahl von 661 Durchläufen der Tasks bestimmt. So hat die ermittelte Ablaufplanung des Schedulers im Gegensatz zum naiven Ansatz nicht nur das Ziel von 60 Minuten erreicht, sondern auch noch mehr Durchläufe erreicht. Dies ist ein Gewinn von 7 % in Bezug auf die Anzahl der Taskdurchläufe. Die Länge der Idle-Phase legte der Scheduler, dabei auf 4.5 Sekunden fest. Auch diese Ergebnisse des Vergleichs lassen sich wieder mit den eingangs beschriebenen Untersuchungen begründen, denn da zeigten die Ereignisse, dass eine größere Idle-Phase zu mehr Erholung (Recovery Effekt) führt. Dies spiegelt sich auch in den ermittelten Werten für die entnommene Ladung wieder, mit 0.76 mAh für die naive Ablaufplanung und 0.87 mAh für den vom offline Scheduler bestimmten Ablaufplan.

Im Zusammenhang mit dieser Untersuchung wurde mit Hilfe des offline Schedulers und der Fitnessfunktion *Targetlifetime* innerhalb der theoretischen Grenzen (obere Schranken) für die Betriebszeit und Anzahl der Tasksdurchläufe in Schritten von einer Stunde, die entsprechenden Werte für die Betriebszeit und Anzahl der Tasksdurchläufe für das System aus Listing 7.1 ermittelt. Die entsprechenden Werte sind in der Abbildung 7.1 dargestellt (Pareto-Front). Wobei nur die Werte bis 42 Stunden in der Abbildung zu sehen sind, da für die Stunden 43-46 sowohl für die Betriebszeit als auch für Anzahl der Tasksdurchläufe sich keine Veränderungen in den Werten ergeben haben. Auch mit Hilfe der Fitnessfunktion *Lifetime*, welche die Betriebszeit maximiert, durchgeführten Untersuchungen lieferten keine Ablaufplanungen mit einer noch längeren Betriebszeit. Angenommen wurde hier, dass das Verfahren hier an seine Grenzen gestoßen ist. Die maximale Anzahl von Tasksdurchläufen, welche der offline Scheduler berechnet hatte, sind 665 (s. erste Untersuchung). Für die obere Schranke der Betriebszeit wurde angenommen, dass sich das System nur im Idle-Zustand befindet und für die obere Schranke der Anzahl der Tasksdurchläufe wurde die Formel 7.1 ohne Betrachtung der Kosten des Idle-Zustands verwendet. Dies führt bei der Betriebszeit zu einem Wert von 46 Stunden und für die Anzahl der Tasksdurchläufe zu einem Wert von 1544 Durchläufen. Gut zu sehen ist an den ermittelten Werten des offline Schedulers, dass eine der beiden Größen immer auf Kosten der anderen Größe wächst. Der Abfall der Punkte von links und rechts ist dadurch begründet, dass die Kosten für die Idle-Phasen immer mehr ins Gewicht fallen. Weiterhin ist in der Abbildung auch der Wert der naiven Ablaufplanung aus der zweiten Untersuchung in blau eingezeichnet zu sehen.

Für die dritte Untersuchung wurde das zweite System aus Listing 7.4 genommen und die Zielsetzung aus der vorherigen Untersuchung. Nur, dass dieses Mal zwei Tasks berücksichtigt werden. Die Ergebnisse dieser Untersuchung zeigen, dass der offline Scheduler auch für etwas komplexere Systeme ein, in Bezug auf die Batterie und naive Ablaufplanung, eine effizientere Ablaufplanung finden kann. Für die naive Ablaufplanung wurde, wie oben beschrieben, angenommen, dass die beiden Tasks parallel ausgeführt werden und die Verteilung der parallelen Tasksausführungen, entsprechend analog zu der vorherigen Untersuchung, ermittelt werden. So wurde ein Wert von 1258 für die theoretisch maximale Anzahl von Durchläufen ermittelt und für die Länge der Idle-Phase zwischen der erneuten Ausführung beider Tasks eine Länge von fünf Sekunden festgelegt. Mit diesem Ablaufplan wurde eine Anzahl von 522 Ausführungen und eine Laufzeit von 24 Minuten erreicht. Die ermittelte Ablaufplanung des offline Schedulers für dieses Szenario erreichte eine Anzahl von 574 für den Mess-Task und 544 für den Funk-Task. Weiterhin wurde auch wieder das Ziel von 60 Minuten erreicht. Mit Zeiten für die Pausen von 5,3 Sekunden für den Mess-Task und von 5,7 für Funk-Task. So wurde mit dem naiven

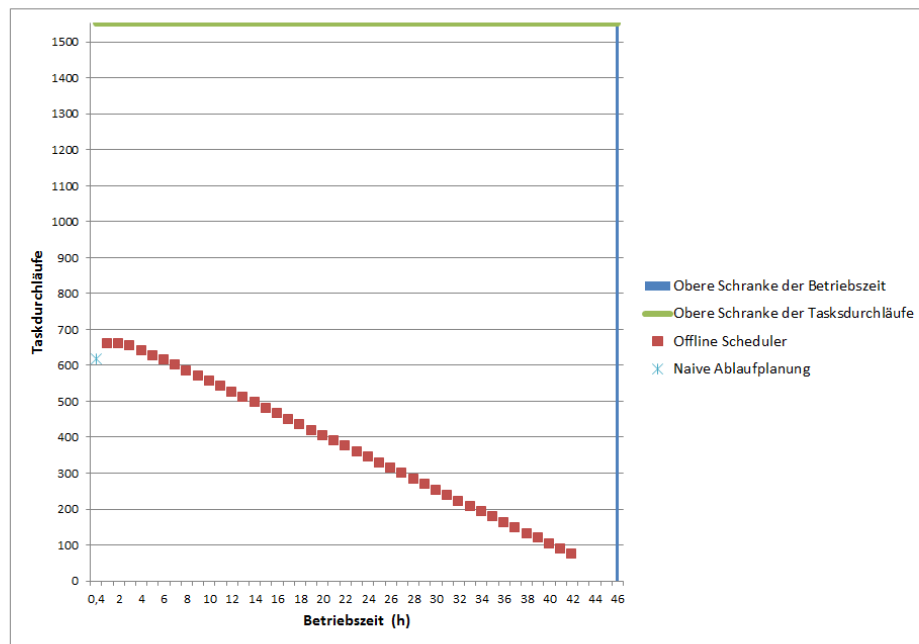


Abbildung 7.1.: Ermittelte Werte des offline Schedulers mit Hilfe der Fitnessfunktion *Targetlifetime* für die Zeiträume von 1-24 Stunden. Zum Vergleich ist der Wert der entsprechenden naiven Ablaufplanung für die anvisierte Betriebszeit von einer Stunde mit angegeben.

Ablaufplan das Ziel von 60 Minuten um 60 % verfehlt. Der offline Scheduler lieferte keine Ablaufplanung, in der die Tasks parallel ausgeführt wurden, um den Rate Capacity Effekt entgegen zu wirken, wobei auch die naive Ablaufplanung aufgrund der Pausenlänge nicht messbar unter diesem Effekt leidet. Was auch an der ermittelten entnommenen Ladung von 0.76 mAh im Vergleich zur vorherigen naiven Ablaufplanung zu sehen ist. Nichts desto trotz, führt der vom offline Scheduler ermittelte Ablaufplan zu einer effizienteren Nutzung der Batterie, weil beide Task öfter ausgeführt werden als durch den naiven Ablaufplan und trotzdem auch noch das System 60 Minuten am „Leben“ bleibt.

Ursprünglich war an dieser Stelle angedacht, ausgewählte ermittelte Ablaufplanung des offline Schedulers auf echter Hardware zu testen. Da aber zum Ende dieser Arbeit keine funktionsfähige Batterie mehr zum Testen vorhanden war, musste darauf verzichtet werden. Angedacht war es, eine Parallelschaltung von Widerständen aufzubauen, welche die Zustände der hier modellierten Hardware repräsentieren. Dabei sind die einzelnen Widerstände über Transistoren schaltbar. Die Transistoren werden über das, in den Grundlagen beschriebene, LaunchPad angesteuert. Auf dem LaunchPad wiederum läuft das Betriebssystem Kratos mit entsprechenden Tasks, welche die ermittelten Pausen wie in die Kapitel 3 beschrieben, implementiert haben. So kann das LaunchPad gemäß einem ermittelten Scheduling die Widerstände ansteuern. Weiterhin ist eine Enfilm-Batterie parallel zu den Widerständen geschaltet. Deren Zustand wird (SoC) während eines Testlaufes mit Hilfe des in den Grundlagen beschriebenen Messgerätes (U2722A USB Modular Source Measure Unit) überwacht. Für jeden Testlauf werden die Werte für die Betriebszeit und Anzahl der Tasksdurchläufe mit den ermittelten Werten des offline Schedulers verglichen.

Aus der Evaluation ist zu erkennen, dass der offline Scheduler eine gute Möglichkeit bietet eine batteriegewahre Ablaufplanung zu bestimmen und, dass es sich auch lohnt, dies zu tun. So führt schon eine einmalige Ausführung des offline Schedulers zu einer Ablaufplanung, welche die Batterie effizienter als eine naive Ablaufplanung nutzt. Im Allgemeinen liefern mehrere Ausführungen des offline Schedulers auch ähnlich gute, wenn nicht sogar dieselben, Ergebnisse. So ist z.B. möglich, die Anzahl der Tasksdurchläufe um bis 19 % zu steigern, verglichen mit einer naiven Ablaufplanung. Weiterhin ist es mit dem offline Scheduler möglich, eine anvisierte Betriebszeit mit einer maximierten Anzahl von Tasksdurchläufen in dieser Zeit zu erreichen<sup>6</sup> und verglichen mit einer naiven entsprechenden Ablaufplanung, welche im obigen Beispiel 7 % weniger Tasksdurchläufe und um 60 % die anvisierte Betriebszeit unterschritt. Auch liefert der Scheduler für komplexere Systeme eine

---

<sup>6</sup>Solange es vom Verhältnis bezogen auf die Betriebszeit, Kapazität und Energieverbrauch generell möglich ist.

batteriegewahre Ablaufplanung, welche zu mehr Tasksdurchläufen und gleichzeitig zu einer höheren Betriebszeit führt, als eine entsprechende naive Ablaufplanung erreicht. Generell ist zu sehen, dass die Stärken des offline Schedulers bei komplexeren Systemen liegen, denn wie die zweite Untersuchung gezeigt hat, ist es auch mit einer naiven Ablaufplanung möglich, in die Nähe der Werte des Schedulers zu gelangen bzogen auf die Anzahl der Tasksdurchläufe.

Für komplexe Systeme ist es nicht so einfach, „per Hand“ eine Ablaufplanung so zu bestimmen, dass sie zu ähnlich guten Werten führt wie eine Ablaufplanung des offline Schedulers, denn durch mehr Hardware und Tasks steigt die Komplexität und dies ist von Hand kaum zu bewältigen. Ferner zeigen auch die Untersuchungen bezüglich des Einflusses der Frequenz von Lasten und der Länge einer Pause auf die entnehmbare Ladung aus der Batterie, dass es nicht trivial ist, eine batteriegewahre Ablaufplanung zu finden. Aber auch beim offline Scheduler besteht noch Potential für Verbesserungen, denn der Recovery Effekt ist nicht über den gesamten Zeitraum der Entladung einer Batterie konstant. So ist er beispielsweise vom SoC und von der vorangegangenen Höhe der Belastung abhängig. Damit dieses Verhalten auch beachtet werden kann, darf es nicht nur einen Wert für eine spezifische Pause geben. Hier muss entweder ein dynamischer Scheduling-Ansatz gewählt werden oder eine Möglichkeit, dass der offline Scheduler eine Menge von Werten für eine spezifische Pause abhängig vom SoC liefert, gefunden werden. Auch kritisch zu sehen sind die sehr langen Laufzeiten des offline Schedulers, um für komplexere Systeme eine gute Ablaufplanung zu bestimmen. Diese Kosten fallen aber nur einmal zur Bestimmung des statischen Scheduling an. Also besteht der Overhead des offline Schedulers im Wesentlichen aus der Laufzeit für die Bestimmung eines Scheduling. Weiterhin besteht statisch noch der Overhead für die Header-Dateien und für den Einbau der Sleep-Methoden.

Ferner zeigt die Evaluation, dass das *KiBaM* dem offline Scheduler bzw. generischen Algorithmus passende Informationen zur Verfügung stellt, um eine batteriegewahre Ablaufplanung zu bestimmen. Zusammenfassend kann gesagt werden, dass sich eine batteriegewahre Ablaufplanung in Bezug auf die Verlängerung der Betriebszeit und Erhöhung der Anzahl der Tasksdurchläufe im Vergleich zu einer Ablaufplanung, welche das nichtlineare Verhalten von Batterien nicht beachtet, lohnt. Weiterhin ist auch der Ansatz des offline Schedulers mit dem *KiBaM* geeignet, um batteriegewahre Ablaufplanungen, welche den Recovery und Rate Capacity Effekt berücksichtigen, zu bestimmen. Im Allgemeinen zeigen die Evaluationsergebnisse, dass in Bezug auf die Menge der entnehmbaren Ladung, es besser ist, Pausen bzw. Phasen mit niedriger Last im Entladestromprofil einer Batterie zu integrieren. Gibt es keine gegenläufigen Effekte wie z.B. die Selbstentladung einer Batterie oder Idle-Phasen, die auf Dauer dem Recovery Effekt entgegenwirken, ist die Bestimmung der Pausenlängen, innerhalb der, durch die Anwendung vorge-



gebenen Grenzen, trivial.<sup>7</sup> Dies liegt daran, dass der maximale erlaubte Wert für die entsprechenden Pausen genommen werden kann. Wobei sich alle diese Aussagen auf die Simulationsergebnisse beziehen und nicht auf die Realität. Aussagen über den Nutzen der batteriegewahren Ablaufplanung bei echter Hardware konnten aufgrund des oben beschriebenen Problems nicht gegeben werden. Zumindest in Bezug auf das *KiBaM* ist mit den Validierungsergebnissen, welche im Rahmen dieser Arbeit möglich waren, gezeigt worden, wie nahe das *KiBaM* an die echte EnFilm-Batterie herankommt. Offen bleiben Aussagen zur Güte der Simulation in Bezug auf die restliche simulierte Hardware.

---

<sup>7</sup>Wobei diese in dieser Arbeit nicht betrachtet wurde, weil die Selbstentladung der EnFilm-Batterie mit 3 % relativ gering ist.



## 8. Fazit und Ausblick

In diesem Kapitel erfolgt zuerst eine Zusammenfassung der Ergebnisse dieser Masterarbeit und danach ein Ausblick für zukünftige Forschungsmöglichkeiten im Gebiet der batteriegefahrenen Ablaufplanung.

### 8.1. Fazit

Im Rahmen dieser Masterarbeit wurde eine Modellierung von Lithium-Dünnschichtbatterien für die Ablaufplanung im Betriebssystem Kratos entwickelt, um nichtlineare Effekte von Batterien, zu deren effizienteren Nutzung bei der Ablaufplanung berücksichtigen zu können.

Hierzu war es erst einmal nötig, ein Modell für Lithium-Dünnschichtbatterien zu finden. Da es im Gebiet der Batteriemodelle eine große Auswahl von Modellen gibt, erschien es vorteilhafter, ein Modell aus der Literatur einzusetzen. Problematisch dabei ist, dass die Modelle zum Teil für sehr unterschiedliche Anwendungsfälle und Batterietypen entwickelt wurden. Deshalb wurden zunächst die Anforderungen, die ein Batteriemodell für die Zwecke dieser Arbeit erfüllen muss, festgelegt und anschließend ein Überblick über Batteriemodelle, mit ihren jeweiligen Vor- und Nachteilen, aus der Literatur gegeben. Schließlich fiel die Entscheidung für das Kinetic Battery Model von Manwell und McGowan, da es am ehesten den Anforderungen entspricht. Da dieses Modell nicht ohne Weiteres für die Zwecke dieser Arbeit und auch für die in dieser Arbeit als Exemplar einer Lithium-Dünnschichtbatterie verwendete EnFilm-Batterie eingesetzt werden konnte, waren Anpassungen am Modell nötig, um auch nichtlineare Effekte im Spannungsmodell des *KiBaMs* abbilden zu können. Weiterhin musste das Modell an die EnFilm-Batterie angepasst werden, d.h. es mussten batterie-spezifische Parameter für das Kapazitäts- und Spannungsmodell ermittelt werden. Hierbei sind aber Probleme aufgetreten, da es bei der Beschaffung der nötigen Messdaten aufgrund des Einflusses von Temperaturschwankungen zu starken Schwankungen in den gemessenen Werten kam. Diese ermittelten Daten waren zur Ermittlung der Parameter nicht zu gebrauchen. Da es im Rahmen dieser Arbeit keine Möglichkeiten gab, dem Einfluss der Temperaturschwankungen entgegen zu wirken, mussten zwangsweise die benötigten Messdaten aus dem Datenblatt des Herstellers verwendet werden. Weiterhin war es auch mit

den Messdaten aus dem Datenblatt nicht ohne Weiteres möglich, ein hinreichend genaues Modell zu bestimmen, da auch diese Daten nicht von hoher Qualität waren. Aufgrund dessen mussten Heuristiken entwickelt werden, um mit den gegebenen Mitteln ein hinreichend genaues Modell finden zu können. Eine wichtige Erkenntnis bei der Parameterbestimmung ist, dass es entscheidend ist, passende Daten bezüglich der Entladeströme bei der Schätzung zu verwenden, mit denen auch das Modell arbeiten soll, um ein genaues Abbild der Realität zu erhalten. Kleinere Änderungen in den Parametern durch Daten, die nicht zu dem späteren Einsatz des Modells passen, führen zu großen Ungenauigkeiten im Modell. Die Validierung des Modells zeigt, dass das Modell sowohl für konstante als auch variable Entladeprofile im Vergleich zur realen Batterie recht akkurat ist und auch die nichtlinearen Effekte der realen Batterie gut abbilden kann.

Damit jetzt mit Hilfe des entwickelten Batteriemodells eine batteriegewahre Ablaufplanung erstellt werden kann, wurde das Batteriemodell in den von Michael Hesse entwickelten offline Scheduler integriert und dieser für eine batteriegewahre Ablaufplanung angepasst. Dieser Scheduler ermittelt mit Hilfe des *KiBaMs*, der Hardware und der Tasks eines Systems Ablaufpläne, welche durch Verzögerungen von Treiberaufrufen einen Einfluss auf die Zeitpunkte haben, zu denen Energie aus einer Batterie genommen wird. Weiterhin wird durch Vermeidung von Parallelität in den Ablaufplänen Einfluss auf die Höhe des Entladestroms genommen.

Die Evaluation des offline Schedulers zeigt, dass es sich lohnt, bezüglich einer effizienteren Nutzung einer Batterie, eine batteriegewahre Ablaufplanung im Vergleich zu einer Ablaufplanung, welche eine Batterie als ideales Bauteil betrachtet, einzusetzen. So ist es möglich, mit einer batteriegewahren Ablaufplanung, welche unter Berücksichtigung der nichtlinearen Effekte wie des Recovery und Rate Capacity Effekts erstellt wurde, die Betriebszeit und Anzahl der Taskdurchläufe gegenüber einer naive Ablaufplanung zu vergrößern. Dies ist deshalb möglich, weil die mit dem offline Scheduler ermittelten Ablaufplanungen mehr Ladung durch Beachtung der nichtlinearen Effekte aus der Batterie entnehmen können. So wurde gezeigt, dass die Frequenz der Belastung einer Batterie und die Länge von Pausen der Belastungen einen maßgeblichen Einfluss auf die entnehmbare Ladung einer Batterie, haben. Genau dies berücksichtigt der offline Scheduler. Ferner ist es möglich, vorgegebene Betriebszeiten mit einer erhöhten Anzahl von Taskdurchläufen zu erreichen, als dies im Vergleich mit naiven Ablaufplanungen möglich ist. Diese Ergebnisse gelten auch für komplexere Systeme mit mehreren Tasks und Hardwarekomponenten. Zu sehen ist aber auch, dass für einfache Systeme mit nur einem Task und einer möglichen Pause pro Durchlauf auch eine naive berechnete Gleichverteilung der Taskausführung zu Ergebnissen führt, die nahe der Ergebnisse des offline Schedulers, bezogen auf die Anzahl der Taskdurchläufe, liegen. Weiterhin wurde auch gezeigt, dass der verwendete offline Scheduler ein geeigneter Ansatz

ist, um ein batteriegewahres Scheduling zu bestimmen, da schon eine einmalige Ausführung des offline Schedulers zu guten Ergebnisse führt. So ist auch der Overhead der vorgestellten batteriegewahren Ablaufplanung nicht besonders hoch und im Wesentlichen nur einmal bei der Ermittlung eines Scheduling vorhanden. Des Weiteren hat sich durch die Evaluation auch gezeigt, dass das *KiBaM* dazu geeignet ist, mit dem offline Scheduler batteriegewahre Ablaufplanungen zu erzeugen.

Insgesamt kann also gesagt werden, dass der offline Scheduler durch Vermeidung von Parallelität und gezieltes Einsetzen von Idle-Phasen mit niedrigen Entladeströmen über die Ablaufplanung eine Batterie effizienter nutzen kann. So kann als Fazit festgehalten werden, dass es sich lohnt, eine Batterie in der Ablaufplanung von Hardwarezugriffen bzw. Hardwareaktivitäten zu berücksichtigen, um die Batterie möglichst effizient ausnutzen zu können, und dass *KiBaM* ein geeignetes Modell zur Ermittlung einer solchen Ablaufplanung ist.

## 8.2. Ausblick

Ausgehend von dieser Arbeit sind weitere Forschungsarbeiten denkbar. So ist es zum Beispiel noch notwendig, das Konzept des offline Schedulers bzw. der batteriegewahren Ablaufplanung auf realer Hardware zu testen und anschließend zu evaluieren, da dies aus den genannten Problemen im Rahmen dieser Arbeit nicht mehr möglich war. Erst durch diese zusätzliche Evaluation sind wirklich Aussagen darüber möglich, inwiefern sich ein batteriegewahrer Ablauf wirklich gegenüber einer naiven Ablaufplanung lohnt.

Weiterhin ist es denkbar, das eingesetzte *KiBaM* in Bezug auf den Recovery Effekt im Vergleich zur Enfilm-Batterie genauer zu validieren und ggf. weiter anzupassen. Generell wäre eine Parameterbestimmung und Validierung unter konstanten Temperaturbedingungen sinnvoll. In diesem Zusammenhang ist es auch möglich, die temperaturbedingten Einflüsse im Modell zu berücksichtigen bzw. ein Modell zu verwenden, welches temperaturbedingte Effekte modelliert, da diese, wie in dieser Arbeit gezeigt wurde, einen großen Einfluss auf das Verhalten der Batterie haben.

Generell bleibt die Frage offen, ob es möglich ist, auch online ein batteriegewahres Scheduling durchzuführen. So können die nichtlinearen Effekte dynamisch ausgenutzt werden. Dies wäre von Vorteil, da sich die Effekte über die Zeit der Entladung einer Batterie nicht gleichbleibend verhalten. Dies gilt besonders für den Recovery Effekt. Für ein online Scheduling ist es aber notwendig, einfache Heuristiken zu finden, um die Gewinne des batteriegewahren Scheduling nicht durch den Mehraufwand im Scheduler wieder zu verlieren. Das ist jedoch aufgrund der komplexen Abläufe einer Batterie nicht so einfach. Generell kann aber gesagt wer-

den, dass es von Vorteil ist, Idle-Phasen einzufügen und möglichst wenig Peripherie parallel zu verwenden. Auch ist es denkbar, hier den vorgestellten Ansatz auf Systeme, die mehrere Batterien verwenden, zu erweitern. So ist es zusätzlich möglich, Batterie-Scheduling zu betreiben.

# Literatur

- [1] L. Benini und G. DeMicheli. *Dynamic power management: design techniques and CAD tools*. Springer Science & Business Media, 2012.
- [2] L. Benini u. a. “A discrete-time battery model for high-level power estimation”. In: *Proceedings of the conference on Design, automation and test in Europe*. ACM. 2000, S. 35–41.
- [3] L. Benini u. a. “Battery-driven dynamic power management”. In: *IEEE Design & Test of Computers* 18.2 (2001), S. 53–60.
- [4] L. Benini u. a. “Extending lifetime of portable systems by battery scheduling”. In: *Proceedings of the conference on Design, automation and test in Europe*. IEEE Press. 2001, S. 197–203.
- [5] H. J. Bergveld, W. Kruijt und P. Notten. “Electronic-network modelling of rechargeable NiCd cells and its application to the design of battery management systems”. In: *Journal of power sources* 77.2 (1999), S. 143–158.
- [6] I. Buchmann. *Batteries in a portable world: a handbook on rechargeable batteries for non-engineers*. Ec & M Books, 1997.
- [7] M. Buschhoff. “Kratos - a resource aware, tailored operating system”. In: *Technical report for Collaborative Research Center SFB 876 - Graduate School*. Hrsg. von K. Morik und W. Rhode. 10. Dezember 2014.
- [8] A. P. Chandrakasan, S. Sheng und R. W. Brodersen. “Low-power CMOS digital design”. In: *IEICE Transactions on Electronics* 75.4 (1992), S. 371–382.
- [9] C.-K. Chau u. a. “Harnessing battery recovery effect in wireless sensor networks: Experiments and analysis”. In: *IEEE Journal on Selected Areas in Communications* 28.7 (2010), S. 1222–1232.
- [10] C.-F. Chiasserini und R. R. Rao. “Energy efficient battery management”. In: *IEEE journal on selected areas in communications* 19.7 (2001), S. 1235–1245.
- [11] C.-F. Chiasserini und R. R. Rao. “Improving battery performance by using traffic shaping techniques”. In: *IEEE Journal on Selected Areas in Communications* 19.7 (2001), S. 1385–1394.

- 
- [12] C.-F. Chiasserini und R. R. Rao. “A model for battery pulsed discharge with recovery effect”. In: *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*. IEEE. 1999, S. 636–639.
- [13] C.-F. Chiasserini und R. R. Rao. “Importance of a pulsed battery discharge in portable radio devices”. In: *Proc. of ACM MOBICOM 1999*. 1999.
- [14] C.-F. Chiasserini und R. R. Rao. “Pulsed battery discharge in communication devices”. In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM. 1999, S. 88–95.
- [15] C.-F. Chiasserini und R. R. Rao. “Routing protocols to maximize battery efficiency”. In: *MILCOM 2000. 21st Century Military Communications Conference Proceedings*. Bd. 1. IEEE. 2000, S. 496–500.
- [16] P. Chowdhury und C. Chakrabarti. “Static task-scheduling algorithms for battery-powered DVS systems”. In: *IEEE transactions on very large scale integration (VLSI) systems* 13.2 (2005), S. 226–237.
- [17] O. Corporation. *Theorie der nichtlinearen Kurvenanpassung*. [Online; Stand 2. September 2016]. URL: <http://www.originlab.de/doc/Origin-Help/NLFit-Theory>.
- [18] S. Dasika u. a. “A framework for battery-aware sensor management”. In: *Proceedings of the conference on Design, automation and test in Europe-Volume 2*. IEEE Computer Society. 2004, S. 20962.
- [19] P. C. Dow. “An analysis of certain errors in electronic differential analyzers ii-capacitor dielectric absorption”. In: *IRE Transactions on Electronic Computers* 1 (1958), S. 17–22.
- [20] M. Doyle, T. F. Fuller und J. Newman. “Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell”. In: *Journal of the Electrochemical Society* 140.6 (1993), S. 1526–1533.
- [21] T. I. Evans, T. Nguyen und R. E. White. “A mathematical model of a lithium/thionyl chloride primary cell”. In: *Journal of the Electrochemical Society* 136.2 (1989), S. 328–339.
- [22] Excellatron. *Advantage of Thin Film Batteries*. [Online; Stand 16. August 2016]. URL: <http://www.excellatron.com/advantage.htm>.
- [23] R. Falkenberg. “Entwurf eines energiegewahren Treibermodells für eingebettete Betriebssysteme”. Magisterarb. Technische Universität Dortmund, 2014.
- [24] T. F. Fuller, M. Doyle und J. Newman. “Relaxation Phenomena in Lithium-Ion-Insertion Cells”. In: *Journal of the Electrochemical Society* 141.4 (1994), S. 982–990.



- 
- [25] T. F. Fuller, M. Doyle und J. Newman. “Simulation and optimization of the dual lithium ion insertion cell”. In: *Journal of the Electrochemical Society* 141.1 (1994), S. 1–10.
- [26] M. C. Glass. “Battery electrochemical nonlinear/dynamic SPICE model”. In: *Energy Conversion Engineering Conference, 1996. IECEC 96., Proceedings of the 31st Intersociety*. Bd. 1. IEEE. 1996, S. 292–297.
- [27] S. Gold. “A PSPICE macromodel for lithium-ion batteries”. In: *Battery Conference on Applications and Advances, 1997., Twelfth Annual*. IEEE. 1997, S. 215–222.
- [28] W. Gu und C. Wang. “Thermal-Electrochemical Modeling of Battery Systems”. In: *Journal of The Electrochemical Society* 147.8 (2000), S. 2910–2922.
- [29] W. Gu u. a. “Computational fluid dynamics modeling of a lithium/thionyl chloride battery with electrolyte flow”. In: *Journal of The Electrochemical Society* 147.2 (2000), S. 427–434.
- [30] S. Gunther u. a. “Managing the impact of increasing microprocessor power consumption”. In: *Intel Technology Journal* 5.1 (2001), S. 1–9.
- [31] S. C. Hageman. “Simple PSpice models let you simulate common battery types”. In: *EDN* 38.22 (1993), S. 117.
- [32] G. Hagmann. *Grundlagen der Elektrotechnik : das bewährte Lehrbuch für Studierende der Elektrotechnik und anderer technischer Studiengänge ab 1. Semester. Mit 4 Tabellen, Aufgaben und Lösungen*. Aula-Verlag, 2013. ISBN: 9783891047798. URL: <https://books.google.de/books?id=Z1K3ngEACAAJ>.
- [33] M Handy und D Timmermann. “Simulation of mobile wireless networks with accurate modelling of non-linear battery effects”. In: *Proc. Int’l. Conf. Applied Simulation and Modeling*. 2003, S. 532–537.
- [34] M. Hesse. “Berücksichtigung nicht-linearer Energiequelleneffekte im eingebetteten Betriebssystem Kratos”. Magisterarb. Technische Universität Dortmund, 2015.
- [35] S. Hong, D. Kim und J.-e. Kim. “Battery aware real time task scheduling in wireless sensor networks”. In: *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA’05)*. IEEE. 2005, S. 269–272.
- [36] E Hyman, W. Spindler und J. Fatula. “Phenomenological discharge voltage model for lead-acid batteries.” In: (1987).
- [37] E Hyman u. a. “Modeling and computerized characterization of lead-acid battery discharges”. In: *BEST Facility Topical Report, RD* (1986), S. 83–1.
-

- 
- [38] T. Instruments. *MSP430FR5969 LaunchPad Development Kit*. [Online; Stand 17. August 2016]. URL: <http://www.ti.com/tool/msp-exp430fr5969#0>.
- [39] H. Jaksch. *Batterie-Lexikon: Primär- und Sekundärsysteme, Ladetechnik, Fertigung, Meßtechnik*. Pflaum, 1993. ISBN: 9783790506501. URL: <https://books.google.de/books?id=ROSqAgAACAAJ>.
- [40] P. K. S. Jones. *State of Solid-State Batteries (Präsentation)*. [Online; Stand 16. August 2016]. URL: <http://ceramics.org/wp-content/uploads/2011/08/energy-ss-batteries-jones.pdf>.
- [41] M. R. Jongerden und B. R. Haverkort. “Which battery model to use?” In: *IET software* 3.6 (2009), S. 445–457.
- [42] M. Jongerden und B. Haverkort. “Battery modeling”. In: (2008).
- [43] G. K. K. Küpfmüller. *Theoretische Elektrotechnik und Elektronik*. 14. Auflage. Berlin: Springer-Verlag, 1993.
- [44] J. Khan und R. Vemuri. “An iterative algorithm for battery-aware task scheduling on portable computing platforms”. In: *Proceedings of the conference on Design, Automation and Test in Europe-Volume 1*. IEEE Computer Society. 2005, S. 622–627.
- [45] A. Lahiri u. a. “Recovery-based real-time static scheduling for battery life optimization”. In: *19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID’06)*. IEEE. 2006, 4–pp.
- [46] K. Lahiri u. a. “Battery-driven system design: A new frontier in low power design”. In: *Proceedings of the 2002 Asia and South Pacific Design Automation Conference*. IEEE Computer Society. 2002, S. 261.
- [47] C. Leonhartsberger. *Innenwiderstand einer Spannungsquelle*. [Online; Stand 16. August 2016]. URL: [http://schulen.eduhi.at/riedgym/physik/10/elektrizitaet/innenwider/innenwider\\_batt.htm](http://schulen.eduhi.at/riedgym/physik/10/elektrizitaet/innenwider/innenwider_batt.htm).
- [48] D. Linden und T. Reddy. *Handbook of batteries*. McGraw-Hill handbooks. McGraw-Hill, 2002. ISBN: 9780071359788. URL: <https://books.google.de/books?id=M-9SAAAAMAAJ>.
- [49] J. Luo und N. K. Jha. “Battery-aware static scheduling for distributed real-time embedded systems”. In: *Proceedings of the 38th annual Design Automation Conference*. ACM. 2001, S. 444–449.
- [50] C. Ma, Z. Zhang und Y. Yang. “Battery-aware scheduling in wireless mesh networks”. In: *Mobile Networks and Applications* 13.1-2 (2008), S. 228–241.

- 
- [51] S. K. Mandal u. a. “IntellBatt: Towards smarter battery design”. In: *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*. IEEE. 2008, S. 872–877.
- [52] J. Manwell u. a. “Evaluation of battery models for wind/hybrid power system simulation”. In: *Proceedings of EWEC*. 1994.
- [53] J. F. Manwell und J. G. McGowan. “Extension of the kinetic battery model for wind/hybrid power systems”. In: *Proceedings of EWEC*. 1994, S. 284–289.
- [54] J. F. Manwell und J. G. McGowan. “Lead acid battery storage model for hybrid energy systems”. In: *Solar Energy* 50.5 (1993), S. 399–405.
- [55] J. F. Manwell u. a. “Improvements to the hybrid2 battery model”. In: *Wind-power 2005 Conference. American Wind Energy Association*. 2005.
- [56] J. Manwell u. a. “Hybrid2—a hybrid system simulation model—theory manual”. In: *Renewable Energy Research Laboratory, University of Massachusetts* (2006).
- [57] D. W. Marquardt. “An algorithm for least-squares estimation of nonlinear parameters”. In: *Journal of the society for Industrial and Applied Mathematics* 11.2 (1963), S. 431–441.
- [58] T. L. Martin und D. P. Siewiorek. “Balancing batteries, power, and performance: system issues in cpu speed-setting for mobile computing”. Diss. PhD thesis, Department of Electrical und Computer Engineering, Carnegie Mellon University, 1999.
- [59] P. Marwedel und L. Wehmeyer. *Eingebettete Systeme*. EXamen. press Series. Springer Berlin Heidelberg, 2008. ISBN: 9783540340492. URL: <https://books.google.de/books?id=94L961HZ51YC>.
- [60] J. Newman. *Fortran programs for the simulation of electrochemical systems*. [Online; 25. August 2016]. URL: <http://www.cchem.berkeley.edu/jsngrp/fortran.html>.
- [61] D. Panigrahi u. a. “Battery life estimation of mobile embedded systems”. In: *VLSI Design, 2001. Fourteenth International Conference on*. IEEE. 2001, S. 57–63.
- [62] C. Park, K. Lahiri und A. Raghunathan. “Battery discharge characteristics of wireless sensor nodes: An experimental analysis”. In: *power* 20 (2005), S. 21.
- [63] M. Pedram und Q. Wu. “Design considerations for battery-powered electronics”. In: *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*. ACM. 1999, S. 861–866.

- 
- [64] R. Pregla. *Grundlagen der Elektrotechnik*. Heidelberg: Hüthig Verlag, 2004.
- [65] J. M. Rabaey und M. Pedram. *Low power design methodologies*. Bd. 336. Springer Science & Business Media, 2012.
- [66] A. Raghunathan, N. K. Jha und S. Dey. *High-level power analysis and optimization*. Springer Science & Business Media, 2012.
- [67] D. Rakhmatov und S. Vrudhula. “Time-to-failure estimation for batteries in portable electronic systems”. In: *Proceedings of the 2001 international symposium on Low power electronics and design*. ACM. 2001, S. 88–91.
- [68] D. Rakhmatov, S. Vrudhula und D. A. Wallach. “A model for battery lifetime analysis for organizing applications on a pocket computer”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 11.6 (2003), S. 1019–1030.
- [69] D. Rakhmatov, S. Vrudhula und D. A. Wallach. “Battery lifetime prediction for energy-aware computing”. In: *Proceedings of the 2002 international symposium on Low power electronics and design*. ACM. 2002, S. 154–159.
- [70] D. N. Rakhmatov und S. B. Vrudhula. “An analytical high-level battery model for use in energy management of portable electronic systems”. In: *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*. IEEE Press. 2001, S. 488–493.
- [71] R. Rao und S. Vrudhula. “Battery optimization vs energy optimization: which to choose and when?” In: *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005*. IEEE. 2005, S. 439–445.
- [72] R. Rao, S. Vrudhula und D. N. Rakhmatov. “Battery modeling for energy aware system design”. In: *Computer* 36.12 (2003), S. 77–87.
- [73] V. Rao u. a. “Battery aware dynamic scheduling for periodic task graphs”. In: *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. IEEE. 2006, 8–pp.
- [74] V. Rao u. a. “Battery model for embedded systems”. In: *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. IEEE. 2005, S. 105–110.
- [75] P. Rong und M. Pedram. “An analytical model for predicting the remaining battery capacity of lithium-ion batteries”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 14.5 (2006), S. 441–451.
- [76] T. Simunic, L. Benini und G. De Micheli. “Energy-efficient design of battery-powered embedded systems”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9.1 (2001), S. 15–28.

- 
- [77] L. Song und J. W. Evans. “Electrochemical-Thermal Model of Lithium Polymer Batteries”. In: *Journal of the Electrochemical Society* 147.6 (2000), S. 2086–2095.
- [78] O. Spinczyk. *Software ubiquitärer Systeme, Hardware: Microcontrollersysteme*. Technische Universität Dortmund, Fakultät Informatik, Lehrstuhl 12. Vortragsfolien. 2013.
- [79] STMicroelectronics. *Enfilm - rechargeable solid state lithium thin film battery (Datenblatt)*. [Online; Stand 16. August 2016]. 2014. URL: [http://www.st.com/content/st\\_com/en/products/power-management/battery-management-ics/enfilm-thin-film-batteries/efl700a39.html](http://www.st.com/content/st_com/en/products/power-management/battery-management-ics/enfilm-thin-film-batteries/efl700a39.html).
- [80] STMicroelectronics. *Enfilm: the Energy of Things - Thin-film rechargeable battery (Präsentation)*. [Online; Stand 16. August 2016]. URL: [http://www.st.com/content/st\\_com/en/products/power-management/battery-management-ics/enfilm-thin-film-batteries/efl700a39.html](http://www.st.com/content/st_com/en/products/power-management/battery-management-ics/enfilm-thin-film-batteries/efl700a39.html).
- [81] I. Synopsys. *Low-Power Flow User Guide, Version D-2010.03*. März 2010.
- [82] K. C. Syracuse und W. D. Clark. “A statistical approach to domain performance modeling for oxyhalide primary lithium batteries”. In: *Battery Conference on Applications and Advances, 1997., Twelfth Annual*. IEEE. 1997, S. 163–170.
- [83] M. E. V. Team. *A Guide to Understanding Battery Specifications*. [Online; Stand 16. August 2016]. URL: [http://web.mit.edu/evt/summary\\_battery\\_specifications.pdf](http://web.mit.edu/evt/summary_battery_specifications.pdf).
- [84] K. Technologie. *Keysight Technologies The Parametric Measurement Handbook*. [Online; Stand 16. August 2016]. URL: <http://www.keysight.com/en/pd-1369935-pn-U2722A/usb-modular-source-measure-unit?cc=DE&lc=ger>.
- [85] K. Technologie. *Keysight Technologies U2722A/U2723A USB Modular Source Measure Unit - Data Sheet*. [Online; Stand 16. August 2016]. URL: <http://www.keysight.com/en/pd-1369935-pn-U2722A/usb-modular-source-measure-unit?cc=DE&lc=ger>.
- [86] K. Technologie. *Keysight Technologies Ultra Low Current Measurement Equipment - Selection Guide*. [Online; Stand 16. August 2016]. URL: <http://www.keysight.com/en/pd-1369935-pn-U2722A/usb-modular-source-measure-unit?cc=DE&lc=ger>.
- [87] K. Technologie. *U2722A USB Modular Source Measure Unit*. [Online; Stand 16. August 2016]. URL: <http://www.keysight.com/en/pd-1369935-pn-U2722A/usb-modular-source-measure-unit?cc=DE&lc=ger>.
-

- 
- [88] M. Theokharidis. *Energiemessung von ARM7TDMI Prozessor-Instruktionen*. Technische Universität Dortmund, Fakultät Informatik, Lehrstuhl 12, Diplomarbeit. 2000.
- [89] K.-C. Tsaur und R. Pollard. “Mathematical Modeling of the Lithium, Thionyl Chloride Static Cell I. Neutral Electrolyte”. In: *Journal of the Electrochemical Society* 131.5 (1984), S. 975–984.
- [90] M. Weiser. *Some Computer Science Issues of Ubiquitous Computing*. Communications of the ACM, Vol. 36, No. 7. Juli 1993.
- [91] M. Weiser. *The Major Trends in Computing*. [Online; Stand 12. August 2016]. URL: <http://www.ubiq.com>.
- [92] K. Wu. *The Quick Guide to the Battery Parameter Finder and the Standalone Battery Model Software*. 2014.
- [93] Q. Wu, Q. Qiu und M. Pedram. “An interleaved dual-battery power supply for battery-operated electronics”. In: *Proceedings of the 2000 Asia and South Pacific Design Automation Conference*. ACM. 2000, S. 387–390.
- [94] S. Yin u. a. “Extending lifetime of battery-powered coarse-grained reconfigurable computing platforms”. In: *Proceedings of the conference on Design, Automation & Test in Europe*. European Design und Automation Association. 2014, S. 337.

# Abbildungsverzeichnis

1.1. Prognose von Mark Weiser über die Trends in der Informationstechnologie. . . . .	3
1.2. Erweiterung der Battery Gap, durch schnell wachsende Leistungsanforderungen und langsame Verbesserungen in der Batterietechnologie. . . . .	4
2.1. Beispiel einer Schaltung in CMOS-Technologie. . . . .	11
2.2. Funktionsweise des CMOS-Inverters. . . . .	12
2.3. Auftreten von Schaltstrom. . . . .	13
2.4. Auftreten eines Kurzschlussstroms. . . . .	14
2.5. Auftreten von Leckströmen. . . . .	15
2.6. PTA einer fiktiven Kommunikationsschnittstelle. Die Übergänge und Zustände tragen ihre jeweiligen Kosten. Durch Eintreten eines Ereignisses oder durch Erfüllen einer zeitlichen Bedingung können Zustandsübergänge ausgelöst werden. . . . .	16
2.7. Links im Bild ist ein klassischer Treiberansatz, welcher keine Informationen über den Energieverbrauch bereitstellt. Rechts im Bild ist der DFA-Treiberansatz, welcher Rückschlüsse über den Leistungsbedarf des aktuellen Hardwarezustandes liefern kann. . . . .	17
2.8. Schichtenstruktur von Kratos. . . . .	18
2.9. Grundaufbau einer Batterie am Beispiel Lithium/Thionylchlorid Batterie. . . . .	19
2.10. Ersatzschaltbild einer Batterie. . . . .	22
2.11. Schematische Darstellung einer realen Batterie mit einem Lastwiderstand $R_L$ . Dabei stellt diese gestrichelte Umrandung die reale Batterie dar. . . . .	23
2.12. Eigenschaften einer idealen Batterie: Konstante Spannung und Kapazität . . . . .	24
2.13. Die Spannung einer Batterie als eine Funktion über der Entladezeit für zwei verschiedene Lasten. . . . .	25
2.14. Der Rate Capacity Effekt. . . . .	26

2.15. Der Recovery Effekt bei einer Batterie. Die untere Kurve zeigt den zeitlichen Verlauf der Klemmspannung bei einem konstanten Entladestrom. Die obere Kurve zeigt den Verlauf der Klemmspannung bei einem intermittierenden Entladestrom, wobei die Pausen zur besseren Darstellung nicht in der Kurve dargestellt sind. . . . .	28
2.16. Schematischer Querschnitt einer Dünnschichtbatterie. . . . .	29
2.17. Vergleich der verschiedenen existierenden Batterietypen. . . . .	30
2.18. Die EnFilm EFL700A39 von ST. . . . .	32
2.19. Die Bestandteile der Enfilm. . . . .	32
2.20. Typische Entladekurven der EnFilm. . . . .	33
2.21. Typische Ladekurve der Enfilm. . . . .	33
2.22. U2722A USB Modular Source Measure Unit. . . . .	35
2.23. Aufbau einer Source Measure Unit. . . . .	35
2.24. 4-Quadranten Operationen. . . . .	36
2.25. Kelvin (4-Kabel) Messung. . . . .	36
2.26. Keysight Measurement Manager (AMM). . . . .	37
2.27. Das MSP430FR5969 LaunchPad. Zu sehen ist unten zum einen die Hauptplatine mit dem MSP430FR5969 Mikrocontroller und oben die integrierte Programmierschaltung zum Aufspielen der Firmware und Debuggen. . . . .	38
3.1. Schematische Darstellung des offline Scheduler mit seinen Bestandteilen zur Ermittlung einer energiequellengewahren Ablaufplanung. . . . .	41
3.2. Das Kammermodell zur Modellierung eines Kondensators. . . . .	44
4.1. Basisschaltungen zum Modellieren einer beliebigen Zelle . . . . .	59
4.2. Das Basis Markow-Ketten-Batteriemodell von Chiasserini und Rao. . . . .	60
4.3. Das erweiterte Markow-Ketten Batteriemodell von Chiasserini und Rao. . . . .	61
4.4. Beispiel eines stückweise konstanten Entladeprofils. . . . .	65
4.5. Physikalisches Bild des Modells nach Rakhmatov und Vrudhula. . . . .	66
4.6. Schematische Darstellung eines allgemeinen Batteriesystems. . . . .	69
4.7. Das Kinetic Battery Model modelliert eine Batterie als Behälter. . . . .	70
5.1. Der Messaufbau mit dem Messgeräte (links) und der Enfilm-Batterie auf der Platine (rechts). . . . .	80



5.2. GUI des <i>KiBaM Parameter Finder</i> . Zu sehen ist oben die Möglichkeit, Daten über den Entlade-/Ladeprozess einer Batterie einzugeben. In der oberen Mitte ist die graphische Darstellung der Eingabedaten bzw. der Modelle (Spannungs- und Kapazitätsmodell) möglich. In der unteren Mitte links kann zwischen den Eingabedaten und verschiedenen Typen von Graphen gewechselt werden. In der unteren Mitte rechts kann die betrachtete Menge der Eingabedaten begrenzt werden und ganz unten werden die ermittelten Parameter angezeigt. . . . .	87
5.3. Entladekurven entsprechend des Entladestroms 3.5mA aus dem Datenblatt und Modell 1. . . . .	92
5.4. Entladekurven entsprechend des Entladestroms 3.5mA aus dem Datenblatt und Modell 2. . . . .	92
5.5. Vergleich der entnehmbaren Ladung für verschiedene Entladeströme zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	96
5.6. Vergleich des Spannungsverlaufs für den Entladestrom von 3.5 mA zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	97
5.7. Vergleich des Spannungsverlaufs für den Entladestrom von 1.4 mA zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	97
5.8. Vergleich des Spannungsverlaufs für den Entladestrom von 1 mA zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	98
5.9. Vergleich des Spannungsverlaufs für den Entladestrom von 70 $\mu$ A zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	98
5.10. Vergleich des Spannungsverlaufs für den Entladestrom von 1 $\mu$ A zwischen dem <i>KiBaM</i> und der EnFilm-Batterie. . . . .	99
5.11. Spannungsverlauf der EnFilm-Batterie bei einer Minute mit 5 mA Entladestrom und drei Minuten mit 1 mA Entladestrom im Wechsel bis die Batterie erschöpft ist. . . . .	100
5.12. Spannungsverlauf des Modells bei einer Minute mit 5 mA Entladestrom und drei Minuten mit 1 mA Entladestrom im Wechsel bis die Batterie erschöpft ist. . . . .	100
5.13. Spannungsverlauf der EnFilm-Batterie bei einer Minute mit 5 mA Entladestrom und 1 Stunde Pause im Wechsel bis die Batterie erschöpft ist. . . . .	101
5.14. Spannungsverlauf des Modells bei einer Minute mit 5 mA Entladestrom und 1 Stunde Pause im Wechsel bis die <i>KiBaM</i> erschöpft ist. . . . .	102

- 
- 7.1. Ermittelte Werte des offline Schedulers mit Hilfe der Fitnessfunktion *Targetlifetime* für die Zeiträume von 1-24 Stunden. Zum Vergleich ist der Wert der entsprechenden naiven Ablaufplanung für die anvisierte Betriebszeit von einer Stunde mit angegeben. . . . . 120

# Tabellenverzeichnis

2.1. Wichtige Merkmale der EnFilm EFL700A39 [79]. . . . .	34
5.1. Die zur Enfilm-Batterie ermittelte Batterielaufzeit der entsprechenden Entladeströme. . . . .	82
5.2. Die ermittelten Werte der Parameter des Kapazitätsmodells für die Enfilm-Batterie. . . . .	85
5.3. Die Wertepaare, welche zur Bestimmung der Parameter verwendet wurden. . . . .	86
5.4. Beispiel einer Eingabedatenmenge für den <i>KiBaM Parameter Finder</i> . . . . .	89
5.5. Ein Ausschnitt aus den verwendeten Kombinationen zur Bestimmung der Parameter des Spannungsmodells. . . . .	90
5.6. Die ermittelten Werte der Parameter des Spannungsmodells für die Enfilm-Batterie. . . . .	93
7.1. Verwendete Einstellungen des offline Schedulers in der Evaluation. . . . .	110
7.2. Die für verschiedene Frequenzen verwendete Zeiten. . . . .	116
7.3. Die für verschiedene Frequenzen ermittelten Endladezeiten und entnommene Ladungen. . . . .	117



# Listingverzeichnis

3.1.	Definition eines Systems in XML. . . . .	42
3.2.	Definition eines Moduls in XML. . . . .	44
3.3.	Task mit Angabe von Pausen. . . . .	46
3.4.	Definition eines Tasks in XML. . . . .	47
7.1.	XML-Definition des Systems mit einem Modul und einem Task. . .	110
7.2.	XML-Definition des Funk-Moduls mit den zwei Zuständen Idle und Transmit. . . . .	111
7.3.	XML-Definition des Funk-Task. . . . .	112
7.4.	XML-Definition des System mit zwei Modulen und zwei Tasks. . . .	112
7.5.	XML-Definition des Mess-Moduls mit den zwei Zuständen Off und Measure. . . . .	113
7.6.	XML-Definition des Mess-Tasks. . . . .	114



# A. Anhang

Für die im Rahmen dieser Masterarbeit entwickelter Software gelten nachfolgende Bedingungen:

Copyright (c) <2016> <David Schmidt>

Hiermit wird unentgeltlich jeder Person, die eine Kopie der Software und der zugehörigen Dokumentationen (die „Software“) erhält, die Erlaubnis erteilt, sie uneingeschränkt zu nutzen, inklusive und ohne Ausnahme mit dem Recht, sie zu verwenden, zu kopieren, zu verändern, zusammenzufügen, zu veröffentlichen, zu verbreiten, zu unterlizenzieren und/oder zu verkaufen, und Personen, denen diese Software überlassen wird, diese Rechte zu verschaffen, unter den folgenden Bedingungen:

Der obige Urheberrechtsvermerk und dieser Erlaubnisvermerk sind in allen Kopien oder Teilkopien der Software beizulegen.

DIE SOFTWARE WIRD OHNE JEDE AUSDRÜCKLICHE ODER IMPLIZIERTE GARANTIE BEREITGESTELLT, EINSCHLIEßLICH DER GARANTIE ZUR BENUTZUNG FÜR DEN VORGESEHENEN ODER EINEM BESTIMMTEN ZWECK SOWIE JEDLICHER RECHTSVERLETZUNG, JEDOCH NICHT DARAUF BESCHRÄNKT. IN KEINEM FALL SIND DIE AUTOREN ODER COPYRIGHTINHABER FÜR JEDLICHEN SCHADEN ODER SONSTIGE ANSPRÜCHE HAFTBAR ZU MACHEN, OB INFOLGE DER ERFÜLLUNG EINES VERTRAGES, EINES DELIKTES ODER ANDERS IM ZUSAMMENHANG MIT DER SOFTWARE ODER SONSTIGER VERWENDUNG DER SOFTWARE ENTSTANDEN.





# Eidesstattliche Versicherung

---

Name, Vorname

---

Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/Masterarbeit\* mit dem Titel

---

---

---

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

---

Ort, Datum

---

Unterschrift

\*Nichtzutreffendes bitte streichen

## Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG - )

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

---

Ort, Datum

---

Unterschrift