



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Timo Gerken

Untersuchung der spektralen Eigenschaften von
optischen Effekten an Lithium-Eisenphosphat-
Elektroden

Timo Gerken

Untersuchung der spektralen Eigenschaften von
optischen Effekten an Lithium-Eisenphosphat-
Elektroden

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Karl-Ragmar Riemschneider
Zweitgutachter : Prof. Dr.-Ing. Jürgen Vollmer

Abgegeben am 24. November 2016

Timo Gerken

Thema der Bachelorthesis

Untersuchung der spektralen Eigenschaften von optischen Effekten an Lithium-Eisenphosphat-Elektroden

Stichworte

Lithium-Eisenphosphat-Akkumulator, LiFePO_4 , optische Messtechnik, Anode, Kathode, Spektrometer, Kamera, elektrochemischer Effekt, Lithium-Interkalation

Kurzzusammenfassung

In dieser Arbeit werden die spektralen Eigenschaften von optischen Effekten an Lithium-Eisenphosphat-Elektroden im Verlauf von Lade- und Entladezyklen untersucht. Dazu ist ein Zykliergerät in Betrieb zu nehmen und umfangreiche Messreihen durchzuführen. Die Skripte der Datenauswertung sind zu optimieren und die Ergebnisse durch geeignete Konzepte darzustellen. Durch unterschiedliche Zusammensetzungen und Ladeverfahren der Zellen sollen Zusammenhänge zwischen elektrischen Werten und optischen Effekten ermittelt werden.

Timo Gerken

Title of the paper

Investigation of the spectral characteristics of optical effects on lithium iron phosphate electrodes

Keywords

Lithium iron phosphate battery, LiFePO_4 , optical metrology, anode, cathode, spectrometer, camera, electrochemical effect, Lithium intercalation

Abstract

Inside this report the spectral characteristics of optical effects of lithium iron phosphate electrodes are being investigated during charge and discharge cycles. A cycle system is put into operation to perform extensive measurements. The scripts of the data evaluation have to be optimized and the results presented through suitable concepts. The relationship of electrical data and spectral effects should be determined through different compositions and loading methods of the cells.

Inhaltsverzeichnis

Abkürzungsverzeichnis	7
1. Einleitung	8
1.1. Motivation	8
1.2. Arbeitsgruppe BATSEN	10
1.3. Ziel dieser Arbeit	10
2. Theorie	12
2.1. Grundlagen der Batterietechnik	12
2.2. Lithium-Ionen-Batterietechnologie	14
2.2.1. Aufbau und Grundlagen	14
2.2.2. Optische Effekte an Lithium-Eisenphosphat-Batterien	17
2.3. Optische Messtechnik	18
2.4. Bildverarbeitung	20
2.5. Referenzwertbildung	21
3. Messplatz	22
3.1. Aufbau des Messplatzes	22
3.1.1. Aufbau der Messschaltung	22
3.1.2. Elektrischer Aufbau	25
3.1.3. Systematischer Messfehler	26
3.2. Zyklersystem	27
3.3. Optische Messungen	30
3.3.1. Mikroskopkamera	30
3.3.2. Spektrometer	32
3.4. Messsoftware	35
3.4.1. Software des Zyklersystems und Kamera	35
3.4.2. Spektrometersoftware	36
3.5. Aufbau der Zellen	38
3.6. Aufgetretene Probleme und deren Lösungen	41

4. Datenverarbeitung	42
4.1. Konzepte	42
4.1.1. Rauschunterdrückung	42
4.1.2. Ableitung	44
4.1.3. Mittelwertbildung	46
4.1.4. Schwerpunkt der Intensität	46
4.2. Verarbeitungssoftware	47
4.2.1. CurrentIntegrator	47
4.2.2. TimestampMatcher	47
4.2.3. HaloStack	48
4.3. Kameradatenverarbeitung	49
4.4. Spektrometerdatenverarbeitung	50
5. Auswertung	52
5.1. Übersicht der Zellaufbauten	52
5.2. Messungen mit der Kamera	56
5.2.1. ITO - typischer Verlauf	56
5.2.2. LiFePO ₄ - typischer Verlauf	61
5.3. Messungen mit dem Spektrometer	68
5.3.1. ITO - typischer Verlauf	68
5.3.2. ITO - Suche nach Wellenlängenabhängigkeit	72
5.3.3. ITO - Verschiebung des Spektrumschwerpunktes	74
5.3.4. ITO - Nachanalyse vorhergehender Messungen	75
5.3.5. LiFePO ₄ - Eigene Messungen	78
6. Fazit	82
6.1. Zusammenfassung und Bewertung der Messreihen	82
6.2. Generelle Bewertung	83
6.3. Ausblick	85
Literaturverzeichnis	87
Tabellenverzeichnis	88
Abbildungsverzeichnis	89
A. Quellcode	92
A.1. Zyklersystem	92
A.1.1. bc_param.cfg	92
A.1.2. start_all.sh	94
A.2. CurrentIntegrator	95
A.3. TimestampMatcher	96

A.4. Shellskripte	97
A.4.1. chmod.sh	97
A.4.2. stacking-100.sh	98
A.4.3. DataOperator.sh	98
A.5. Matlaskripte	99
A.5.1. config.m	99
A.5.2. m01_imageRead.m	100
A.5.3. m02_plotCreate.m	103
A.5.4. m03_videoCreate.m	109
A.5.5. m12_spectrometer2matlab.m	112
A.5.6. m13_showSpectra.m	113
B. Ordnerstruktur	117
C. Aufgabenstellung	118

Abkürzungsverzeichnis

SoH State of Health

SoC State of Charge

LiCoO₂ Chemische Summenformel für Lithium-Cobaltdioxid

LiNiO₂ Chemische Summenformel für Lithium-Nickeldioxid

LiMn₂O₂ Chemische Summenformel für Lithium-Manganoxid

LiFePO₄ Chemische Summenformel für Lithium-Eisenphosphat

LiPF₆ Chemische Summenformel für Lithiumhexafluorophosphat

SEI Solid Electrolyte Interface

PEEK Polyetheretherketon, ein thermoplastischer Kunststoff

EPDM Ethylen-Propylen-Dien-Kautschuk, ein terpolymeres Elastomer (Gummi)

1. Einleitung

1.1. Motivation

Durch die Industrialisierung und dem daraus folgenden Wirtschaftswachstum steigt der Energieverbrauch, vor allem in Industrie- und Schwellenländern, rapide an. Ein Großteil dieser Energie wird aus fossilen Energiequellen gewonnen, bei deren Verbrennung Kohlendioxid (CO₂) freigesetzt wird. Sein Anstieg in der Erdatmosphäre wird als Hauptgrund für den Klimawandel angesehen.

Um eine weitere Erhöhung des Treibhausgasausstoßes zu verhindern, wurde von den Vereinten Nationen am 11.12.1997 im Rahmen des Kyoto-Protokolls ein Rahmenübereinkommen beschlossen, welches vorsah, die CO₂-Emissionen in den Industrieländern bis zum Jahr 2012 um durchschnittlich 5,2% zum Stand von 1990 zu reduzieren. Am 16.02.2005 trat das Abkommen in Kraft und sah für Deutschland eine Reduzierung um 21% zum Jahr 1990 vor. Dieses Ziel wurde von Deutschland bereits 2007 erreicht.

Ungeachtet dessen ist der weltweite Treibhausgasausstoß bis zum Jahr 2014 um über 50% zum Jahr 1990 auf 35,5 Mrd. t CO₂-Äquivalenten¹ gestiegen.

Als ein Vorreiter in der Energiewende hat es sich Deutschland zur Aufgabe gemacht, bis zum Jahr 2020 die Emissionen um weitere 19% zu senken. Das Hauptaugenmerk liegt hierbei auf dem Umstieg von fossilen Energiequellen zu klimaneutralen erneuerbaren Energien. Doch auch in anderen Bereichen wird versucht, die CO₂-Emissionen zu senken. Wie in Abbildung 1.1 zu erkennen ist, betrug der Anteil der Treibhausgasemissionen des Verkehrs in Deutschland im Jahr 2014 17,9% am Gesamtausstoß von 902 Mio. t CO₂-Äquivalenten. Um diesen zu senken, hat es sich die Bundesregierung zur Aufgabe gemacht, bis zum Jahr 2020 rund eine Millionen Elektroautos auf die deutschen Strassen zu bringen. Aktuellen Statistiken zufolge liegt die Anzahl der Elektroautos zur Zeit bei etwa 25000 Fahrzeugen [1].

¹CO₂-Äquivalente, Maßeinheit des relativen Treibhauspotentials von Gasen

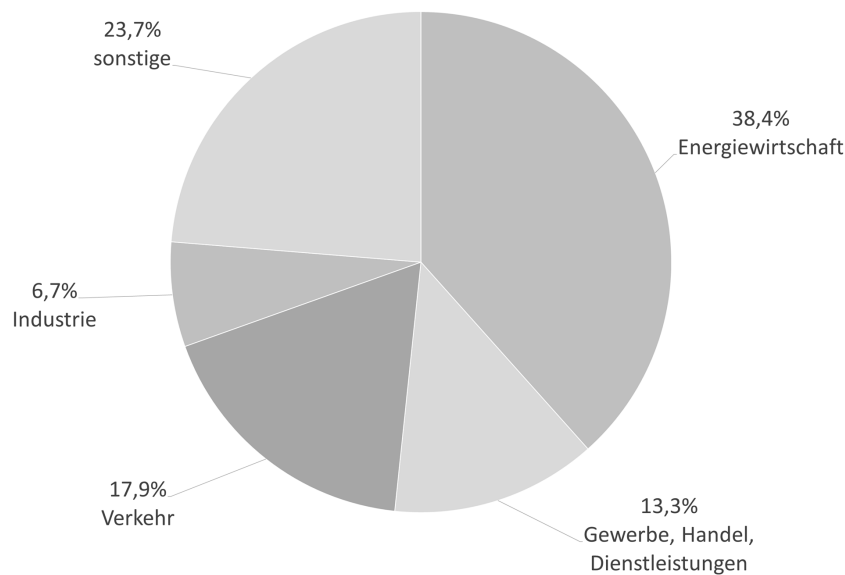


Abbildung 1.1.: Anteil der Sektoren an den Treibhausgasemissionen in Deutschland 2014 [2]

Neben den hohen Anschaffungskosten und der begrenzten Anzahl von Ladestationen gehören vor allem die hohen Ansprüche an die Traktionsbatterien zu den großen Problemen der Elektromobilität. Die Batterie muss eine hohe Energiedichte aufweisen, um bei geringem Gewicht weite Fahrtstrecken zu ermöglichen. Dabei ist sie sowohl hohen Entladeströmen beim Beschleunigen des Fahrzeugs, als auch hohen Ladeströmen, um die Ladezeiten kurz zu halten, ausgesetzt.

Ein weiterer wichtiger Punkt ist die Lebensdauer der Batterie. Eine hohe Zyklenfestigkeit und kalendarische Lebensdauer ist wünschenswert, um die Wirtschaftlichkeit für den Kunden zu erhöhen. Wichtig ist ebenfalls die Temperaturbeständigkeit, bedingt durch die jahreszeitlichen Schwankungen, die ebenfalls die Lebensdauer beeinflussen kann.

Eine Batterietechnologie, die alle diese Anforderungen erfüllt, gibt es gegenwärtig nicht. Hersteller von Elektroautos setzen derzeit nahezu ausschließlich auf Lithium-Ionen-Batterien. Sie besitzen eine relativ hohe Energiedichte und vertragen eine hohe Anzahl von Ladezyklen. Des Weiteren haben sie keinen nennenswerten Memory-Effekt, bei dem aufgrund häufiger Teilentladungen ein Kapazitätsabfall auftritt.

Jedoch sind sie heutzutage noch teuer in der Herstellung und haben eine lange Ladezeit. Ein weniger augenscheinlicher Nachteil ist, dass sich verlässliche Aussagen über den SOH¹

¹State of Health, engl. für Alterungszustand

und SOC¹ alleine durch die elektrischen Messgrößen, wie Zellspannung und Ströme, nicht ausreichend genau treffen lassen. Dies ist jedoch notwendig, um die Betriebssicherheit und Zuverlässigkeit zu gewährleisten.

Ein Ansatz diese Größen zu ermitteln, ist die optische Beobachtung der Elektrodenmaterialien der Zellen. Hierbei wird versucht, über Änderungen der Farbintensität eine Aussage über den Zustand der Zelle zu machen.

1.2. Arbeitsgruppe BATSEN

Die von Partnern in der Industrie und durch das Bundesministerium für Bildung und Forschung (BMBF) geförderte Arbeitsgruppe BATSEN befasst sich mit der Entwicklung von drahtlosen Zellsensoren für Fahrzeugbatterien. Ziel ist es, ein Sensornetz zur Überwachung jeder einzelnen Zelle der Batterie zu erstellen um ihre Wirtschaftlichkeit, Sicherheit und betriebliche Verfügbarkeit zu verbessern. Der Anwendungsbereich umfasst neben Starter- und Antriebsbatterien für Fahrzeuge auch Bleibatterien mit vielen Zellen und Lithiumbatterien. Das Konzept sieht vor, die Spannungs- und Temperaturmessungen für jede Zelle vorzunehmen und die Daten zusammen mit einer zentralen Strommessung in einem Batteriesteuersystem zu verarbeiten.

Eine weitere Idee war, durch die optische Beobachtung der Elektrodenmaterialien von Lithium-Ionen-Batterien eine Aussage über den Lade- und Gesundheitszustand der Zellen machen zu können. Hierzu wurden Rahmen von Bachelorarbeiten bereits einige Vorarbeiten getätigt.

1.3. Ziel dieser Arbeit

Die getätigten Vorarbeiten im Projekt BATSEN sind im Rahmen dieser Arbeit weiterzuführen. Ein für diesen Zweck entwickeltes Zyklersystem, welches die Auf- und Entladephasen von Lithium-Eisenphosphat-Batterien steuert, die elektrischen Messdaten aufnimmt und die Kameraansteuerung der optischen Beobachtung übernimmt, ist in Betrieb zu nehmen. Die daraus resultierenden Ergebnisse der Messungen sollen ausgewertet und analysiert werden. Dieses beinhaltet sowohl die erfolgreichen Messungen, als auch die Unterstützung bei der Behebung eventuell auftretender Fehler in der Soft- oder Hardware des Messaufbaus.

¹State of Charge, engl. für Alterungszustand

Das Hauptaugenmerk der Arbeit liegt jedoch bei der Planung und Durchführung von Messreihen, sowie der Auswertung und Beurteilung der Ergebnisse. Insbesondere das spektrale Verhalten der Elektroden soll mit einer Mikroskopkamera und einem Spektrometer untersucht werden. Zu diesem Zweck sind geeignete Konzepte der Datenverarbeitung zu nutzen.

Die einzelnen Punkte sind im Folgenden aufgeführt:

- Einarbeitung in die Grundlagen und Effekte von Lithium-Batterien, sowie den getätigten Vorarbeiten
- Entwurf und Modifikation von Matlab-Skripten zur Auswertung und Darstellung
- Vorabmessungen, Platinenaufbau und Inbetriebnahme des Zyklrier- und Erfassungssystems
- Montage der Testzellen mit den Elektrodenproben
- Herstellen des Messaufbaus
- Versuchsplanung und Durchführung von umfangreichen Messreihen
- Spektrumsauswertung und Darstellung von spektralen Änderungen während des Zyklrierbetriebs
- Zusammenfassen und beurteilen der Messergebnisse
- Bewertung der gewählten Konzepte sowie offener Punkte

2. Theorie

In diesem Kapitel werden die theoretischen Grundlagen der Batterietechnik und Lithium-Ionen-Batterien dargestellt. Hierbei wird anhand eines Batteriemodells besonders auf den chemischen Aufbau, sowie die elektrochemischen Abläufe der Batterie eingegangen. Des Weiteren wird auf die optische Messtechnik und Referenzwertbildung eingegangen.

2.1. Grundlagen der Batterietechnik

Als Batterie wird ein elektrischer Energiespeicher bezeichnet, der aus einer in einem Gehäuse untergebrachten Zusammenschaltung mehrerer galvanischer Zellen besteht. Hierbei wird grundsätzlich in Primär- und Sekundärzellen unterschieden.

Primärzellen

Eine Primärbatterie ist in der Lage, ihre gespeicherte elektrochemische Energie in elektrische Energie umzuwandeln. Der elektrochemische Prozess ist jedoch nicht umkehrbar, weshalb sie nach einer Entladung nicht mehr wieder aufgeladen werden kann.

Sekundärzellen

Sekundärbatterien, auch Akkumulatoren genannt, funktionieren nach demselben Prinzip wie Primärbatterien. Im Gegensatz zu ihnen können sie jedoch wieder aufgeladen werden. Da beim Aufladen der Sekundärbatterien Wärme entsteht, kann nicht die gesamte Energie in chemischer Form gespeichert werden. Hier spricht man vom Ladewirkungsgrad. Dieser liegt gewöhnlich zwischen 70 und 85 %, wird häufig aber auch als sein Kehrwert, dem Ladefaktor angegeben.

Ist in dieser Arbeit der Begriff „Batterie“ verwendet, so ist damit die Sekundärbatterie gemeint.

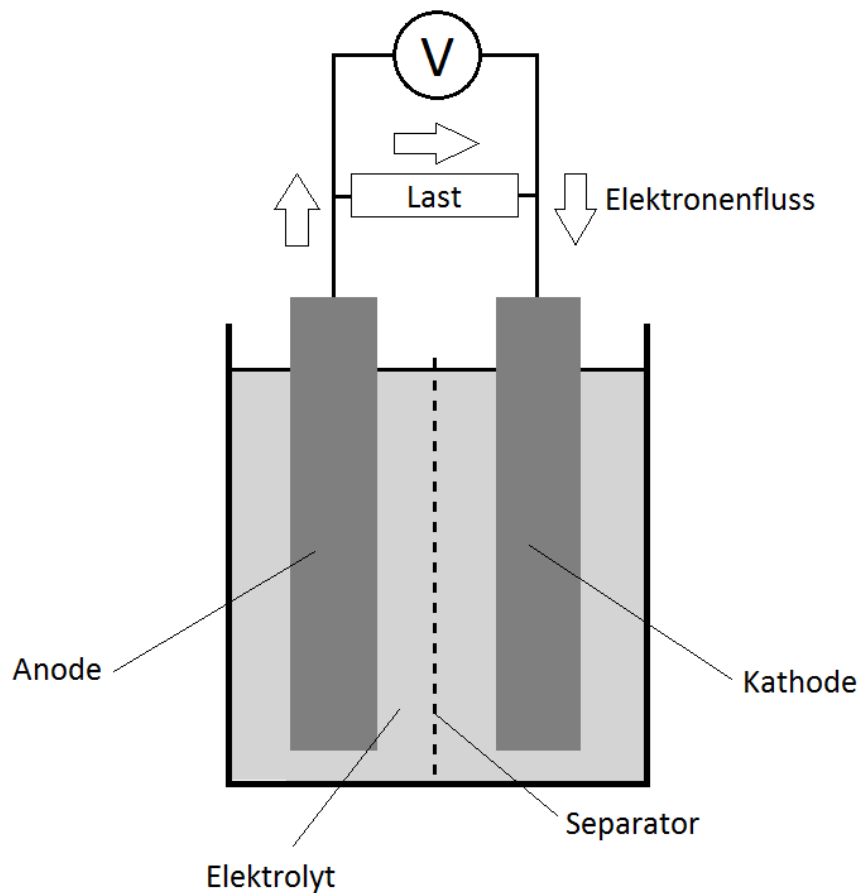


Abbildung 2.1.: Aufbau einer galvanischen Zelle

Abbildung 2.1 zeigt den Aufbau einer galvanischen Zelle. Sie besteht aus einer positiven Elektrode (Kathode) und einer negativen Elektrode (Anode). Diese sind umgeben von einem Elektrolyten und durch einen Separator voneinander getrennt.

Das Prinzip der galvanischen Zelle beruht auf einer Redoxreaktion. Hierbei gibt die negative Elektrode Elektronen ab, das heißt sie oxidiert. Die positive Elektrode nimmt diese Elektronen auf, sie wird reduziert.

Bei einer Aufladung der Zelle, kehren sich die Polaritäten der Elektroden um. Die positive Elektrode wird zu Anode und die negative Elektrode zur Kathode. Wird in dieser Arbeit einer der Begriffe verwendet, so wird, wenn nicht anders angegeben, vom Fall der Entladung ausgegangen.

2.2. Lithium-Ionen-Batterietechnologie

Aufgrund ihres hohen Energieverbrauchs bei möglichst geringem Gewicht und Volumen werden in tragbaren Unterhaltungsmedien sowie der Elektromobilität bevorzugt Lithium-Ionen-Batterien verwendet. Diese zeichnen sich durch eine hohe Energiedichte und Stromverträglichkeit aus und besitzen keinen nennenswerten Memory-Effekt. Lithium eignet sich besonders als Aktivmaterial, da es:

- das leichteste Material in Feststoffform ist
- das größte negative Standardpotential der elektrochemischen Spannungsreihe besitzt.

Die Eigenschaften von Lithium-Ionen-Batterien lassen sich durch unterschiedliche Zusammensetzungen weiter beeinflussen. Eine wesentliche Rolle spielen hierbei:

- Kathodenmaterial
- Anodenmaterial
- Elektrolyt
- Separator

Auf diese Bauteile wird im Folgenden weiter eingegangen.

2.2.1. Aufbau und Grundlagen

Negative Elektrode (Anode):

Die negative Elektrode besteht aus einer als Stromableiter dienenden Kupferfolie mit einer aufgetragenen Schicht Graphit oder Lithium-legiertem Material. In diese Schicht werden beim Aufladen die positiven Lithium-Ionen eingelagert. Am verbreitetsten sind Graphitanoden, da sie ein niedriges Elektrodenpotential¹ aufweisen und sich bei der Einlagerung von positiven Lithium-Ionen nur sehr geringfügig ausdehnen. Aluminium hat zwar eine geringere Dichte und ist günstiger als Kupfer, kann jedoch aufgrund von parasitären Ausbildungen von Lithium-Aluminium-Legierungen durch das niedrige Potential der negativen Elektrode nicht als Stromableiter verwendet werden.

¹Elektrodenpotential, nach der chemischen Spannungsreihe kann jeder Elektrode ein Potential zugeordnet werden. Gemessen wird es gegenüber dem konstanten Potential von metallischem Lithium.

Positive Elektrode (Kathode):

Für die negative Elektrode kann, im Gegensatz zur Anode, Aluminium als Stromableiter verwendet werden. Auf ihm wird das Aktivmaterial aufgebracht. Das meist verbreitete Kathodenmaterial ist Lithium-Cobaltdioxid. Es wurde auch im ersten kommerziell erhältlichen Lithium-Ionen-Akkumulator verwendet, der 1991 von Sony auf den Markt gebracht wurde. Bei einer mittleren Entladespannung von 3,9 V können relativ hohe Energiedichten erreicht werden. Es bedarf jedoch einer Überwachung durch ein Batteriemanagementsystem, da dieses Material sehr anfällig gegen Überhitzung bei hohen Belastungen oder unkontrolliertem Laden ist.

Ein weiteres Kathodenmaterial ist Lithium-Nickeldioxid. Trotz einer sehr guten Energiedichte bei einer mittleren Entladespannung von 3,8 V wird dieses Material nicht in Traktionsbatterien verwendet. Der Grund hierfür sind seine sehr kritischen Sicherheitseigenschaften.

Gute Eigenschaften im Hochstrombereich sowie gutes Sicherheitsverhalten besitzen Akkumulatoren mit Lithium-Manganoxid als Kathodenmaterial. Sie werden häufig als Traktionsbatterie verwendet. Aufgrund ihrer relativ geringen Energiedichte wird dieses Material jedoch meist im gemischten Zustand mit Lithium-Cobaltdioxid verwendet.

Eine im Vergleich zu Lithium-Cobaltdioxid-Akkumulatoren ähnliche Energiedichte weisen Akkumulatoren mit Lithium-Eisenphosphat auf. Die mittlere Entladespannung beträgt 3,4 V. Ein großer Vorteil sind die sehr guten Sicherheitseigenschaften. Da keine exothermen Reaktionen mit diesem Material bestehen, können diese Akkumulatoren nicht brennen und explodieren. Lithium-Eisenphosphat-Akkumulatoren vertragen hohe Lade- und Entladeströme, weshalb sie wegen relativ kurzen Ladezeiten häufig als Traktionsbatterien eingesetzt werden.

Material	Zellspannung [V]	Kapazität [mAh/g]	Gewicht [g/cm^3]
Lithium-Cobaltdioxid ($LiCoO_2$)	3,9	160	5,05
Lithium-Nickeldioxid ($LiNiO_2$)	3,8	220	4,80
Lithium-Manganoxid ($LiMn_2O_2$)	4,0	110	4,20
Lithium-Eisenphosphat ($LiFePO_4$)	3,4	160	3,70

Tabelle 2.1.: Vergleich von Kathodenmaterialien nach [3]

Elektrolyt

Für den Ionentransport zwischen den Elektroden des Akkumulators dient der Elektrolyt als Leitmedium. Neben einer guten Leitfähigkeit für Ionen, muss er sich im Ladespannungsbereich des Akkumulators stabil verhalten. In Lithium-Ionen-Akkumulatoren sind Elektrolyten in verschiedenen Zuständen anzutreffen:

- Flüssig: Der meist organische Elektrolyt besteht aus einem Lithium-Ionen enthaltenen Leitsalz und nicht wässrigen Lösungsmitteln. Wasser darf nicht enthalten sein, da es in Verbindung mit den enthaltenen Salzen zur Bildung von Flusssäure kommen kann und die Elektroden schädigt.
- Polymer: Im Gegensatz zu flüssigen Elektrolyten können Polymere nicht austreten, was die Sicherheit erhöht, jedoch leidet darunter die Ionen-Leitfähigkeit
- Fest: Feste Elektrolyten finden in der Praxis kaum eine Anwendung. Ihre Nachteile sind schlechte Ionen-Leitfähigkeit und schwieriges Einbringen in die Batterie.

Im Elektrolyt bildet sich an der Anode eine Lithium-Ionen durchlässige Grenzschicht. Diese sogenannte SEI¹-Schicht ist für Elektronen undurchlässig und schützt die Elektrode vor Korrosion. Bei Lade- und Entladeprozessen wächst sie weiter und erhöht somit ihren Widerstand. Dadurch ist sie maßgeblich am Alterungsprozess der Zelle beteiligt.

Separator

An den Separator sind hohe Anforderungen gestellt, da er die Elektroden voneinander trennt und somit einen Kurzschluss verhindert. Dabei muss er für Ionen durchlässig bleiben. Bei einer Übertemperatur schmilzt der Separator und unterbricht somit den Ionen-Fluss, wodurch kein Strom mehr fließen kann. Dies wird auch als „shut down“-Mechanismus bezeichnet. Die Zelle ist anschließend zerstört.

Als Material werden in der Regel thermoplastische Kunststoffe wie Polypropylen (PP) oder Polyethylen (PE) verwendet.

¹SEI, Solid Electrolyte Interface

2.2.2. Optische Effekte an Lithium-Eisenphosphat-Batterien

Bei Lithium-Eisenphosphat-Batterien wird LiFePO_4 als Kathodenmaterial und Graphit als Anodenmaterial verwendet. Beim Laden der Zelle wandern die Lithium-Ionen durch den Elektrolyt und den Separator und lagern sich in der Graphitanode ein. Dieser Vorgang wird Interkalation genannt. Beim Entladen, der Deinterkalation, wird dieser Prozess umgekehrt. Die Lithium-Ionen wandern zurück zur Kathode. Durch den ablaufenden Oxidationsprozess werden gleichzeitig Elektronen freigesetzt, die über die elektrische Verbindung zur Kathode fließen. Beim Reduktionsprozess an der Kathode werden diese Elektronen aufgenommen.

Im Folgenden ist von links nach rechts die Reaktionsgleichung der Kathode bei vollständiger Ladung dargestellt. Im Falle der Entladung, wird die Gleichung umgekehrt, also von rechts nach links durchlaufen.



Die Graphitanode nimmt bei der Aufladung Lithium-Ionen auf, für sie gilt folgende Gleichung.



Durch die Änderung der Zusammensetzung der Kathode ändert sich auch ihre Farbe. Während FePO_4 gelb-bräunliche Farbe besitzt, ist LiFePO_4 grau bis schwarz.

Abbildung 2.2 zeigt die optischen Veränderungen an einer Kathode bei der Entladung. Zu erkennen ist ein optischer Effekt, der von außen in Richtung der Mitte der Elektrode eintritt. Hier lagern sich die Lithium-Ionen in die Kathode ein, wodurch eine Verfärbung eintritt.

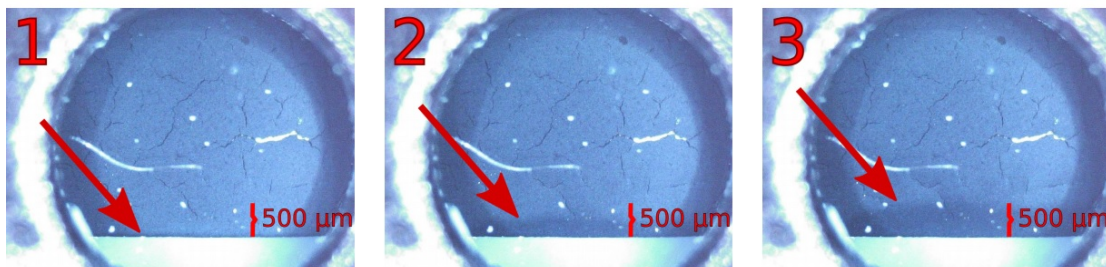


Abbildung 2.2.: Optischer Effekt der Entladung an der Kathode einer Lithium-Eisenphosphat Batterie [4]

2.3. Optische Messtechnik

Eine elektromagnetische Welle besteht aus elektrischen und magnetischen Feldern. Die Wellen unterscheiden sich in ihrer Frequenz und sind in Abbildung 2.3 als elektromagnetischen Spektrum dargestellt. Das für das menschliche Auge sichtbare Licht stellt nur einen kleinen Teil des Spektrums dar.

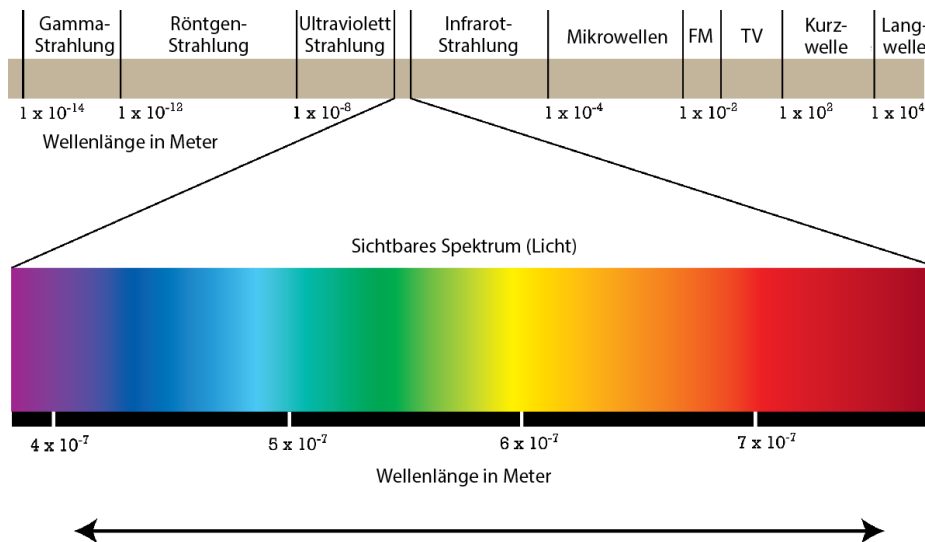


Abbildung 2.3.: Elektromagnetisches Spektrum [5]

Zur optischen Beobachtung von Stoffen gibt es unterschiedliche Methoden. Eine von ihnen wird bei der Pulsoximetrie angewandt. Zur Feststellung des Sauerstoffgehalts im Blut werden elektromagnetische Wellen im roten bis infraroten Bereich durch das menschliche Gewebe, meist ein Finger oder Ohrläppchen, gestrahlt und auf der anderen Seite von einer Photodiode registriert. Hämoglobin, ein eisenhaltiger Proteinkomplex, der in den roten Blutkörperchen Sauerstoff bindet, hat je nach Sauerstoffgehalt eine andere Farbe und somit ein anderes Absorptionsverhalten auf das Licht. Anhand der Messwerte und einer Referenztafel kann das Pulsoximeter dann den Sauerstoffgehalt feststellen.

In dieser Arbeit kann diese Methode jedoch nicht angewendet werden, da die Zellen das Licht vollständig absorbieren. Stattdessen werden die Elektroden der Testzellen von einer Lichtquelle im Bereich des sichtbaren Lichts, sowie der angrenzenden Wellenlängen in den Bereichen von ultravioletter und Infrarotstrahlung angestrahlt und das reflektierte Licht von der Kamera oder dem Spektrometer aufgenommen.

Spiegelreflexion und diffuse Reflexion

Fällt ein Bündel von Lichtstrahlen vom einem Punkt P auf eine glatte Oberfläche, so werden sie mit dem selben Winkel reflektiert, mit dem sie eingefallen sind (Abbildung 2.4 (a)). Für einen Betrachter erscheint es, als würden die Strahlen vom Punkt P' hinter der Oberfläche des reflektierenden Objekts ausgehen. Hierbei wird von einer Spiegelreflexion gesprochen. Sie unterscheidet sich von diffuser Reflexion an rauen Oberflächen, bei der durch die unterschiedliche Ausrichtung der einzelnen Oberflächenpunkte eine Streuung der reflektierten Strahlung entsteht.

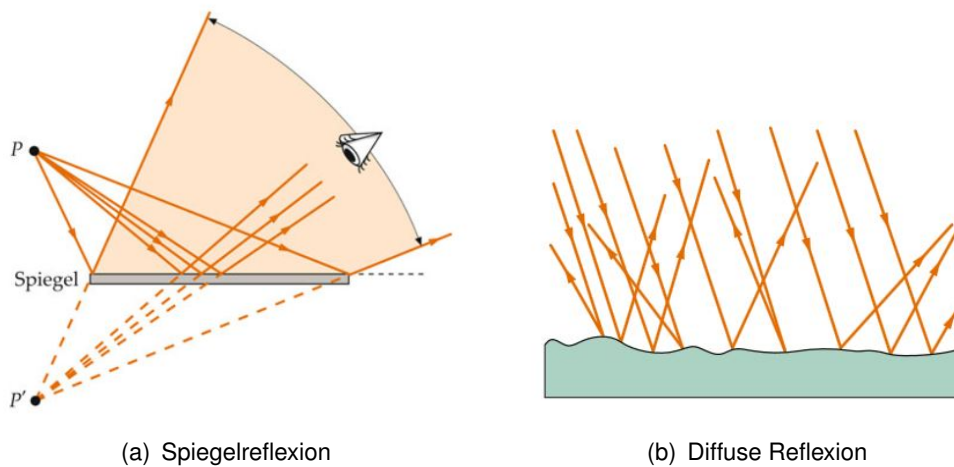


Abbildung 2.4.: Reflexionen an glatten (a) und rauen (b) Oberflächen [6]

2.4. Bildverarbeitung

Zur Bilderfassung der Mikroskopkamera wird das Programm „fswebcam“ genutzt, welches die Bilder im „Portable Network Graphics“-Format, kurz PNG, speichert.

Das PNG-Format ist ein verlustfreies Grafikformat für Rastergrafiken. Gespeichert werden die Farben im „24-Bit RGB + 8-Bit Alpha“-Format, bei dem für jedes Pixel 3 mal 8-Bit Informationen für die Farben Rot, Grün und Blau zur Verfügung stehen. Über additives Mischen der drei Grundfarben ist es möglich, die Farbwahrnehmung des menschlichen Auges nachzubilden. Die Intensitätswerte jeder der drei Farben werden im Bereich von 0 bis 255 gespeichert und ermöglichen es durch Mischen über 16,7 Millionen Farbwerte zu erhalten. Mithilfe von Matlab können die Farbintensitätsverläufe der PNG-Dateien ausgewertet und über die Zeit dargestellt werden.

Der zusätzliche 8-Bit Alpha-Kanal enthält Informationen über die Transparenz des Bildobjekts, ist aber für diese Untersuchungen uninteressant.

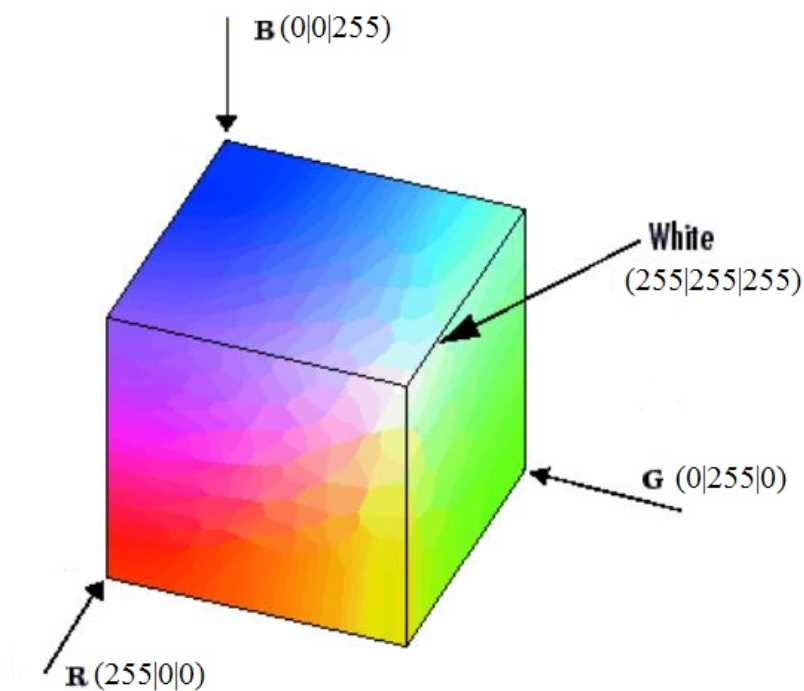


Abbildung 2.5.: Visualisierung des RGB-Farbmodells in Form eines Würfels modifiziert nach [7]

2.5. Referenzwertbildung

Aufgrund von Belichtungsschwankungen im Verlauf der Messungen ist es notwendig die erfassten optischen Messungen durch Verwenden von Referenzwerten zu korrigieren. In dieser Arbeit werden hierzu 2 verschiedene Methoden verwendet, die in diesem Kapitel beschrieben werden.

Kameraaufnahme

Zur Auswertung der Kamerabilder wird neben der Elektrodenfläche auch eine Referenzfläche ausgewählt und die RGB-Werte ausgelesen. Um die Helligkeitsänderungen der Elektrode auf der Referenzfläche abzubilden, sollten die Flächen optisch möglichst gleich beschaffen sein.

Für jede Aufnahme der Messung werden durch ein Matlab-Skript die RGB-Werte je Kanal für beide Flächen aufsummiert. Anschließend werden die RGB-Werte der Elektrodenfläche durch die der Referenzfläche dividiert und mit dem Mittelwert der Referenzfläche über den gesamten Messverlauf multipliziert.

Hier ist die Berechnung durch Matlab exemplarisch am roten Kanal dargestellt:

$$R_{\text{elektrode,korrigiert}} = (R_{\text{elektrode}} ./ R_{\text{ref}}) * \text{mean}(R_{\text{ref}}) \quad (2.3)$$

Spektrometeraufnahme

Bei der Spektrometeraufnahme ist es möglich, die Fehler in der Belichtung durch das Bilden des Mittelwertes über alle Wellenlängen einer Messung zu minimieren. Abschließend wird die Differenz der einzelnen Wellenlängen vom Mittelwert berechnet. Zwar sind hierdurch die Intensitäten des Spektrums nicht mehr darzustellen, in der Auswertung dieser Arbeit ist es jedoch möglich aus den Differenzen zum Mittelwert die Verschiebungen des Spektrums im Bereich der Wellenlängen abzubilden.

Ist „Spektrum“ der Vektor der Wellenlängenintensitäten einer Messaufnahme, lautet die Syntax der Berechnung in Matlab:

$$\text{Spektrum}_{\text{diff}} = \text{Spektrum} - \text{mean}(\text{Spektrum}); \quad (2.4)$$

3. Messplatz

Dieses Kapitel befasst sich mit dem Aufbau der zu messenden Zellen und der Messplätze. Verwendet werden die Messzellen „ECC-Opto-Std“ der Firma EI-Cell GmbH. Diese Zellen werden vom Messsystem mit Licht bestrahlt und die reflektierte Strahlung von einem Spektroskop oder einer Mikroskopkamera aufgenommen. Des Weiteren sorgt der Messaufbau durch Vorgabe einer Spannung an die Elektroden für die Auf- und Entladung der Zelle. Neben dem mechanischen Aufbau wird im Folgenden weiter auf die Messsoftware eingegangen.

3.1. Aufbau des Messplatzes

Im Folgenden wird der Aufbau des Messplatzes sowie der elektrische Aufbau beschrieben. Anschließend wird auf den systematischen Messfehler eingegangen, der durch die Reihenschaltung der Testzelle mit dem externen Amperemeter entsteht.

3.1.1. Aufbau der Messschaltung

Die einzelnen Messschaltungen bestehen aus der Testzelle, dem Zyklersystem mit angeschlossener externer Festplatte, einem DMM4020 Digitalmultimeter zur Strommessung und einer Mikroskopkamera oder Spektrometer mit Lichtquelle. Um die Zyklersysteme über eine Maus und einer Tastatur zu bedienen und nur einen Monitor verwenden zu müssen, sind diese über ein 4-Kanal KVM¹-Switch von Digitus verbunden. Der Anschluss an den Monitor findet über VGA statt. Um die Zyklersysteme an das Switch anzuschließen, ist ein Adaptor von VGA auf HDMI + USB notwendig.

Das Zyklersystem, beziehungsweise der Raspberry Pi 2 Model B, besitzt 4 USB Anschlüsse, an die folgende Geräte angeschlossen sind:

- Mikroskopkamera
- externes Amperemeter

¹KVM: Abkürzung für Keyboard-Video-Mouse

- externe Festplatte
- KVM Switch

Um ein Verrutschen des Kamerabildes zu verhindern, wurde ein Metallgestell entworfen, an welches die Kamera befestigt und der Testzellenhalter mittels Magneten fixiert ist.

Für die Verwendung des Spektrometers ist eine Kappe für die Testzellen vorgesehen, an der die Lichtwellenleiter angeschraubt werden und welche sich mittels Schrauben an der Zelle fixieren lässt. Dabei zu beachten ist, dass der Lichtwellenleiteranschluss genau zentriert wird, um den zu untersuchenden Bereich der Zelle aufzunehmen.

Alle Messungen finden zur Abschirmung von externen Lichtquellen wie Tageslicht oder Deckenbeleuchtung des Labors in dafür angefertigten Holzkästen oder unter Kartons statt.

Abbildung 3.1 zeigt den Laboraufbau des Messsystems ohne elektrischen Aufbau.

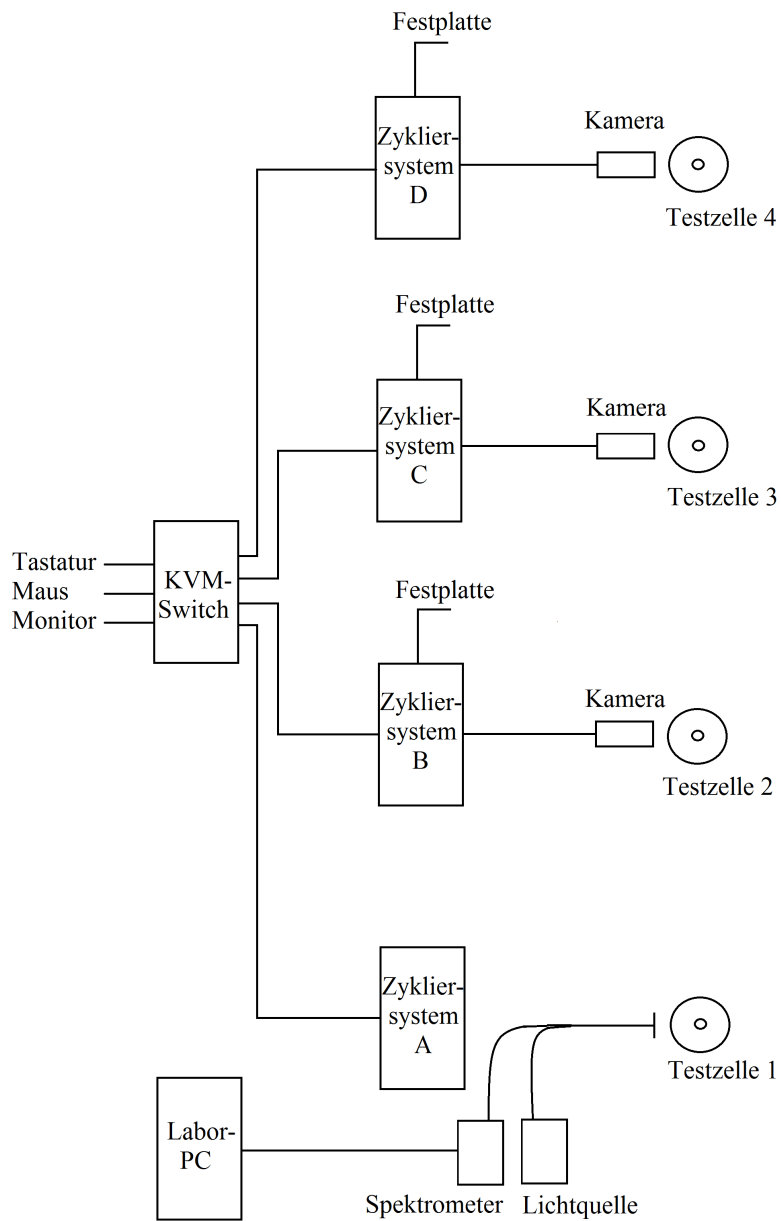


Abbildung 3.1.: Laboraufbau des Messsystems ohne elektrischen Aufbau

3.1.2. Elektrischer Aufbau

Der elektrische Aufbau eines einzelnen Messaufbaus besteht neben dem Zyklersystem mit internem Voltmeter aus der Testzelle und einem externen Amperemeter. Die Gründe für die Verwendung der externen Strommessung sind in Kapitel 3.2 beschrieben. Für die Verdrahtung sind am Zyklersystem sowie an der Testzelle Buchsen für 2 mm Bananenstecker vorgesehen. Da das als Amperemeter verwendete Digitalmultimeter „DMM4020“ von Tektronix nur über 4 mm Anschlüsse verfügt, werden hier Adapter verwendet. In Abbildung 3.2 ist der elektrische Aufbau der Messschaltung dargestellt.

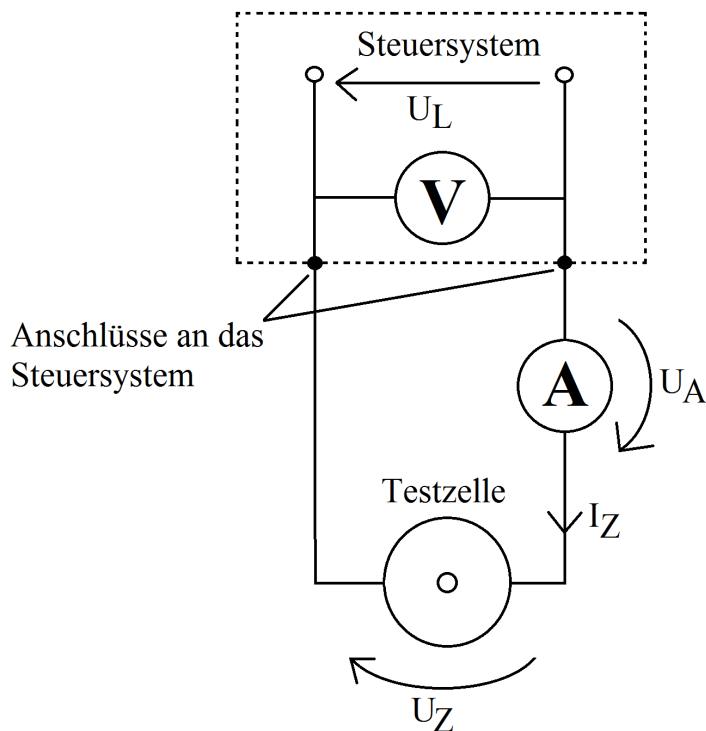


Abbildung 3.2.: Elektrischer Aufbau der Messungen

3.1.3. Systematischer Messfehler

Wie in Abbildung 3.2 dargestellt, erfolgt die Verdrahtung der Reihenschaltung aus Amperemeter und Testzelle parallel zum internen Voltmeter. Die Messungen erfolgen also stromrichtig, jedoch mit einem Fehler in der Spannungsmessung. Die Ladespannung U_L ist gleich der vom Voltmeter gemessenen Spannung U_V . Da über das Amperemeter eine Spannung U_A abfällt, ist die Zellspannung U_Z ungleich der Ladespannung U_L . Die Messungengenauigkeiten lassen sich durch die Angaben im Datenblatt des „DMM4020“ berechnen. Im seinem eingestellten Messbereich bis $200 \mu A$ Gleichstrom fällt über das Amperemeter eine Spannung von maximal 5 mV ab [8].

Die allgemeine Spannungsformel für das Aufladen lautet:

$$U_L = U_V = U_Z + U_A \quad (3.1)$$

Für die Zellspannung ergibt sich somit:

$$U_Z = U_L - U_A \quad (3.2)$$

Die größte Auswirkung hat der Messfehler bei kleinen Entladespannungen. Für U_L beträgt die geringste Spannung in diesen Untersuchungen $1,5 \text{ V}$. Im Falle der Entladung kehrt sich das Vorzeichen der Spannung U_A um, da die Spannung der Zelle jetzt höher ist als die Spannung U_L und sich dadurch der Stromfluss ändert. Die Zellspannung errechnet sich in diesem Fall mit:

$$U_Z = U_L + U_A \quad (3.3)$$

$$U_Z = 1,5 \text{ V} + 5 \text{ mV} = 1,505 \text{ V} \quad (3.4)$$

Der Messfehler beträgt:

$$\frac{U_A}{U_L} = \frac{5 \text{ mV}}{1,5 \text{ V}} = 0,3\% \quad (3.5)$$

Mit $0,3\%$ ist der Messfehler so gering, dass er vernachlässigt werden kann. Andere Faktoren, wie Toleranzen an Bauteilen des Zyklersystems, Ungenauigkeiten der Messgeräte und Verdrahtungswiderstände haben einen größeren Effekt auf die Messungen.

3.2. Zyklersystem

Im Rahmen vorher gegangener Arbeiten im Projekt BATSEN wurde ein System entwickelt, welches den Lade- und Entladezyklus der zu untersuchenden Testzellen steuert, die elektrischen Messdaten aufnimmt [9] [10] [11] und die Ansteuerung der Kamera übernimmt. Die Hauptkomponenten sind ein Raspberry Pi 2 und eine Steuerplatine, entworfen von Torsten Geist [10], die in einem Kasten untergebracht sind. Abbildung 3.3 zeigt den Aufbau mit seinen Komponenten. Der Raspberry Pi 2 ist hier mit einem A markiert und die Steuerplatine mit einem B. Die Spannungsversorgung erfolgt über einen 230V-Anschluss. Für den Raspberry wird ein 5 V Schaltnetzteil [12] vorgeschaltet und die Platine über ein 2x12 V Netztransformator [13]. Diese Bauteile sind hier mit einem C gekennzeichnet.

Zum Anschluss der Testzellen sind die Anschlüsse der Steuerplatine nach außen geführt (D).

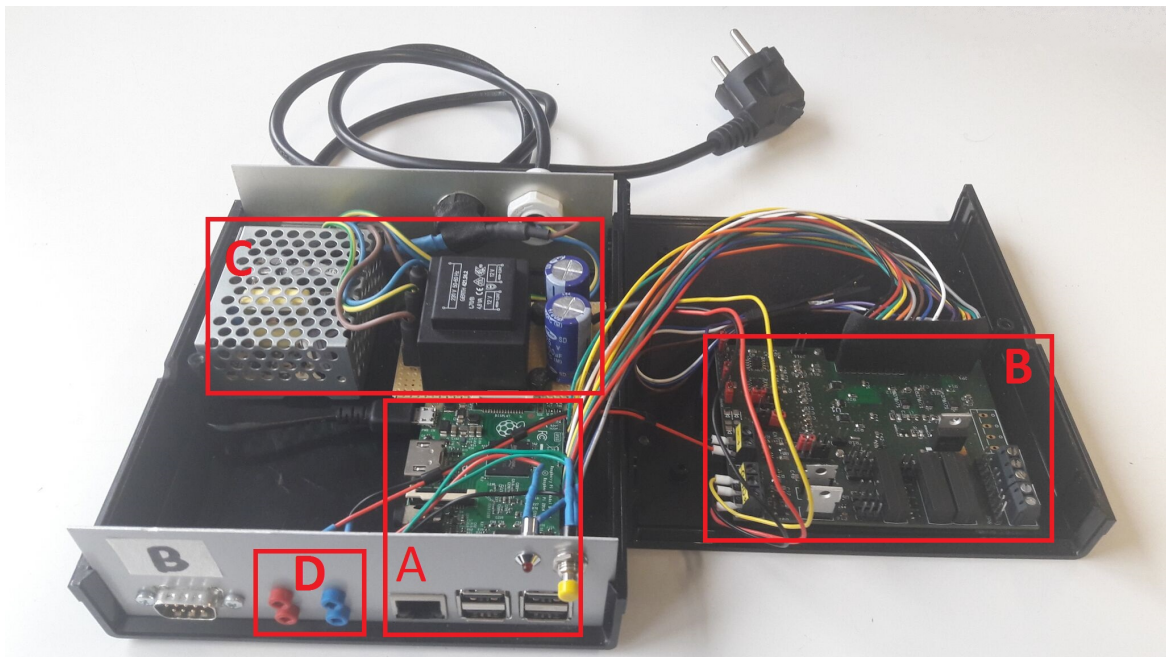


Abbildung 3.3.: Geöffnetes Steuergerät des Messaufbaus

- A Raspberry PI
- B Steuerplatine
- C Spannungsversorgung
- D Anschluss der Testzelle

Die Software zur Regelung der Lade- und Entladezyklen der Testzellen wurde in der Bachelorarbeit von Nico Rieckmann entwickelt. Es gibt verschiedene Modi, die über ein Parameter-File eingegeben werden. Die Folgenden sind möglich [11]:

cC-Regelung

Bei der cC-Regelung (constant Current) wird die Testzelle mit konstantem Strom geladen und entladen. Mit diesem vorher eingegebenen Stromwert läuft die Ladung oder Entladung bis eine ebenfalls definierte Schwellspannung erreicht oder eine ebenfalls zuvor eingegebene Zeit abgelaufen ist.

cV-Regelung

Im cV-Regel (constant Voltage) Modus wird wie im cC-Modus die Zelle geladen bis eine Schwellspannung erreicht oder eine definierte Zeit abgelaufen ist. Hier wird jedoch nicht mit einem konstanten Strom, sondern einer konstanten Spannung geladen.

cCcV-Regelung

Die cCcV-Regelung stellt eine Kombination aus der cC- und cV-Regelung dar. Die Ladung, bzw. Entladung, erfolgt im ersten Schritt mit einem festen Stromwert. Hat die Zellspannung eine bestimmte Schwelle erreicht, wird in den cV-Modus übergegangen. Die Zellspannung wird konstant gehalten, bis eine eingegebene Zeit abgelaufen ist. Der minimale Lade- oder Entladestrom beträgt in dieser Zeit 10% vom im cC-Modus definierten Strom, mindestens jedoch $10\mu\text{A}$.

Zur Regelung braucht das Zykliersystem aktuelle Strom und Spannungswerte, die über je einen ADC ermittelt werden. Zur Auswertung des Versuchs werden diese Messwerte in Textfiles gespeichert. Des Weiteren kann vom System eine Mikroskopkamera angesteuert werden, die in festgelegten Sequenzen Bilder der zu untersuchenden Elektrode macht. Da die Messungen in vielen Fällen mehrere Tage laufen und der Speicherplatz auf der micro-SD des Raspberry Pi sehr begrenzt ist, werden diese Datensätze auf einer externen Festplatte gespeichert.

Externe Strommessung

Abbildung 3.4 zeigt den Vergleich der Messgenauigkeiten von interner Strommessung und einem digitalen Multimeter. Hierzu wurde ein Tektronix DMM 4020, mit einer Auflösung im Messbereich bis $200 \mu\text{A}$ DC von bis zu $0,001 \mu\text{A}$, verwendet [8]. Die Messung wurde bei einem Sollstrom von $20 \mu\text{A}$ über einem $4,7 \text{k}\Omega$ Widerstand durchgeführt.

Deutlich zu erkennen ist das Messrauschen des internen Messgeräts. Dies führt dazu, dass es die Regelung nicht schafft, genau $20 \mu\text{A}$ einzustellen. Im Gegensatz dazu stellt die Regelung bei Verwendung des digitalen Multimeters im eingeschwungenen Zustand einen Wert ein, der sich ohne äußere Einflüsse nicht ändert. Aus diesem Grund wird in diesen Versuchen das Tektronix DMM4020 verwendet.

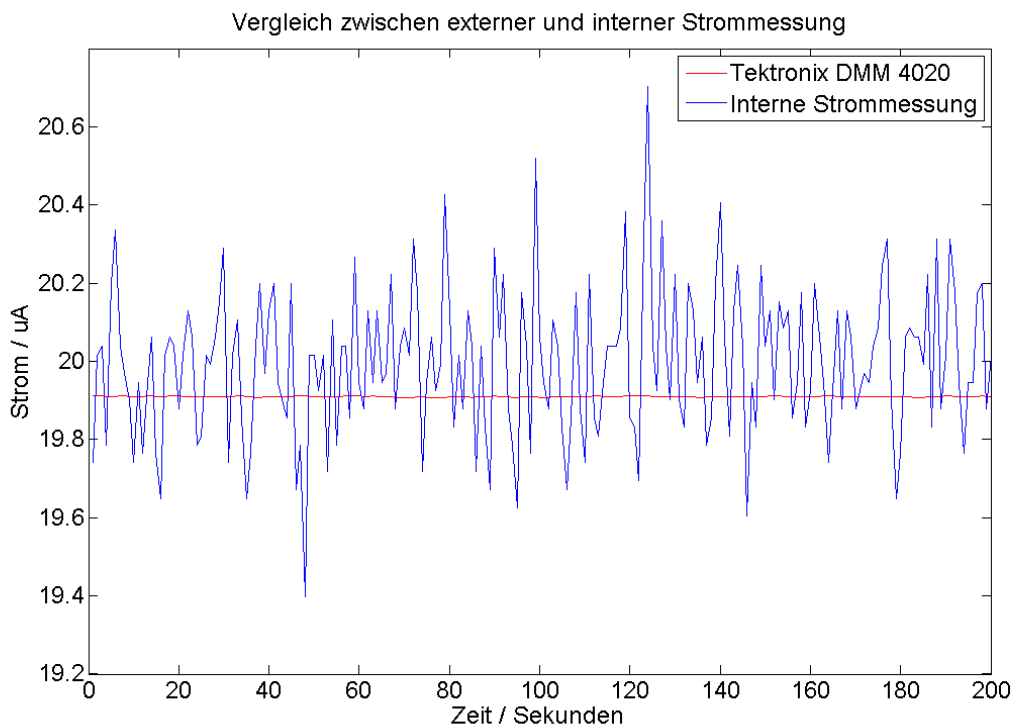


Abbildung 3.4.: Vergleich der Messgenauigkeiten von internem und externem Messgerät bei $20 \mu\text{A}$ in eingeschwungenem Zustand an einem $4,7 \text{k}\Omega$ Widerstand

3.3. Optische Messungen

Die optische Beobachtung der Testzellen erfolgt in dieser Arbeit mittels zweier unterschiedlicher Methoden. Zum einen werden Mikroskopkameras verwendet, zum anderen die optischen Veränderungen der Elektrode mit einem Spektrometer aufgenommen. Beide Vorgehensweisen werden im Folgenden erläutert.

3.3.1. Mikroskopkamera

Die Bilderfassung der Elektroden erfolgt durch die Mikroskopkamera „MZ902“ von oowltech. Die maximal 200fache Vergrößerung der erfassten Elektrode wird vom Abstand der Kameralinse bestimmt. Anschließend muss das Bild über die Drehung des gummierten Rades an der Kamera scharf gestellt werden. Da die Messungen in einem bestenfalls lichtundurchlässigem Raum stattfinden, werden zur Belichtung die 8 LED der Kamera genutzt. Diese verfügt über die Helligkeitsstufen 0 bis 2, wobei Stufe 2 die hellste ist und Stufe 0 das Ausstellen der Belichtung bedeutet. Tests mit der Kamera haben gezeigt, dass es bei der Stufe 2 aufgrund der Nähe der Testobjekte zu einer Überbelichtung kommen kann. Aus diesem Grund wurde die Belichtungsstufe 1 gewählt [14].

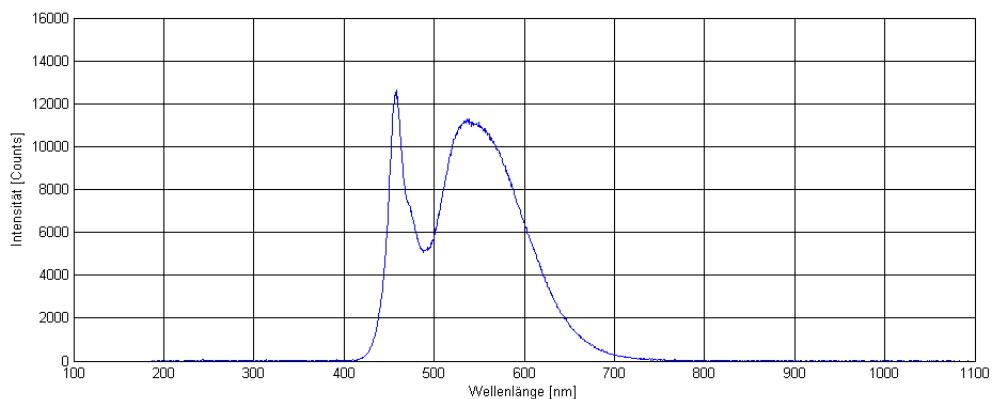


Abbildung 3.5.: Spektrum der LED-Beleuchtung der Mikroskopkamera

In Abbildung 3.5 ist das Spektrum der Kamerabeleuchtung dargestellt. Heutzutage ist es nicht möglich eine LED herzustellen, die direkt weißes Licht erzeugt. Aus diesem Grund wird eine mit Phosphor beschichtete blaue LED verwendet. In dem dargestellten Spektrum ist die erste Spitze bei circa 450 nm das blaue von der LED erzeugte Licht. Das angestrahlte

Phosphor erzeugt die weitere Erhebung im Spektrum zwischen 500 nm und 650 nm. Durch das additive Mischen der beiden Spektren, erscheint das Licht weiß.

Angegeben ist die Auflösung der Kamera mit 9 Megapixel, effektiv wird jedoch nur eine physikalische Auflösung von 800x600 Pixel durch den Sensor erzeugt. Auf 9 Megapixel kann das Bild interpoliert werden, was jedoch zu einem erheblichen Schärfeverlust führt. Für die Messungen wurde deshalb die physikalische Auflösung des Sensors genutzt.

Des Weiteren ist es der Kamerasoftware möglich, einen Weißabgleich und Gegenlichtkorrektur zu machen. In der Praxis haben sich diese beiden Einstellungen als problematisch erwiesen, da sie die Originalaufnahmen verfälschen, weshalb beide deaktiviert wurden.

Die Ansteuerung der Kamera erfolgt über das Programm „fswebcam“, welches die Bilder im PNG-Format speichert.

3.3.2. Spektrometer

Zur Aufnahme der Spektren wird das Spektrometer „HR2000+“ von Ocean Optic verwendet. In seinem Arbeitsbereich von 187 nm im ultravioletten Bereich bis 1091 nm Infrarotstrahlung werden 2048 Pixel des Spektrums mit einer optischen Auflösung von 0,035 bis 6,8 nm aufgenommen und über einen USB 2.0 Anschluss auf einen Computer übertragen. Über die Software „OceanView“ werden die Daten ausgewertet und abgespeichert. In Abbildung 3.6 ist der Aufbau des Spektrometers dargestellt. Es besteht aus folgenden Bauteilen:

1. SMA-Anschluss an den Lichtleiter
2. Schlitzblende
3. Filter
4. Kollimator-Linse
5. Beugungsgitter
6. Fokussierlinse
7. Sammellinse
8. CCD¹-Sensor

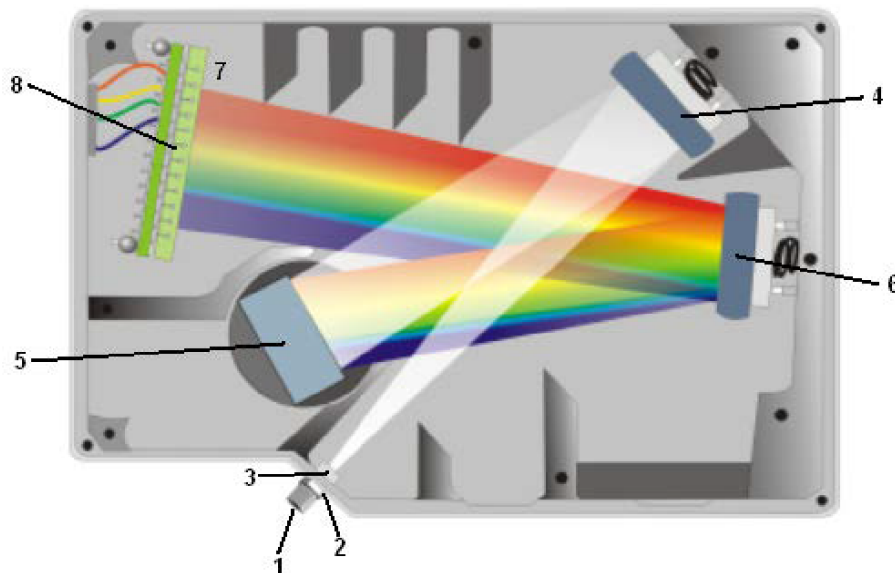


Abbildung 3.6.: Das Spektrometer „HR2000+“ mit seinen Komponenten nach [15]

¹charge-coupled device, engl. für ladungsgekoppeltes Bauteil

An den SMA-Anschluss des Spektrometers wird das reflektierte Licht der Testzelle eingespeist und die Menge und Strahlrichtung durch eine Schlitzblende reguliert. Um nur die relevanten Wellenlängen aufzunehmen, durchläuft der Strahl einen optischen Bandpass und wird von einer Kollimator-Linse parallelisiert. Zur Aufspaltung des Lichts in seine Wellenlängen dient ein optisches Gitter. Diese sind in verschiedenen Gitterdichten erhältlich, wodurch die Auflösung variiert werden kann. Über eine Fokussierlinse und eine Sammellinse gelangt das aufgespaltene Licht auf einen CCD-Sensor. Dieses lichtempfindliche Bauteil besteht aus einer Reihe von Photodioden, die mittels des photoelektrischen Effekts arbeiten. Trifft Licht auf eine Photodiode, werden Elektronen durch die eintretenden Photonen aus dem Silizium geschlagen. Je nach Energie und Anzahl der Photonen entsteht eine Ladung. Diese wird in Kondensatoren gespeichert und kann gemessen und digitalisiert werden.

Je nach Strahlungsintensität der Lampe und der Entfernung zwischen Testzelle und Aus- und Eintrittspunkt in den Lichtleiter ändert sich auch die Intensität des gemessenen Lichts. Um diese Werte in einen sinnvollen Bereich zu bringen, ist in der PC-Software die Integrationszeit zwischen 1 ms und 65 s einstellbar. Die Datenübertragung zum Computer ist über eine serielle Schnittstelle oder einen USB 2.0 Anschluss möglich. Der serielle Anschluss braucht zum Übertragen einen kompletten Datensatzes jedoch 600 ms, wodurch in diesem Fall nicht der komplette Bereich der einstellbaren Integrationszeit genutzt werden kann. In diesen Versuchen wurde stets der USB 2.0 Anschluss genutzt, wodurch es möglich war, den kompletten Stellbereich zu nutzen.

Lichtquellen

Die Wellenlängenbereiche, die vom Spektrometer aufgenommen werden und untersucht werden können, sind abhängig von der Lichtquelle. Nur Wellenlängen, die von ihr ausgestrahlt werden, können auch von der Testzelle reflektiert werden. Im Labor an der HAW Hamburg wird dafür eine Leuchte „SL1 Visible Light Source“ von StellarNet verwendet. Das breitbandige Spektrum reicht von 350 nm bis 2200 nm [16], die Intensitäten der einzelnen Wellenlängen unterscheiden sich jedoch erheblich. So ist eine Untersuchung im Bereich der Wellenlängen unter 500 nm aufgrund der schwachen Strahlung problematisch.

Im Fachbereich Chemie der Universität Hamburg steht eine Deuteriumlampe „L10290 High Power UV-VIS fiber light source“ von Hamamatsu zur Verfügung. In dieser Lampe steht das Gas Deuterium unter leichtem Druck. Wird an die Elektroden eine Spannung angelegt, entsteht ein Lichtbogen, mit dem es möglich ist Wellenlängen im Bereich von 200 nm bis 1600 nm zu untersuchen [17].

In Abbildung 3.7 sind die Spektren beider Lampen bei einer Reflexion auf einer weißen Fläche dargestellt. Dies stellt die Lampenspektren jedoch nur annähernd da. Um die Spektren unverfälscht aufnehmen zu können muss ein verlustfreier Spiegel verwendet werden, denn auch handelsübliche Spiegel mit einer Aluminiumbeschichtung absorbieren beim Reflektieren des Lichts Teile der Strahlung.

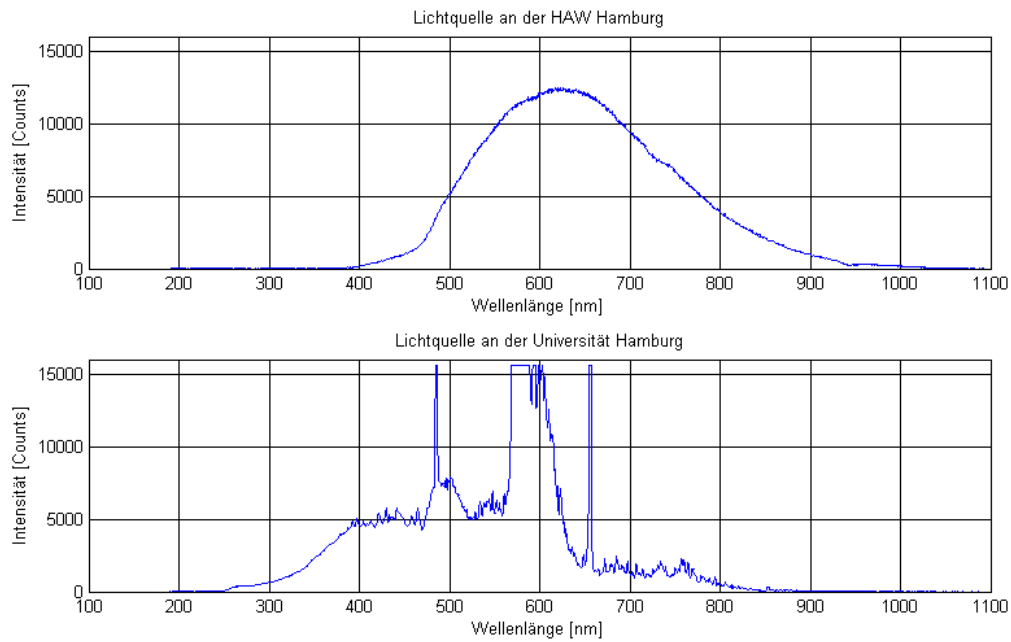


Abbildung 3.7.: Vergleich der Leuchtenspektren, reflektiert von einer weißen Fläche

3.4. Messsoftware

Zum Durchführen der Messungen sind Einstellungen in der Software nötig. So wird das Auf- und Entladeverhalten in einem Parameterfile festgelegt und allgemeine Einstellungen im Shell-Skript geändert. Durch Ausführen dieses Shell-Skripts werden die Messungen anschließend gestartet. Die weitere Software des Zyklersystems ist in der Bachelorarbeit von Herrn Rieckmann [11] beschrieben, jedoch für das Durchführen der Messungen nicht relevant. Zum Betreiben des Spektrometers ist die Herstellersoftware „OceanView“ von Ocean Optics notwendig und wird in diesem Kapitel ebenfalls beschrieben.

3.4.1. Software des Zyklersystems und Kamera

bc_param.cfg

Das Parameterfile „bc_param.cfg“ enthält alle Einstellungen, die den Zykletrieb der Zelle betreffen und wird beim Start der Software ausgelesen. Die Daten sind in drei Gruppen aufgeteilt:

1. **batterygroup**
Enthält Daten über die Zelle.
2. **limitgroup**
Gibt Grenzen an, die bei der Eingabe in der cyclegroup eingehalten werden müssen.
3. **cyclegroup**
In dieser Datengruppe wird der Zykletrieb definiert.

Die Parameter der batterygroup und limitgroup sind an die Testzellen angepasst und ändern sich bei den Messungen nicht. Die notwendigen Eingaben in der cyclegroup werden in folgender Tabelle zusammengefasst. Die Eingaben für die Farbansteuerung der LED sind vernachlässigt, da diese noch nicht in der Software implementiert wurden.

Parameter	Beschreibung
mode	Die Eingabe entscheidet, welcher Zykletriebmodus durchlaufen wird (Abschnitt 3.2)
mode_time	Zeitliche Begrenzung der Modi in Sekunden
threshold_voltage	Ladeschlussspannung, Schwellspannung in Volt
nominal_current	Ladestrom in Ampere
convalescence	Erholungszeit nach Durchlaufen des Zyklus in Sekunden
cycles	Gibt an, wie oft die Zyklen durchlaufen werden

Tabelle 3.1.: Eingaben in das Parameterfile für den Zykletrieb

start_all.sh

Das Shell-Skript enthält Angaben zum Speicherort der Messungen und zum Ausführen der Programme. Es erstellt die Ordnerstruktur in der die Daten abgespeichert werden und kopiert alle eingestellten Parameter in eine „cycle_plan.log“-Datei. Anschließend werden die einzelnen Programme des Zyklersystems aufgerufen. Die hierzu benötigten Parameter finden sich in der Bachelorarbeit von Herrn Rieckmann [11].

3.4.2. Spektrometersoftware

Zur Aufnahme der Spektrometerdaten wird die Software „OceanView“ von Ocean Optics verwendet. Die Daten werden in Echtzeit erfasst und je nach Einstellungen abgespeichert. Abbildung 3.8 zeigt die grafische Oberfläche der Software.

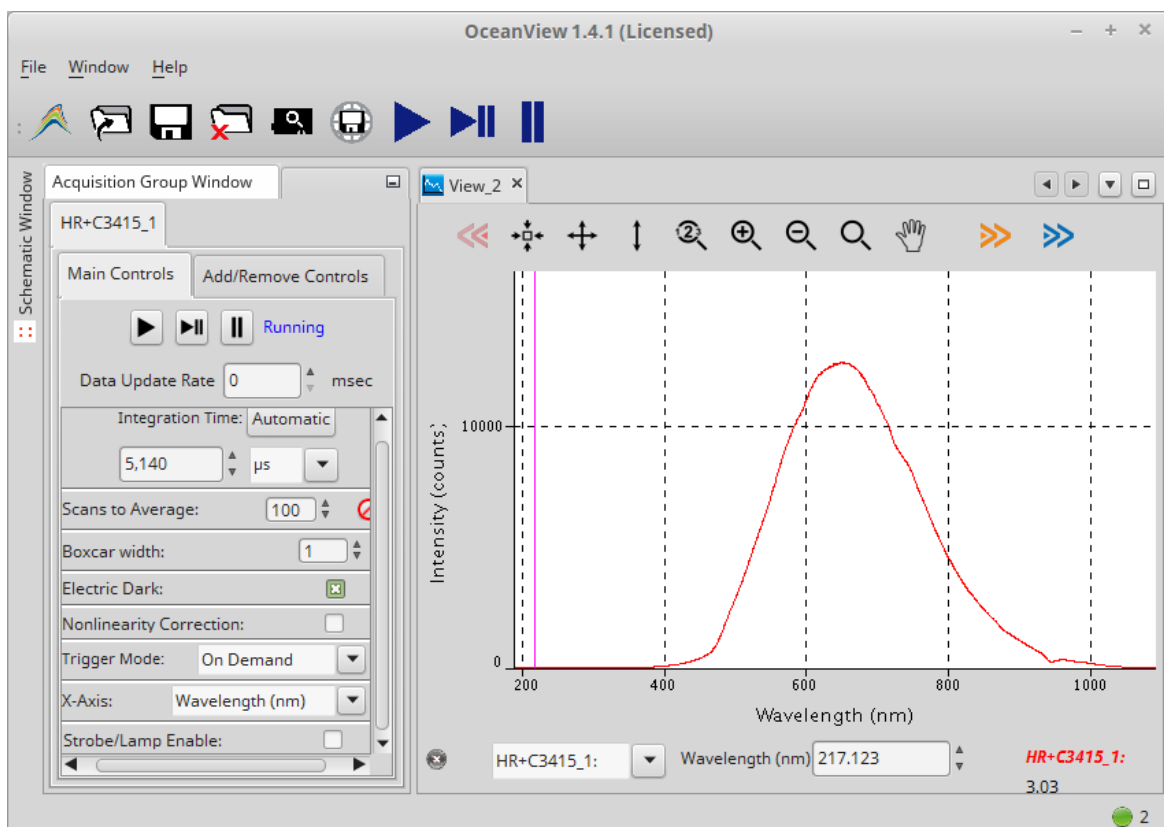


Abbildung 3.8.: Grafische Oberfläche der Software „OceanView“ [18]

Vor dem Starten der Messung sind folgende Eingaben zu tätigen:

Integration Time:	Die Integrationszeit gibt die Zeit an, in der die Photonen vom CCD-Sensor erfasst werden und bestimmt somit die Intensität der Aufnahme.
Scans to Average:	Zur Rauschunterdrückung können mehrere Aufnahmen gemacht werden, deren Durchschnitt die Ausgabe ist.
Boxcar width:	Dies dient dem Glätten der Aufnahme. Jedes aufgenommene Pixel wird über die angegebene Anzahl von benachbarten Pixeln gemittelt.
Electric Dark:	Um Änderungen im zeitlichen Offset zu verhindern kann die Funktion „Elektric Dark“ genutzt werden. Dazu erfassen abgedunkelte Pixel den Offset, der anschließend von den einzelnen Pixeln subtrahiert wird.
Nonlinearity Correction:	Mit dieser Funktion wird überprüft, wie konstant die gemessene Strahlung über den zeitlichen Verlauf von dem Spektrometer erfasst wird. Die Nichtlinearität des Detektors kann somit korrigiert werden.
Trigger Mode:	Der Trigger Mode gibt an, wann die Messaufnahme gestartet wird.
X-Axis:	Zur Einstellung der Intensitäten über der Wellenlänge oder der Frequenz.
Strobe/Lamp Enable:	Hiermit können zeitgleich mehrere Messungen mit demselben Spektrometer durchgeführt werden.

Tabelle 3.2.: Grundeinstellungen der Spektrometersoftware „OceanView“

Da sich die Intensitäten bei verschiedenen Zellen und Lichtquellen sehr unterscheiden, wird die Integrationszeit automatisch ermittelt. Zu beachten ist jedoch, dass eine längere Zeit auch eine höhere Intensität mit sich bringt und diese maximal 15532 Counts betragen darf, da das Spektrometer hier in Sättigung geht.

In der Einstellung „Scans to Average“ werden 5 Messungen gewählt, um ein mögliches Rauschen zu unterdrücken. Eine mögliche Drift der Messung soll mit der „Electric dark correction“ verhindert werden. Die „Boxcar width“ wird auf 0 gesetzt, um die Messung nicht zu verfälschen.

Die Intensitäten der Wellenlängen sollen in dieser Arbeit ausgewertet werden. Deshalb werden diese als X-Achse angenommen.

Unter „Configure graph saving“ lassen sich die Speichereinstellungen verwalten. Abbildung 3.9 zeigt die für die Messungen gewählten Einstellungen. Gespeichert werden die Daten als Textdatei im ASCII-Format mit Header. In diesem Header sind Informationen zum Zeitpunkt der Messung und den gewählten Grundeinstellungen (Abschnitt 3.2) enthalten. Als Name wird die Bezeichnung der aktuellen Messung eingetragen. Durch den „File counter“ wird hier eine 5-stellige Zahl angehängt, die bei jeder Messungen um eins erhöht wird.

Da es sich bei den optischen Änderungen der Elektroden um langsame Effekte handelt, reicht es alle 10 Sekunden eine Aufnahme des Spektrums zu machen. Als Abbruchkriterium wird eine hohe Anzahl von Stunden gewählt und die Messung bei Bedarf von Hand gestoppt.

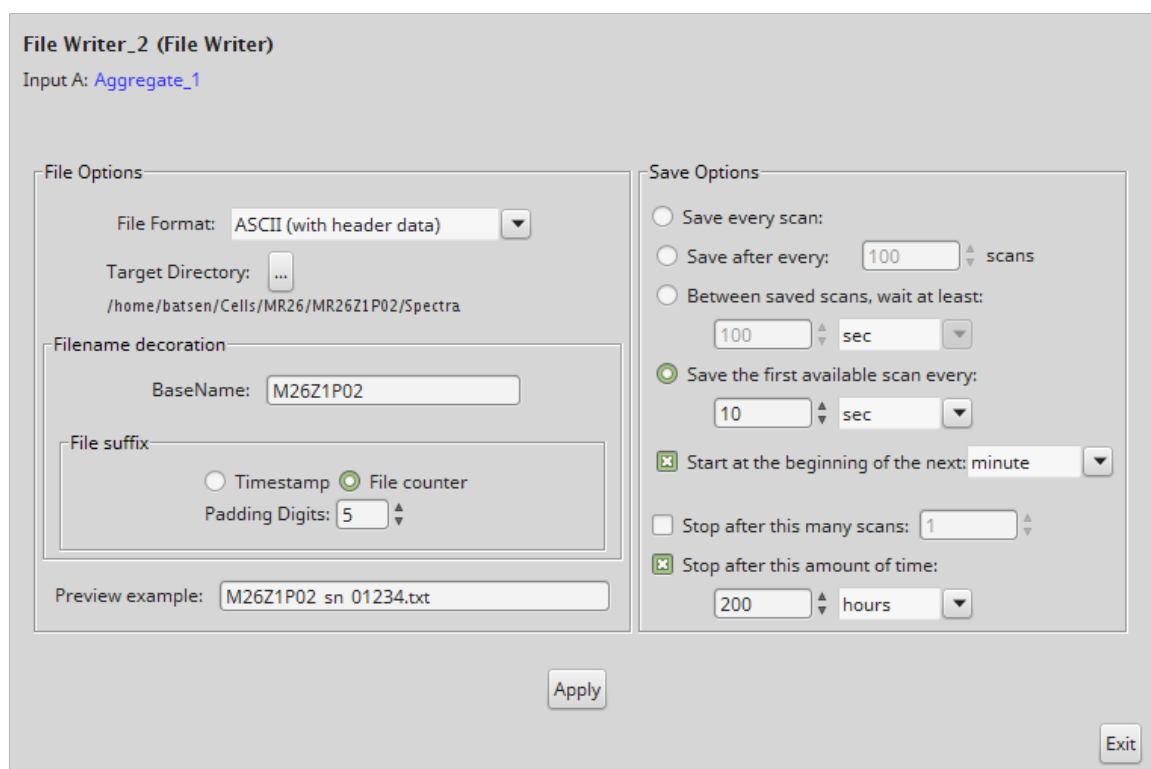


Abbildung 3.9.: Grafische Oberfläche der Software „OceanView“ [18]

3.5. Aufbau der Zellen

Die in dieser Arbeit verwendeten Messzellen „ECC-Opto-Std“ wurden von der Firma El-Cell speziell zur optischen Beobachtung von Elektroden entwickelt. Diese Zellen stellen das Grundgerüst dar, in die das Aktivmaterial eingebracht wird. Da an sie hohe thermische und

chemische Ansprüche gestellt werden, bestehen sie aus Edelstahl, PEEK¹ und EPDM². Abbildung 3.10 zeigt den Aufbau der Testzelle als Explosionszeichnung.

Um einen sicheren und stabilen Versuchsaufbau zu gewährleisten, wird die Testzelle auf einem Stativ platziert. Um weitere Stabilität zu erreichen, werden in diesen Versuchen die Füße des Stativs mit Magneten an ein Gestell befestigt. Dies verhindert ein Verrutschen der Testzelle, um zu gewährleisten, dass sie bei jeder Aufnahme im gleichen Bereich der Kameraauflösung liegt.

Zusätzlich zur Kathode und Anode besitzt die Testzelle eine Referenzelektrode, die in diesen Versuchen jedoch keine Verwendung fand. Der Anschluss der Elektroden an den Messaufbau erfolgt über die dafür vorgesehenen 2mm-Laborstecker-Anschlüsse.

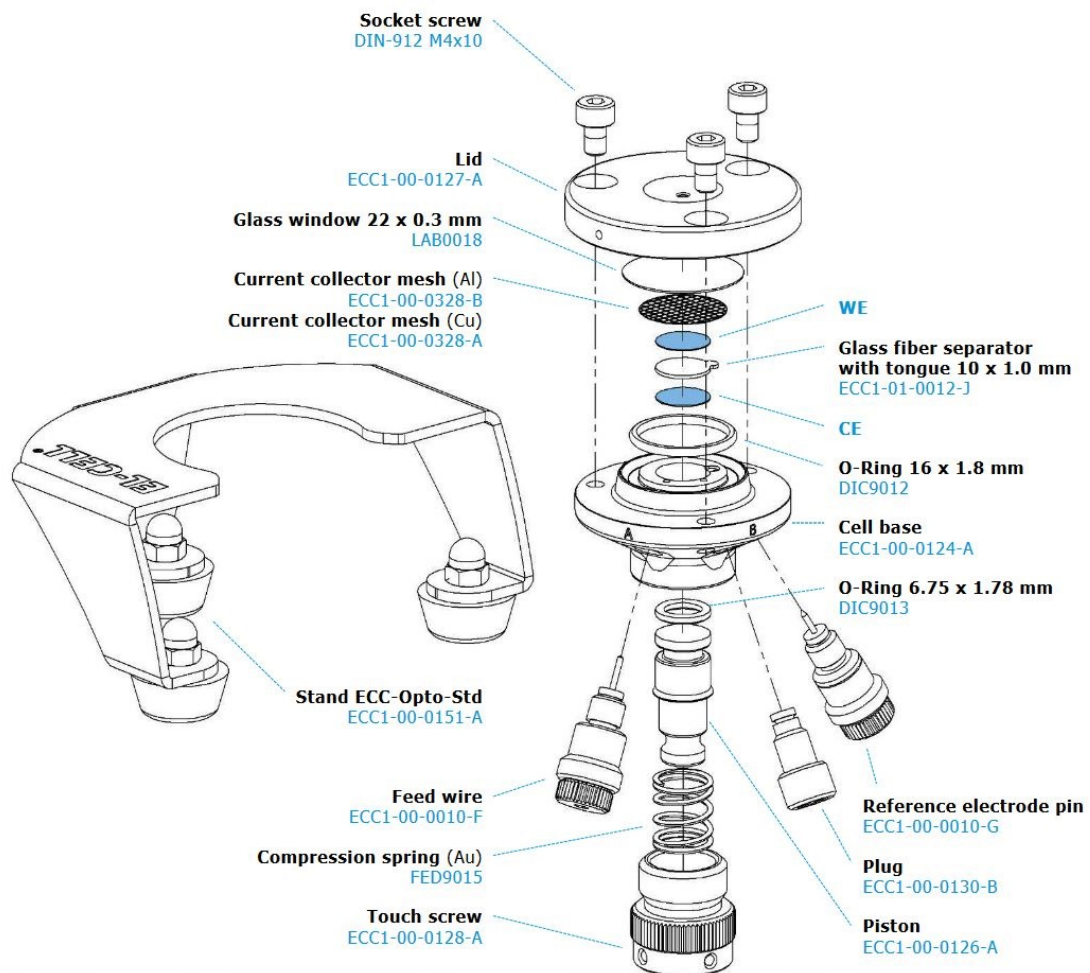


Abbildung 3.10.: Explosionszeichnung der Testzelle „ECC-Opto-Std“ [19]

¹PEEK, Polyetheretherketon, ein thermoplastischer Kunststoff

²EPDM, Ethylen-Propylen-Dien-Kautschuk, ein terpolymeres Elastomer (Gummi)

Aufbau der Testzellen

Da sich am Lithium aufgrund seiner hohen Reaktivität mit feuchter Luft sehr schnell eine Lithiumhydroxid-Schicht bilden würde, muss der Aufbau der Zellen unter Schutzatmosphäre geschehen. Hierzu wird ein Handschuhkasten [20] der Firma M. Braun Inertgas-Systeme GmbH verwendet. Das hierin enthaltene Inertgas Argon steht unter leichtem Überdruck, so dass auch bei kleinen Lecks am Kasten keine Luft eindringen kann. Über eine Schleuse gelangen die Bauteile der Zelle und das Werkzeug hinein.

Der Aufbau der Testzelle erfolgt in vier Einzelschritten und wird im Folgenden beschrieben.



Abbildung 3.11.: Aufbau der Testzellen nach [18]

1. Einsetzen der Anode
2. Einbringen des Separators
3. Einsetzen der Kathode
4. Füllen der Zelle mit dem Elektrolyten

1. Da in diesen Versuchen die Kathode untersucht werden soll, wird die Anode zuerst in die Testzelle platziert.
2. Anschließend wird hierauf der Separator, in Form einer Scheibe aus Glaswolle, eingesetzt.
3. Die zu beobachtende Kathode wird eingelegt und die Testzelle geschlossen.
4. Durch den dafür vorgesehenen Stutzen wird der Elektrolyt eingebracht. Hierzu wird dieser in eine Spritze gefüllt, mit der anschließend das Argon aus dem Inneren der Zelle gesaugt wird. Durch den entstehenden Unterdruck füllt sich die Zelle beim Entspannen der Spritze mit dem Elektrolyten.

Die zu untersuchenden Zellen werden mit unterschiedlichen Kathodenmaterialien aufgebaut. Die Tabelle 5.1 zeigt eine vollständige Übersicht aller in den Messungen verwendeten Testzellen.

3.6. Aufgetretene Probleme und deren Lösungen

Da es im Laufe der Messungen immer wieder zu Problemen mit der Bildaufnahme durch die Mikroskopkamera kam, bei denen die Aufnahme abgebrochen ist, wurden entsprechende Zeilen des Shell-Skripts auskommentiert und die Aufnahme manuell in einem Terminal gestartet. Durch diese Maßnahme konnten diese Probleme behoben werden.

Ein weiteres Problem ergab sich bei der Nutzung des internen Voltmeters parallel zum Betrieb des externen Amperemeters. In diesen Fällen war der Analog-Digital-Wandler der internen Strommessung nicht initialisiert, sein Chip-Select lag permanent auf low, weshalb er auf die Datenleitung schrieb. Die Daten des Analog-Digital-Wandlers des internen Voltmeters, die auf derselben Datenleitung übertragen wurden, wurden dadurch unbrauchbar. Die Regelung braucht zum Ansteuern der Testzellen den aktuellen Spannungswert, der somit nicht zuverlässig war. Die Lösung ist eine Initialisierung der beiden internen ADC, auch wenn die interne Strommessung nicht genutzt wird. Im Skript zeigt der Pointer der Strommessung auf den ungenutzten Temperatursocket. Dieser gibt keine sinnvollen Ergebnisse, die Daten werden deshalb nicht gespeichert.

Im Verlauf der Arbeit kam es zu Problemen mit dem von Herrn Grießbach [9] entworfenen und in Abschnitt 4.2.2 beschriebenen TimestampMatcher. Diese Probleme traten auf, wenn die Zeit zwischen den Aufnahmen der Mikroskopkamera oder des Spektrometers auf eine Sekunde eingestellt war. Der TimestampMatcher konnte die Zeitstempel der elektrischen und optischen Daten nicht mehr richtig zuordnen und gab entweder immer wieder die gleichen Werte aus oder machte gar keine Ausgabe. Durch Löschen jeder zweiten Aufnahme der Mikroskopkamera oder des Spektrometers konnte das Problem vorläufig behoben werden. Wünschenswert wäre jedoch eine Lösung in der keine Daten verloren gehen. Ohne den TimestampMatcher zu bearbeiten, konnte diese jedoch noch nicht gefunden werden.

4. Datenverarbeitung

Um die Rohdaten des Stroms, der Spannung und der optischen Erfassung auswerten zu können, sind einige Bearbeitungsschritte notwendig. Neben der Rauschunterdrückung liegt dabei das Hauptaugenmerk auf einer sinnvollen Darstellung der Ergebnisse. In diesem Kapitel werden die Konzepte und die dafür erstellte Software erläutert.

4.1. Konzepte

4.1.1. Rauschunterdrückung

Durch äußere Einflüsse und Ungenauigkeiten im Messsystem tritt in den Spektrometerdaten ein ungewolltes Rauschen auf. Um die Daten dennoch grafisch sinnvoll darstellen zu können werden diese geglättet.

In Matlab steht hierfür die Smooth-Funktion zur Verfügung. Sie glättet die einzelnen Wellenlängenintensitäten über ihre benachbarten Wellenlängen.

Syntax:

$$yy = smooth(y, span) \tag{4.1}$$

Anstelle von „y“ wird der Vektor der verrauschten Spektrometerdaten eingegeben. „Span“ gibt an, über welche Anzahl von benachbarten Wellenlängen geglättet wird. Diese Zahl sollte ungerade sein, sonst wird automatisch um 1 reduziert. Ist gar kein Wert angegeben, wird er als 5 angenommen. Die verbesserten Werte werden in Form eines Vektors als „yy“ wieder ausgegeben.

Folgende Berechnung gilt für „span“ = 3:

$$\begin{aligned}yy(1) &= y(1) \\yy(2) &= (y(1) + y(2) + y(3))/3 \\yy(3) &= (y(2) + y(3) + y(4))/3\end{aligned}\tag{4.2}$$

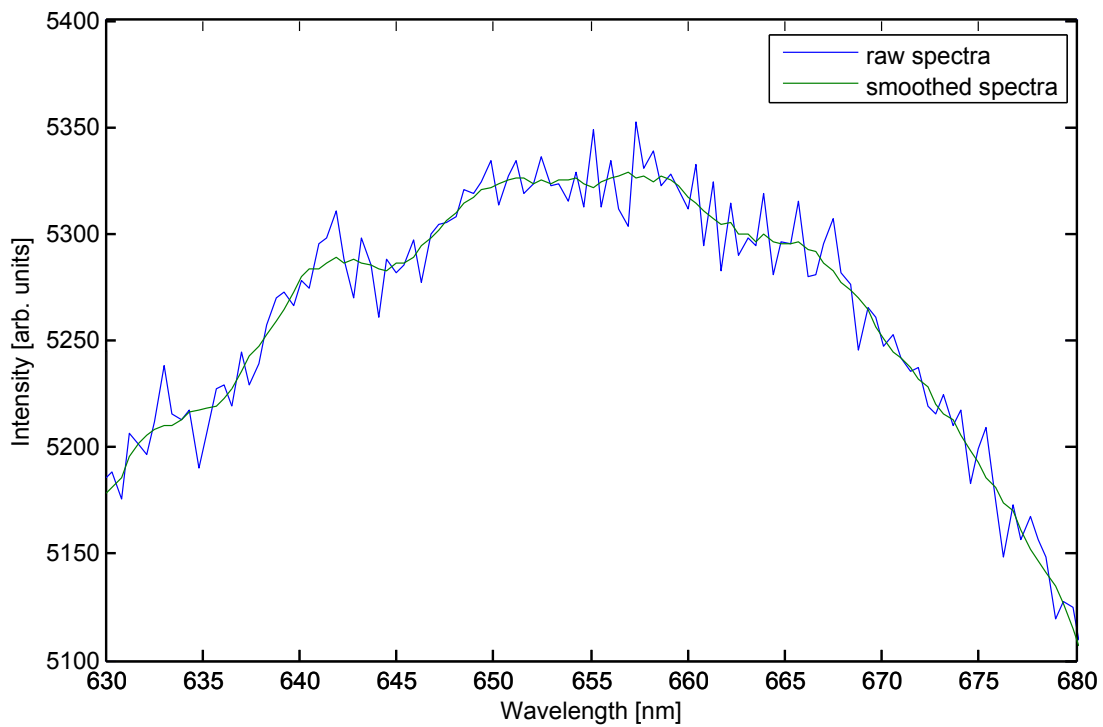


Abbildung 4.1.: Vergleich von ungeglättetem zu über 9 Messwerte geglättetem Spektrum

4.1.2. Ableitung

In den Messungen mit dem Spektrometer kann es zu Drifts in den Intensitäten kommen. Abbildung 4.2(a) zeigt das Abdriften der Intensität bei einer von Herrn Pan durchgeführten Messung [18]. Der Lade- und Entladezyklus ist relativ konstant, die Intensität hingegen fällt nach ca. 13 Stunden stark ab. Um den Zusammenhang zwischen Ladung und Intensität dennoch beobachten zu können, werden die Differenz- und die Signumfunktion genutzt. Die Differenzfunktion berechnet die Differenz von aufeinander folgenden Elementen des Intensitätsvektors einer Wellenlänge. Mit „X“ als Eingangsvektor und „Y“ als Ausgangsvektor lautet die Syntax in Matlab [21]:

$$Y = \text{diff}(X) \quad (4.3)$$

Da immer die Differenzen zwischen zwei Elementen berechnet werden, ist der Ausgangsvektor um ein Element kleiner. Im Folgenden ist die Berechnung für ein Vektor der Länge „m“ dargestellt:

$$Y = [X(2) - X(1) \quad X(3) - X(2) \quad \dots \quad X(m) - X(m - 1)] \quad (4.4)$$

Auf den Differenzenvektor wird anschließend die Signumfunktion angewendet. Sie untersucht die Elemente des Vektors auf ihre Vorzeichen. Die Matlab-Syntax lautet [21]:

$$Y = \text{sign}(X) \quad (4.5)$$

Definiert ist die Ausgabe der Signumfunktion wie folgt:

$$Y = \begin{cases} +1 & \text{wenn } X > 0 \\ 0 & \text{wenn } X = 0 \\ -1 & \text{wenn } X < 0 \end{cases} \quad (4.6)$$

Wie in Abbildung 4.2 dargestellt ist durch Anwenden dieser Methode der Zusammenhang zwischen Ladung und Intensitäten besser zu erkennen. Jedoch ist in dieser Darstellung die Stärke der Veränderungen nicht mehr ersichtlich.

4.1.3. Mittelwertbildung

Zur besseren grafischen Darstellung der Lade- und Entladekurven, wird für jeden Zeitpunkt der Messung ein Mittelwert über alle Wellenlängenintensitäten erstellt. In diesem Fall wird der Mittelwert von der Intensität subtrahiert. Die Syntax mit dem Eingangsvektor „X“ und dem Mittelwert Y in Matlab lautet:

$$Y = \text{mean}(X) \quad (4.7)$$

Für einen Vektor der Länge „n“ berechnet sich der Mittelwert wie folgt:

$$Y = \frac{1}{n} \sum_{i=1}^n X_i \quad (4.8)$$

4.1.4. Schwerpunkt der Intensität

Um festzustellen ob und in welcher Weise sich der Schwerpunkt der Intensitäten im Lade- und Entladevorgang verändert, wird dieser wie ein Masseschwerpunkt ermittelt. Die dafür verwendete Formel mit „n“-vielen Wellenlängen, der Gesamtmasse „M“, dem Wellenlängenvektor „m“ und dem Intensitätsvektor „x“ lautet:

$$X_s = \frac{1}{M} \sum_{i=1}^n X_i * m_i \quad (4.9)$$

Der berechnete Wert X_s entspricht der Wellenlänge, auf dem der Schwerpunkt liegt. Wird dieser Wert für jeden Messpunkt ermittelt, ergibt sich der Schwerpunktsverlauf, der zur besseren Beobachtbarkeit zusammen mit dem Intensitäts- oder Ladungsverlauf dargestellt werden kann.

4.2. Verarbeitungssoftware

4.2.1. CurrentIntegrator

Das Messsystem erfasst während des Betriebes sowohl die Spannungs- als auch die Stromwerte. Die optischen Veränderungen der Elektroden gehen jedoch mit dem Ladezustand der Zelle einher. Um diesen Zusammenhang betrachten zu können, wurde von Herrn Griebach das Programm „CurrentIntegrator“ entworfen [9].

Die Ladung einer Batterie lässt sich berechnen durch:

$$Q(t) = Q(t_0) + \int_{t_0}^t I(t) dt \quad (4.10)$$

Die Software basiert auf dem Prinzip der Integration nach der Trapezregel. Der in einer Textdatei gespeicherte Stromverlauf wird über die Zeit integriert und ergibt somit den Ladungsverlauf der Zelle. Dies bezieht jedoch nicht die Ladung der Zelle zu Beginn der Messung mit ein und hat dadurch keine Aussage über ihren absoluten Ladezustand. Die errechneten Ladungen werden zusammen mit ihren Zeitstempeln, die ebenfalls aus der Textdatei des Stroms stammen, in einer neu angelegten Datei gespeichert.

Der Aufruf des Programms erfolgt durch:

```
./CurrentIntegrator < infile.txt > outfile.txt
```

4.2.2. TimestampMatcher

Die Strom- und Spannungsaufnahme des Messsystems erfolgt zeitdiskret im Sekundentakt. Die Zeitspanne zwischen den Aufnahmen des Spektrometers und der Mikroskopkamera können variiert werden und betragen in den Versuchen dieser Arbeit in der Regel 10 Sekunden. Um die Datensätze zeitlich zusammenführen zu können, wurde von Herrn Griebach das Programm „TimestampMatcher“ erstellt. Es liest die Zeitstempel der Ausgabedateien des Spektrometers oder der von den Kameras erzeugten Bilder ein und sucht aus der Strom- oder Spannungsdatei die Werte, die diesem am nächsten liegen. Die Ausgabe erfolgt in einer neu erzeugten Datei, in der die Zeitstempel der Bilder mit den passenden elektrischen Messwerten gespeichert sind.

Aufgerufen wird das Programm durch:

```
./TimestampMatcher -f timestampfile < infile.txt > outfile.txt
```

4.2.3. HaloStack

„HaloStack“ ist eine einfache plattformunabhängige Software zum Bearbeiten von Bildern. Als grafische Oberfläche ist sie nicht erhältlich, ihr Aufruf erfolgt über die Kommandozeile des Terminals. Die Parametereingabe entscheidet über die Funktion, die zur Bearbeitung genutzt wird. In diesen Versuchen wird der Parameter „-a“ gewählt. Dies sorgt dafür, dass die Software den Durchschnitt der eingegebenen Bilder berechnet, indem sie die Werte für jedes Pixel aufsummiert und anschließend durch die Anzahl der Bilder teilt. Die Abspeicherung erfolgt als PNG im 16-bit-Format.

Der Aufruf erfolgt durch:

```
python halostack.py -a outputfile.png inputfiles.png
```

4.3. Kameradatenverarbeitung

Vor Beginn der Datenauswertung mit Matlab wird mittels Ausführen des Shell-Skripts „chmod.sh“ eine Datei mit dem Ladungsverlauf der Zelle erzeugt (Abschnitt 4.2.1). Bei längeren Messungen und somit großen Datensätzen der Bilder empfiehlt es sich das Shell-Skript „stacking.sh“ ebenfalls auszuführen. Es fasst alle Bilder aus je 100 Sekunden zu einem Bild zusammen und verringert den zeitlichen Aufwand beim Durchlaufen der Matlab-Skripte erheblich.

Abbildung 4.3 zeigt das Flussdiagramm der Datenvorverarbeitung bei Verwendung der Mikroskopkamera. Ein Zusammenführen der Kamera- mit den elektrischen Messdaten durch den „TimestampMatcher“ findet erst während der Ausführung von Matlab statt.

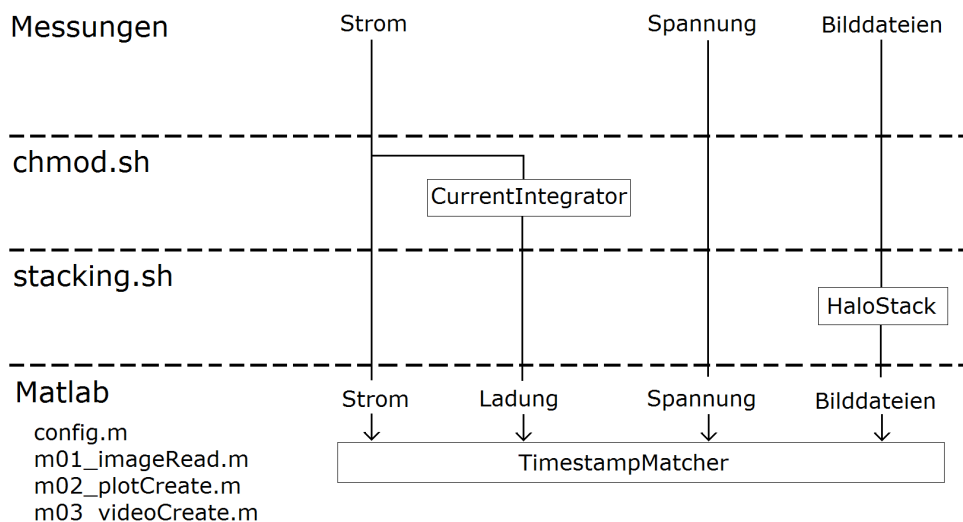


Abbildung 4.3.: Vorverarbeitung der Daten bei Nutzung der Mikroskopkamera

Zur Darstellung der Ergebnisse wurden folgende Matlab-Skripte verwendet. Der vollständige Quellcode befindet sich im Anhang A.5.

- **Config.m**

In der Konfigurationsdatei werden die Dateipfade der vom Shell-Skript „Chmod.sh“ (Abschnitt A.4.1) erstellten Ladungsdatei, sowie den Strom- und Spannungsdateien eingegeben. Des Weiteren ist der Pfad zum Ordner der Kamerabilder angegeben, sowie der x- und y-Positionen der Elektrode und einer Referenzfläche auf den Bildern.

- **m01_imageRead.m**

Das Matlab-Skript „m01_imageRead.m“ liest die in der Konfigurationsdatei gemachten Eingaben aus und lädt mit Hilfe dieser Angaben die Daten in Matlab ein. Anschließend werden diese in einer mat-Datei abgespeichert.

- **m02_plotCreate.m**

Die von „m01_imageRead.m“ erstellte mat-Datei wird eingelesen und die Daten grafisch dargestellt.

- **m03_videoCreate.m**

Entgegen seiner Bezeichnung erstellt „m03_videoCreate.m“ kein Video, sondern nur für jeden Zeitpunkt der Kameramessung ein Bild der Elektrode, welches zusammen mit ihren dazugehörigen elektrischen Daten in einem Plot dargestellt wird. Die Plots werden als Bilddateien gespeichert und können durch Aneinanderfügen zu einem Video zusammengestellt werden.

Korrigieren der Belichtungsänderung

Bei den RGB-Werten der Messungen kann es über die Zeit zu geringen Belichtungsänderungen kommen. Um diese zu minimieren wird, neben der Kathodenfläche, auch eine Referenzfläche aufgenommen. Für jede Aufnahme werden die Summen der Farbwerte RGB der Kathodenfläche berechnet und in einen Vektor gespeichert. Dasselbe gilt für die Referenzfläche. Anschließend wird der Vektor der Kathodenfläche elementweise durch die Referenzfläche dividiert und mit dem Durchschnitt des Referenzvektors multipliziert.

4.4. Spektrometerdatenverarbeitung

Die Datenvorverarbeitung erfolgt bei den Messungen mit Spektrometer mittels des Shell-Skripts „DataOperator“. Es erstellt aus durch Aufrufen des „CurrentIntegrators“ (Abschnitt A.2) aus den Daten der Stromdatei eine Ladungsdatei. Die Zusammenführung der Messdaten erfolgt mit dem „TimestampMatcher“. Dieser fügt die elektrischen Messdaten zu einer Datei zusammen und erstellt eine Datei, die den zeitlichen Verlauf des Spektrums beinhaltet. Der vollständige Quellcode des Skripts ist in Anhang A.4.3 enthalten. Abbildung 4.4 zeigt das Flussdiagramm der Datenvorverarbeitung.

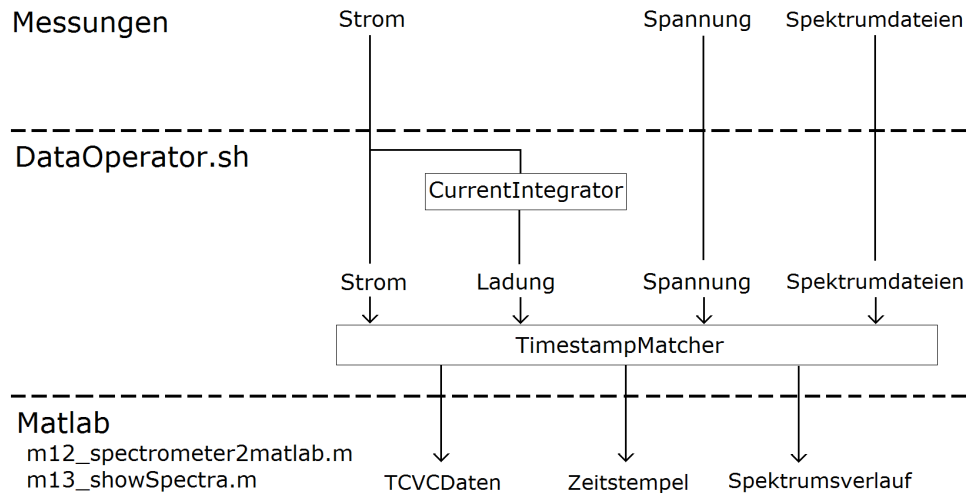


Abbildung 4.4.: Vorverarbeitung der Daten bei Nutzung des Spektrometers

Die weitere Verarbeitung und Darstellung der Daten erfolgt mittels Matlab. Folgende zwei Matlabskripte wurden verwendet, deren vollständiger Quellcode sich im Anhang A.5 befindet:

- **m12_spectrometer2matlab.m**

Mit diesem Skript werden die elektrischen Messdaten, sowie der Verlauf des Spektrums in Matlab geladen. Eine Verarbeitung findet hier nicht statt, die Daten werden lediglich als Matlab-Datei abgespeichert.

- **m13_showSpectra.m**

Hier findet die Verarbeitung der Daten nach den in Kapitel (Abschnitt 4.1) vorgestellten Verfahren statt. Es werden verschiedene Darstellungen erstellt und anschließend gespeichert.

Rauschunterdrückung

Aufgrund eines hohen Rauschens auf dem Spektrum, wird dieses vor weiteren Berechnungen in Matlab mittels der „smooth“-Funktion über 8 benachbarte Werte im Bereich der Wellenlänge und anschließend im zeitlichen Bereich geglättet (Abschnitt 4.1.1).

5. Auswertung

In diesem Kapitel werden die durchgeführten Messreihen vorgestellt und die Ergebnisse der Messungen ausgewertet.

Vorab muss jedoch geklärt werden, welche optischen Ergebnisse mit den gewählten Hilfsmitteln erzielt werden können. Der erste Unterschied zwischen den Messungen der Kamera und des Spektrometers besteht bereits im erfassten Kathodenbereich. Während man diesen bei den Aufnahmen der Kamera mittels Matlab präzise bestimmen kann, erfasst das Spektrometer die komplette reflektierte Strahlung der Kathode. Aufgrund des im 90° Winkel einfallenden Lichts der Quelle ist mit einer hohen Intensität der Spiegelreflexion (Abschnitt 2.3) zu rechnen. Bei optimal ausgerichteter Messung sind dadurch gute Ergebnisse zu erwarten. Jedoch spielt auch die diffuse Reflexion eine Rolle, da auch benachbarte Bereiche der Kathodenfläche das erfasste Spektrum beeinflussen. Sind die optischen Effekte am Elektrodenrand besonders ausgeprägt, das Spektrometer jedoch auf den Mittelpunkt der Elektrode ausgerichtet, führt dieses zu weniger eindeutigen Ergebnissen.

Einen unmittelbaren Einfluss auf die Ergebnisse der Messungen haben die Lichtquellen. Nur von ihnen ausgesendete Strahlung kann vom Testobjekt reflektiert werden. Optimal wäre deshalb ein linearer Intensitätsverlauf über alle Wellenlängen. Tatsächlich haben die in diesen Untersuchungen verwendeten Lichtquellen jedoch sehr ausgeprägte Kurvenverläufe. Diese sind in den Kapiteln Mikroskopkamera (Abschnitt 3.3.1) und Spektrometer (Abschnitt 3.3.2) dargestellt.

5.1. Übersicht der Zellaufbauten

Im Rahmen der Bachelorarbeit wurden insgesamt 12 Testzellen aufgebaut. Ihr Aufbau und die gewählten Materialien basieren auf den Vorarbeiten in der Arbeitsgruppe BATSEN [18]

[9] [22]. Die Anoden dieser Testzellen bestanden stets aus metallischem Lithium. Durch unterschiedliche Zusammensetzungen der Kathode werden unterschiedliche Effekte und Intensitäten der optischen Änderungen erreicht. So wurden Zellen aus reinem ITO¹ als Kathode erstellt, da ITO für diese Versuche sehr deutlich erkennbare optische Effekte aufweist. Den Zellen aus LiFePO₄ wurde ebenfalls ITO beigefügt um die Beobachtbarkeit ihrer Effekte zu verbessern. Als Bindemittel findet in diesen Zellenaufbauten NMP² mit einer Zumischung von 10 % PVDF³ Verwendung.

Die Aufzeichnung der elektrischen Messdaten erfolgte bei allen Messungen im Sekundentakt. Aufnahmen der Mikroskopkameras und des Spektrometers hingegen variierten zwischen 5 und 10 Sekunden.

Eine vollständige Aufzählung der aufgebauten Zellen ist in folgender Tabelle zu finden. Die Bezeichnung der Zellen basiert auf den vorangegangenen Arbeiten. Da bereits 24 Messreihen durchgeführt wurden [18] [9] [22], beginnt diese Arbeit der der Messreihe 25 (MR25). Für jede Reihe werden 4 Zellen aufgebaut, die von 1 bis 4 durchnummeriert werden (Z1 - Z4). In den Messreihen wurden einzelne Messungen durchgeführt, bei denen die Parameter variiert werden, diese Messungen sind mit einem (P) für „Part“ gekennzeichnet. Aus diesen einzelnen Benennungen ergibt sich die komplette Bezeichnung einer Messung (MRxxZxPxx).

Zelle	Datum des Aufbaus	Anode	Kathode	
MR25Z1	14.07.2016	Li	ITO	
MR25Z2	14.07.2016	Li	ITO	
MR25Z3	14.07.2016	Li	15 % ITO	83 % LiFePO ₄
MR25Z4	14.07.2016	Li	15 % ITO	83 % LiFePO ₄
MR26Z1	29.08.2016	Li	ITO	
MR26Z2	29.08.2016	Li	ITO	
MR26Z3	29.08.2016	Li	15 % ITO	83 % LiFePO ₄
MR26Z4	29.08.2016	Li	15 % ITO	83 % LiFePO ₄
MR27Z1	07.10.2016	Li	kein Aktivmaterial	
MR27Z2	07.10.2016	Li	26 % ITO	71,7 % LiFePO ₄
MR27Z3	07.10.2016	Li	40,8 % ITO	55,5 % LiFePO ₄
MR27Z4	07.10.2016	C	LiFePO ₄	

Tabelle 5.1.: Aufbau und Materialien der Testzellen

¹indium tin oxide, engl. für Indiumzinnoxid

²N-Methyl-2-pyrrolidon, ein organisches Lösungsmittel

³Polyvinylidenfluorid, ein thermoplastischer Fluorkunststoff

Messreihe 25

Die Messreihe 25 ist die erste, die im Rahmen dieser Bachelorarbeit durchgeführt wurde. Während die optischen Veränderungen der Zelle 1 mit einem Spektrometer aufgenommen wurden, wurden die restlichen 3 mit jeweils einer Kamera beobachtet. In dieser Messreihe wurden erstmals die Zykliersysteme 3.2 genutzt. Aus diesem Grund gab es bei der Durchführung noch einige Probleme. So kam es bei einigen Messungen mit der Kamera zu Abbrüchen bei der Bildaufnahme. Auch die Ansteuerung der Zellen funktionierte nicht einwandfrei. Zum Teil wurden zu hohe oder zu niedrige Spannungen an die Zelle angelegt, die zur Zerstörung dieser führten. Eine Zusammenfassung der Probleme und ihrer Lösungen findet sich im Kapitel „Aufgetretene Probleme“ (Abschnitt 3.6).

Messreihe 25	Zelle 1 (ITO)	Zelle 2 (ITO)	Zelle 3 (83 % LiFePO ₄)	Zelle 4 (83 % LiFePO ₄)
Part 01 Dauer: Aufnahme durch:	Zyklusart: cCcV 21 Zyklen Spektrometer	Zyklusart: cCcV 1 Zyklus Kamera	Zyklusart: cC 31 Zyklen Kamera	Zyklusart: cC 25 Zyklen Kamera
Part 02 Dauer: Aufnahme durch:	Zyklusart: cCcV 11 Zyklen Spektrometer	Zyklusart: cCcV 1 Zyklus Kamera	Zyklusart: cC 13 Zyklen Kamera	Zyklusart: cC 15 Zyklen Kamera
Part 03 Dauer: Aufnahme durch:	Zyklusart: cCcV 6 Zyklen Spektrometer	Zyklusart: cCcV Zykliefehler Kamera	Zyklusart: cC 1 Zyklus Kamera	Zyklusart: cC 1 Zyklus Kamera
Part 04 Dauer: Aufnahme durch:		Zyklusart: cCcV Zykliefehler Kamera	Zyklusart: cC Zykliefehler Kamera	Zyklusart: cC Zykliefehler Kamera

Tabelle 5.2.: Messungen der Messreihe 25

Messreihe 26

Der Zellenaufbau und die optische Beobachtung der Zellen erfolgt bei der Messreihe 26, mit Ausnahme der Parts 04 und 05, wie bei der Messreihe 25. Part 04 und 05 wurden mit einem Spektrometer im Fachbereich Chemie an der Universität Hamburg aufgenommen. Diese Messungen unterscheiden sich zu den anderen Aufnahmen mit dem Spektrometer anhand der Lichtquelle. Hier wurde eine Deuteriumlampe mit breiterem Spektrum verwendet, welche eine für die Untersuchungen ausreichende Intensität bis zu einem unteren Wellenlängenbereich von 400 nm aufweist (Abschnitt 3.3.2).

Messreihe 26	Zelle 1 (ITO)	Zelle 2 (ITO)	Zelle 3 (83 % LiFePO ₄)	Zelle 4 (83 % LiFePO ₄)
Part 01 Dauer: Aufnahme durch:	Zyklusart: cCcV 5,5 Zyklen Spektrometer	Zyklusart: cCcV 1,5 Zyklen Kamera	Zyklusart: cC 30 Zyklen Kamera	Zyklusart: cC 41 Zyklen Kamera
Part 02 Dauer: Aufnahme durch:	Zyklusart: cCcV 5 Zyklen Spektrometer	Zyklusart: cCcV 3 Zyklen Kamera	Zyklusart: cC 20 Zyklen Kamera	Zyklusart: cC 48 Zyklen Kamera
Part 03 Dauer: Aufnahme durch:	Zyklusart: cCcV 1,5 Zyklen Spektrometer	Zyklusart: cCcV Zykliefehler Kamera	Zyklusart: cC 1 Zyklus Kamera	Zyklusart: cC 8 Zyklen Kamera
Part 04 Dauer: Aufnahme durch:	Zyklusart: cCcV 0,5 Zyklen Spektrometer		Zyklusart: cC 4 Zyklen Spektrometer	
Part 05 Dauer: Aufnahme durch:	Zyklusart: cCcV 1,5 Zyklen Spektrometer		Zyklusart: cC 1 Zyklus Spektrometer	

Tabelle 5.3.: Messungen der Messreihe 26

Messreihe 27

Alle optischen Aufnahmen der Messreihe 27 wurden mit einer Kamera angefertigt. Die Zellen 2 und 3 wurden jeweils aus einer Mischung von ITO und LiFePO₄ erstellt. Die Zelle 1 enthält ausschließlich den PVDF-Binder auf dem Stromableiter, welcher in diesem Fall beobachtet wird.

Messreihe 27	Zelle 1 Kein Aktivmaterial	Zelle 2 (71,7 % LiFePO ₄)	Zelle 3 (71,7 % LiFePO ₄)	Zelle4 (LiFePO ₄)
Part 01 Dauer: Aufnahme durch:	Zyklusart: cC 10 Zyklen Kamera	Zyklusart: cC 5 Zyklen Kamera	Zyklusart: cC 1,5 Zyklen Kamera	Zyklusart: cC Zykliefehler Kamera
Part 02 Dauer: Aufnahme durch:	Zyklusart: cC 15 Zyklen Kamera	Zyklusart: cC 28 Zyklen Kamera	Zyklusart: cC 46 Zyklen Kamera	Zyklusart: cC 5 Zyklen Kamera
Part 03 Dauer: Aufnahme durch:	Zyklusart: cCcV 6 Zyklen Kamera			

Tabelle 5.4.: Messungen der Messreihe 27

5.2. Messungen mit der Kamera

5.2.1. ITO - typischer Verlauf

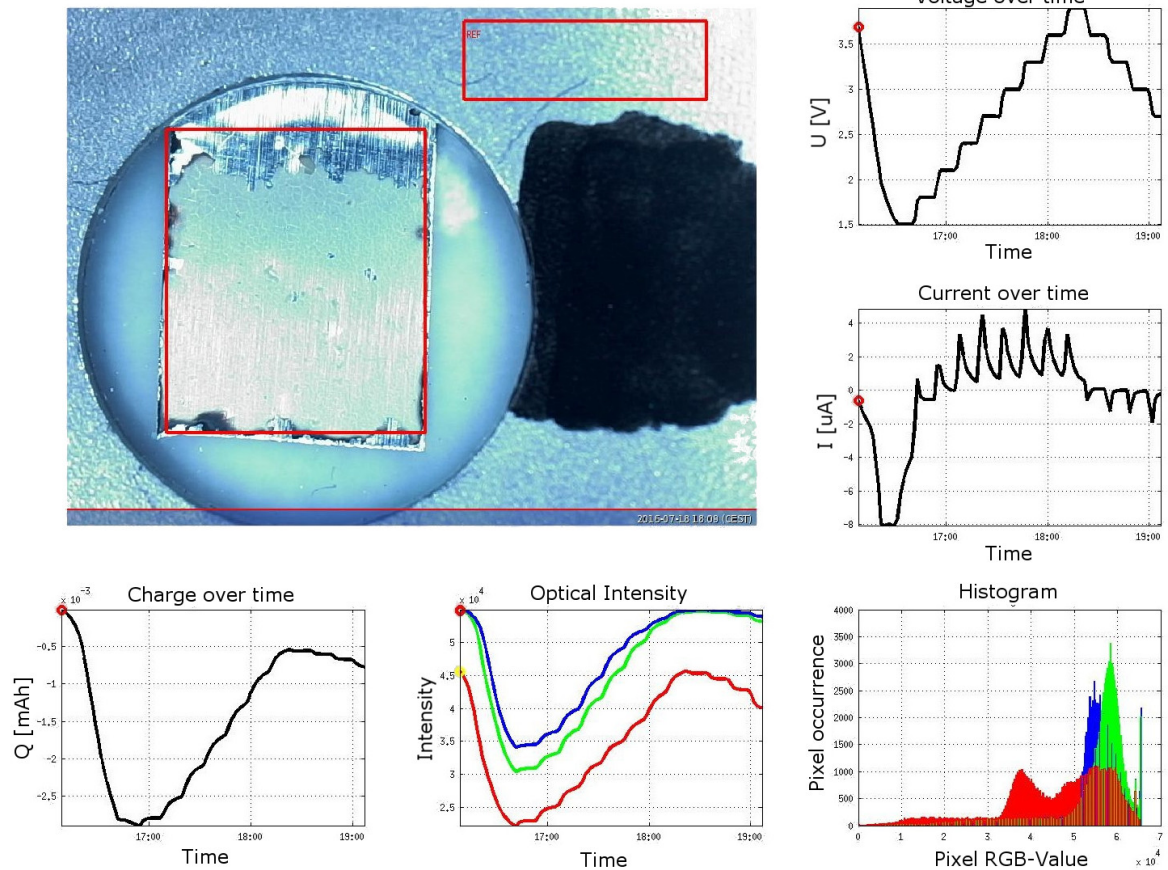


Abbildung 5.1.: MR25Z2P02 - Flächenmarkierungen auf der Elektrode

Der typische Verlauf einer ITO-Zelle unter der Mikroskopkamera wird hier exemplarisch an der Messung MR25Z2P02 dargestellt. Abbildung 5.1 zeigt die Flächenmarkierungen der Elektrode und ihrer Referenzfläche. Des Weiteren sind von oben rechts nach unten links abgebildet:

1. zeitlicher Verlauf der Spannung
2. zeitlicher Verlauf des Stroms
3. Histogramm, quantitatives Auftreten der RGB-Werte
4. zeitlicher Verlauf der RGB-Kanäle
5. zeitlicher Verlauf der Ladung

Die Messung MR25Z2P02 wurde im cCcV-Modus (siehe Abschnitt 3.2) und für eine Dauer von mehreren vollständigen Zyklen betrieben. Aufgrund von in Kapitel 3.6 beschriebenen Problemen der Bildaufnahme, kann jedoch nur ein Zyklus ausgewertet werden.

Zu Beginn soll die Zelle mit $10 \mu\text{A}$ entladen werden. Aufgrund der langsamen Ansteuerung des Zykliersystems werden jedoch nur $8 \mu\text{A}$ erreicht, bevor die Schwellspannung von $1,5 \text{ V}$ der Zelle erreicht ist. Diese wird anschließend für 10 Minuten gehalten. Durch die relativ kurzen Zyklierzeiten wird eine Ladungsänderung von lediglich $3 \mu\text{Ah}$ erreicht.

Der cCcV-Modus führt durch einem treppenförmigen Anstieg der Zellspannung um jeweils $0,3 \text{ V}$ zu steilen Stromverläufen, die nach Erreichen der jeweiligen Schwellspannung wieder abfallen. Daraus resultiert ein ebenfalls annähernd treppenförmiger Ladungsverlauf, der deutliche optische Änderungen zur Folge hat.

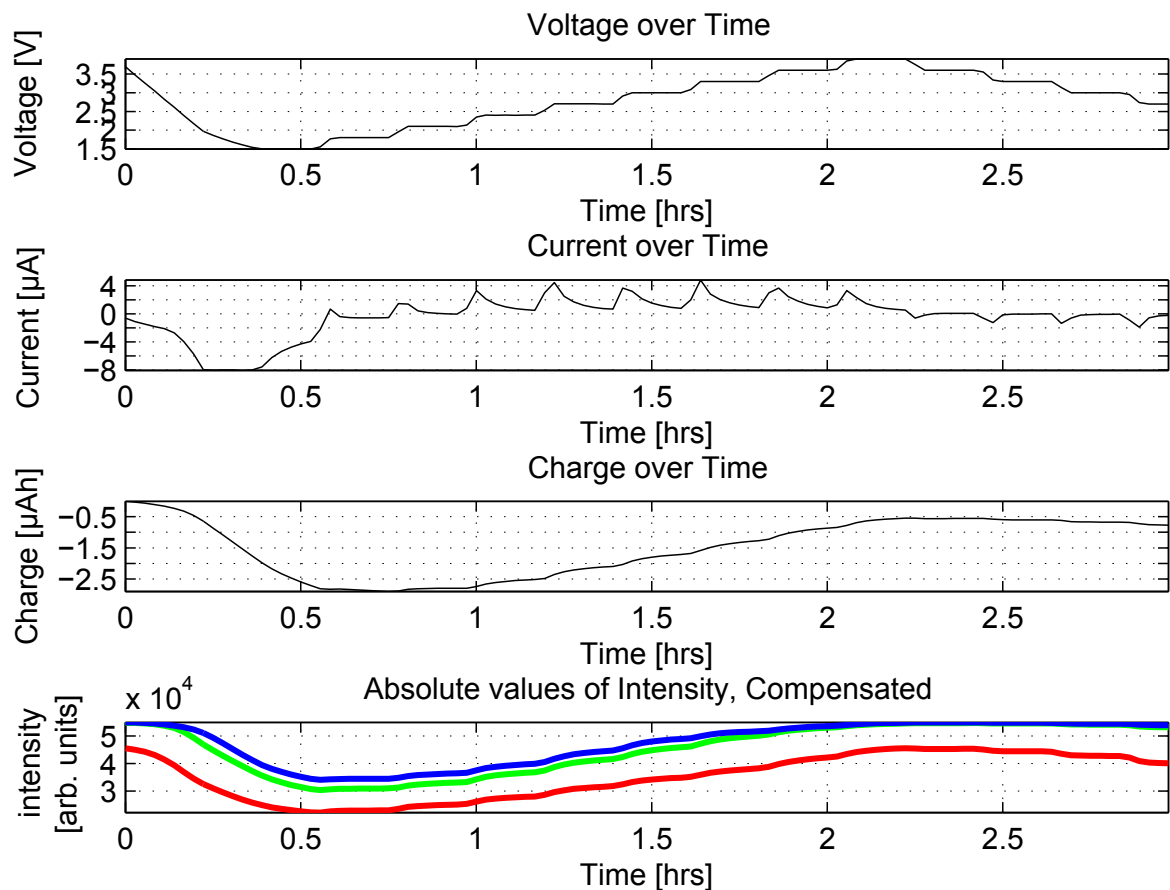


Abbildung 5.2.: MR25Z2P02 - Zeitliche Verläufe der elektrischen Messdaten und der Intensitäten der RGB-Kanäle während der Messung im cCcV-Modus

Die unterste Darstellung in Abbildung 5.2 stellt den Intensitätsverlauf der RGB-Kanäle dar. Zu erkennen ist, dass ihre Verläufe annähernd dem Ladungsverlauf entsprechen, sich jedoch sowohl in ihrer Intensitätsänderung unterscheiden, als auch zeitlich versetzt sind. Während der rote Kanal unmittelbar auf Veränderungen des Ladezustands reagiert, beginnen der blaue und der grüne Kanal erst nach ungefähr 10 Minuten deutlich abzufallen.

Gut zu erkennen ist in folgender Abbildung, dass trotz einer höheren Intensität des blauen Kanals gegenüber dem roten Kanal seine Änderungen nicht so stark ausgeprägt sind.

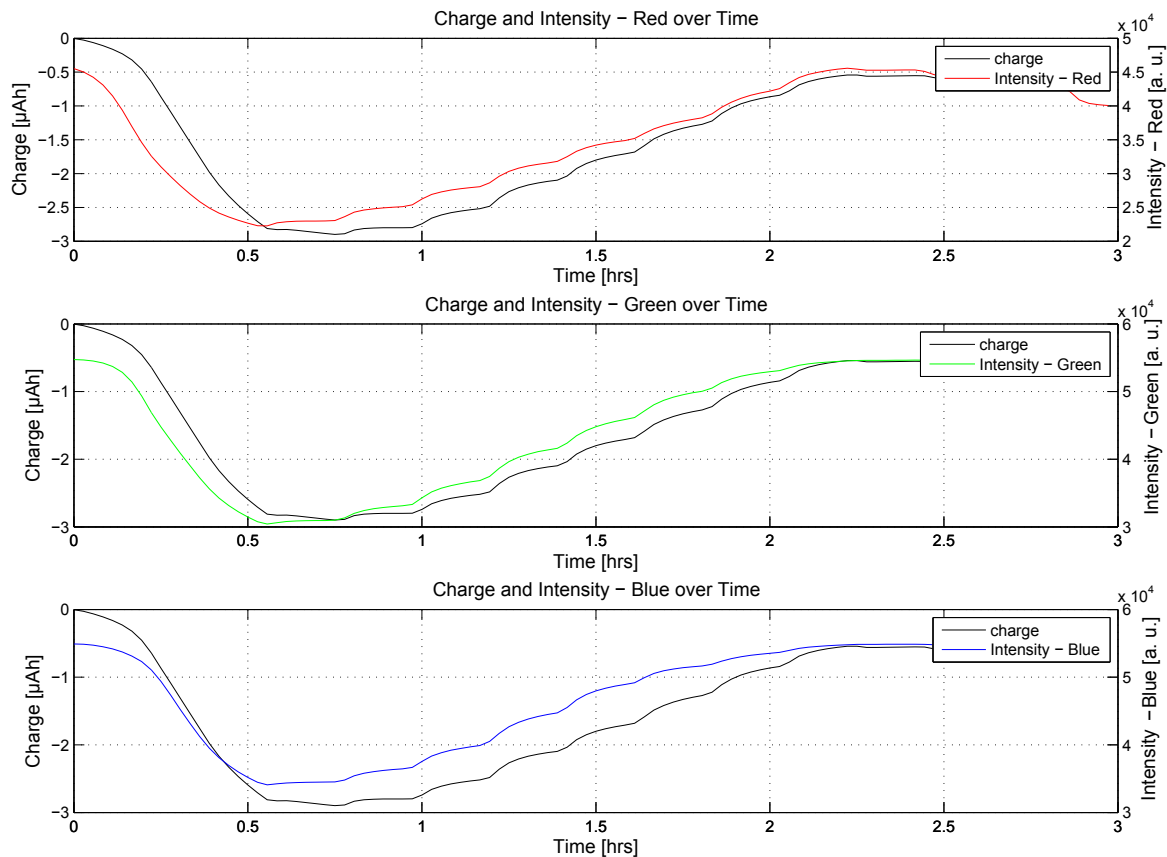


Abbildung 5.3.: MR25Z2P02 - Sichtbarwerden der Unterschiede im Zeitverlauf der RGB-Farbkanäle bei gleichem Ladungsverlauf

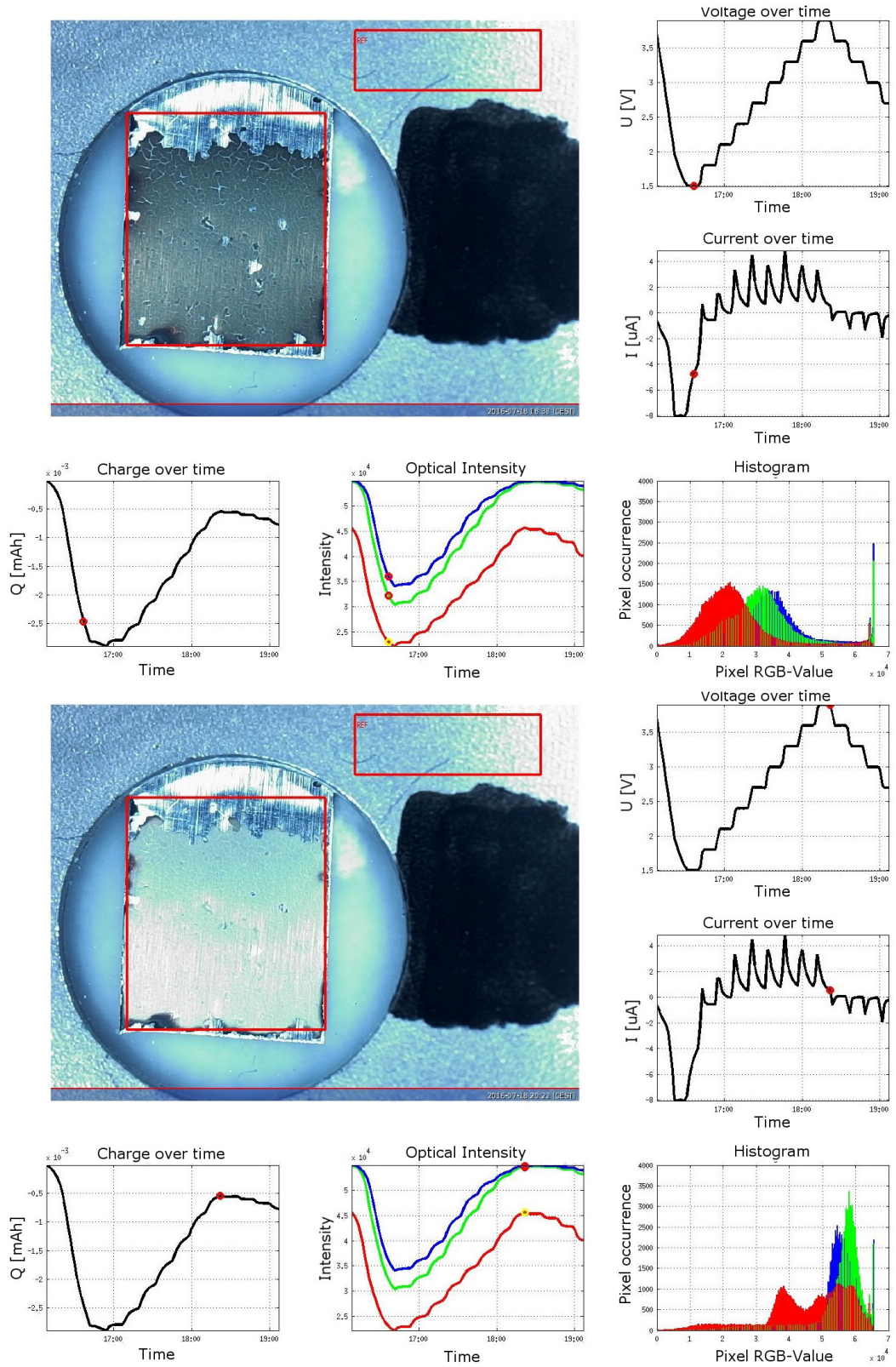


Abbildung 5.4.: MR25Z2P02 - Optische Änderungen der Elektrode im Verlauf der Messung
 oben: Zustand kurz vor tiefstem Ladezustand der Zelle
 unten: Zustand nach dem Laden der Zelle

Abbildung 5.1 zeigt die Kathode der Testzelle zu Beginn des Zyklivorgangs. Sie ist auffallend hellgrau bis silbern. Zwei weitere Aufnahmen aus dem Verlauf der Messung sind in Abbildung 5.4 dargestellt, wobei die Zelle in der oberen Darstellung ihren Ladungstiefpunkt erreicht hat. Deutlich zu erkennen ist die Verfärbung hin zum Schwarzen. Lediglich kleine Fehler in der Kathodenoberfläche und der Stromableiter am oberen Rand der ausgewählten Fläche sind weiterhin hell. Daraus resultiert auch der gleichbleibende Peak im dargestellten Histogramm, denn ein heller Wert ist im RGB-Format das Maximum der Bits. Da dieser Datensatz durch das Programm HaloStack (Abschnitt 4.2.3) zusammengefasst wurde, sind die RGB-Daten im 16 bit-Format dargestellt, was ein Wert von bis zu $2^{16} = 65.536$ ergibt.

5.2.2. LiFePO_4 - typischer Verlauf

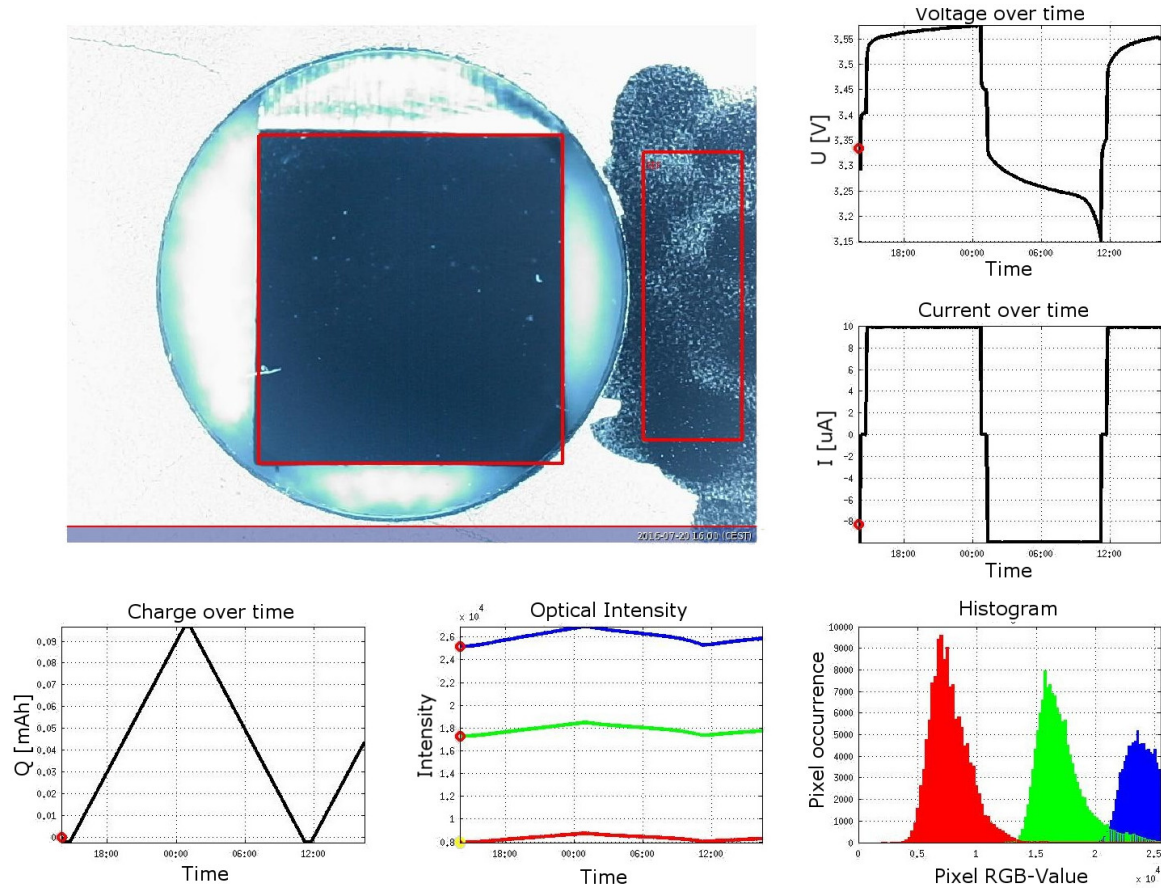


Abbildung 5.5.: MR25Z3P03 - Flächenmarkierungen auf der Elektrode und von oben rechts nach unten links:

1. zeitlicher Verlauf der Spannung
2. zeitlicher Verlauf des Stroms
3. Histogramm, quantitatives Auftreten der RGB-Werte
4. zeitlicher Verlauf der RGB-Kanäle
5. zeitlicher Verlauf der Ladung

Die Durchführung der Untersuchung an einer Testzelle aus LiFePO_4 ist hier exemplarisch an der Messung MR25Z3P03 dargestellt. Die verwendete Zelle besteht zu 83 % aus LiFePO_4 und zu 15 % aus ITO. Die in Abbildung 5.5 dargestellte Elektrodenfläche ist, im Gegensatz zu einer Elektrode aus ITO (Abschnitt 5.2.1), durch das Einbringen von LiFePO_4 sehr dunkel. Im PNG-Format gehen dunkle Werte mit einer kleineren Intensität der RGB-Äquivalenten einher (Abschnitt 2.4), wodurch die Farbkanäle in ihrer Größenordnung sowie der Änderung ihrer Wertigkeiten, deutlich weniger ausgeprägt sind.

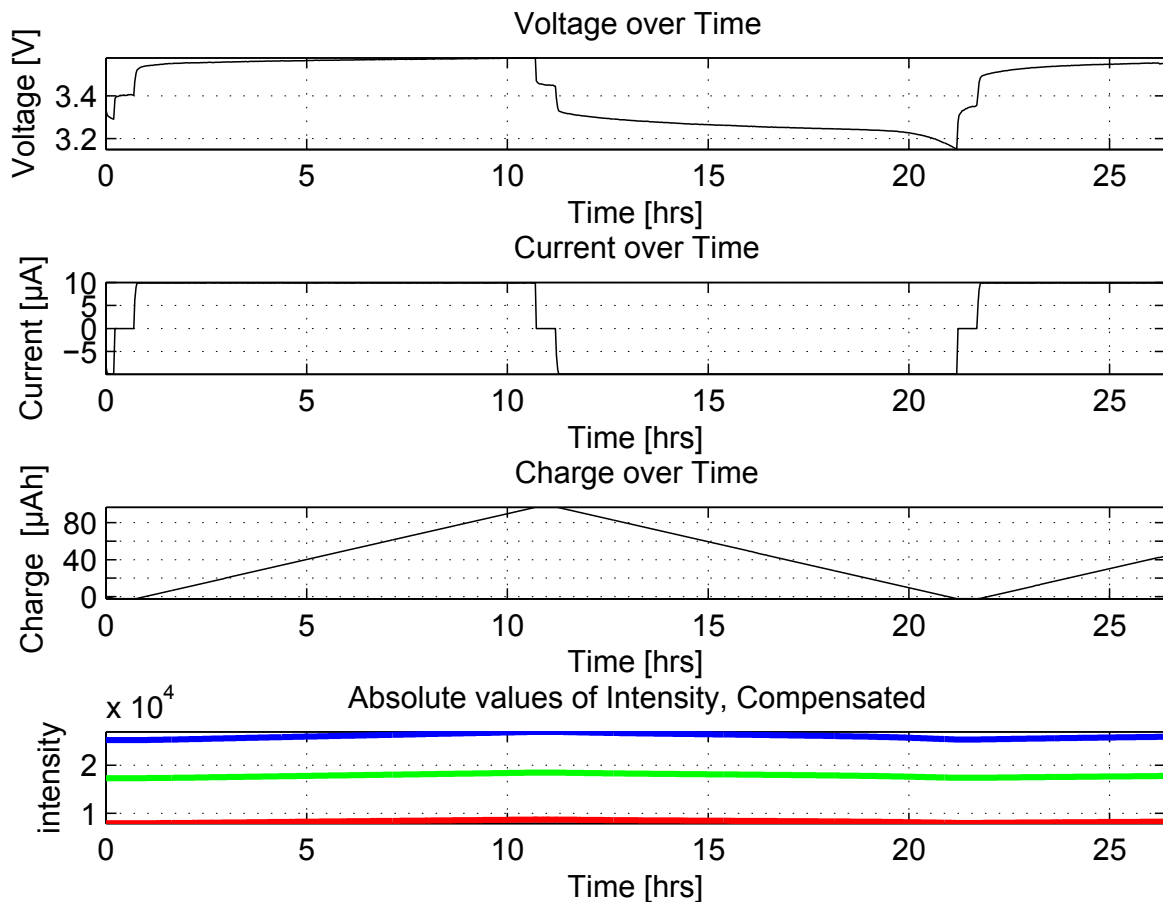


Abbildung 5.6.: MR25Z3P03 - Zeitliche Verläufe der elektrischen Messdaten und der Intensitäten der RGB-Kanäle während der Messung

Zur Versuchsdurchführung ist der cC-Modus gewählt, der die Zelle für 10 Stunden auflädt, ehe sie anschließend für dieselbe Zeit entladen wird. Durch die langsame Ansteuerung des Zyklersystems werden statt der erwarteten 100 μAh nur ungefähr 90 μAh Ladungsänderung erreicht.

Deutlich zu erkennen ist in der Messwertaufnahme ein Einbruch der Spannung nach ca. 20 Stunden. Dies deutet darauf hin, dass die Zelle fast vollständig entladen ist. Trotz der, im Gegensatz zur Messung mit ITO-Zellen (Abschnitt 5.2.1) starken Ladungsänderung, sind die Intensitätsänderungen in den RGB-Kanälen der LiFePO_4 -Zellen, im Vergleich zu den Messungen der ITO-Zellen eher schwach ausgeprägt, jedoch noch deutlich erkennbar.

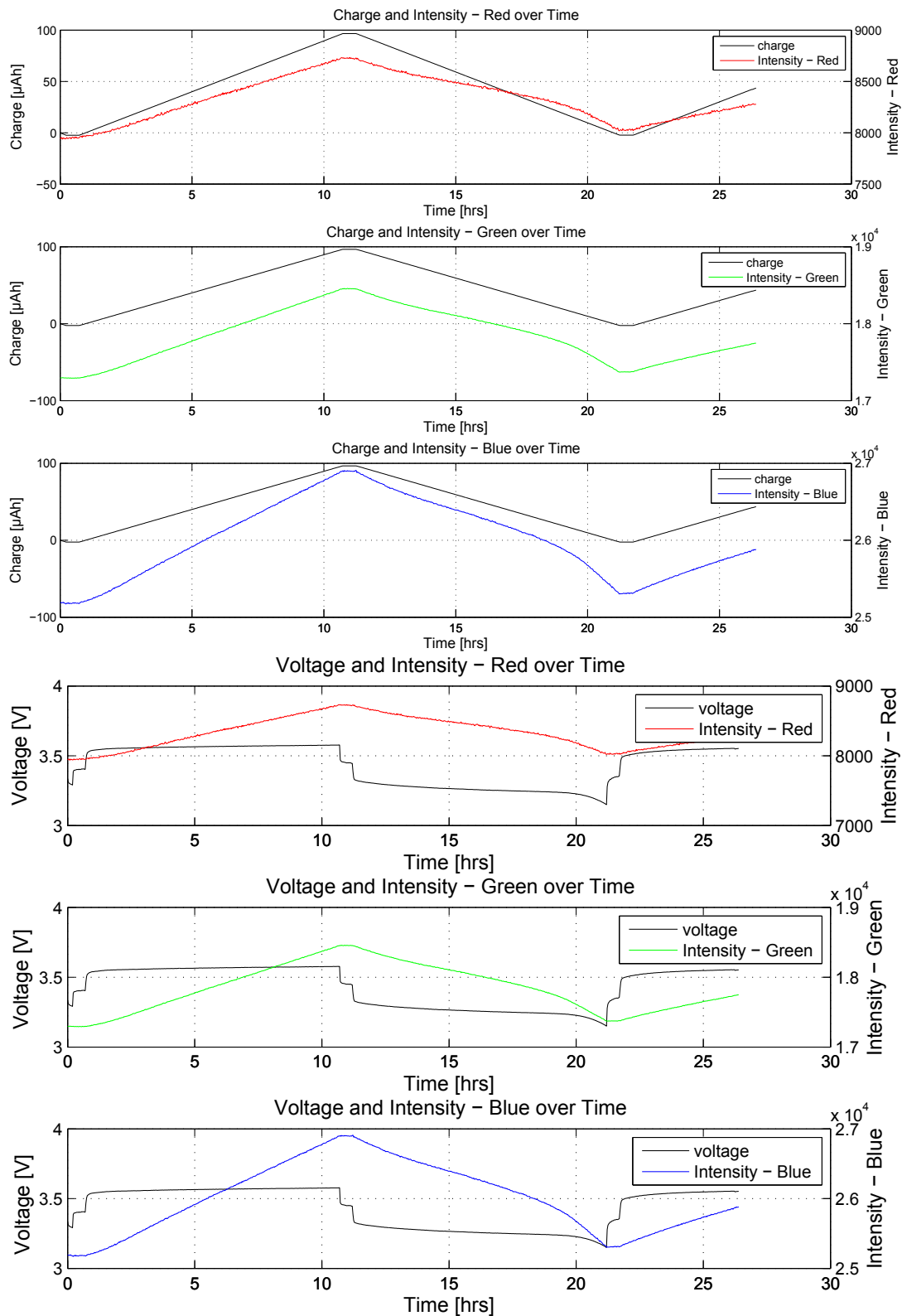


Abbildung 5.7.: MR25Z3P03 - Zusammenhang der zeitlichen Verläufe der RGB-Kanalintensitäten und
 - in den oberen 3 Abbildungen: dem Ladungsverlauf
 - in den unteren 3 Abbildungen: dem Spannungsverlauf

Die Intensitätsverläufe der RGB-Kanäle entsprechen größtenteils den Erwartungen. Sie steigen bei einem linearen Anstieg der Ladung ebenfalls linear an. Bei der Entladung gibt es jedoch Differenzen in der Ausprägung der abfallenden Ladungs- und Intensitätskurven. Zwar beginnen die Intensitätskurven mit ungefähr der negativen Steigung der Aufladekurve abzufallen, im Verlauf der Entladung werden sie jedoch weniger steil, bevor sie zum Ende hin sehr stark absinken.

Eine mögliche Erklärung für den starken Abfall zum Ende der Entladung ist in Abbildung 5.7 dargestellt. Im Bereich des konstanten Spannungsabfalls während der Entladung, fallen auch die RGB-Werte konstant ab. Durch die fortschreitende Entladung beginnt die Zelle schneller an Spannung zu verlieren. Mit Beginn dieses stärkeren Spannungsabfalls, verlieren auch die RGB-Werte schneller an Intensität.

Auf der nachfolgenden Seite sind die optischen Änderungen der Elektrodenfläche im Verlauf der Messung dargestellt. Während die obere Darstellung die Zelle in aufgeladenem Zustand zeigt, hat diese zum Zeitpunkt der Aufnahme des unteren Bildes ihren tiefsten Ladungszustand erreicht.

Zu erkennen ist, dass der optische Effekt beim Laden der Zelle von den Seiten und von unten eintritt. Hieraus kann geschlossen werden, dass die Lithium-Ionen, die sich während der Zellenladung zur Anode bewegen, zuerst aus den Randregionen der Kathode stammen und sich der Effekt bei einer weiteren Ladung in Richtung Kathodenmitte konzentriert.

Im umgekehrten Fall, der Entladung, setzen sich die Lithium-Ionen von Rand her wieder in die Kathode ein, bis die Zelle vollständig entladen ist und die Kathode ihren ursprünglichen optischen Zustand wieder einnimmt.

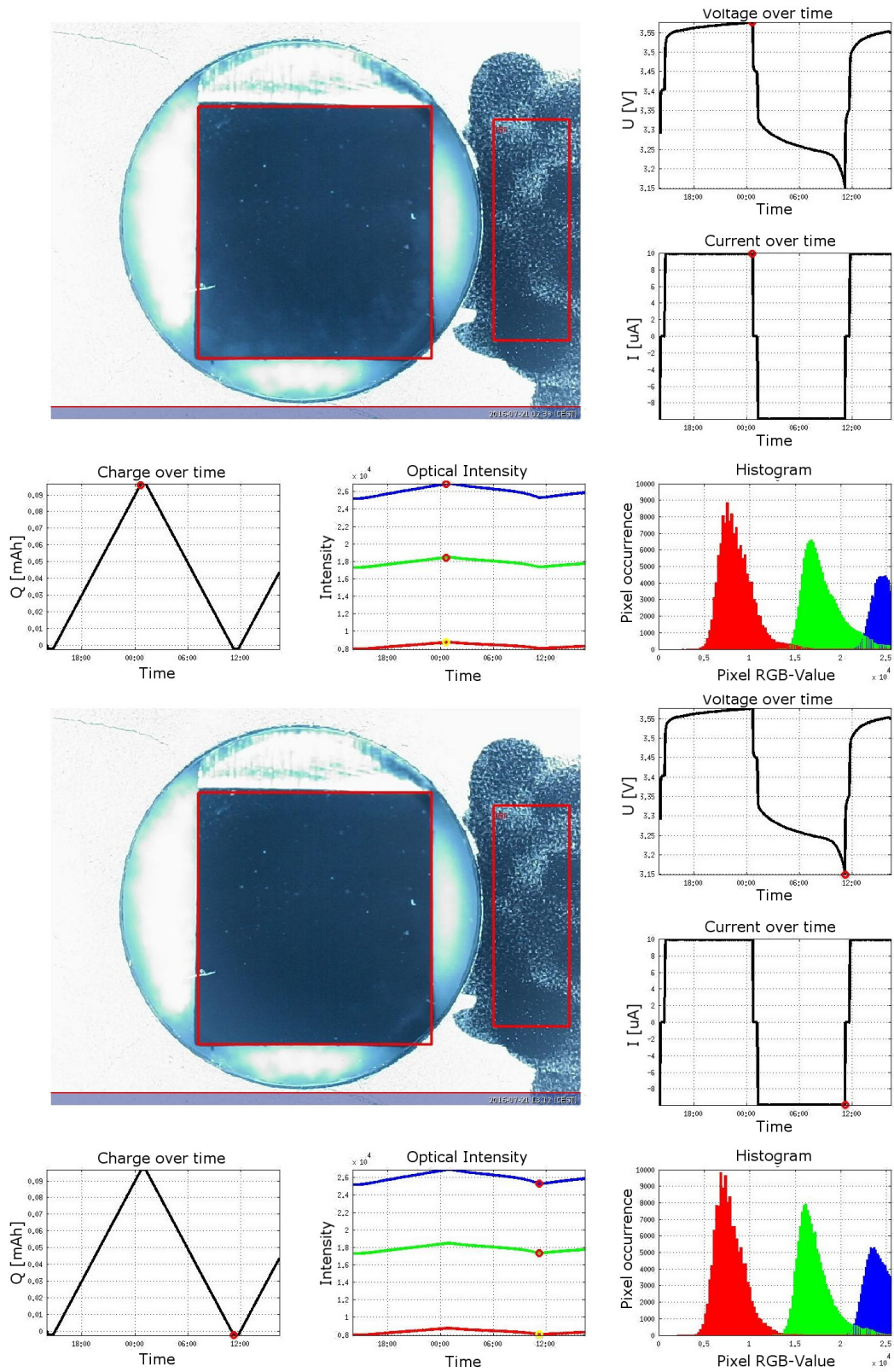


Abbildung 5.8.: MR25Z3P03 - Optische Änderungen der Elektrode im Verlauf der Messung
 oben: Zustand zum Zeitpunkt der maximalen Ladung
 unten: Zustand zum Zeitpunkt der maximalen Entladung

In folgender Abbildung ist mit der Messung MR26Z4P02 ein längerer Verlauf der Zyklisierung dargestellt. Auch diese Zelle besteht aus 83 % LiFePO_4 und 15 % ITO und wird im cC-Modus zyklisiert. Die Ladung erfolgt mit einem Strom von $5 \mu\text{A}$ und die Entladung mit $10 \mu\text{A}$. Bei Erreichen der oberen, beziehungsweise unteren, Schwellspannung endet der jeweilige Zyklus.

Am Verlauf der Spannung ist deutlich eine Alterung der Zelle zu erkennen. Die Zykluszeiten verkürzen sich aufgrund des immer schnelleren Erreichens der Schwellspannung bei gleich bleibenden Strömen. Die Zelle nimmt dadurch im Ladefall weniger Ladung auf, die sie im Entladezyklus wieder abgeben kann. Anfangs bestätigen die Verläufe der RGB-Farbkanäle die in Abbildung 5.6 gemachten Erkenntnisse. Bei zunehmender Dauer der Messung werden aufgrund der immer geringer werdenden Ladungsänderungen jedoch auch ihre Intensitätsänderungen geringer, bis nach circa 80 Stunden in dieser Darstellung keine Änderungen der Intensität mehr zu erkennen sind.

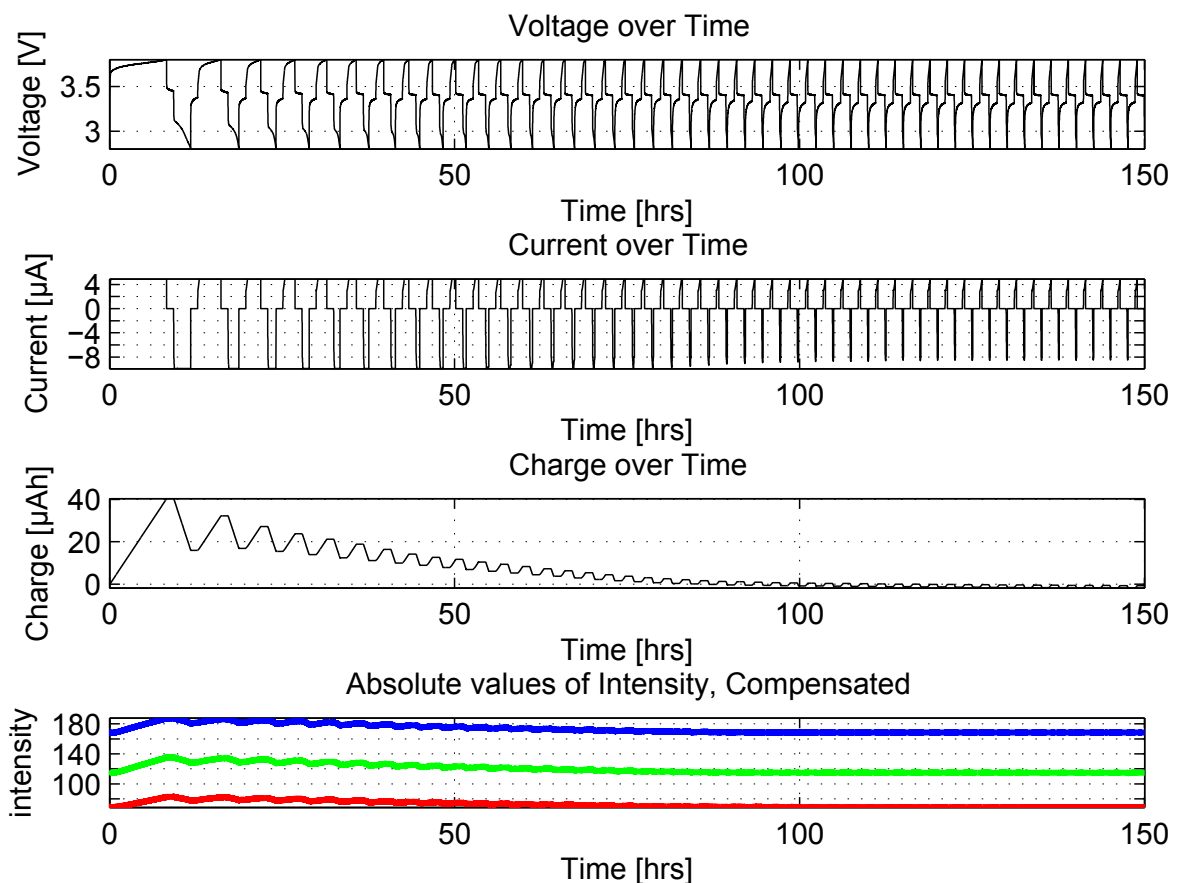


Abbildung 5.9.: MR26Z4P02 - Zeitlicher Verlauf der elektrischen Messdaten und RGB-Farbkanäle über einen längeren Zeitraum und deutlich erkennbare Alterung der Zelle

In der Abbildung 5.10 ist der Zusammenhang zwischen der Ladung der Zelle und den RGB-Farbkanalwerten veranschaulicht. Über die X-Achse ist der Verlauf der Ladung dargestellt. Das Maximum von circa $40 \mu\text{A}$ wird nur im ersten Zyklus erreicht. Die Verläufe der Zyklen werden zwar mit der Zeit sowohl in der Intensität des Kanals als auch im Ladungswert kleiner, das Verhältnis zwischen den beiden Werten bleibt aber gleich, wodurch die Verläufe übereinander liegen. Lediglich der Verlauf der ersten Ladephase liegt deutlich daneben. Vermutlich dauert es einen Zyklus, bevor die Zelle ihren eingeschwingenen Zustand erreicht hat.

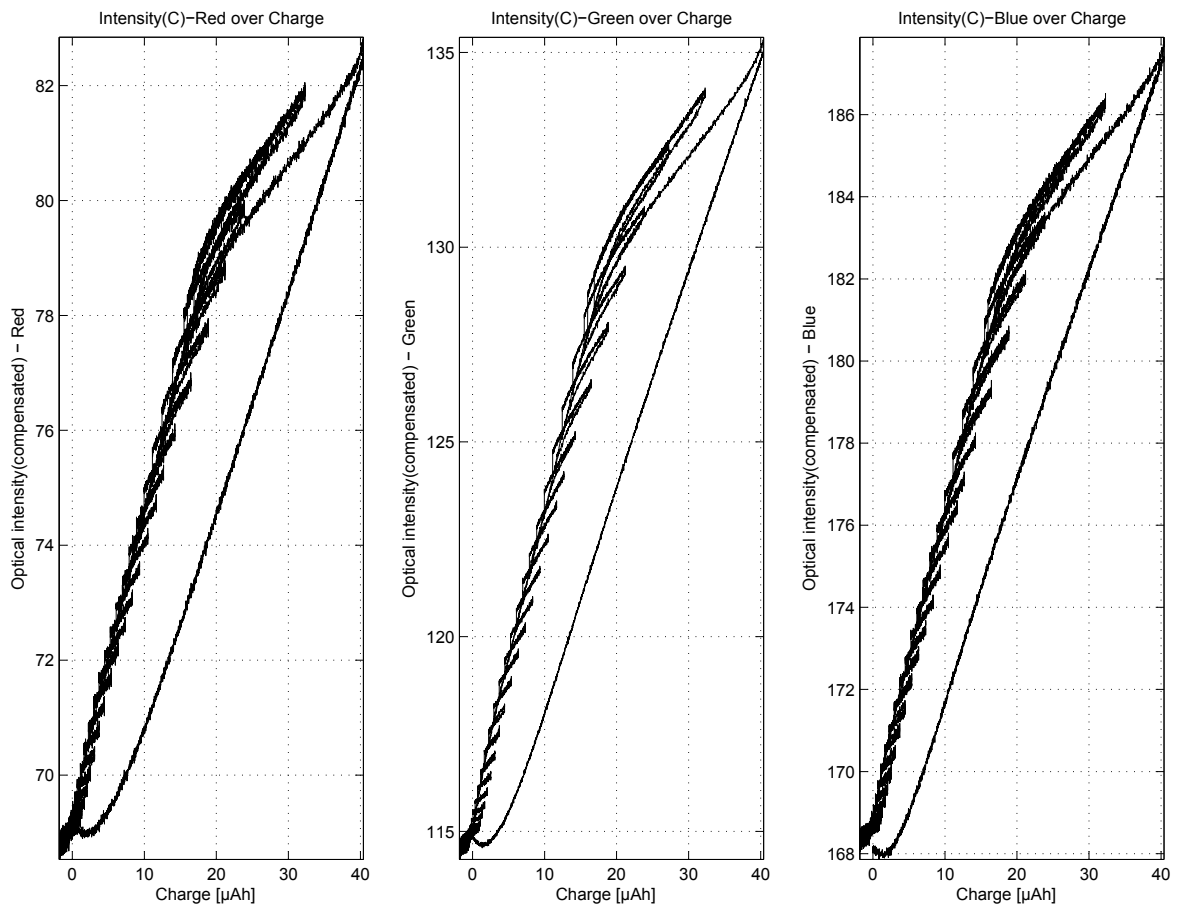


Abbildung 5.10.: MR26Z4P02 - Zusammenhang der RGB-Farbkanäle mit dem Ladezustand der Zelle

links: roter Farbkanal
mitte: grüner Farbkanal
rechts: blauer Farbkanal

5.3. Messungen mit dem Spektrometer

5.3.1. ITO - typischer Verlauf

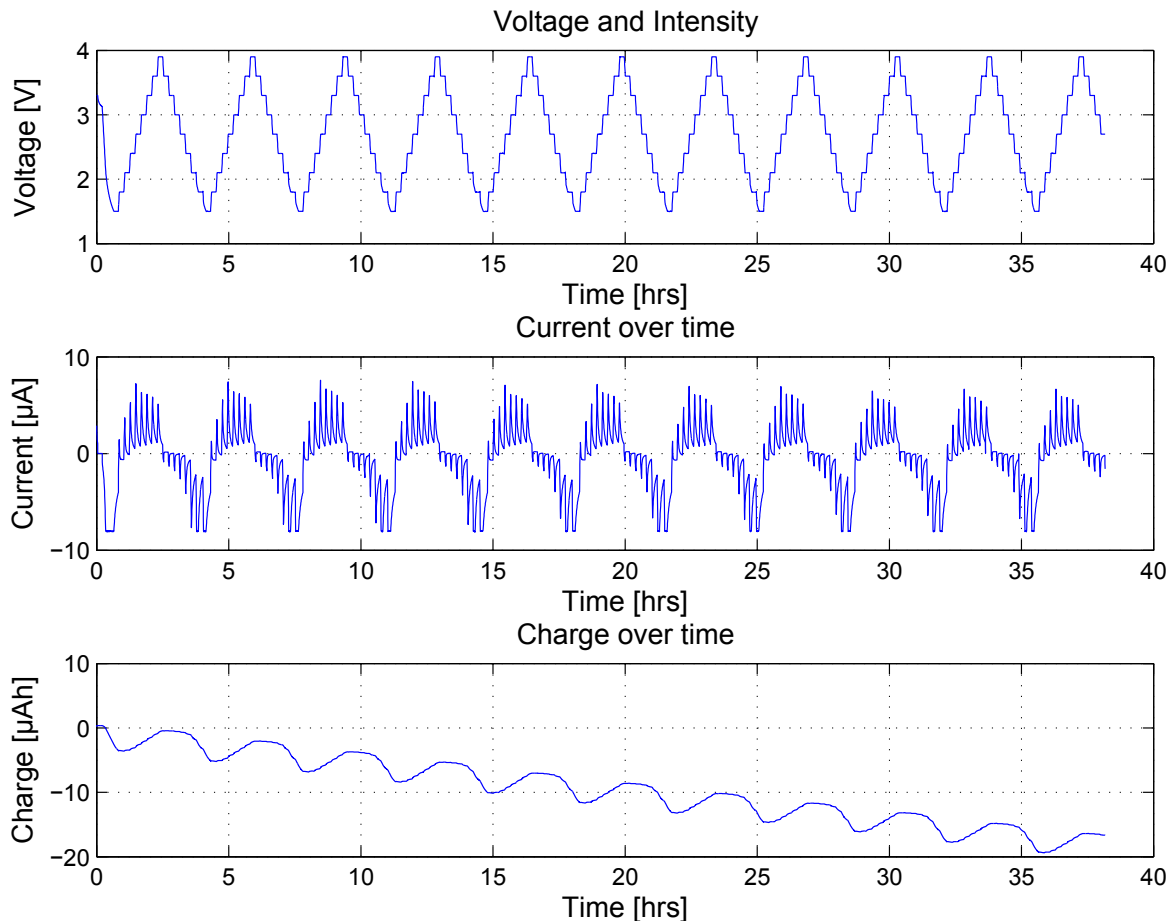


Abbildung 5.11.: MR25Z1P02 - Typische Verläufe der elektrischen Messdaten im cCcV-Modus und deutlich erkennbarer Abfall der Ladung über den Zeitraum der Messung

Bei der Beobachtung der Testzelle mit dem Spektrometer wird diese, wie in der zuvor beschriebenen Durchführung mit der Kamera, im cCcV-Modus zyklisiert. Daraus resultiert der, in Abbildung 5.12 dargestellte, treppenförmige Spannungsverlauf. Der während der Konstantstromphase des cCcV-Modus erwünschte Strom von $10 \mu\text{A}$ wird aufgrund des schnellen Erreichens der Schwellspannungen und relativ langsamen Ansteuerung nicht erreicht, ist aber für eine erfolgreiche Auswertung nicht ausschlaggebend. Durch ein überwiegendes Betreiben der Messungen unterhalb der Zellspannung von circa $3,3 \text{ V}$ sind die Ströme im Entladefall größer als im Ladevorgang. Daraus ergibt sich ein langsamer Abfall der Ladung, der in dieser Messung nach 24 Stunden ungefähr $10 \mu\text{Ah}$ beträgt.

In folgender Darstellung ist der Verlauf des Spektrum mit dazugehörigen Spannungs- und Ladungsverläufen abgebildet. Über die y-Achse sind die Wellenlängen dargestellt, deren zeitlicher Verlauf durch die x-Achse beschrieben wird. Die unterschiedlichen Intensitäten des Spektrums sind durch eine farbliche Änderung kenntlich gemacht.

Durch die Helligkeitsänderung der Zellenelektrode während des Lade- und Entladevorgangs ändert sich auch das Reflexionsverhalten ihrer Oberfläche. Je heller diese ist, desto mehr Strahlung der Lichtquelle wird reflektiert. Hieraus ergibt sich die typische Darstellung des Spektrums.

Deutlich zu erkennen ist, dass dieser Verlauf dem treppenförmigen Spannungsverlauf der Zelle entspricht.

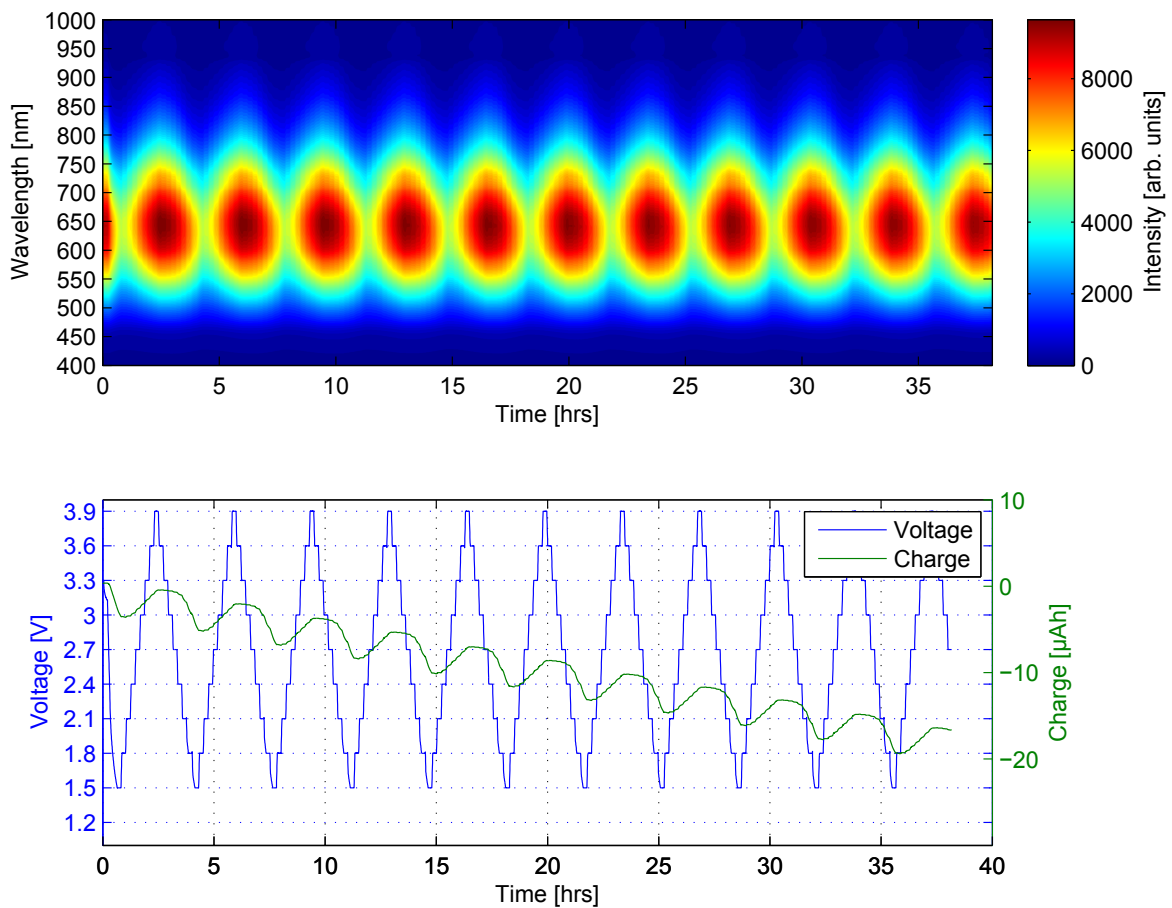


Abbildung 5.12.: MR25Z1P02 - Zusammenhang zwischen den Intensitäten des Spektrums (oben) und der Spannung und Ladung (unten) im zeitlichen Verlauf der Messung

Die folgende Abbildung zeigt den zeitlichen Verlauf der Intensitäten zweier Wellenlängenbereiche während der Messung. Da die Intensität der Wellenlänge 560 nm deutlich geringer ist als die der Wellenlänge 710 nm, werden Bereiche der nachfolgenden Wellenlängen aufaddiert, bis die Summen der Wellenlängen sich in ihrem Maximum annähern. Hierzu sind die Bereiche der Wellenlängen von 510 nm bis 568 nm und von 710 nm bis 715 nm gewählt worden.

Aus der Grafik ist deutlich eine Verschiebung zwischen den Wellenlängen zu erkennen. Der Bereich um 710 nm reagiert schneller auf eine Ladung oder Entladung der Zelle. Neben dem zeitlichen Versatz zwischen den beiden Bereichen ist auch ein Unterschied in der Dauer des Verbleibens um die Maxima ersichtlich. Die Intensität des Wellenlängenbereichs von 510 nm bis 568 nm fällt erst deutlich später wieder ab.

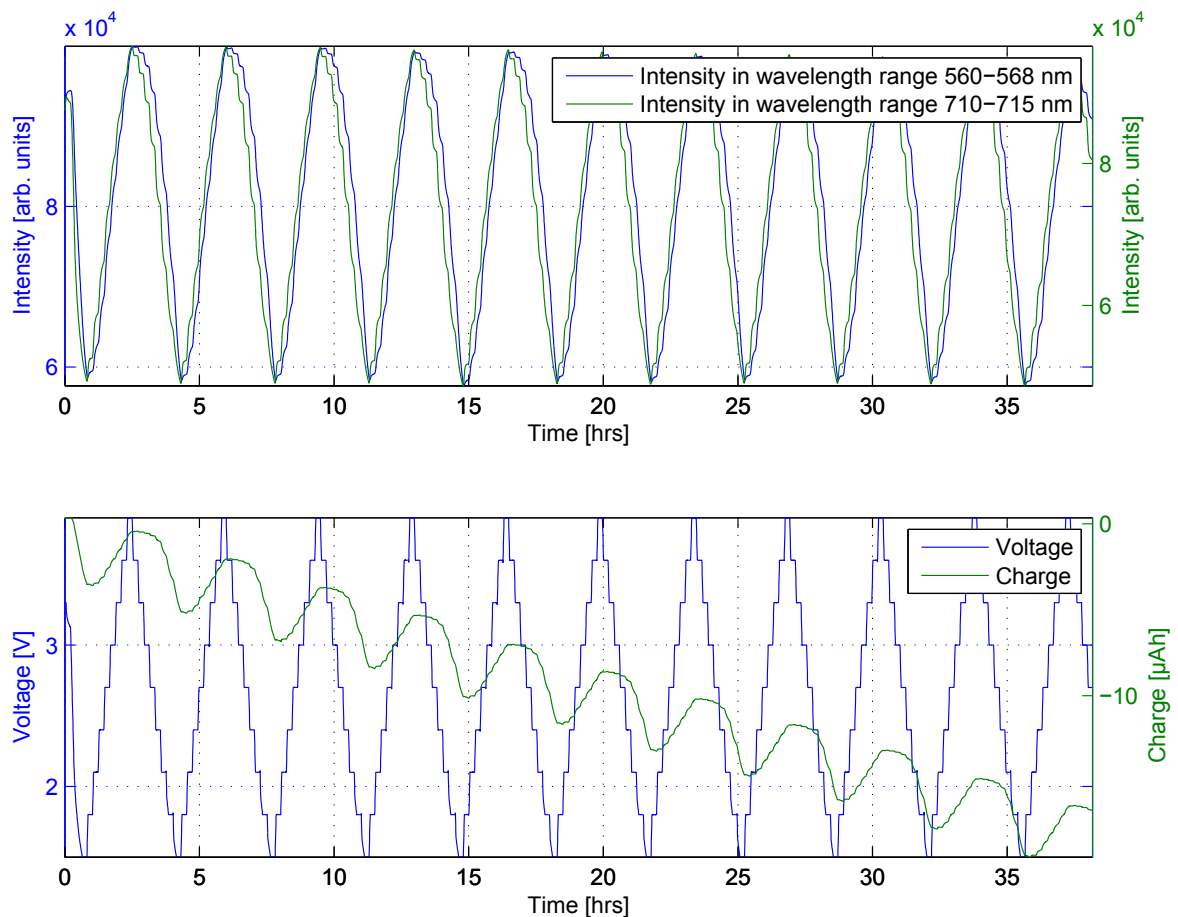
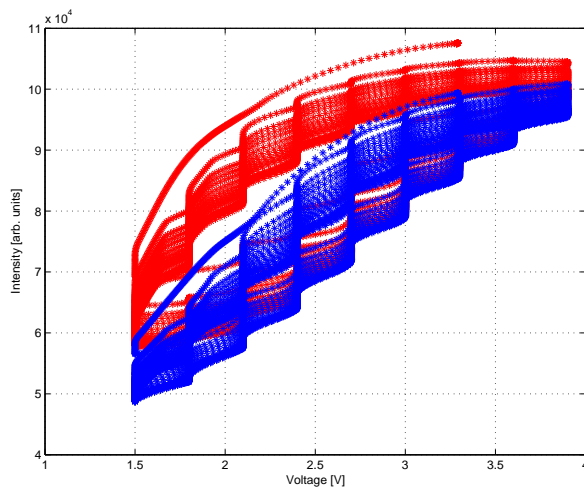


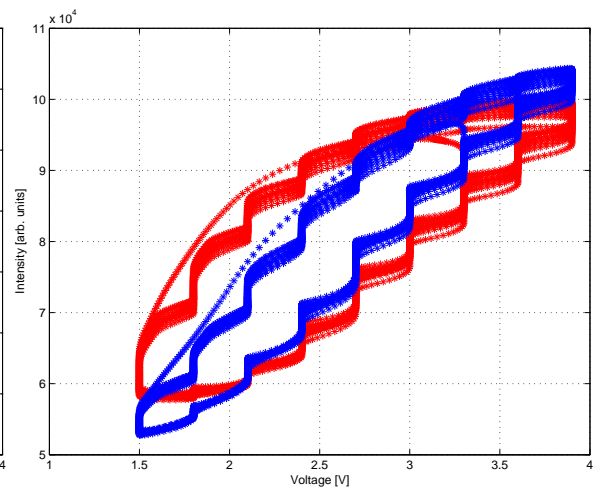
Abbildung 5.13.: MR25Z1P02 - Darstellung des zeitlichen Verlaufs der Wellenlängenbereiche von 560 nm bis 568 nm und 710 nm bis 715 nm in der oberen Abbildung und den Zeitverläufen von Spannung und Ladung im Verlauf der Messung

Zusammenhang zwischen Spannung und Intensität

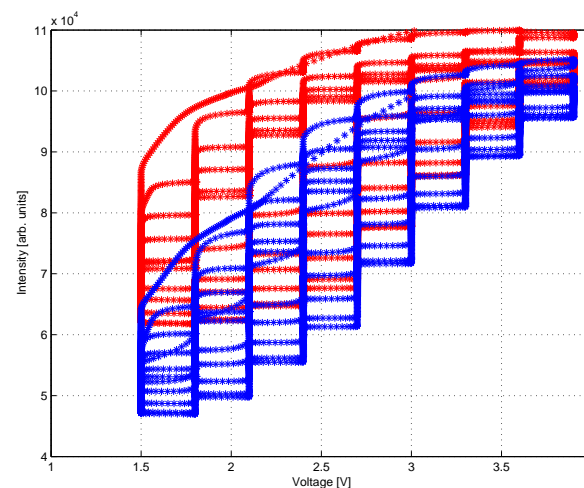
Der Zusammenhang zwischen der Spannung und den Intensitäten der Wellenlängenbereiche von 560 nm bis 568 nm und 710 nm bis 715 nm ist in den folgenden Abbildungen dargestellt. Alle vier Messungen sind im cCcV-Modus zwischen der oberen Schwellspannung von 3,9 V und der unteren Schwellspannung von 1,5 V durchlaufen. Das Hysterese-Verhalten der Verläufe entsteht vermutlich, da die Vorgänge der inneren Zellchemie langsamer als das Spannungsverhalten der Zelle sind und sie bei der Ladung und Entladung ihren Ruhezustand nicht erreichen können. Deutlich zu erkennen ist auch das Einschwingverhalten zu Beginn der Messung. Besonders auffallend ist, dass die jeweils ersten Messungen einer Messreihe wesentlich größere Verschiebungen bei gleichen Spannungen aufweisen, als die darauffolgenden.



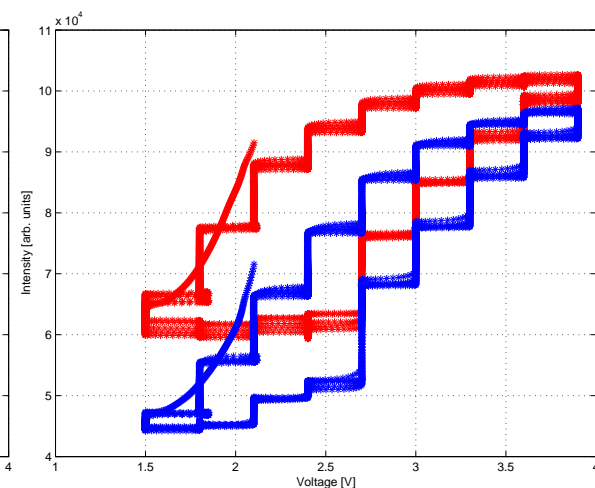
Messreihe 25 Zelle 1 Part 01



Messreihe 25 Zelle 1 Part 02



Messreihe 26 Zelle 1 Part 01



Messreihe 26 Zelle 1 Part 02

Abbildung 5.14.: Zusammenhang zwischen der Spannung und der Intensität des Spektrums
 rot: Wellenlängenbereich von 510 nm bis 568 nm
 blau: Wellenlängenbereich von 710 nm bis 715 nm

5.3.2. ITO - Suche nach Wellenlängenabhängigkeit

Aufgrund der in Kapitel 5.3.1 beschriebenen Beobachtung des zeitlichen Versatzes zwischen den Intensitäten der Wellenlängenbereiche von 510 nm bis 516 nm und von 710 nm bis 715 nm, wurden hierzu weitere Untersuchungen an der Messung MR26Z1P01 vorgenommen. Die Zellaufbauten dieser beiden Messungen sind identisch, jedoch wurden die Haltezeiten der Spannung im cCcV-Modus von 10 Minuten auf eine Stunde erhöht. Zu bewerten gilt es, ob der hieraus resultierende höhere Ladungsunterschied auch eine deutlichere Ausprägung der vorangegangenen Ergebnisse mit sich bringt.

In der folgenden Darstellung ist der Unterschied der Wellenlängen zum Durchschnitt der Messung abgebildet. Um die zeitliche Verschiebung deutlicher darzustellen ist nur ein Ausschnitt aus der Messung ausgewählt worden. Der typische treppenförmige Verlauf des cCcV-Modus ist auch hier klar zu erkennen und sorgt dafür, dass auch eine Verschiebung im Wellenlängenbereich ersichtlich wird. Besonders bei der Entladung, in dieser Darstellung dem Farbwechsel von rot über gelb zum Blauen, ist in den Abstufungen eine farbliche Änderung über die Wellenlängen zu beobachten.

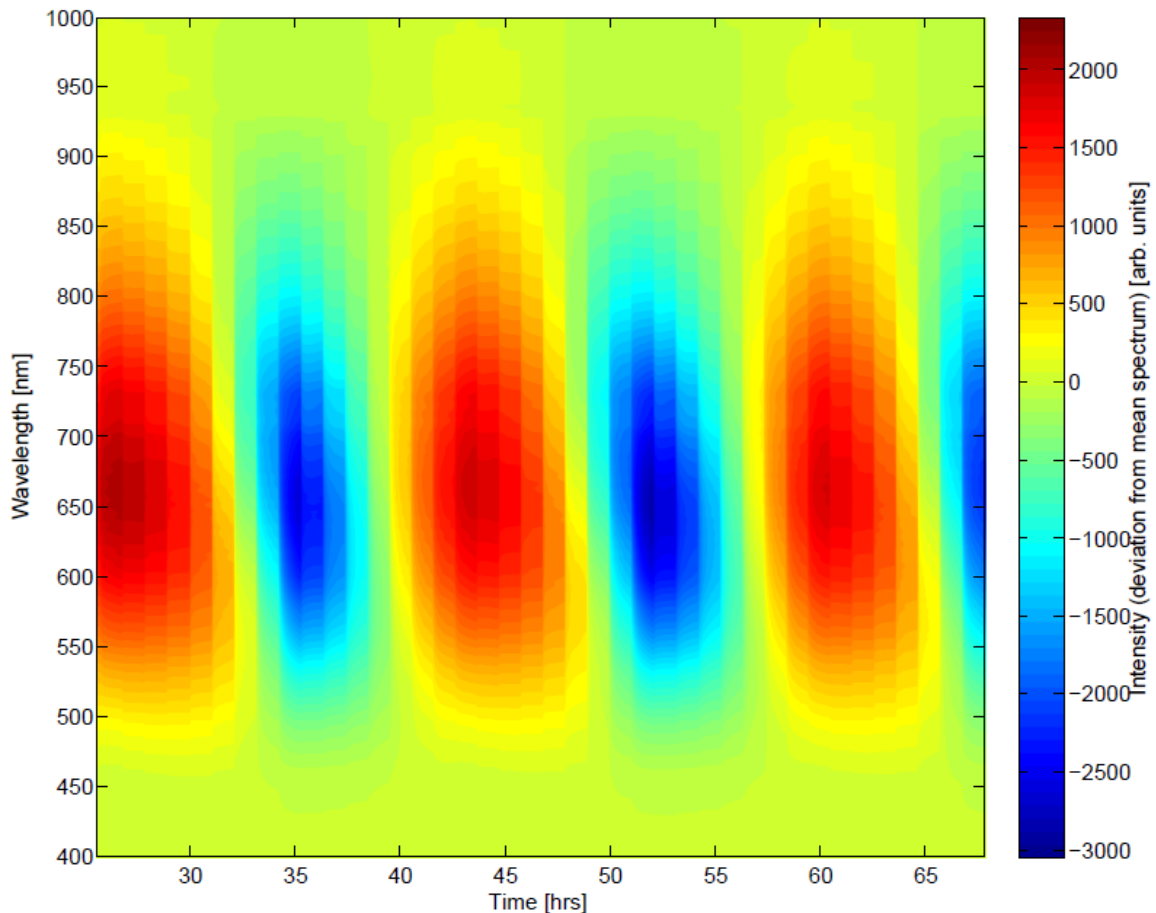


Abbildung 5.15.: MR26Z1P01 - Ausschnitt aus dem Verlauf der Messung. Dargestellt ist die Differenz zum Mittelwert des Spektrums zum jeweiligen Zeitpunkt

Der vollständige Verlauf der Spannung und Ladung von Messung MR26Z1P01 ist in Abbildung 5.16 zusammen mit den Wellenlängenbereichen von 560 nm bis 568 nm und von 710 nm bis 715 nm dargestellt. Die Verschiebung zwischen den Wellenlängenbereichen aus Abbildung 5.13 konnte durch diesen Versuch wiederholt werden.

Aufgrund der langen Aufnahme der Messung über 90 Stunden ist auch deutlich zu erkennen, dass die Intensität über den Gesamtverlauf abfällt. Sowohl die Minima zu den Zeitpunkten der tiefsten Entladung, als auch die Maxima bei Ladeschluss sinken in ihren Intensitätswerten ab. Der Grund liegt in einem Abfall der Ladung, denn die Stromwerte bei der Entladung sind im cCcV-Modus größer als die Ladeströme. Dies ist in Abbildung 5.12 sehr deutlich zu erkennen.

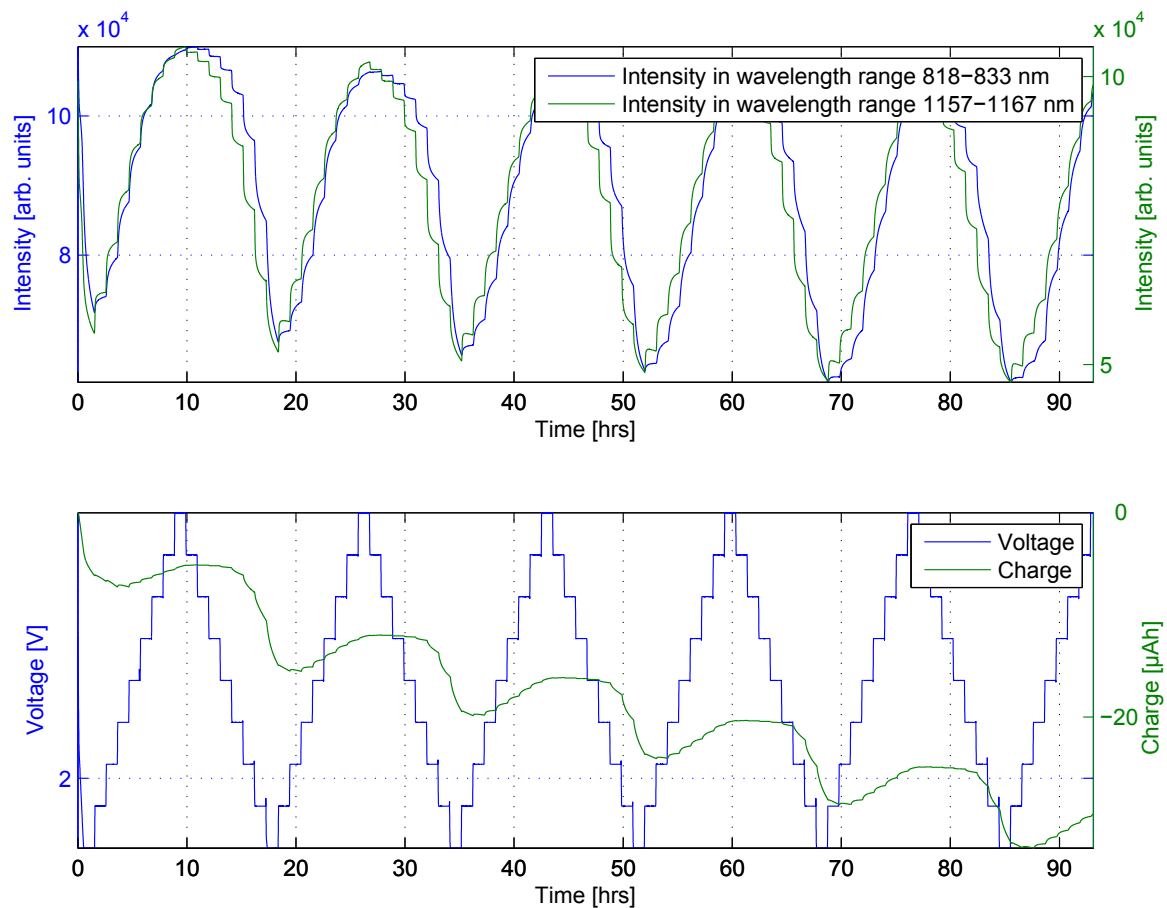


Abbildung 5.16.: MR26Z1P01 - Zeitlicher Verlauf zweier Wellenlängenbereiche (oben) mit den zeitlichen Verläufen von Spannung und Ladung

5.3.3. ITO - Verschiebung des Spektrumschwerpunktes

Die Messung MR25Z1P02 wurde, wie in Kapitel 5.3.1 beschrieben, im cCcV-Modus mit einer Haltezeit der Schwellspannungen für 10 Minuten betrieben. Aufgrund der ungleichen Verschiebungen der Wellenlängen im Spektrum wurde vermutet, dass auch der Schwerpunkt des Spektrums zeitlich nichtlinear verläuft. Aus diesem Grund ist in der oberen Darstellung von Abbildung 5.17 zum Verlauf des Spektrums noch sein Schwerpunkt (Abschnitt 4.1.4) dargestellt und darunter der Verlauf des Schwerpunktes zusammen mit der Spannung abgebildet.

Der Schwerpunkt variiert sehr deutlich zwischen 642 nm zum Zeitpunkt der tiefsten Entladung und 664 nm beim Ladungsmaximum dieses Versuchs. Auch das treppenförmige Verhalten der Spannung ist in der Schwerpunktskurve, besonders im Falle der Aufladung der Zelle, stark ausgeprägt.

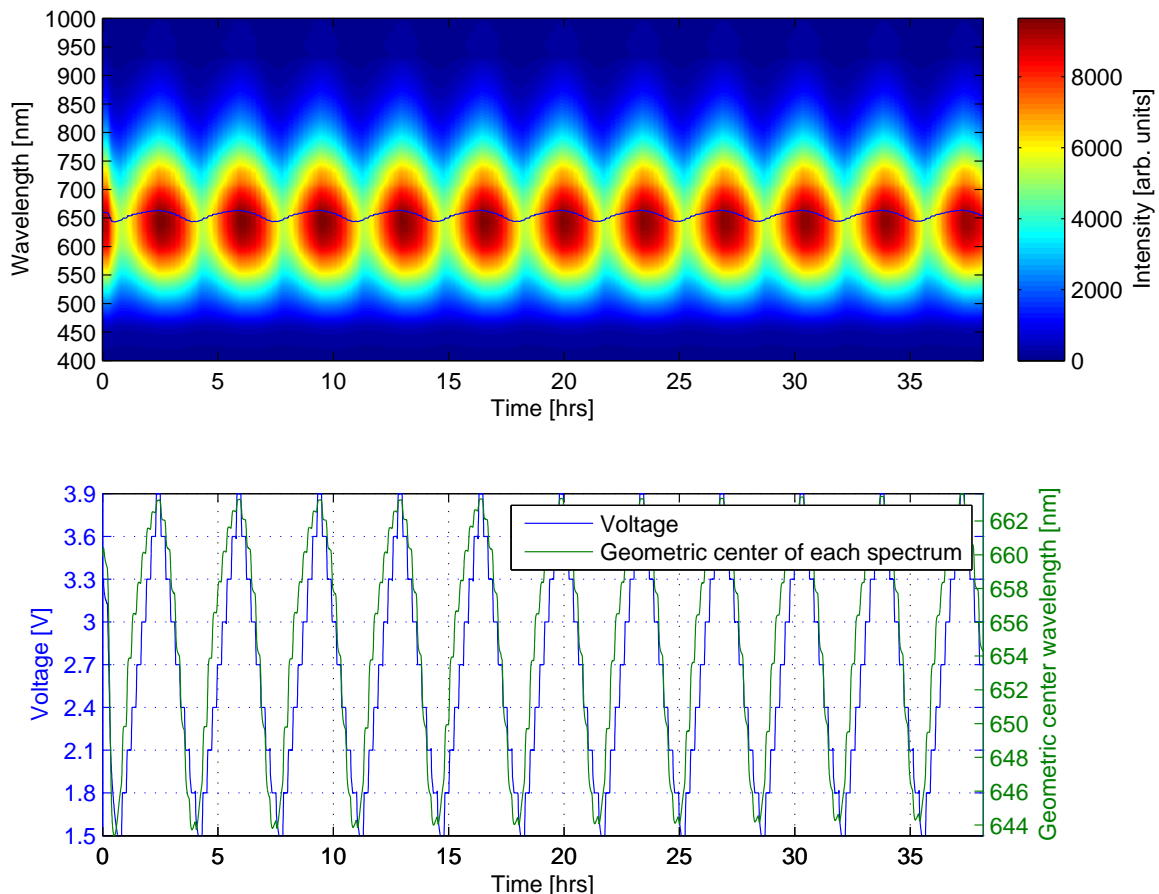


Abbildung 5.17.: MR25Z1P02 - In der oberen Darstellung ist der zeitliche Verlauf des Spektrums mit seinem Schwerpunkt dargestellt, darunter sind die zeitlichen Spannungs- und Ladungsverläufe abgebildet

5.3.4. ITO - Nachanalyse vorhergehender Messungen

In der Arbeit von Herrn Pan [18] wurden mit der Messreihe MR22Z2 die optischen Effekte an einer Testzelle mit ITO als Kathode und Lithium als Anode untersucht. Dabei wurden Gegenläufigkeiten in den Intensitäten der Wellenlängen 650 nm und 450 nm festgestellt und beschrieben.

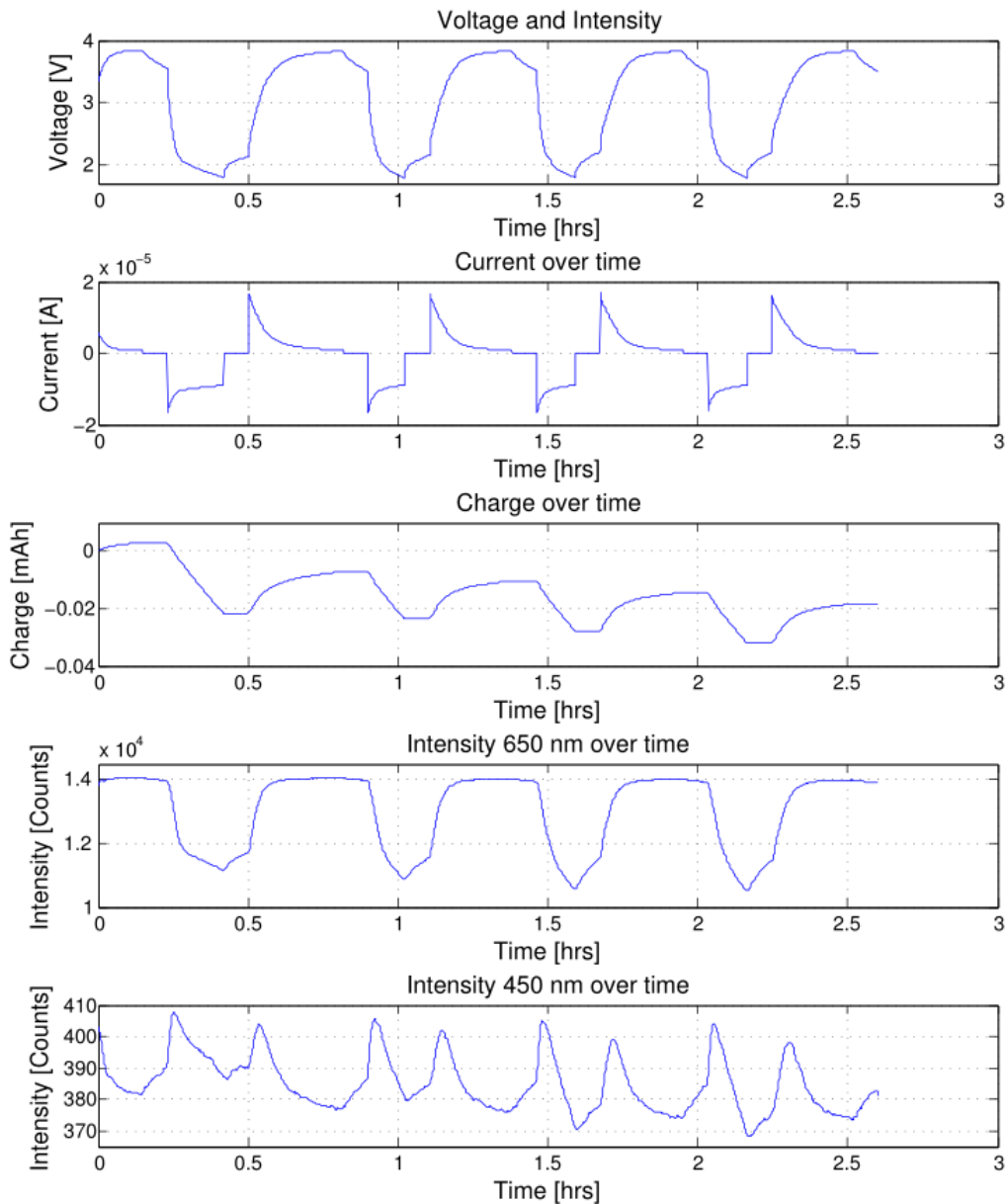


Abbildung 5.18.: MR22Z2 - Zeitlicher Verlauf der elektrische Messdaten in den oberen drei Abbildungen sowie zeitliche Darstellung und Gegenläufigkeit der Wellenlängen 450 nm und 650 nm darunter nach [18]

Die Ansteuerung der Testzelle erfolgte hier nicht mit dem in dieser Arbeit verwendeten Zyklersystem, sie wurde stattdessen über eine Relaiskarte realisiert. Diese lädt die Zelle für 15 Minuten mit anschließender 8-minütiger Entladephase. Zwischen diesen Schritten ist jeweils eine Erholungsphase von 5 Minuten eingefügt. Aus diesem Spannungsverlauf folgt, dass der Strom zu Beginn einer Lade- oder Entladephase sprunghaft ansteigt und anschließend wieder abfällt.

In den unteren zwei Darstellungen von Abbildung 5.18 sind die Unterschiede in den Verläufen der Wellenlängenintensitäten dargestellt. Die Intensität der Wellenlänge 650 nm folgt wie erwartet dem Ladungsverlauf der Zelle. Mit einer Differenz von circa 3000 Counts zwischen Minimum und Maximum des Verlauf ist das optische Verhalten in dieser Wellenlänge sehr ausgeprägt. In der Wellenlänge von 450 nm ist die Ausprägung der Intensitäten und deren Änderungen wesentlich geringer. Für eine Bewertung der Ergebnisse ist die Ausleuchtung dieser Wellenlängen zu gering. Aus diesem Grund ist eine Messung an einer ITO-Testzelle mit der Deuteriumlampe als Lichtquelle durchgeführt worden. Diese hat eine Ausleuchtung mit wesentlich größerer Bandbreite und lässt auch Untersuchungen im Bereich von 450 nm zu (Abschnitt 3.3.2).

In Abbildung 5.19 sind die Intensitätsverläufe der Wellenlängen 450 nm und 650 nm, sowie der Spannungs- und Ladungsverlauf dargestellt. Die Messung wurde über einen Zeitraum von 2 Stunden betrieben. In dieser Zeit wurden zwei Zyklen des cCcV-Modus durchlaufen. Auch in diesen Wellenlängenbereichen ist eine, in Kapitel 5.3.2 beschriebene, Wellenlängenverschiebung deutlich zu erkennen. Der Bereich um die Wellenlänge 650 nm steigt bei einer Zellenladung deutlich früher an als die Wellenlänge 450 nm, fällt jedoch auch früher wieder ab.

Die von Herrn PAN gemachten Gegenläufigkeiten konnten in dieser Messung nicht bestätigt werden. Es wird vermutet, dass es in seiner Auswertung zu diesen Effekten gekommen ist, da die Ausleuchtung nicht ausreichend war und möglicherweise äußere Einflüsse eine weitere Rolle gespielt haben. Möglicherweise ist auch die Kalibrierung des Spektrometers ein Einflussfaktor, da hier eine Glättung erfolgt.

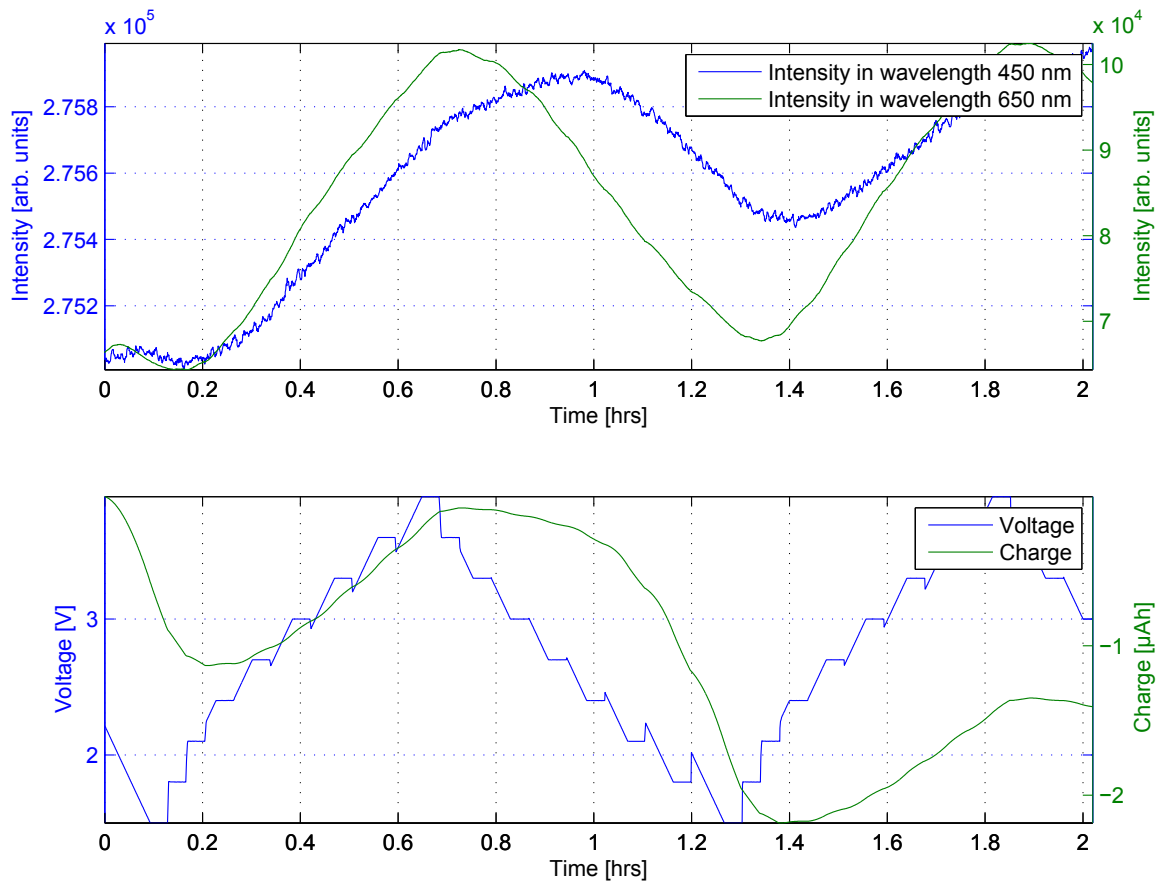


Abbildung 5.19.: MR26Z1P05 - Zeitlicher Verlauf der Wellenlängen 450 nm und 650 nm sowie die dazugehörigen Spannungs- und Ladungsverläufe

5.3.5. LiFePO₄ - Eigene Messungen

In der Bachelorarbeit von Herrn Pan [18] ist mit der Zelle 2 der Messreihe 23 eine Testzelle aus 100 % LiFePO₄ untersucht worden. Die beschriebenen optischen Beobachtungen mit dem Spektrometer sollen erneut durchgeführt und wenn möglich bestätigt werden. Aus diesem Grund ist im Fachbereich Chemie der Universität Hamburg mithilfe einer Deuteriumlampe (Abschnitt 3.3.2) die hier beschriebene Messung MR26Z3P05 durchgeführt worden. Der fast 2 Stunden dauernde Zyklervorgang umfasst einen Lade- und Entladezyklus im cC-Modus. Der eingestellte Strom von 5 μA wird im Ladefall aufgrund der langsamen Ansteuerung des Zyklersystems erst nach nach 15 Minuten erreicht, im Falle der Entladung jedoch deutlich früher. Dadurch fällt der Ladungsverlauf im Laufe des Zyklus erkennbar ab. Auf das Ergebnis der Messung sollte dies jedoch keinen negativen Einfluss haben.

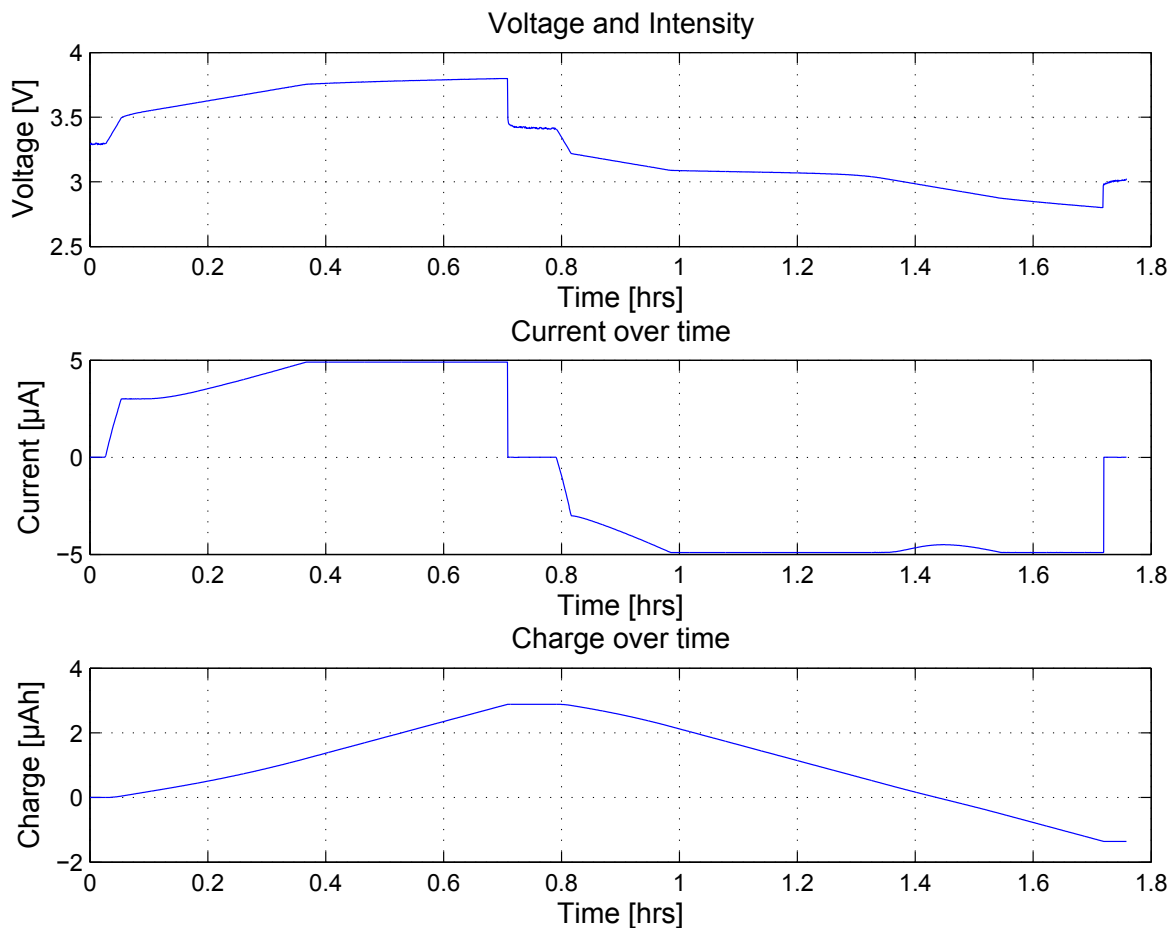


Abbildung 5.20.: MR26Z3P05 - Zeitlicher Verlauf der Spannungs- und Strommessung und der hieraus errechnete Ladungsverlauf im cC-Modus mit 5 μA Ladestrom über einen Zyklus

Aufgrund von schmalbandigen Spitzen im Wellenlängenbereich des Spektrums der Deuteriumlampe, ist die Intensität in der Software des Spektrometers so hoch gesetzt worden, dass einige Bereiche des Spektrums bei einer Intensität von 15000 a.u. in die Aussteuerung gegangen sind. Diese Bereiche sind im Spektrum als dunkelrote Linien ersichtlich.

In der folgenden Abbildung sind das Spektrum und die Spannungs- und Ladungswerte des Verlaufs der Messung dargestellt. Trotz Spannungsunterschied von über 1 V und 4 μAh Ladungsdifferenz sind im Spektrum keine Änderungen der Intensitäten ersichtlich. Die Linien der unterschiedlichen Intensitäten verlaufen linear von Beginn der Messung bis zu ihrem Ende.

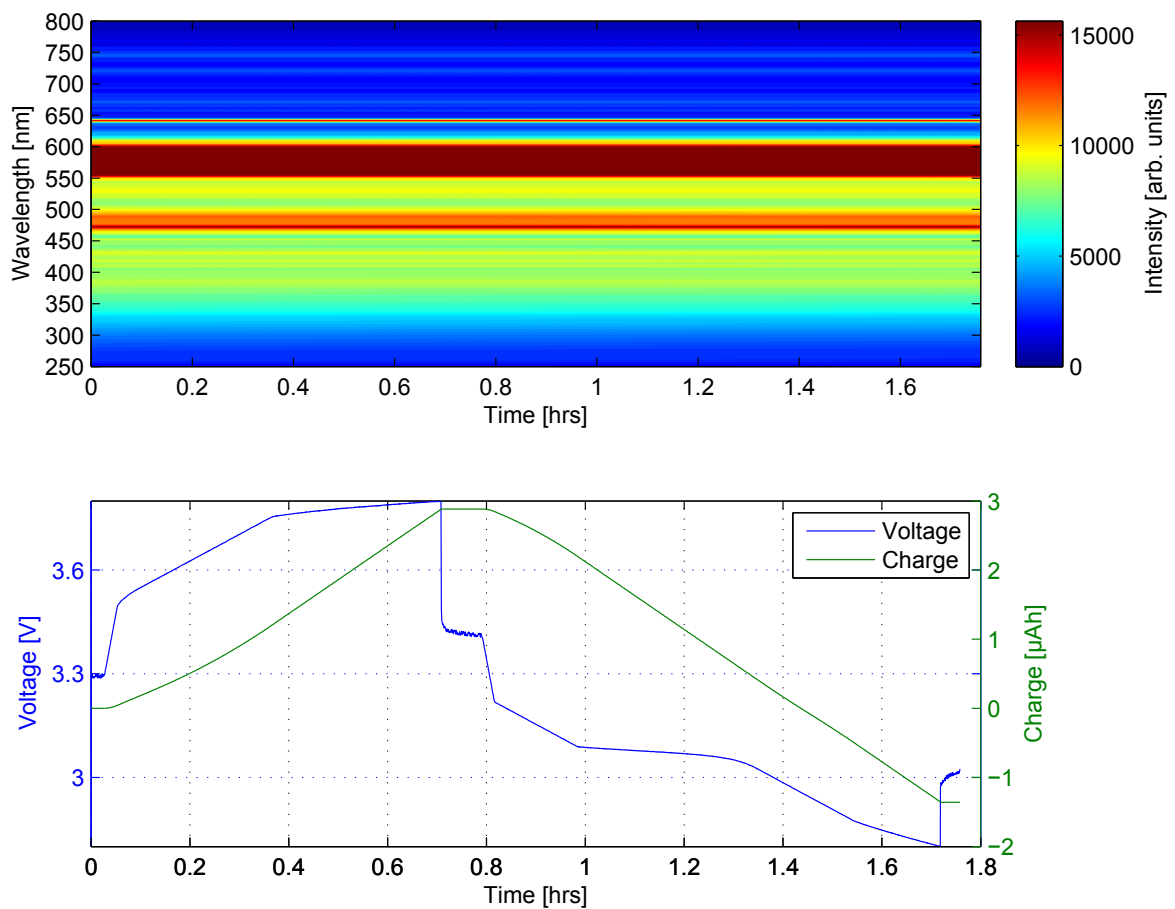


Abbildung 5.21.: MR26Z3P05 - Zeitlicher Verlauf des Spektrums mit den dazugehörigen Spannungs- und Ladungsverläufen in der unteren Abbildung

Wird der komplette Spektrumsverlauf abgebildet, sind aufgrund der großen Intensitätsunterschiede zwischen den Wellenlängen kleine optische Änderungen durch den Lade- und Entladezyklus kaum ersichtlich. Deshalb sind in der folgenden Abbildung 2 Wellenlängenbereiche dargestellt. An ihnen sind die optischen Änderungen sehr gut ersichtlich. Diese folgen jedoch nicht dem Verlauf der Ladung oder Spannung fallen über den gesamten Ladevorgang ab.

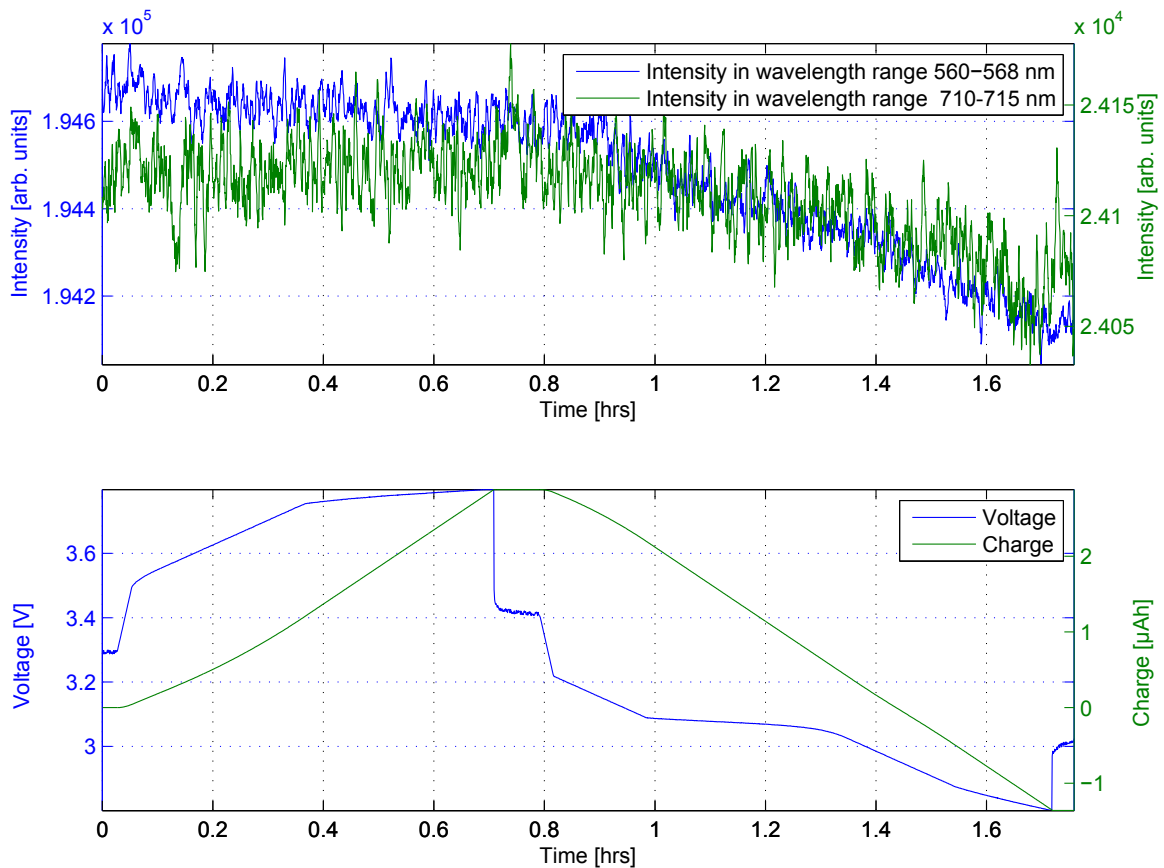


Abbildung 5.22.: MR26Z3P05 - zeitlicher Verlauf zweier Wellenlängen mit den dazu gehörigen Spannungs- und Ladungsverläufen

Aufgrund der nicht erkennbaren Änderungen des in Abbildung 5.21 dargestellten Spektrums, wird an dieser Stelle die Differenz zu seinem Mittelwert abgebildet. Auf diese Weise ist es möglich, auch kleine Änderungen kenntlich zu machen.

Der hier zu erkennende Abfall der Spannung über den gesamten Spannungsverlauf passt zeitlich nicht zu den elektrischen Messwerten, sodass davon ausgegangen werden muss, dass dieser aus Schwankungen in der Intensität der Lichtquelle stammt.

Der lineare grüne Bereich zwischen 550 nm und 600 nm stammt von der Aussteuerung der Spektrometersoftware. Die Intensitäten der Wellenlängen größer als 650 nm ändern sich vermutlich aufgrund der schwachen Ausleuchtung der Lampe in diesem Bereich nicht.

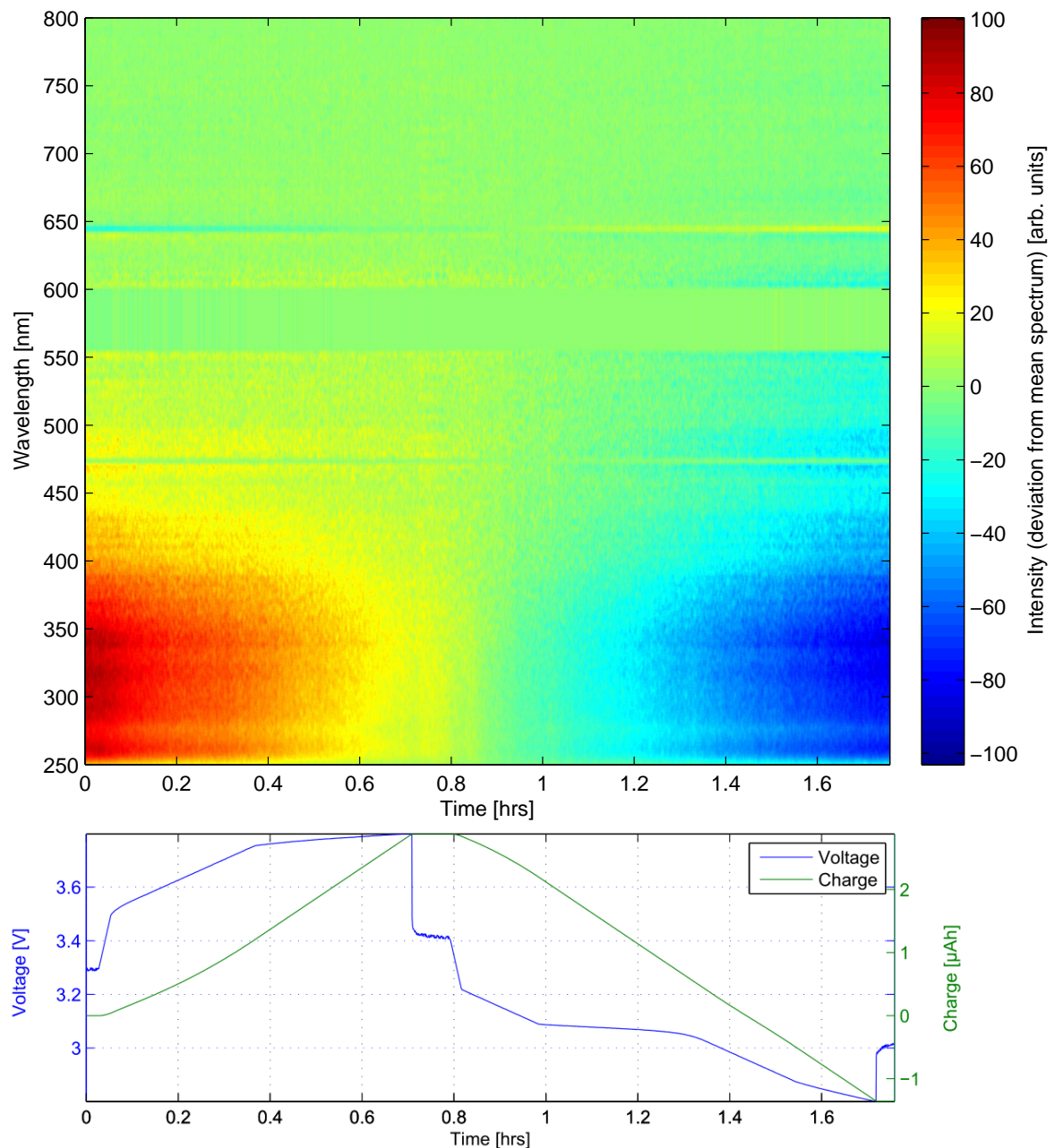


Abbildung 5.23.: MR26Z3P05 - Zeitlicher Verlauf der Differenz des Spektrums zu seinem Mittelwert und die Verläufe der Spannung und Ladung

6. Fazit

Abschließend werden in dieser Arbeit die Ergebnisse der Untersuchungen zusammengefasst und bewertet. Des Weiteren wird ein Ausblick auf weitere Untersuchungen und offene Punkte gegeben.

6.1. Zusammenfassung und Bewertung der Messreihen

In dieser Arbeit wurden erstmalig Messreihen mit dem Zyklersystem aufgenommen. Abschließend lässt sich sagen, dass sich dieses System sehr gut zur parallelen Aufnahme von elektrischen Messdaten und der optischen Beobachtung eignet. Jedoch gibt es in einigen Punkten noch Verbesserungsbedarf. Zum einen wäre eine schnellere Regelung wünschenswert, um den gewählten Sollstrom auch bei kürzeren Zykluszeiten erreichen zu können. Auch die Ansteuerung der Kamera bereitet noch Probleme. Da die Aufnahme bei einem Aufruf aus dem Skript „start_all.sh“ bei einigen Messungen nach kurzer Zeit abgebrochen ist, konnten diese Versuche nur teilweise ausgewertet werden. Aus diesem Grund wurde die Bildaufnahme in späteren Messreihen in einem eigenem Terminal gestartet. Dies löst zwar die Probleme, ein automatisierter Start wäre jedoch wünschenswert.

Die optische Beobachtung mittels Mikroskopkamera hat sowohl bei Versuchen mit ITO als Kathodenmaterial, als auch bei LiFePO_4 -Kathoden, deutliche Ergebnisse gezeigt. Aufgrund der stark ausgeprägten optischen Effekte von ITO, die sogar mit dem menschlichen Auge wahrgenommen werden können, weisen diese Testzellen besonders hohe Intensitätsänderungen in den RGB-Kanälen auf. Wie in Abbildung 5.4 zu erkennen ist, verteilen sich die Effekte ganzflächig auf der ITO-Kathode. Daher ist eine Auswertung der gesamten Fläche sinnvoll. In der Auswertung sind die Änderungen der RGB-Kanäle so eindeutig, dass sogar die treppenförmigen Ladeverläufe in ihnen sichtbar werden.

Zur besseren Beobachtbarkeit der Testzellen mit LiFePO_4 als Kathode, wurde diesen Kathoden in den Messreihen MR25 und MR26 15 % ITO beigemischt. Obwohl sich die optischen Effekte bei ITO-Testzellen über die gesamte Kathode verteilen, sind die Effekte in diesem gemischten Kathodenmaterial fast ausschließlich auf ihren Rand beschränkt. Die Auswertung der Kameradaten umfasst auch diese Bereiche und macht die Effekte in den RGB-Kanälen

sichtbar, wenn auch weniger ausgeprägt als bei reinen ITO-Kathoden. Eine Ansteuerung in cCcV-Modus wäre hier nicht geeignet, da die Ladungs- und Spannungsunterschiede der einzelnen Zyklusschritte so gering sind, dass die optischen Effekte noch weniger ausgeprägt wären. Im cC-Modus wird die Zelle jedoch in einem Schritt einer hohen Ladung- und Spannungsänderung ausgesetzt. Dadurch sind auch an ihr die Effekte deutlich zu erkennen.

Die Erfassung der optischen Effekte durch das Spektrometer ist deutlich detaillierter als die der Kameraaufnahme. Statt 3 RGB-Kanäle werden 2048 Pixel im Bereich von 187 nm bis 1091 nm erfasst. Dies lässt eine ausführlichere Auswertung der erfassten Daten zu. Die Ergebnisse der ITO-Testzellen sind so ausgeprägt, dass Untersuchungen zu Wellenlängenabhängigkeiten und Schwerpunkten des Spektrums gemacht werden können. Die Auswertung von Effekten an LiFePO_4 -Kathoden gestaltet sich hingegen schwierig. Aufgrund des vorrangigen Auftretens der Effekte am Kathodenrand und der Ausrichtung des Spektrometers auf den Mittelpunkt der Elektrode, ist eine zufriedenstellende Auswertung nicht möglich.

6.2. Generelle Bewertung

In dieser Arbeit hat sich gezeigt, dass es zu unterschiedlichen Ergebnissen bei der Beobachtung der Verteilung der optischen Effekte im Spektrum kommt. Diese sind in der folgenden Tabelle zusammengefasst.

	breitbandig	linear ausgeprägt	breitbandig verschoben
Kamera ITO	deutliche Ausprägung	konnte nicht nachgewiesen werden	Verschiebung in RGB-Kanalverläufen
Kamera LiFePO_4	deutliche Ausprägung	konnte nicht nachgewiesen werden	konnte nicht nachgewiesen werden
Spektrometer ITO	deutliche Ausprägung	deutliche Ausprägung	deutliche Ausprägung
Spektrometer LiFePO_4	deutliche Ausprägung	konnte nicht nachgewiesen werden	konnte nicht nachgewiesen werden

Tabelle 6.1.: Nachweisbarkeit der Verteilung optischer Effekte im Spektrum

Eine breitbandige Änderung der optischen Beschaffenheit ist sowohl mit der Kamera, als auch mit dem Spektrometer nachzuweisen. Eine lineare Ausprägung dieser Effekte kann mit der Kamera nicht erfasst werden. Auch mit dem Spektrometer war dies in diesen Versuchen nur an den Testzellen mit ITO als Kathodenmaterial möglich. Die Effekte an der LiFePO_4 -Testzelle sind vermutlich so schwach ausgeprägt, dass sie nicht nachgewiesen

werden konnten.

Eine breitbandige Verschiebung der Spektren konnte in den hier getätigten Untersuchungen an den ITO-Zellen beobachtet werden. Bei den Messungen mit der Kamera ist dies, wie zum Beispiel in Abbildung 5.2 in den Verschiebungen zwischen den RGB-Kanälen zu erkennen. Diese Verschiebungen resultieren vermutlich aus der in der Messung mit dem Spektrometer beobachteten Verschiebung des Schwerpunktes in Abbildung 5.17.

Ob die hier gemachten Beobachtungen zum Bilden eines Referenzwertes für zukünftige Untersuchungen genutzt werden können, dafür gibt es einige Punkte, die in der folgenden Tabelle zusammengefasst werden.

Dafür	Dagegen
<ul style="list-style-type: none"> Die Änderungen der Intensitäten des Spektrums sind deutlich erkennbar 	<ul style="list-style-type: none"> Die aufgetretenen Effekte sind mehrheitlich breitbandig
<ul style="list-style-type: none"> Eine Verschiebung zwischen den Wellenlängen ist bei ITO-Zellen deutlich erkennbar 	<ul style="list-style-type: none"> Bei Zellen mit gemischten ITO- und LiFePO_4-Kathoden ist eine Verschiebung des Schwerpunktes noch nicht beobachtet. Vermutlich ist sie sehr schwach ausgeprägt.
<ul style="list-style-type: none"> Eine Veränderung des Schwerpunktes des Spektrums konnte beobachtet werden 	<ul style="list-style-type: none"> Einige Effekte, wie Gegenläufigkeiten wie in der Arbeit von Herrn, sind nicht reproduzierbar

Tabelle 6.2.: Argumente für und gegen eine Nutzung als Referenzwert

Ein wichtiger Punkt bei der Übertragung der hier gewonnenen Erkenntnisse auf die Anwendung im praktischen Fall ist, dass ein Großteil der in dieser Arbeit beobachteten Effekte im Spannungsbereich der Batterie liegen, die im normalen Batteriebetrieb nicht erreicht werden. In diesem Fall ist also damit zu rechnen, dass die Effekte deutlich weniger ausgeprägt sind.

6.3. Ausblick

Trotz der in dieser Arbeit gewonnenen Erkenntnisse, bleiben noch einige Punkte offen. Zum einen stellt sich die Frage, in welcher Weise sich äußere Einflüsse in den Ergebnissen der Messungen bemerkbar machen. Da sich die Messungen in den Räumlichkeiten der HAW befinden, unterliegen sie auch dem täglichen Verlauf des Raumklimas. Eine Aufnahme der Temperatur der Zelle ist sinnvoll, um einen möglichen Zusammenhang mit zyklischen Schwankungen feststellen zu können. Besser noch wäre zum Beispiel ein Temperaturschrank, der den Messaufbau vor äußeren Einflüssen schützt.

Ein anderer Punkt ist, dass die Ergebnisse der Messungen sehr stark von der eingesetzten Lichtquelle anhängig sind. Die hier bisher eingesetzten Leuchten und LED der Mikroskopkamera haben jeweils sehr ausgeprägte Kurvenverläufe. Optimal wäre eine Lichtquelle mit einem linearem Spektrum. Mithilfe des Surface Lighting Controllers „Sulico“ ist es auch die Testzelle mit rotem, grünem oder blauem Licht von den LED der Kamera zu bestrahlen. Auch ein Mischen dieser Farben ist möglich, um das Spektrum der Beleuchtung in ihrem festgelegten Rahmen zu verändern.

In dieser Arbeit ist es gelungen, lineare Verschiebungen und Verschiebungen des Schwerpunktes der Spektren, der ITO-Zellen nachzuweisen. Dies gilt jedoch nicht für die LiFePO_4 -Testzellen. Zwar konnten Zusammenhänge zwischen dem Ladezustand der Zelle und den optischen Kathodeneffekten dargestellt werden, lineare Verschiebungen und Schwerpunktsänderungen der Intensitäten konnten jedoch nicht beobachtet werden. Zumindest bei Zellen aus einem Gemisch aus LiFePO_4 und ITO wird vermutet, dass diese Effekte dennoch auftreten. Jedoch sind sie so schwach ausgeprägt, dass sie in den Untersuchungen dieser Arbeit nicht nachzuweisen waren. Möglicherweise wäre dies durch genauere Beobachtungen möglich.

Literaturverzeichnis

- [1] Statista. Anzahl der Elektroautos in Deutschland von 2006 bis 2016. <http://de.statista.com/statistik/daten/studie/265995/umfrage/anzahl-der-elektroautos-in-deutschland>.
- [2] Umweltbundesamt. Jährliche Treibhausgasemissionen in Deutschland 1990-2014 nach Kategorie. https://www.umweltbundesamt.de/sites/default/files/medien/376/bilder/dateien/jaehrliche_treibhausgasemissionen_in_deutschland_1990-2014_nach_kategorie.xlsx.
- [3] Masaki Yoshio, Ralph J. Brodd, Akiya Kozawa. *Lithium-Ion-Batteries: Science and Technologies*. Springer, 2010.
- [4] Valentin Roscher. *Battery Monitoring Using Optical Electrode Observation*. 2015.
- [5] INFRALINE s.r.o. Elektromagnetisches Spektrum. <http://www.pion.cz/de/article/?catID=33>.
- [6] Paul A. Tipler, Gene Mosca, Jenny Wagner (Hrsg.). *Physik für Wissenschaftler und Ingenieure*. Springer Spektrum, 2015.
- [7] Matlab. MathWorks Dokumentation. <https://de.mathworks.com/help/images/reducing-the-number-of-colors-in-an-image.html>.
- [8] Tektronix. Datenblatt: DMM4020 digital Multimeter. <http://www.tek.com/sites/tek.com/files/media/media/resources/Tektronix-DMM4020-Digital-Multimeter-Datasheet-6.pdf>.
- [9] Jan Griebbach. *Messaufbau mit Steuer- und Analysesoftware für die optische Zustandsbeobachtung von Lithiumbatterien*. Bachelor Thesis - HAW Hamburg, 2014.
- [10] Torsten Geist. *Entwicklung eines embedded Einplatinensteuersystems für optische Batterieuntersuchung*. Bachelor Thesis - HAW Hamburg, 2015.
- [11] Nico Rieckmann. *Entwicklung einer Regelsoftware auf Linux-Basis für eine optische Batterie-Analyse*. Bachelor Thesis - HAW Hamburg, 2016.

- [12] Mean Well. Datasheet: 15W Single Output Switching Power Supply RS-15 series. <http://www.kosmodrom.com.ua/pdf/RS-15-3.3.pdf>.
- [13] Gerth. Datenblatt: Sicherheits-Netztransformator Baureihe 421.XX2. http://cdn-reichert.de/documents/datenblatt/C500/DS_421XX2.pdf.
- [14] oowltech. Datenblatt: Digital-Mikroskop-Kamera USB 9.0MP 200X. http://www.produktinfo.conrad.com/datenblaetter/175000-199999/191377-an-01-ml-DIGITAL_MIKROSKOPKAMERA_USB_de_en_fr_nl.pdf.
- [15] Ocean Optics. HR2000+ Spectrometer, Installation and Operation Manual. <http://oceanoptics.com/wp-content/uploads/hr2000-.pdf>.
- [16] StellarNet. SL1 Tungsten Halogen Light Sources for VIS - NIR. <http://www.stellarnet.us/wp-content/uploads/SL1-Halogen-light.pdf>.
- [17] Hamamatsu. Datasheet: L10290 High Power UV-VIS fiber light source. http://www.hamamatsu.com/resources/pdf/etd/L10290_TLSZ1010E.pdf.
- [18] Fengjunjie PAN. *Untersuchung des Zusammenhangs zwischen elektrischen und optischen Messdaten von Lithium-Ionen-Mikrozellen*. Bachelor Thesis - HAW Hamburg, 2016.
- [19] El-Cell GmbH. User Manual: ECC-Opto-Std, Release 2.5 . http://el-cell.com/wp-content/uploads/downloads/manuals/Manual_ECC-Opto-Std_Release_2.5.pdf.
- [20] M. Braun Inertgas-Systeme GmbH. LABstar Glove Box Workstation. <http://mbraun.de/products/glovebox-workstations/labstar-glovebox/>.
- [21] Matlab. MathWorks Dokumentation. <http://de.mathworks.com/help/matlab/>.
- [22] Don Mithila Meshan Palliyaguruge. *mage processing for investigation of effects in Lithium battery electrodes*. Bachelor Thesis - HAW Hamburg, 2015.

Tabellenverzeichnis

2.1. Vergleich von Kathodenmaterialien nach [3]	15
3.1. Eingaben in das Parameterfile für den Zyklusbetrieb	35
3.2. Grundeinstellungen der Spektrometersoftware „OceanView“	37
5.1. Aufbau und Materialien der Testzellen	53
5.2. Messungen der Messreihe 25	54
5.3. Messungen der Messreihe 26	55
5.4. Messungen der Messreihe 27	55
6.1. Nachweisbarkeit der Verteilung optischer Effekte im Spektrum	83
6.2. Argumente für und gegen eine Nutzung als Referenzwert	84

Abbildungsverzeichnis

1.1. Anteil der Sektoren an den Treibhausgasemissionen in Deutschland 2014 [2]	9
2.1. Aufbau einer galvanischen Zelle	13
2.2. Optischer Effekt der Entladung an der Kathode einer Lithium-Eisenphosphat Batterie [4]	17
2.3. Elektromagnetisches Spektrum [5]	18
2.4. Reflexionen an glatten (a) und rauen (b) Oberflächen [6]	19
2.5. Visualisierung des RGB-Farbmodells in Form eines Würfels modifiziert nach [7]	20
3.1. Laboraufbau des Messsystems ohne elektrischen Aufbau	24
3.2. Elektrischer Aufbau der Messungen	25
3.3. Geöffnetes Steuergerät des Messaufbaus A Raspberry PI B Steuerplatine C Spannungsversorgung D Anschluss der Testzelle	27
3.4. Vergleich der Messgenauigkeiten von internem und externem Messgerät bei $20 \mu\text{A}$ in eingeschwungenem Zustand an einem $4,7 \text{k}\Omega$ Widerstand	29
3.5. Spektrum der LED-Beleuchtung der Mikroskopkamera	30
3.6. Das Spektrometer „HR2000+“ mit seinen Komponenten nach [15]	32
3.7. Vergleich der Leuchtspektren, reflektiert von einer weißen Fläche	34
3.8. Grafische Oberfläche der Software „OceanView“ [18]	36
3.9. Grafische Oberfläche der Software „OceanView“ [18]	38
3.10. Explosionszeichnung der Testzelle „ECC-Opto-Std“ [19]	39
3.11. Aufbau der Testzellen nach [18]	
1. Einsetzen der Anode	
2. Einbringen des Separators	
3. Einsetzen der Kathode	
4. Füllen der Zelle mit dem Elektrolyten	40
4.1. Vergleich von ungeglättetem zu über 9 Messwerte geglättetem Spektrum	43
4.2. Abdriften der Intensität einer nicht näher definierten Wellenlänge [18]	45
4.3. Vorverarbeitung der Daten bei Nutzung der Mikroskopkamera	49
4.4. Vorverarbeitung der Daten bei Nutzung des Spektrometers	51
5.1. MR25Z2P02 - Flächenmarkierungen auf der Elektrode	56

5.2. MR25Z2P02 - Zeitliche Verläufe der elektrischen Messdaten und der Intensitäten der RGB-Kanäle während der Messung im cCcV-Modus	57
5.3. MR25Z2P02 - Sichtbarwerden der Unterschiede im Zeitverlauf der RGB-Farbkanäle bei gleichem Ladungsverlauf	58
5.4. MR25Z2P02 - Optische Änderungen der Elektrode im Verlauf der Messung oben: Zustand kurz vor tiefstem Ladezustand der Zelle unten: Zustand nach dem Laden der Zelle	59
5.5. MR25Z3P03 - Flächenmarkierungen auf der Elektrode und von oben rechts nach unten links: 1. zeitlicher Verlauf der Spannung 2. zeitlicher Verlauf des Stroms 3. Histogramm, quantitatives Auftreten der RGB-Werte 4. zeitlicher Verlauf der RGB-Kanäle 5. zeitlicher Verlauf der Ladung	61
5.6. MR25Z3P03 - Zeitliche Verläufe der elektrischen Messdaten und der Intensitäten der RGB-Kanäle während der Messung	62
5.7. MR25Z3P03 - Zusammenhang der zeitlichen Verläufe der RGB-Kanalintensitäten und - in den oberen 3 Abbildungen: dem Ladungsverlauf - in den unteren 3 Abbildungen: dem Spannungsverlauf	63
5.8. MR25Z3P03 - Optische Änderungen der Elektrode im Verlauf der Messung oben: Zustand zum Zeitpunkt der maximalen Ladung unten: Zustand zum Zeitpunkt der maximalen Entladung	65
5.9. MR26Z4P02 - Zeitlicher Verlauf der elektrischen Messdaten und RGB-Farbkanäle über einen längeren Zeitraum und deutlich erkennbare Alterung der Zelle	66
5.10. MR26Z4P02 - Zusammenhang der RGB-Farbkanäle mit dem Ladezustand der Zelle links: roter Farbkanal mitte: grüner Farbkanal rechts: blauer Farbkanal	67
5.11. MR25Z1P02 - Typische Verläufe der elektrischen Messdaten im cCcV-Modus und deutlich erkennbarer Abfall der Ladung über den Zeitraum der Messung .	68
5.12. MR25Z1P02 - Zusammenhang zwischen den Intensitäten des Spektrums (oben) und der Spannung und Ladung (unten) im zeitlichen Verlauf der Messung	69
5.13. MR25Z1P02 - Darstellung des zeitlichen Verlaufs der Wellenlängenbereiche von 560 nm bis 568 nm und 710 nm bis 715 nm in der oberen Abbildung und den Zeitverläufen von Spannung und Ladung im Verlauf der Messung	70
5.14. Zusammenhang zwischen der Spannung und der Intensität des Spektrums rot: Wellenlängenbereich von 510 nm bis 568 nm blau: Wellenlängenbereich von 710 nm bis 715 nm	71
5.15. MR26Z1P01 - Ausschnitt aus dem Verlauf der Messung. Dargestellt ist die Differenz zum Mittelwert des Spektrums zum jeweiligen Zeitpunkt	72
5.16. MR26Z1P01 - Zeitlicher Verlauf zweier Wellenlängenbereiche (oben) mit den zeitlichen Verläufen von Spannung und Ladung	73

5.17.MR25Z1P02 - In der oberen Darstellung ist der zeitliche Verlauf des Spektrums mit seinem Schwerpunkt dargestellt, darunter sind die zeitlichen Spannungs- und Ladungsverläufe abgebildet	74
5.18.MR22Z2 - Zeitlicher Verlauf der elektrische Messdaten in den oberen drei Abbildungen sowie zeitliche Darstellung und Gegenläufigkeit der Wellenlängen 450 nm und 650 nm darunter nach [18]	75
5.19.MR26Z1P05 - Zeitlicher Verlauf der Wellenlängen 450 nm und 650 nm sowie die dazugehörigen Spannungs- und Ladungsverläufe	77
5.20.MR26Z3P05 - Zeitlicher Verlauf der Spannungs- und Strommessung und der hieraus errechnete Ladungsverlauf im cC-Modus mit 5 μ A Ladestrom über einen Zyklus	78
5.21.MR26Z3P05 - Zeitlicher Verlauf des Spektrums mit den dazugehörigen Spannungs- und Ladungsverläufen in der unteren Abbildung	79
5.22.MR26Z3P05 - zeitlicher Verlauf zweier Wellenlängen mit den dazu gehörigen Spannungs- und Ladungsverläufen	80
5.23.MR26Z3P05 - Zeitlicher Verlauf der Differenz des Spektrums zu seinem Mittelwert und die Verläufe der Spannung und Ladung	81

A. Quellcode

A.1. Zyklersystem

A.1.1. bc_param.cfg

```
1  /*****
2  * File :          bc_param.cfg
3  * Version:        0.3
4  * Date:           2016-04-29
5  * Author:        Nico Rieckmann
6  *
7  * Discription:   This is a configuration file to parameterize the inner settings of battery
8  *                cycling software. It will be read by MainController process.
9  * Info:          Syntax Symbols for comments are '//', '#' and '/' *...*/'. Trailing simicolon
10 *                is optional and whitespace is not significant.
11 * References:    This configuration file uses the C lib "libconfig-1.5" from Marc A. Lindner.
12 *
13 * Changes:       0.3: NR tidy up all and added some more
14 *                0.2: NR added impedance and mode_time parameters
15 *                0.1: NR first version.
16 *
17 * *****/
18
19 version = 0.3;
20
21
22 batterygroup:
23 {
24     type           = "LiFePO";           //Lithium -Eisenphosphat Zelle
25     impedance      = 320.0;             //Toleranzwiderstand in [Ohm]
26 }
27
28
29 limitgroup:      # Grenzwerte
30 {
31     max_voltage    = 3.90;              //obere Spannungsgrenze in [V]
32     min_voltage    = 2.70;              //untere Spannungsgrenze in [V]
33     max_temp       = 4.0564;           //obere Temperaturgrenze in [Â°C]
34     min_temp       = 4.0564;           //untere Temperaturgrenze in [Â°C]
35     max_current_charge = 1000.0;       //maximaler Ladestrom in [µA]
36     max_current_discharge = 1000.0;   //maximaler Entladestrom in [µA]
37 }
38
39
40 cyclegroup:      # Zyklierungs Parameter
41 {
42     //"mode":       Three dis-/charging modes are available: cCcV (const. Current const. Voltage),
43     //             cC (const. Current) or cV (const. Volatage).
44     //"cycles":     Number of battery cycling iterations: int value or "0" for startphase only.
45     //"mode_time":  Countdown to end the mode: int value or "0" for no timing limit.
46     //             The activation of mode_time is depending on modes:
47     //             1. "cV":   mode_time begins after current falls below nominal current.
48     //             2. "cC":   mode_time begins at start of this mode.
49     //             3. "cCcV": mode_time is only for latter cV phase possible.
50
51     startphase:    # Startphase soll Zelle auf folgende Werte bringen
52     {
```



```

53     mode           = "cC";           //Ladeverfahren
54     mode_time      = 5;              //Zeitliche Modus Begrenzung in [s]
55     threshold_voltage = 3.16;       //Schwellspannung, Ladeschlussspannung in [V]
56     nominal_current = 1.0;         // Sollstrom in [ $\mu$ A]
57     convalescence  = 1;            // Erholungszeit in [s]
58     led_R          = 10.68;        //Led-Rot-Wert in [%]
59     led_G          = 90.50;        //Led-Gr $\frac{1}{4}$ n-Wert in [%]
60     led_B          = 100.0;        //Led-Blau-Wert in [%]
61 }
62
63
64 cyclephase:      # Zyklierungsphase
65 {
66     cyclelist = (
67     {
68         //Entladung mit konst. Strom
69         mode           = "cC";       //konst Strom Entladung
70         mode_time      = 0;          //Zeitliche Modus Begrenzung in [s]
71         threshold_voltage = 2.80;    //Schwellspannung, Entladeschlussspannung in [V]
72         nominal_current = -1.0;     // Sollstrom in [ $\mu$ A]
73         convalescence  = 10;        // Erholungszeit in [s]
74         led_R          = 0.0;       //Led-Rot-Wert in [%]
75         led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
76         led_B          = 0.0;       //Led-Blau-Wert in [%]
77     },
78     {
79         //Ladung mit konst. Strom und konst. Spannung
80         mode           = "cC";       //Ladeverfahren
81         mode_time      = 100;        //Zeitliche Modus Begrenzung in [s]
82         threshold_voltage = 3.80;    //Schwellspannung, Ladeschlussspannung in [V]
83         nominal_current = 1.0;      // Sollstrom in [ $\mu$ A]
84         convalescence  = 10;        // Erholungszeit in [s]
85         led_R          = 0.0;       //Led-Rot-Wert in [%]
86         led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
87         led_B          = 0.0;       //Led-Blau-Wert in [%]
88     },
89     {
90         //Entladung mit konst. Strom
91         mode           = "cC";       //konst Strom Entladung
92         mode_time      = 100;        //Zeitliche Modus Begrenzung in [s]
93         threshold_voltage = 2.80;    //Schwellspannung, Entladeschlussspannung in [V]
94         nominal_current = -20.0;     // Sollstrom in [ $\mu$ A]
95         convalescence  = 10;        // Erholungszeit in [s]
96         led_R          = 0.0;       //Led-Rot-Wert in [%]
97         led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
98         led_B          = 0.0;       //Led-Blau-Wert in [%]
99     },
100    {
101        //Ladung mit konst. Strom und konst. Spannung
102        mode           = "cC";       //Ladeverfahren
103        mode_time      = 100;        //Zeitliche Modus Begrenzung in [s]
104        threshold_voltage = 3.80;    //Schwellspannung, Ladeschlussspannung in [V]
105        nominal_current = 20.0;     // Sollstrom in [ $\mu$ A]
106        convalescence  = 10;        // Erholungszeit in [s]
107        led_R          = 0.0;       //Led-Rot-Wert in [%]
108        led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
109        led_B          = 0.0;       //Led-Blau-Wert in [%]
110    },
111    {
112        //Entladung mit konst. Strom
113        mode           = "cC";       //konst Strom Entladung
114        mode_time      = 100;        //Zeitliche Modus Begrenzung in [s]
115        threshold_voltage = 2.80;    //Schwellspannung, Entladeschlussspannung in [V]
116        nominal_current = -100.0;    // Sollstrom in [ $\mu$ A]
117        convalescence  = 10;        // Erholungszeit in [s]
118        led_R          = 0.0;       //Led-Rot-Wert in [%]
119        led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
120        led_B          = 0.0;       //Led-Blau-Wert in [%]
121    },
122    {
123        //Ladung mit konst. Strom und konst. Spannung
124        mode           = "cC";       //Ladeverfahren
125        mode_time      = 100;        //Zeitliche Modus Begrenzung in [s]
126        threshold_voltage = 3.80;    //Schwellspannung, Ladeschlussspannung in [V]
127        nominal_current = 100.0;     // Sollstrom in [ $\mu$ A]
128        convalescence  = 10;        // Erholungszeit in [s]
129        led_R          = 0.0;       //Led-Rot-Wert in [%]
130        led_G          = 0.0;       //Led-Gr $\frac{1}{4}$ n-Wert in [%]
131        led_B          = 0.0;       //Led-Blau-Wert in [%]
132    }
133 );

```

```

129     cycles = 1;           //Anzahl an Wiederholungen der Cyclelist
130   }
131 };

```

A.1.2. start_all.sh

```

1 # This skript starts the batcycle programs and kill camera process when cycling is finished
2 # Version 1.0 2016-01-01 NR
3 # Version 1.1 2016-06-27 VR      added copying of template folder
4
5
6 # User definable variables
7
8 CELL="MR25Z3P00"
9 COMMENT="LFP microcell"
10 STARTDATE=$(date +%m-%d-%Y)
11 DATA_LOCATION="/media/B3Mass2/Cells/MR25"
12 FOLDER=$DATA_LOCATION/$CELL
13
14 initFile="in/bc_init.cfg"      #input files
15 paramFile="in/bc_param.cfg"
16
17 # Folder preparation
18
19 mkdir -p $FOLDER              #create new folder on HDD
20 cp -pr template/* $FOLDER
21
22 # Copy cycle plan and supplementary information into data folder
23 echo $COMMENT >> $FOLDER/Data/cycle_plan.log
24 cat start_all.sh >> $FOLDER/Data/cycle_plan.log
25 cat in/bc_param.cfg >> $FOLDER/Data/cycle_plan.log
26
27 # Obsolete, enable if there is no folder template
28 # mkdir -p $FOLDER
29 # mkdir -p $FOLDER/Images
30 # mkdir -p $FOLDER/Data
31
32 # Fixed variables
33
34 sockFile="/tmp/lsock.uds"     #socket file of Amperemeter
35 sockFileV="/tmp/vsock.uds"   #socket file of Voltmeter
36 sockFileT="/tmp/tsock.uds"   #socket file of Temperature
37 sockFileD="/tmp/dsock.uds"   #socket file of VoltageController
38 sockFileR="/tmp/rsock.uds"   #socket file of RelayController
39
40 logFile=$FOLDER'/Data/'$CELL'_ '$STARTDATE'_logI.txt'      #output files on HDD
41 logFileV=$FOLDER'/Data/'$CELL'_ '$STARTDATE'_logV.txt'
42 logFileT=$FOLDER'/Data/'$CELL'_ '$STARTDATE'_logT.txt'
43
44 PICS=$FOLDER/Images/'%s'.png
45
46 #Amperemeter
47 Ixterminal --command="./BC_DMM4020Reader_SP -d /dev/ttyUSB0 -p $sockFileI -m 2" --title="DMM4020
  Amperemeter" --geometry=74x20
48 #Ixterminal --command="sudo ./BC_Amperemeter_SP -p $sockFileI -n 3" --title="BC_Amperemeter_SP" --
  geometry=74x20
49 Ixterminal --command="sudo ./BC_Amperemeter_SP -p $sockFileT -n 3" --title="BC_Amperemeter_SP" --
  geometry=74x20
50
51 #Voltmeter
52 #Ixterminal --command="./BC_DMM4020Reader_SP -d /dev/ttyUSB0 -p $sockFileV -m 0" --title="DMM4020
  Voltmeter" --geometry=74x20
53 Ixterminal --command="sudo ./BC_Voltmeter_SP -p $sockFileV -n 3" --title="BC_Voltmeter_SP" --geometry
  =74x20
54
55 #Temperature
56 #Ixterminal --command="sudo ./BC_Temperature_SP -p $sockFileT -n 3" --title="BC_Temperature_SP" --
  geometry=74x20
57
58 #controllers
59 Ixterminal --command="sudo ./BC_VoltageController_SP -p $sockFileD -n 3" --title="
  BC_VoltageController_SP" --geometry=74x20
60 Ixterminal --command="sudo ./BC_RelayController_SP -p $sockFileR -n 3" --title="BC_RelayController_SP
  " --geometry=74x20

```

```

61
62 sleep 8           #wait for servers
63
64 #DataLogger
65 #!xterminal --command="/BC_DataLogger_CP $sockFile:$logFile:2:1000 $sockFileV:$logFileV:2:1000
    $sockFileT:$logFileT:0:0" --title="BC_DataLogger_CP" --geometry=74x20
66 !xterminal --command="/BC_DataLogger_CP $sockFile:$logFile:2:1000 $sockFileV:$logFileV:2:1000" --
    title="BC_DataLogger_CP" --geometry=74x20
67
68 #MainController
69 !xterminal --command="/BC_MainController_CP -i $initFile -p $paramFile" --title="
    BC_MainController_CP" --geometry=74x20 &
70 PID_MAIN=$!
71
72 #start camera tool fswebcam in background
73 #!xterminal --comand='fswebcam -d /dev/video0 --loop 10 -F5 --png 0 --scale 640x480 --set "White
    Balance Temperature, Auto=False" --set "Backlight Compensation=0" --save $PICS' --title="
    fswebcam" --geometry=74x20
74 fswebcam -d /dev/video0 --loop 10 -F5 --png 0 --scale 800x600 --set "White Balance Temperature, Auto
    =False" --set "Backlight Compensation=0" --save $PICS &
75 PID_CAM=$!
76
77
78
79 sleep 1
80
81 echo "kill fswebcam"
82 #loop checks if MainController has finished, for kill fswebcam
83 while [ true ]; do
84     var=`ps a | grep BC_MainController_CP | wc -l`      #filter prozess view for BC_MainController_CP
    and count results
85
86     if test $var -le 1; then          #if results less than 1, MainController has finished
87         echo "MainController is finished"
88         sudo kill -SIGINT $PID_CAM      #kill fswebcam process
89         break
90     fi
91     sleep 10
92 done
93

```

A.2. CurrentIntegrator

```

1  /*****
2  * Filter to calculate the integral of current measurements made
3  * with "DMM4020_reader" or "DMM4020_reader_v2"
4  *
5  * USAGE: ./CurrentIntegrator < infile.txt > outfile.txt
6  *
7  *
8  * File:    CurrentIntegrator.c
9  * Version: 0.1
10 * Date:    2014-05-07
11 * Author:  Jan Griessbach
12 *
13 *****/
14
15 #include <stdio.h>
16 #include <stdlib.h>
17
18 #define BUFFERSIZE 32
19 #define LINESIZE 2
20
21 int main( void )
22 {
23     char    strTime[BUFFERSIZE]; /* #####.###\0 */
24     char    strValue[BUFFERSIZE]; /* +##.###E-# XXXX\0 */
25     char    line[LINESIZE+BUFFERSIZE];
26     double  value, prevValue;
27     double  time, prevTime;
28     double  charge = 0;
29     unsigned long lineCnt = 0;
30

```

```

31  /* loop through all lines */
32  while( NULL != fgets(line, LINESIZE*BUFFERSIZE, stdin)) {
33      /* current integration at non-comment lines only */
34      if(line[0] != '#') {
35
36          /* isolate timestamp and value */
37          sscanf(line, "%s\t%s\n", strTime, strValue);
38          time = atof(strTime);
39          value = atof(strValue);
40
41          /* calculate charge, except on first line*/
42          if(lineCnt > 0)
43              charge += (time - prevTime) * (prevValue + value) / 2;
44
45          /* backup data */
46          prevTime = time;
47          prevValue = value;
48          lineCnt++;
49
50          /* output */
51          fprintf(stdout, "%s\t%E\n", strTime, charge);
52      }
53      else {
54          /* copy comment lines */
55          fprintf(stdout, "%s", line);
56      }
57      fflush(stdout); /* line buffering, if output is redirected via pipe */
58  }
59  return EXIT_SUCCESS;
60  }

```

A.3. TimestampMatcher

```

1  /******
2  * Filter to extract measurements that are close to given
3  * timestamps
4  *
5  * USAGE: ./TimestampMatcher -f timestampfile < infile.txt > outfile.txt
6  *
7  *
8  * File:    TimestampMatcher.c
9  * Version: 0.1
10 * Date:    2014-10-09
11 * Author:  Jan Griessbach
12 *
13 * *****/
14
15 #include <stdio.h>
16 #include <unistd.h>
17 #include <stdlib.h>
18
19 #define BUFFERSIZE 32
20 #define LINESIZE 2
21
22 void usage(char *progname) {
23     fprintf(stderr, "\tUSAGE:\t %s -f timestampfile\n"
24             "\n\tOPTIONS:"
25             "\n\t\t%s\t -f\tfile containing the timestamps to match\n",
26             progname
27     );
28 }
29
30 int main( int argc, char *argv[] )
31 {
32     char          mandatoryOptions = 1;
33     FILE          *fp;
34     char          *tmstpfile;
35     int           opt;
36     char          strTime[BUFFERSIZE]; /* #####.###\0 */
37     char          line1[LINESIZE*BUFFERSIZE];
38     char          line2[LINESIZE*BUFFERSIZE];
39     char          *line = line1, *prevLine = line2;
40     double        imgTime, time;

```

```

41
42
43  /* evaluate commandline options */
44  while( (opt = getopt(argc, argv, "f:")) != -1 ) {
45      switch( opt ) {
46          case 'f': tmstmpfile = optarg;
47                  mandatoryOptions--;
48                  break;
49          default: usage(argv[0]);
50                  exit(EXIT_FAILURE);
51      }
52  }
53
54  if(mandatoryOptions > 0) {
55      usage(argv[0]);
56      exit(EXIT_FAILURE);
57  }
58
59  /* open timestamp file */
60  if( (fp = fopen(tmstmpfile, "r")) == NULL ) {
61      fprintf(stderr, "Can't open timestamp file.\n");
62      return EXIT_FAILURE;
63  }
64
65  /* process the timestamp file */
66  while(NULL != fgets(line, LINESIZE+BUFFERSIZE, fp)) {
67      /* skip empty lines */
68      if(line[0] == '\n')
69          continue;
70      imgTime = atof(line);
71
72      /* process standard input */
73      while( NULL != fgets(line, LINESIZE+BUFFERSIZE, stdin)) {
74          /* skip empty lines */
75          if(line[0] == '\n')
76              continue;
77
78          /* process non-comment lines only */
79          if(line[0] != '#') {
80              /* isolate timestamp */
81              sscanf(line, "%s\t*\n", strTime);
82              time = atof(strTime);
83
84              if( imgTime < time )
85                  break;
86
87              /* backup line */
88              line = (line == line1) ? line2 : line1;
89              prevLine = (line == line2) ? line1 : line2;
90          }
91          else {
92              /* copy comment lines */
93              fprintf(stdout, "%s", line);
94              fflush(stdout);
95          }
96      }
97      fprintf(stdout, "%s", prevLine);
98      fflush(stdout);
99  }
100  return EXIT_SUCCESS;
101  }

```

A.4. Shellskripte

A.4.1. chmod.sh

```

1  chmod +x ./Data/CurrentIntegrator
2  chmod +x ./Scripts/TimestampMatcher
3  tempfile=./Data/*logI.txt
4  ./Data/CurrentIntegrator < $tempfile > ./Data/logQ.txt

```

A.4.2. stacking-100.sh

```

1 # Edit 3 zeros and one ? for different
2 # Script needs pythonmagick installed to run.
3 # Install via sudo apt-get install python-pythonmagick
4
5
6 # Image prefix
7 part="MR27Z2P02_"
8
9 # Image read/write loop
10 for f in {14758496..14762217}
11 do
12     echo "Processing $f"
13     TMSTMP=`stat -c "%Y" ../Images/$part$(printf $f)00.png`
14     python halostack.py -a ../Images-100s-stack/$part$(printf $TMSTMP).png ../Images/$part$(printf $f)
15     echo $TMSTMP
16     touch -d @$TMSTMP ../Images-100s-stack/$part$(printf $TMSTMP).png
17 done

```

A.4.3. DataOperator.sh

```

1 # 2016-10-26 VR - Collected various procedures and merged into this script
2 #
3 # Starts SpectrumCollectors and creates TCVCDaten file
4 echo -e 'Welcome to m10_DataOperator. I will prepare electrical data and spectra.'
5 Spannung=" ../Data/*logV.txt"
6 Strom=" ../Data/*logI.txt"
7
8 echo -e 'Setting File Permissions...'
9
10 chmod +x ../Data/TimestampMatcher
11 chmod +x ../Data/CurrentIntegrator
12
13 echo -e 'Collecting Timestamps...'
14
15 for file in ../Spectra/*.txt; do
16     timestmp=$(stat -c %Y $file)
17     echo "$timestmp" >> ../Output/Data/tempimagetimestamp.dat
18 done
19
20 tr -d '\r' < ../Output/Data/tempimagetimestamp.dat > ../Output/Data/imagetimestamp.dat
21
22
23 echo -e 'Collecting Electrical Data...'
24 cd ../Data/
25
26 tr -d '\r' < $Spannung|grep -v '#'|awk '{print $2}'> tempspannung.dat
27 tr -d '\r' < $Strom|grep -v '#'|awk '{print $2}'> tempstrom.dat
28 ./CurrentIntegrator <$Strom |grep -v '#'>logQ.txt
29
30 paste -d " " logQ.txt tempspannung.dat tempstrom.dat > tempTCVCDaten.dat
31 cat tempTCVCDaten.dat |./TimestampMatcher -f ../Output/Data/imagetimestamp.dat > ../Output/Data/
32   TCVCDaten.dat
33 rm tempspannung.dat
34 rm tempstrom.dat
35 rm tempTCVCDaten.dat
36 rm ../Output/Data/tempimagetimestamp.dat
37
38 cd ../Scripts/
39
40 # Collects OceanOptics spectra and outputs format compatible with matlab script "spectrometer2matlab"
41
42 echo -e 'Collecting Spectra...'
43
44 for file in ../Spectra/*.txt; do
45     timestmp=$(stat -c %Y $file)
46     echo "#Time of acquisition: $timestmp" >> spectrumfor3d.tmp
47     tail -n 2048 $file >> spectrumfor3d.tmp

```

```

48     echo "" >> spectrumfor3d.tmp
49     echo "" >> spectrumfor3d.tmp
50 done
51
52 tr -d '\r' < spectrumfor3d.tmp > ../Output/Data/spectrumfor3d.dat
53
54 rm spectrumfor3d.tmp
55
56 echo -e 'Done'

```

A.5. Matlabskripte

A.5.1. config.m

```

1 % Version 1.2 2016-09-09 VR - added shift_from_UTC
2
3
4 function param = config()
5
6
7 param.xposl = 337;
8 param.xposr = 681;
9 param.yposu = 241;
10 param.yposd = 550;
11
12 param.ref_xposl = 52;
13 param.ref_xposr = 178;
14 param.ref_yposu = 49;
15 param.ref_yposd = 297;
16
17 % One mm in pixels (calculate from hole diameter!)
18 param.onemm = 90;
19
20 % 1 for winter, 2 for summer, 8 for China
21 param.shift_from_UTC = 2;
22
23 param.tempVolt = 'temp_Voltdata.txt';
24 param.tempCurrent = 'temp_Currentdata.txt';
25 param.tempCharge = 'temp_Chargedata.txt';
26
27 param.ImageRGB_data = 'temp_RGBdata';
28 param.voltageFile = '../Data/MR27Z2P02_10-07-2016_logV.txt';
29 param.currentFile = '../Data/MR27Z2P02_10-07-2016_logI.txt';
30
31 % use CurrentIntegrator or chmod.sh to generate chargefile!
32 param.chargeFile = '../Data/logQ.txt';
33 param.imageSource = '../Images/';
34 param.imageDest = '../Output/Images/MR27Z2P02-100s-stack-plotframe';
35
36 %Image Processing results
37 param.imageProcessingDest1 = '../Output/Images/BineryImage/MR27Z2P02-100s-stack-plotframe';
38 param.imageProcessingDest2 = '../Output/Images/BineryImagePercentage/MR27Z2P02-100s-stack-plotframe';
39
40 %charge_vs_percentage
41 param.imageProcessingDest3 = '../Output/Images/charge_vs_percentage/MR27Z2P02-100s-stack-plotframe';
42
43 %binaryImage_affectedRegion_pixel_countgraph_vidgen
44 param.imageProcessingDest6 = '../Output/Images/binaryImage_affectedRegion_pixel_countgraph_vidgen/
45 MR27Z2P02-100s-stack-plotframe';
46
47 %Q_percent_intensity
48 param.imageProcessingDest4 = '../Output/Images/Plots/MR27Z2P02-100s-stack-plotframe';
49
50 %charge_vs_percentage_noisecheck
51 param.imageProcessingDest5 = '../Output/Images/charge_vs_percentage_noisecheck/MR27Z2P02-100s-stack-
52 plotframe';
53
54 % parameters for grant plotter
55 param.imageArea_data = 'Area_in_percent';
56 param.grantPlotDest = '../Output/Plots/MR27Z2P02-100s-stack';

```

```
57 param.multithresImageDest = '../Output/Images/Multithreshold'
```

A.5.2. m01_imageRead.m

```
1 %% v1.6 2016-04-25 DMMP, VR,WSQ, VR
2 %% v1.6: Added shift_from_UTC from config.m file
3 % v1.5: Removed most Plotting functions, only kept the basic plots
4 % v1.4:
5 % v1.3: Removed Plot function, renamed
6 % v1.2: Merged with Plot function from rgbvoltcurrentchargeplotter
7
8 %% clean up
9 close all;
10 clear all;
11 clc;
12
13 %% this codes links to config.m file in the working directory
14 parser = config;
15
16 %% get the number of parts
17 parts = str2num(char(inputdlg('Number of parts', 'Number of parts', 1, {'1'}) ));
18
19 %% allocate memory for arrays
20 xpos = zeros(parts, 1);
21 ypos = zeros(parts, 1);
22 xsize = zeros(parts, 1);
23 ysize = zeros(parts, 1);
24 xpos_ref = zeros(parts, 1);
25 ypos_ref = zeros(parts, 1);
26 xsize_ref = zeros(parts, 1);
27 ysize_ref = zeros(parts, 1);
28 path = cell(parts, 1);
29 channels = cell(7, parts); % RGB timestamps RGB(reference)
30 noFiles = zeros(parts, 1);
31
32 %% get paths and section information
33 for m=1:1:parts
34     path(m) = cellstr(uigetdir(parser.imageSource));
35
36     xpos(m)= parser.xposl;
37     ypos(m)= parser.yposu;
38     xsize(m)= parser.xposr - parser.xposl;
39     ysize(m)= parser.yposd - parser.yposu;
40
41     xpos_ref(m) = parser.ref_xposl;
42     ypos_ref(m) = parser.ref_yposu;
43     xsize_ref(m) = parser.ref_xposr - parser.ref_xposl;
44     ysize_ref(m) = parser.ref_yposd - parser.ref_yposu;
45     clear invalues;
46 end
47
48 %% process images
49 for m=1:1:parts
50     % get all .png files in given directory
51     files = dir(fullfile(char(path(m)), '*.png'));
52
53     % pick number of files to be read and num of pixels
54     noFiles(m) = size(files, 1);
55     noPixels = xsize(m) * ysize(m);
56     noPixels_ref = xsize_ref(m) * ysize_ref(m);
57
58     % initialize variables
59     red = zeros(noFiles(m), 2);
60     green = zeros(noFiles(m), 2);
61     blue = zeros(noFiles(m), 2);
62     dates = zeros(noFiles(m), 1);
63     imname= cell(noFiles(m), 1);
64
65     % process every file
66     for n=1:1:noFiles(m)
67         % get image one by one and read its data
68         [A, MAP, ALPHA] = imread(fullfile(char(path(m)), files(n).name), 'PNG');
69         dates(n) = files(n).datenum;
70         imname(n) = cellstr(files(n).name);
```



```

71
72 % extract section
73 section = A(ypos(m):ypos(m)+ysize(m)-1, xpos(m):xpos(m)+xsize(m)-1, :);
74 section_ref = A(ypos_ref(m):ypos_ref(m)+ysize_ref(m)-1, xpos_ref(m):xpos_ref(m)+xsize_ref(m)
75 -1, :);
76 clear A MAP ALPHA;
77
78 % sum-up intensities on every channel
79 red(n,1) = sum(sum(section(:, :, 1))) / noPixels;
80 green(n,1) = sum(sum(section(:, :, 2))) / noPixels;
81 blue(n,1) = sum(sum(section(:, :, 3))) / noPixels;
82 red(n,2) = sum(sum(section_ref(:, :, 1))) / noPixels_ref;
83 green(n,2) = sum(sum(section_ref(:, :, 2))) / noPixels_ref;
84 blue(n,2) = sum(sum(section_ref(:, :, 3))) / noPixels_ref;
85 clear section section_ref;
86
87 disp(['Part: ' num2str(m) '/' num2str(parts) ' File: ', num2str(n), '/', num2str(noFiles(m)
88 )]);
89
90 end
91 channels{1, m} = red(:,1);
92 channels{2, m} = green(:,1);
93 channels{3, m} = blue(:,1);
94 channels{4, m} = dates;
95 channels{5, m} = red(:,2);
96 channels{6, m} = green(:,2);
97 channels{7, m} = blue(:,2);
98 clear n red green blue dates files noPixels noPixels_ref;
99
100 end
101
102 % concatenate parts
103 red = vertcat(channels{1, :});
104 green = vertcat(channels{2, :});
105 blue = vertcat(channels{3, :});
106 dates = vertcat(channels{4, :});
107 red_ref = vertcat(channels{5, :});
108 green_ref = vertcat(channels{6, :});
109 blue_ref = vertcat(channels{7, :});
110 clear channels m;
111
112 %% apply correction
113 red_c = (red ./ red_ref) * mean(red_ref);
114 green_c = (green ./ green_ref) * mean(green_ref);
115 blue_c = (blue ./ blue_ref) * mean(blue_ref);
116
117 %% export data to a text and a mat file
118 % [exportfile exportpath] = uiputfile;
119 exportfile = strcat(pwd, '/', parser.ImageRGB_data);
120 udates = round(864e5 * (dates - datenum('1970', 'yyyy')) ./ 1000 - parser.shift_from_UTC + 3600; % Daylight
121 Saving time: -3600, Summer time: -7200
122 exportMatrix = [udates red green blue red_c green_c blue_c];
123 dlmwrite(exportfile, sprintf('#timestamp\tred (raw)\tgreen (raw)\tblue (raw)\tred (compensated)\
124 \tgreen (compensated)\tblue (compensated)', 'delimiter', ''));
125 dlmwrite(exportfile, exportMatrix, '-append', 'delimiter', '\t', 'precision', 15);
126 save(strcat(parser.ImageRGB_data, '.mat'), 'exportMatrix', 'imname');
127 clear exportpath udates exportMatrix;
128
129
130 % get data as a mat file
131 A = load(strcat(parser.ImageRGB_data, '.mat'));
132 filenames = A.imname(:,1);
133
134 % get the image time stamps to a text file
135 [status dates] = system(['sed "1{/^#/d;};s/[\\t].*/g" ' pwd '/' parser.ImageRGB_data]);
136 disp(dates);
137
138 % to do: compare udates, dates for high difference to prevent
139 % TimestampMatcher misbehavior
140
141 % save image timestamps in a temporary text file
142 fid = fopen('imageTimeStamps.txt', 'wt');
143 fprintf(fid, '%s\n', dates);
144
145 %
146 system(['cat ' parser.voltageFile '|grep -v "#"|./TimestampMatcher -f ./imageTimeStamps.txt >'
147 parser.tempVolt]);

```

```

144 system(['cat ' parser.currentFile '|grep -v "#"|./TimestampMatcher -f ./imageTimeStamps.txt >'
        parser.tempCurrent]);
145 system(['cat ' parser.chargeFile '|grep -v "#"|./TimestampMatcher -f ./imageTimeStamps.txt >'
        parser.tempCharge]);
146
147 % To match only correct formatting (I could not get \tab and -? to work, so two greps):
148 % system(['cat ' parser.voltageFile '|grep -E "[0-9]{10}\.[0-9]{3}"|grep -E "[0-9]{1}\.[0-9]{6}E
        [+ -][0-9]{2}$"|./TimestampMatcher -f ./imageTimeStamps.txt >' parser.tempVolt]);
149 % system(['cat ' parser.currentFile '|grep -E "[0-9]{10}\.[0-9]{3}"|grep -E "[0-9]{1}\.[0-9]{6}E
        [+ -][0-9]{2}$"|./TimestampMatcher -f ./imageTimeStamps.txt >' parser.tempCurrent]);
150 % system(['cat ' parser.chargeFile '|grep -E "[0-9]{10}\.[0-9]{3}"|grep -E "[0-9]{1}\.[0-9]{6}E
        [+ -][0-9]{2}$"|./TimestampMatcher -f ./imageTimeStamps.txt >' parser.tempCharge]);
151
152
153 % TimestampMatcher output is faulty if it cannot find a match within ~1000
154 % seconds. First lines or complete output will be garbled. Also check
155 % plausibility of data for forgotten DST/time zone mismatch
156
157 voltMatrix = dlmread(parser.tempVolt);
158 currentMatrix = dlmread(parser.tempCurrent);
159 chargeMatrix = dlmread(parser.tempCharge);
160
161 % delete (parser.tempVolt , parser.tempCurrent , parser.tempCharge); % Data
162 % is needed by m02_plotCreate
163 % delete ('imageTimeStamps.txt'); % Modified m03_videoCreate
164 % needs imageTimeStamps.txt
165
166 %% Plot data
167
168 unixTime = A.exportMatrix(:,1);
169 imageTime = (unixTime - A.exportMatrix(1,1))/3600; % Time from start in hours. If using imageDate,
        uncomment datetick commands and adjust label!
170 date = chargeMatrix(:,1); % Date/time
171 chargeData = chargeMatrix(:,2)/3600*1000*1000; % Charge in  $\mu$ Ah
172 voltData = voltMatrix(:,2); % Voltage in V
173 currentData = currentMatrix(:,2)+1000*1000; % Current in  $\mu$ A
174 color_red_unc = A.exportMatrix(:,2);
175 color_green_unc = A.exportMatrix(:,3);
176 color_blue_unc = A.exportMatrix(:,4);
177 color_red_cor = A.exportMatrix(:,5);
178 color_green_cor = A.exportMatrix(:,6);
179 color_blue_cor = A.exportMatrix(:,7);
180 color_all_cor = (A.exportMatrix(:,5) + A.exportMatrix(:,6) + A.exportMatrix(:,7));
181
182
183 %% Charge and Intensity over time for red, green, blue
184
185 cit = figure(1);
186 set(cit, 'Units', 'centimeters');
187 pos = get(cit, 'Position');
188 set(cit, 'PaperPositionMode', 'auto');
189 set(gca, 'linewidth', 1.5);
190 set(gca, 'GridLineStyle', '-');
191 set(cit, 'Position', [0 0 29.7 21]);
192 set(cit, 'PaperOrientation', 'landscape');
193
194 % Charge and Intensity over time for red
195 subplot(3,1,1);
196 [ax1,p1,p2] = plotyy(imageTime,chargeData,imageTime,color_red_cor,'plot');
197 %datetick(ax1(1),'keeplimits'); datetick(ax1(2),'keeplimits');
198 set(p1,'color','K');set(p2,'color','R')
199 xlabel('Time [hrs]', 'FontSize',12);
200 ylabel(ax1(1),'Charge [ $\mu$ Ah]', 'FontSize',12);
201 ylabel(ax1(2),'Intensity - Red', 'FontSize',12);
202 legend('charge','Intensity - Red');
203 title('Charge and Intensity - Red over Time', 'FontSize',12);
204 grid on;
205
206 % Charge and intensity over time for green
207 subplot(3,1,2);
208 [ax2,p3,p4] = plotyy(imageTime,chargeData,imageTime,color_green_cor,'plot');
209 %datetick(ax2(1),'keeplimits'); datetick(ax2(2),'keeplimits');
210 set(p3,'color','K');set(p4,'color','G')
211 xlabel('Time [hrs]', 'FontSize',12);
212 ylabel(ax2(1),'Charge [ $\mu$ Ah]', 'FontSize',12);
213 ylabel(ax2(2),'Intensity - Green', 'FontSize',12);
214 legend('charge','Intensity - Green');
215 title('Charge and Intensity - Green over Time', 'FontSize',12);

```

```

216 grid on;
217
218 % Charge and intensity over time for blue
219 subplot(3,1,3);
220 [ax3,p5,p6] = plotyy(imageTime,chargeData,imageTime,color_blue_cor,'plot');
221 %datetick(ax3(1),'keeplimits');datetick(ax3(2),'keeplimits');
222 set(p5,'color','k');set(p6,'color','b')
223 xlabel('Time [hrs]','FontSize',12);
224 ylabel(ax3(1),'Charge [ $\mu$ Ah]','FontSize',12);
225 ylabel(ax3(2),'Intensity - Blue','FontSize',12);
226 legend('charge','Intensity - Blue');
227 title('Charge and Intensity - Blue over Time','FontSize',12);
228 linkaxes([ax1,ax2,ax3],'x')
229 grid on;
230
231 %% Voltage , Current , Charge , Intensity over time
232 vcci = figure(2);
233 set(vcci,'Units','centimeters');
234 pos = get(vcci,'Position');
235 set(vcci,'PaperPositionMode','auto');
236 grid on
237 set(vcci,'PaperOrientation','landscape');
238
239 ax1 = subplot(4,1,1);
240 plot(imageTime,voltData,'k');
241 xlabel('Time [hrs]');
242 ylabel('Voltage [V]');
243 title('Voltage over Time');
244 axis tight;
245 grid on;
246
247 ax2 = subplot(4,1,2);
248 plot(imageTime,currentData,'k');
249 xlabel('Time [hrs]');
250 ylabel('Current [ $\mu$ A]');
251 title('Current over Time');
252 axis tight;
253 grid on;
254
255 ax3 = subplot(4,1,3);
256 plot(imageTime,chargeData,'k');
257 xlabel('Time [hrs]');
258 ylabel('Charge [ $\mu$ Ah]');
259 title('Charge over Time');
260 axis tight;
261 grid on;
262
263 ax4 = subplot(4,1,4);
264 plot(imageTime,color_red_cor,'r',imageTime,color_green_cor,'g',imageTime,color_blue_cor,'b','
  LineWidth',2);
265 xlabel('Time [hrs]');
266 ylabel('intensity');
267 title('Absolute values of Intensity , Compensated');
268 axis tight;
269 grid on;
270 grid on;
271
272 linkaxes([ax1,ax2,ax3,ax4],'x')

```

A.5.3. m02_plotCreate.m

```

1 %% 2016-03-29 DMMP, VR, WSQ
2 % v1.5: Add the Plot function
3 % v1.4: Cleaned up functions , fixed graphs
4 % v1.3: Removed Plot function , renamed
5 % v1.2:Merged with Plot function from rgbvoltcurrentchargeplotter
6
7 %% clean up
8 close all;
9 clear all;
10 clc;
11
12 %% this codes links to config.m file in the working directory

```

```

13 parser = config;
14
15 % get data as a mat file
16 A = load(strcat(parser.ImageRGB_data, '.mat'));
17 voltMatrix = dlmread(parser.tempVolt);
18 currentMatrix = dlmread(parser.tempCurrent);
19 chargeMatrix = dlmread(parser.tempCharge);
20 %delete (parser.tempVolt, parser.tempCurrent, parser.tempCharge);
21
22 %% Plot data
23
24 unixTime = A.exportMatrix(:,1);
25 imageTime = (unixTime - A.exportMatrix(1,1))/3600; % Time from start in hours. If using imageDate,
    uncommment datetick commands and adjust label!
26 date = chargeMatrix(:,1); % Date/time
27 chargeData = chargeMatrix(:,2)/3600*1000*1000; % Charge in uAh
28 voltData = voltMatrix(:,2); % Voltage in V
29 currentData = currentMatrix(:,2)*1000*1000; % Current in uA
30 color_red_unc = A.exportMatrix(:,2);
31 color_green_unc = A.exportMatrix(:,3);
32 color_blue_unc = A.exportMatrix(:,4);
33 color_red_cor = A.exportMatrix(:,5);
34 color_green_cor = A.exportMatrix(:,6);
35 color_blue_cor = A.exportMatrix(:,7);
36 color_all_cor = (A.exportMatrix(:,5) + A.exportMatrix(:,6) + A.exportMatrix(:,7));
37 color_min_cor = (A.exportMatrix(:,5) + A.exportMatrix(:,6) + A.exportMatrix(:,7));
38
39
40 %% Charge and Intensity over time for red, green, blue
41
42 cit = figure(1);
43 set(cit, 'Units', 'centimeters');
44 pos = get(cit, 'Position');
45 set(cit, 'PaperPositionMode', 'auto');
46 set(gca, 'linewidth', 1.5);
47 set(gca, 'GridLineStyle', '-');
48 set(cit, 'Position', [0 0 29.7 21]);
49 set(cit, 'PaperOrientation', 'landscape');
50
51 % Charge and Intensity over time for red
52 subplot(3,1,1);
53 [ax1, p1, p2] = plotyy(imageTime, chargeData, imageTime, color_red_cor, 'plot');
54 %datetick(ax1(1), 'keeplimits'); datetick(ax1(2), 'keeplimits');
55 set(p1, 'color', 'K'); set(p2, 'color', 'R');
56 xlabel('Time [hrs]', 'FontSize', 12);
57 ylabel(ax1(1), 'Charge [uAh]', 'FontSize', 12);
58 ylabel(ax1(2), 'Intensity - Red', 'FontSize', 12);
59 legend('charge', 'Intensity - Red');
60 title('Charge and Intensity - Red over Time', 'FontSize', 12);
61 set(ax1, {'ycolor'}, {'k'; 'k'})
62 grid on;
63
64 % Charge and intensity over time for green
65 subplot(3,1,2);
66 [ax2, p3, p4] = plotyy(imageTime, chargeData, imageTime, color_green_cor, 'plot');
67 %datetick(ax2(1), 'keeplimits'); datetick(ax2(2), 'keeplimits');
68 set(p3, 'color', 'K'); set(p4, 'color', 'G');
69 xlabel('Time [hrs]', 'FontSize', 12);
70 ylabel(ax2(1), 'Charge [uAh]', 'FontSize', 12);
71 ylabel(ax2(2), 'Intensity - Green', 'FontSize', 12);
72 legend('charge', 'Intensity - Green');
73 title('Charge and Intensity - Green over Time', 'FontSize', 12);
74 set(ax2, {'ycolor'}, {'k'; 'k'})
75 grid on;
76
77 % Charge and intensity over time for blue
78 subplot(3,1,3);
79 [ax3, p5, p6] = plotyy(imageTime, chargeData, imageTime, color_blue_cor, 'plot');
80 %datetick(ax3(1), 'keeplimits'); datetick(ax3(2), 'keeplimits');
81 set(p5, 'color', 'K'); set(p6, 'color', 'B');
82 xlabel('Time [hrs]', 'FontSize', 12);
83 ylabel(ax3(1), 'Charge [uAh]', 'FontSize', 12);
84 ylabel(ax3(2), 'Intensity - Blue', 'FontSize', 12);
85 legend('charge', 'Intensity - Blue');
86 title('Charge and Intensity - Blue over Time', 'FontSize', 12);
87 set(ax3, {'ycolor'}, {'k'; 'k'})
88 linkaxes([ax1, ax2, ax3], 'x')
89 grid on;

```

```

90
91 %% Voltage, Current, Charge, Intensity over time
92 vcci = figure(2);
93 set(vcci, 'Units', 'centimeters');
94 pos = get(vcci, 'Position');
95 set(vcci, 'PaperPositionMode', 'auto');
96 grid on
97 set(vcci, 'PaperOrientation', 'landscape');
98
99 ax1 = subplot(4,1,1);
100 plot(imageTime, voltData, 'k');
101 xlabel('Time [hrs]');
102 ylabel('Voltage [V]');
103 title('Voltage over Time');
104 axis tight;
105 grid on;
106
107 ax2 = subplot(4,1,2);
108 plot(imageTime, currentData, 'k');
109 xlabel('Time [hrs]');
110 ylabel('Current [uA]');
111 title('Current over Time');
112 axis tight;
113 grid on;
114
115 ax3 = subplot(4,1,3);
116 plot(imageTime, chargeData, 'k');
117 xlabel('Time [hrs]');
118 ylabel('Charge [uAh]');
119 title('Charge over Time');
120 axis tight;
121 grid on;
122
123 ax4 = subplot(4,1,4);
124 plot(imageTime, color_red_cor, 'r', imageTime, color_green_cor, 'g', imageTime, color_blue_cor, 'b',
125       'LineWidth', 2);
126 xlabel('Time [hrs]');
127 ylabel('intensity');
128 title('Absolute values of Intensity, Compensated');
129 axis tight;
130 grid on;
131
132 linkaxes([ax1, ax2, ax3, ax4], 'x')
133
134 %% Intensity Raw Data
135 ird = figure(3);
136 set(ird, 'Units', 'centimeters');
137 pos = get(ird, 'Position');
138 set(ird, 'PaperPositionMode', 'auto');
139 set(ird, 'PaperOrientation', 'landscape');
140
141 ax1 = subplot(3,1,1);
142 plot(imageTime, color_red_unc, 'r', imageTime, color_green_unc, 'g', imageTime, color_blue_unc, 'b',
143       'LineWidth', 2);
144 xlabel('Time [hrs]');
145 ylabel('intensity');
146 title('Optical intensity, uncorrected');
147 axis tight;
148 grid on;
149
150 ax2 = subplot(3,1,2);
151 plot(imageTime, color_red_cor, 'r', imageTime, color_green_cor, 'g', imageTime, color_blue_cor, 'b',
152       'LineWidth', 2);
153 xlabel('Time [hrs]');
154 ylabel('intensity');
155 title('Optical intensity, corrected');
156 axis tight;
157 grid on;
158
159 ax3 = subplot(3,1,3);
160 plot(imageTime, color_red_cor-color_red_unc, 'r', imageTime, color_green_cor-color_green_unc, 'g',
161       imageTime, color_blue_cor-color_blue_unc, 'b', 'LineWidth', 2);
162 xlabel('Time [hrs]');
163 ylabel('intensity');
164 title('Difference between corrected and uncorrected optical intensity');
165 axis tight;
166 grid on;

```

```
164
165 linkaxes([ax1,ax2,ax3], 'x')
166
167 %% Current und voltage over charge
168
169 cvc = figure(4);
170 set(cvc, 'Units', 'centimeters');
171 pos = get(cvc, 'Position');
172 set(cvc, 'PaperPositionMode', 'auto');
173 grid on
174 set(gca, 'linewidth', 1.5);
175 set(gca, 'GridLineStyle', '-');
176 set(cvc, 'Position', [0 0 29.7 21]);
177 set(cvc, 'PaperOrientation', 'landscape');
178
179 subplot(1,2,1)
180 plot(chargeData, currentData, 'k');
181 xlabel('Charge [uAh]');
182 ylabel('Current [uA]');
183 title('Current over Charge');
184 axis tight;
185 grid on;
186
187 subplot(1,2,2)
188 plot(chargeData, voltData, 'k');
189 xlabel('Charge [uAh]');
190 ylabel('Voltage [V]');
191 title('Voltage over Charge');
192 axis tight;
193 grid on;
194
195 %% Intensity (compensated) over Voltage black & seperate
196
197 iov = figure(5);
198 set(iov, 'Units', 'centimeters');
199 pos = get(iov, 'Position');
200 set(iov, 'PaperPositionMode', 'auto');
201 grid on
202 set(gca, 'linewidth', 1.5);
203 set(gca, 'GridLineStyle', '-');
204 set(iov, 'Position', [0 0 29.7 21]);
205 set(iov, 'PaperOrientation', 'landscape');
206
207 subplot(1,3,1);
208 plot(voltData, color_red_cor, 'k');
209 xlabel('Voltage [V]');
210 ylabel('Optical intensity (compensated) - Red');
211 title('Intensity(C)-Red over Voltage');
212 axis tight;
213 grid on;
214
215 subplot(1,3,2);
216 plot(voltData, color_green_cor, 'k');
217 xlabel('Voltage [V]');
218 ylabel('Optical intensity (compensated) - Green');
219 title('Intensity(C)-Green over Voltage');
220 axis tight;
221 grid on;
222
223 subplot(1,3,3);
224 plot(voltData, color_blue_cor, 'k');
225 xlabel('Voltage [V]');
226 ylabel('Optical intensity (compensated) - Blue');
227 title('Intensity(C)-Blue over Voltage');
228 axis tight;
229 grid on;
230
231 %% Intensity (compensated) over Charge
232
233 iocc = figure(6);
234 set(iocc, 'Units', 'centimeters');
235 pos = get(iocc, 'Position');
236 set(iocc, 'PaperPositionMode', 'auto');
237 grid on
238 set(gca, 'linewidth', 1.5);
239 set(gca, 'GridLineStyle', '-');
240 set(iocc, 'Position', [0 0 29.7 21]);
241 set(iocc, 'PaperOrientation', 'landscape');
242
```

```

243
244 subplot(1,3,1);
245 plot(chargeData,color_red_cor,'k');
246 xlabel('Charge [uAh]');
247 ylabel('Optical intensity (compensated) - Red');
248 title('Intensity (C)-Red over Charge');
249 axis tight;
250 grid on;
251
252 subplot(1,3,2);
253 plot(chargeData,color_green_cor,'k');
254 xlabel('Charge [uAh]');
255 ylabel('Optical intensity (compensated) - Green');
256 title('Intensity (C)-Green over Charge');
257 axis tight;
258 grid on;
259
260 subplot(1,3,3);
261 plot(chargeData,color_blue_cor,'k');
262 xlabel('Charge [uAh]');
263 ylabel('Optical intensity (compensated) - Blue');
264 title('Intensity (C)-Blue over Charge');
265 axis tight;
266 grid on;
267
268 %% Intensity (not compensated) over Charge
269
270 iocu = figure(7);
271 set(iocu,'Units','centimeters');
272 pos = get(iocu,'Position');
273 set(iocu,'PaperPositionMode','auto');
274 grid on
275 set(gca,'linewidth',1.5);
276 set(gca,'GridLineStyle','-');
277 set(iocu,'Position',[0 0 29.7 21]);
278 set(iocu,'PaperOrientation','landscape');
279
280 subplot(1,3,1);
281 plot(chargeData,color_red_unc,'k');
282 xlabel('Charge [uAh]');
283 ylabel('Optical intensity (uncompensated) - Red');
284 title('Intensity -Red(UC) over Charge');
285 axis tight;
286 grid on;
287
288 subplot(1,3,2);
289 plot(chargeData,color_green_unc,'k');
290 xlabel('Charge [uAh]');
291 ylabel('Optical intensity - Green(uncompensated)');
292 title('Intensity -Green(UC) over Charge');
293 axis tight;
294 grid on;
295
296 subplot(1,3,3);
297 plot(chargeData,color_blue_unc,'k');
298 xlabel('Charge [uAh]');
299 ylabel('Optical intensity (uncompensated) - Blue (uncorrected)');
300 title('Intensity -Blue(UC) over Charge');
301 axis tight;
302 grid on;
303
304 %% Dreidimension imageTime,intensity ,charge
305
306 ictd = figure(8);
307 set(ictd,'Units','centimeters');
308 pos = get(ictd,'Position');
309 set(ictd,'PaperPositionMode','auto');
310 grid on
311 set(gca,'linewidth',1.5);
312 set(gca,'GridLineStyle','-');
313 set(ictd,'Position',[0 0 29.7 21]);
314 set(ictd,'PaperOrientation','landscape');
315
316 x= imageTime + 1;
317 y1= color_red_cor;
318 y2= color_green_cor;
319 y3= color_blue_cor;
320 z= chargeData;
321

```

```

322 plot3(x,y1,z,'r',x,y2,z,'g',x,y3,z,'b');
323 xlabel('Time [hrs]');
324 ylabel('Optical intensity (corrected)');
325 zlabel('Charge [uAh]');
326 title('Intensity(Compensated), Charge, Time in 3D');
327 axis tight;
328 grid on;
329
330 %% Dreidimension imageTime,intensity ,voltage
331
332 ivtd = figure(9);
333 set(ivtd,'Units','centimeters');
334 pos = get(ivtd,'Position');
335 set(ivtd,'PaperPositionMode','auto');
336 grid on
337 set(gca,'linewidth',1.5);
338 set(gca,'GridLineStyle','-');
339 set(ivtd,'Position',[0 0 29.7 21]);
340 set(ivtd,'PaperOrientation','landscape');
341
342 x= imageTime + 1;
343 y1= color_red_cor;
344 y2= color_green_cor;
345 y3= color_blue_cor;
346 z= voltData;
347
348 plot3(x,y1,z,'r',x,y2,z,'g',x,y3,z,'b');
349 xlabel('Time [hrs]');
350 ylabel('Optical intensity (corrected)');
351 zlabel('Voltage [V]');
352 title('Intensity(Compensated), Voltage, Time in 3D');
353 axis tight;
354 grid on;
355
356 %% Save to file
357
358 set(cit,'InvertHardCopy','on'); % print to jpg exact colors of figure
359 filename = strcat(parser.grantplotDest,'_charge_intensity_over_time');
360 saveas(cit,filename,'pdf')
361
362 set(cit,'InvertHardCopy','on'); % print to jpg exact colors of figure
363 filename = strcat(parser.grantplotDest,'_charge_intensity_over_time');
364 saveas(cit,filename,'fig')
365
366 set(vcci,'InvertHardCopy','on'); % print to jpg exact colors of figure
367 filename = strcat(parser.grantplotDest,'_voltage_current_charge_intensity_over_time');
368 saveas(vcci,filename,'pdf')
369
370 set(vcci,'InvertHardCopy','on'); % print to jpg exact colors of figure
371 filename = strcat(parser.grantplotDest,'_voltage_current_charge_intensity_over_time');
372 saveas(vcci,filename,'fig')
373
374 set(ird,'InvertHardCopy','on'); % print to jpg exact colors of figure
375 filename = strcat(parser.grantplotDest,'_intensity_data_full');
376 saveas(ird,filename,'pdf')
377
378 set(ird,'InvertHardCopy','on'); % print to jpg exact colors of figure
379 filename = strcat(parser.grantplotDest,'_intensity_data_full');
380 saveas(ird,filename,'fig')
381
382 set(cvc,'InvertHardCopy','on'); % print to jpg exact colors of figure
383 filename = strcat(parser.grantplotDest,'_current_voltage_over_charge');
384 saveas(cvc,filename,'pdf')
385
386 set(cvc,'InvertHardCopy','on'); % print to jpg exact colors of figure
387 filename = strcat(parser.grantplotDest,'_current_voltage_over_charge');
388 saveas(cvc,filename,'fig')
389
390 set(iov,'InvertHardCopy','on'); % print to jpg exact colors of figure
391 filename = strcat(parser.grantplotDest,'_intensity_over_voltage');
392 saveas(iov,filename,'pdf')
393
394 set(iov,'InvertHardCopy','on'); % print to jpg exact colors of figure
395 filename = strcat(parser.grantplotDest,'_intensity_over_voltage');
396 saveas(iov,filename,'fig')
397
398 set(iocc,'InvertHardCopy','on'); % print to jpg exact colors of figure
399 filename = strcat(parser.grantplotDest,'_intensity_com_over_charge');
400 saveas(iocc,filename,'pdf')
401

```



```

402 set(ioccc, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
403 filename = strcat(parser.grantplotDest, '_intensity_com_over_charge');
404 saveas(ioccc, filename, 'fig')
405
406 set(iocuc, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
407 filename = strcat(parser.grantplotDest, '_intensity_uncom_over_charge');
408 saveas(iocuc, filename, 'pdf')
409
410 set(iocud, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
411 filename = strcat(parser.grantplotDest, '_intensity_uncom_over_charge');
412 saveas(iocud, filename, 'fig')
413
414 set(ictd, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
415 filename = strcat(parser.grantplotDest, '_intensity_charge_time_3D');
416 saveas(ictd, filename, 'pdf')
417
418 set(ictu, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
419 filename = strcat(parser.grantplotDest, '_intensity_charge_time_3D');
420 saveas(ictu, filename, 'fig')
421
422 set(ivtd, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
423 filename = strcat(parser.grantplotDest, '_intensity_voltage_time_3D');
424 saveas(ivtd, filename, 'pdf')
425
426 set(ivtu, 'InvertHardCopy', 'on'); % print to jpg exact colors of figure
427 filename = strcat(parser.grantplotDest, '_intensity_voltage_time_3D');
428 saveas(ivtu, filename, 'fig')

```

A.5.4. m03_videoCreate.m

```

1 %% v1.2 2015-05-21 DMMP, VR
2 % Removed Plotting function to intensitaetsverlauf_kompensiert
3
4 %% clean up
5 close all;
6 clear all;
7 clc;
8
9 %% read inputdata from configplotter.m file
10 parser = config;
11
12 %% get data as a mat file
13 A = load(strcat(parser.ImageRGB_data, '.mat'));
14 filenames = A.imname(:,1);
15
16 %% get the image time stamps to a text file
17 [status dates] = system(['sed "1{/^#/d;};s/[\\t].*/g" ' pwd '/' parser.ImageRGB_data]);
18 disp(dates);
19
20 %% save image timestamps in a temporary text file
21 %fid = fopen('imageTimeStamps.txt', 'wt');
22 %fprintf(fid, '%s\\n', dates);
23
24 %%
25 % system(['cat ' parser.voltageFile '|grep -v "#".//TimestampMatcher -f ./imageTimeStamps.txt >'
26 % parser.tempVolt]);
27 % system(['cat ' parser.currentFile '|grep -v "#".//TimestampMatcher -f ./imageTimeStamps.txt >'
28 % parser.tempCurrent]);
29 % system(['cat ' parser.chargeFile '|grep -v "#".//TimestampMatcher -f ./imageTimeStamps.txt >'
30 % parser.tempCharge]);
31
32 voltMatrix = dlmread(parser.tempVolt);
33 currentMatrix = dlmread(parser.tempCurrent);
34 chargeMatrix = dlmread(parser.tempCharge);
35
36 %delete (parser.tempVolt, parser.tempCurrent, parser.tempCharge);
37
38 %% plot
39 % try
40 unixTime = A.exportMatrix(:,1);
41 imageDate = unixTime/86400 + datenum(1970,1,1); % convert back to matlab time stamp format
42 time = (unixTime - A.exportMatrix(1,1))+1440/60; % +1440 for minutes; +86400 for seconds
43 sumIntensity_c = (A.exportMatrix(:,5) + A.exportMatrix(:,6) + A.exportMatrix(:,7));
44

```

```

43 %     fig = figure() ;
44     h = figure(1);
45     whitebg('w'); %change figure background color
46
47     %% .....figure
48     settings.....
49     %settings:
50     plotlinewidth = 1.5;
51     gridlinewidth = 1;
52
53 %     % font size
54     set(findall(h,'type','text'),'FontSize',50,'fontWeight','bold')
55
56     %Position
57     set(h,'Units','centimeters');
58     pos = get(h,'Position');
59     set(h,'PaperPositionMode','auto');
60     %set(0,'DefaultAxisFontSize',16)
61
62     %grid-line-width
63     grid on
64     set(gca,'linewidth',gridlinewidth);
65     set(gca,'GridLineStyle','-');
66
67     set(h,'Position',[0 0 29.7 21]);
68     set(h,'PaperOrientation','landscape');
69     a = imread(strcat(parser.imageSource,char(filename(1))));
70
71 %# .....End of figure
72     settings.....
73
74     %% plot volt graph
75     subplot(3,3,[1,2,4,5]);
76     imshow(a);
77
78     subplot(3,3,3);
79     y1 = voltMatrix(:,2);
80     plot(imageDate, y1, 'k','LineWidth',2);
81     datetick('x','keplimits');
82     xlabel('Time','FontSize',20);
83     ylabel('U [V]','FontSize',20);
84     title('Voltage over time','FontSize',20);
85     axis tight;
86     grid on;
87
88     % create moving point + coords text
89
90     hLine1 = line('XData',imageDate(1), 'YData',y1(1), 'Color','r', ...
91                 'Marker','o', 'MarkerSize',5, 'LineWidth',2);
92 %     hTxt1 = text(imageDate(1), y1(1), sprintf('%.2f,%.2f',imageDate(1),y1(1)), ...
93 %               'Color','b', 'FontSize',8, ...
94 %               'HorizontalAlignment','left', 'VerticalAlignment','top');
95
96     %% plot current graph
97
98     subplot(3,3,6);
99     y2 = currentMatrix(:,2)*1000000;
100    plot(imageDate, y2, 'k','LineWidth',2);
101    datetick('x','keplimits');
102    xlabel('Time','FontSize',20);
103    ylabel('I [uA]','FontSize',20);
104    title('Current over time','FontSize',20);
105    axis tight;
106    grid on;
107
108    % create moving point + coords text
109    hLine2 = line('XData',imageDate(1), 'YData',y2(1), 'Color','r', ...
110                'Marker','o', 'MarkerSize',5, 'LineWidth',2);
111 %    hTxt2 = text(imageDate(1), y2(1), sprintf('%.2f,%.2f',imageDate(1),y2(1)), ...
112 %               'Color','b', 'FontSize',8, ...
113 %               'HorizontalAlignment','left', 'VerticalAlignment','top');
114
115    %% plot charge graph
116    subplot(3,3,7);
117    y3 = chargeMatrix(:,2)/3600*1000;
118    plot(imageDate,y3,'k','LineWidth',2);
119    datetick('x','keplimits');
120    xlabel('Time','FontSize',20);

```

```

121 ylabel('Q [mAh]', 'FontSize', 20);
122 title('Charge over time', 'FontSize', 20);
123 axis tight;
124 grid on;
125
126 % create moving point + coords text
127
128 hLine3 = line('XData', imageDate(1), 'YData', y3(1), 'Color', 'r', ...
129             'Marker', 'o', 'MarkerSize', 5, 'LineWidth', 2);
130 % hTxt3 = text(imageDate(1), y3(1), sprintf('%.2f,%.2f', imageDate(1), y3(1)), ...
131 %           'Color', 'b', 'FontSize', 8, ...
132 %           'HorizontalAlignment', 'left', 'VerticalAlignment', 'top');
133
134 %% plot intensity graph
135
136 subplot(3,3,8);
137 y4 = A.exportMatrix(:,5); y5 = A.exportMatrix(:,6); y6 = A.exportMatrix(:,7);
138 plot(imageDate, y4, 'r', imageDate, y5, 'g', imageDate, y6, 'b', 'LineWidth', 2);
139 datetick('x', 'keeplimits');
140 xlabel('Time', 'FontSize', 20);
141 ylabel('intensity', 'FontSize', 20);
142 title('Absolute values', 'FontSize', 20);
143 axis tight;
144 grid on;
145
146 % create moving point + coords text
147 hLine4 = line('XData', imageDate(1), 'YData', y4(1), 'Color', 'y', ...
148             'Marker', 'o', 'MarkerSize', 5, 'LineWidth', 2);
149 hLine5 = line('XData', imageDate(1), 'YData', y5(1), 'Color', 'r', ...
150             'Marker', 'o', 'MarkerSize', 5, 'LineWidth', 2);
151 hLine6 = line('XData', imageDate(1), 'YData', y6(1), 'Color', 'r', ...
152             'Marker', 'o', 'MarkerSize', 5, 'LineWidth', 2);
153 % hTxt4 = text(imageDate(1), y4(1), sprintf('%.2f,%.2f', imageDate(1), y4(1)), ...
154 %           'Color', 'b', 'FontSize', 8, ...
155 %           'HorizontalAlignment', 'left', 'VerticalAlignment', 'top');
156 % hTxt5 = text(imageDate(1), y5(1), sprintf('%.2f,%.2f', imageDate(1), y5(1)), ...
157 %           'Color', 'b', 'FontSize', 8, ...
158 %           'HorizontalAlignment', 'left', 'VerticalAlignment', 'top');
159 % hTxt6 = text(imageDate(1), y6(1), sprintf('%.2f,%.2f', imageDate(1), y6(1)), ...
160 %           'Color', 'b', 'FontSize', 8, ...
161 %           'HorizontalAlignment', 'left', 'VerticalAlignment', 'top');
162
163
164 %% iterate through all images
165 for i = 1:size(A.exportMatrix, 1) %All images
166
167     subplot(3,3,[1,2,4,5]);
168     filepath = strcat(parser.imageSource, char(filenamees(i)));
169     a = imread(filepath);
170     imshow(a);
171     rectangle('Position',[parser.xposl parser.yposu (parser.xposr - parser.xposl) (parser.yposd -
172     parser.yposu)], 'LineWidth', 2, 'EdgeColor', 'r');
173     rectangle('Position',[parser.ref_xposl parser.ref_yposu (parser.ref_xposr - parser.ref_xposl) (
174     parser.ref_yposd - parser.ref_yposu)], 'LineWidth', 2, 'EdgeColor', 'r');
175     rectangle('Position',[50 1100 parser.onemm 50], 'LineWidth', 2, 'EdgeColor', 'w');
176     text(parser.ref_xposl+4, parser.ref_yposu+14, 'REF', 'Color', 'r')
177     text(80, 1117, '1 mm', 'Color', 'w')
178
179     subplot(3,3,9);
180     b(:, :) = a(parser.yposu : parser.yposd, parser.xposl : parser.xposr, 3);
181     hist(double(b(:)), 255);
182     hold on;
183     g(:, :) = a(parser.yposu : parser.yposd, parser.xposl : parser.xposr, 2);
184     hist(double(g(:)), 255);
185     hold on;
186     r(:, :) = a(parser.yposu : parser.yposd, parser.xposl : parser.xposr, 1);
187     hist(double(r(:)), 255);
188     hold off;
189     title('Histogram', 'FontSize', 20);
190     ylabel('Pixel occurrence', 'FontSize', 20);
191     xlabel('Pixel RGB-value', 'FontSize', 20);
192     axis([0 70000 0 4000]);
193     %axis tight;
194     grid on
195
196     h = findobj(gca, 'Type', 'patch');
197     set(h(1), 'facecolor', 'r', 'edgecolor', 'r');
198     set(h(2), 'facecolor', 'g', 'edgecolor', 'g');

```

```

198     set(h(3), 'facecolor', 'b', 'edgecolor', 'b');
199
200     %# update point & text
201     set(hLine1, 'XData', imageDate(i), 'YData', y1(i))
202     %     set(hTxt1, 'Position', [imageDate(i) y1(i)], ...
203     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y1(i)])
204
205     set(hLine2, 'XData', imageDate(i), 'YData', y2(i))
206     %     set(hTxt2, 'Position', [imageDate(i) y2(i)], ...
207     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y2(i)])
208
209     set(hLine3, 'XData', imageDate(i), 'YData', y3(i))
210     %     set(hTxt3, 'Position', [imageDate(i) y3(i)], ...
211     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y3(i)])
212
213
214     set(hLine4, 'XData', imageDate(i), 'YData', y4(i))
215     set(hLine5, 'XData', imageDate(i), 'YData', y5(i))
216     set(hLine6, 'XData', imageDate(i), 'YData', y6(i))
217     %     set(hTxt4, 'Position', [imageDate(i) y4(i)], ...
218     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y4(i)])
219     %     set(hTxt5, 'Position', [imageDate(i) y5(i)], ...
220     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y5(i)])
221     %     set(hTxt6, 'Position', [imageDate(i) y6(i)], ...
222     %     'String', sprintf('%.2f,%.2f'), [imageDate(i) y6(i)])
223
224     drawnow;                    %# force refresh
225     %pause(0.1)                 %# slow down animation
226
227     %Filename
228     numm=int2str(i);
229     set(gcf, 'InvertHardCopy', 'off'); % print to jpg exact colors of figure
230     filename = strcat(parser.imageDest, numm);
231     saveas(gcf, filename, 'jpg')
232     disp(['Part: ' num2str(i) '/' num2str(size(A.exportMatrix, 1))]);
233 end

```

A.5.5. m12_spectrometer2matlab.m

```

1 %% Modifiziert: 3653 Zeilen -> 2048
2 %
3 %%
4 clc;
5 clear all;
6 close all;
7
8 % get data file "spectrumfor3d"
9 [file, path] = uigetfile({'*.dat', 'Data file (*.dat)'}, 'Select a file', '../Output/Data/
    spectrumfor3d.dat');
10
11 % extract timestamps
12 [retval output] = system(['cat ' path file ' |grep "#" |sed -e "s/.*: //g"']);
13 timestamp = cell2mat(textscan(output, '%f\n'));
14
15 % extract spectra and wavelengths
16 [wavelength spectra] = textread(strcat(path, file), '%f\t%f\n', -1, 'commentstyle', 'shell');
17 %[wavelength spectra] = textread(strcat(path, file), '%f\t%f\n', -1, 'commentstyle', 'shell');
18
19 % reshape matrices
20 N = size(wavelength, 1)/2048;
21 spectra = reshape(spectra, 2048, N);
22 wavelength = reshape(wavelength, 2048, N);
23
24 % remove redundant wavelengths
25 wavelength = wavelength(:, 1);
26
27 clear file path retval output N;
28
29
30
31 %% Electrical Measurement Data
32
33 [file, path] = uigetfile({'TCVCDaten.dat', 'Data file (*.dat)'}, 'Select TCVC file to load', '../Output
    /Data/TCVCDaten.dat');
34

```

```

35 [timestamp2 charge voltage current] = textread(strcat(path, file), '%f\t%f\t%f\t%f\n', -1, '
    commentstyle', 'shell');
36
37
38 % clean up
39 clear file path retval output N;
40
41 %save data to *.mat-files
42 uisave({'voltage', 'current', 'charge', 'timestamp2'}, '../Output/Data/m01_electrical.mat');
43 uisave({'spectra', 'timestamp2', 'wavelength'}, '../Output/Data/m02_spectra.mat');

```

A.5.6. m13_showSpectra.m

```

1 %% 2016-10-25 VR
2 % Modifiziert: 3653 Zeilen -> 2048
3 %
4 %%
5
6 clc;
7 clear all;
8 close all;
9
10 m01_electrical = load('../Output/Data/m01_electrical.mat');
11 voltage = m01_electrical.voltage; % Voltage in Volts
12 current = m01_electrical.current*1000*1000; % Current in  $\mu$ A
13 charge = m01_electrical.charge*1000*1000/3600; % Charge in  $\mu$ Ah
14 timestamp = m01_electrical.timestamp2;
15
16 m02_spectra = load('../Output/Data/m02_spectra.mat');
17 spectra = m02_spectra.spectra;
18 wavelength = m02_spectra.wavelength;
19
20
21 spectra_backup=spectra;
22
23
24 %% Smoothing and calculations
25
26 spectra=spectra_backup;
27 filename_string='_smooth10';
28
29 %% Time from beginning of measurement in hours
30 spectrumTime = (timestamp-timestamp(1))/3600;
31
32
33 %% Smoothing in wavelength axis. 1 is no smoothing.
34 for n=1:size(spectra,2)
35     spectra(:,n)=smooth(spectra(:,n),10);
36 end
37 clear n;
38
39 %% Smoothing in temporal axis. 1 is no smoothing.
40 for n=1:size(spectra,1)
41     spectra(n,:)=smooth(spectra(n,:),10);
42 end
43 clear n;
44
45 %% Difference between spectra and meanval of the spectra
46 meanval = mean(spectra,2);
47 diffspectra = zeros(2048, size(spectra,2));
48 for n=1:size(spectra,2)
49     diffspectra(:,n)=spectra(:,n)-meanval;
50 end
51 clear n;
52
53
54 % Gradient of the spectrum
55 gradspectra = zeros(2048, size(spectra,2));
56 for n=2:size(spectra,2)
57     gradspectra(:,n)=spectra(:,n)-spectra(:,n-1);
58 end
59 gradspectra(:,1)=gradspectra(:,2);
60
61 % Geometric centre of Intensity
62 % (intensity * vector index) / sum of intensity = index of geometric centre
63 for n=1:size(spectra,2)

```

```

64     summe = 0;
65     for i=1 : 2048
66         summe = summe + (i* spectra(i,n));
67     end
68     summe2(n) = summe / sum(spectra(:,n));
69 end
70 % index to wavelength
71 for n=1:size(spectra,2)
72     spectra_center(n)=summe2(n)/2048*904 +187;
73 end
74
75 clear n;
76 clear summe1 summe2;
77
78
79 %% plot Spectrum 3D and geometric centre
80 spectrum3d = figure(1);
81 ax1=subplot(2,1,1);
82 clim = [2000 8000];
83 [ap1]=imagesc(spectrumTime, wavelength, spectra);%clim);
84 set(ax1,'YDir','normal')
85 set(ax1,'YTick',400:50:1000)
86 hold on;plot(spectrumTime, spectra_center, 'b');
87 ylabel('Wavelength [nm]');ylim([400 1000]);
88 xlabel('Time [hrs]');
89 ylabel(colorbar, 'Intensity [arb. units]');
90
91 ax2=subplot(2,1,2);
92 [ap2,ah21,ah22]=plotyy(spectrumTime, voltage, spectrumTime, spectra_center);
93 legend('Voltage','Geometric center of each spectrum')
94 set(get(ap2(1),'Ylabel'),'String','Voltage [V]');
95 set(get(ap2(2),'Ylabel'),'String','Geometric center wavelength [nm]');
96 xlabel('Time [hrs]');
97 set(ax2(1),'YTick',0:0.3:10)
98 set(ap2(2),'YTick',0:2:1000);
99 grid on;
100
101 axis(ap2,'tight')
102 pos2 = get(ax2,'Position');
103 pos1 = get(ax1,'Position');
104 pos2(3) = pos1(3);
105 set(ax2,'Position',pos2)
106
107 %% Plot Defference to mean spectra
108 spectrum3ddiff = figure(2);
109 ax1=subplot(1,1,1);
110 clim = [2000 8000];
111 [ap1]=imagesc(spectrumTime, wavelength, diffspectra);%clim);
112 set(ax1,'YDir','normal')
113 set(ax1,'YTick',400:50:1000)
114 hold on;
115 ylabel('Wavelength [nm]');ylim([400 1000]);
116 xlabel('Time [hrs]');
117 ylabel(colorbar, 'Intensity (deviation from mean spectrum) [arb. units]');
118
119 %% Plot gradient of the spectra
120 spectrum3dgrad = figure(3);
121 ax1=subplot(1,1,1);
122 clim = [2000 8000];
123 [ap1]=imagesc(spectrumTime, wavelength, gradspectra);
124 set(ax1,'YDir','normal')
125 set(ax1,'YTick',400:50:1000)
126 hold on;
127 ylabel('Wavelength [nm]');ylim([400 1000]);
128 xlabel('Time [hrs]');
129 ylabel(colorbar, 'Intensity (gradient) [arb. units]');
130
131 examplespectrum = figure(4);
132 plot(wavelength, spectra(:,3));
133 xlabel('Wavelength [nm]');
134 ylabel('Time [hrs]');
135 legend('Exemplary spectrum from beginning of the measurement')
136
137 %% Two chosen wavelength areas
138 spectrum_1=spectra(817,:);
139 for n=818:833
140     spectrum_1=spectrum_1+spectra(n,:);
141 end
142

```

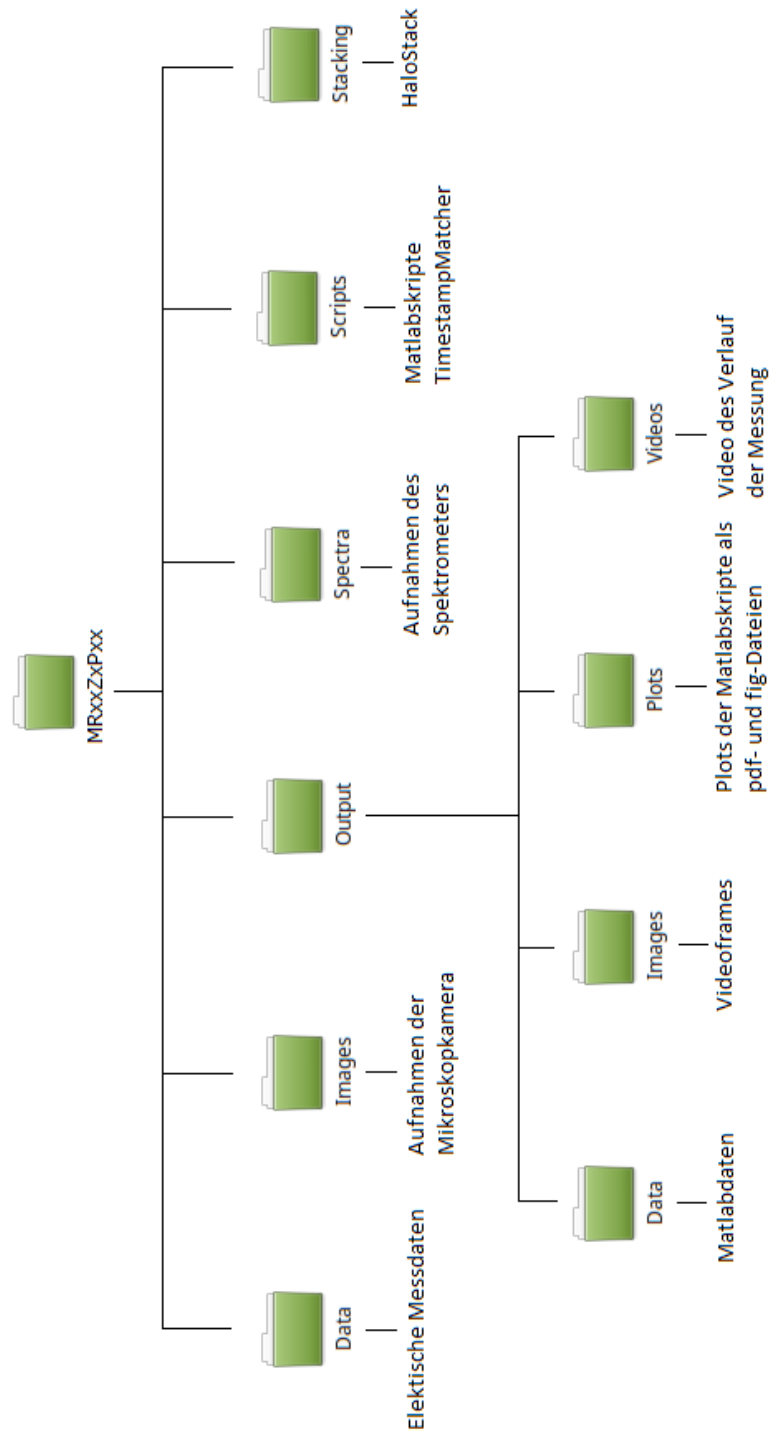
```

143 spectrum_2=spectra(1156,:);
144 for n=1157:1167
145     spectrum_2=spectrum_2+spectra(n,:);
146 end
147
148 %% Plot Intensity over Voltage
149 iov=figure(5);
150 plot(smooth(voltage,1500),smooth(spectrum_1,100), 'r*');
151 hold on;
152 plot(smooth(voltage,1500),smooth(spectrum_2,1000), 'b*');
153 hold off;
154 ylabel('Intensity [arb. units]');
155 xlabel('Voltage [V]');
156 grid on;
157
158 %% Plot two wavelength
159 twowavelengths=figure(6);
160 subplot(2,1,1);
161 ap9=plotyy(spectrumTime, spectrum_450, spectrumTime, spectrum_650);
162 set(get(ap9(1),'Ylabel'),'String','Intensity [arb. units]');
163 set(get(ap9(2),'Ylabel'),'String','Intensity [arb. units]');
164 legend('Intensity in wavelength 450 nm','Intensity in wavelength 650 nm')
165 xlabel('Time [hrs]');
166 grid on; axis(ap9,'tight')
167
168 subplot(2,1,2);
169 ap9=plotyy(spectrumTime, voltage, spectrumTime, charge);
170 set(get(ap9(1),'Ylabel'),'String','Voltage [V]');
171 set(get(ap9(2),'Ylabel'),'String','Charge [ $\mu$ Ah]');
172 xlabel('Time [hrs]');
173 grid on; axis(ap9,'tight')
174 legend('Voltage','Charge')
175
176 %% Save plots to file
177
178 set(spectrum3d, 'InvertHardCopy', 'on'); % print to fig
179 filename = strcat('.. /Output/Plots/spectrum3d',filename_string);
180 saveas(spectrum3d,filename,'fig')
181
182 set(spectrum3d, 'InvertHardCopy', 'on'); % print to pdf
183 filename = strcat('.. /Output/Plots/spectrum3d',filename_string);
184 saveas(spectrum3d,filename,'pdf')
185
186 set(spectrum3ddiff, 'InvertHardCopy', 'on'); % print to fig
187 filename = strcat('.. /Output/Plots/spectrum3ddiff',filename_string);
188 saveas(spectrum3ddiff,filename,'fig')
189
190 set(spectrum3ddiff, 'InvertHardCopy', 'on'); % print to pdf
191 filename = strcat('.. /Output/Plots/spectrum3ddiff',filename_string);
192 saveas(spectrum3ddiff,filename,'pdf')
193
194 set(spectrum3dgrad, 'InvertHardCopy', 'on'); % print to fig
195 filename = strcat('.. /Output/Plots/spectrum3dgrad',filename_string);
196 saveas(spectrum3dgrad,filename,'fig')
197
198 set(spectrum3dgrad, 'InvertHardCopy', 'on'); % print to pdf
199 filename = strcat('.. /Output/Plots/spectrum3dgrad',filename_string);
200 saveas(spectrum3dgrad,filename,'pdf')
201
202 set(examplespectrum, 'InvertHardCopy', 'on'); % print to fig
203 filename = strcat('.. /Output/Plots/examplespectrum',filename_string);
204 saveas(examplespectrum,filename,'fig')
205
206 set(examplespectrum, 'InvertHardCopy', 'on'); % print to pdf
207 filename = strcat('.. /Output/Plots/examplespectrum',filename_string);
208 saveas(examplespectrum,filename,'pdf')
209
210 set(twowavelengths, 'InvertHardCopy', 'on'); % print to fig
211 filename = strcat('.. /Output/Plots/twowavelengths',filename_string);
212 saveas(twowavelengths,filename,'fig')
213
214 set(twowavelengths, 'InvertHardCopy', 'on'); % print to pdf
215 filename = strcat('.. /Output/Plots/twowavelengths',filename_string);
216 saveas(twowavelengths,filename,'pdf')
217
218 set(iov, 'InvertHardCopy', 'on'); % print to fig
219 filename = strcat('.. /Output/Plots/iov',filename_string);
220 saveas(iov,filename,'fig')
221

```

```
222 set(iov, 'InvertHardCopy', 'on'); % print to pdf
223 filename = strcat('../Output/Plots/iov',filename_string);
224 saveas(iov, filename, 'pdf')
225
226
227 set(twowavelengths_2, 'InvertHardCopy', 'on'); % print to fig
228 filename = strcat('../Output/Plots/twowavelengths_2',filename_string);
229 saveas(twowavelengths_2, filename, 'fig')
230
231 set(twowavelengths_2, 'InvertHardCopy', 'on'); % print to pdf
232 filename = strcat('../Output/Plots/twowavelengths_2',filename_string);
233 saveas(twowavelengths_2, filename, 'pdf')
```


B. Ordnerstruktur



C. Aufgabenstellung



Hochschule für Angewandte Wissenschaften Hamburg
Department Informations- und Elektrotechnik
Prof. Dr.-Ing. Karl-Ragmar Riemschneider

14. Mai 2014

Bachelorthesis Timo Gerken

Spektrale Untersuchung für die optische Zustandsbeobachtung von Lithiumbatterien

Motivation

Im Rahmen des Projektes 'Sino-German Electromobility Research - SINGER', das vom Bundesverkehrsministerium gefördert wird, soll der Zustand von Batterien durch Sensorik erfasst werden. Für Lithiumbatterien ist die Verbesserung der Erfassung des Batteriezustandes wünschenswert. Die elektrisch erfassbaren Größen - wie Klemmspannung, Batteriestrom und Temperatur - sollen im Rahmen der Forschungsarbeiten um eine physikalische/chemische Beobachtung des Elektrodenmaterials ergänzt werden, die geeignet ist, in den laufenden Batteriebetrieb integriert zu werden.

Aufgabe

Herr Timo Gerken erhält die Aufgabe, mit einem automatisierten Messaufbau eine optische Beobachtung von Batteriezellen durchzuführen. Dabei soll der Batteriebetrieb in-situ beobachtet werden. Es sind Elektrodenproben in verfügbare Messzellen mit optischem Fenster einzubringen und als Testbatterien gesteuert zu betreiben. Dafür sind die Versuche zu planen, ein Messaufbau einzurichten und die Steuersoftware zu konfigurieren bzw. anzupassen. Für die Auswertung der erfassten Daten sind passende Methoden zu entwickeln, die insbesondere den Zusammenhang zwischen elektrischen Messgrößen (Spannung, Strom, Ladung) und optischen Messgrößen (Farbwerte eines Kamerabildes oder Spektrometerdaten) im zeitlichen Verlauf darstellen. Die wissenschaftliche Fragestellung besteht in der Untersuchung der Ausprägung der optischen Effekte bei verschiedenen Lichtwellenlängen. Bei stabilen und signifikanten Unterschieden bei verschiedenen Lichtwellenlängen könnten Ansätze für eine Referenzgröße gefunden werden, welche die praktische Verwendung der optischen Beobachtung wesentlich erleichtern könnten. Durch umfangreiche Messreihen sind die Ergebnisse zu überprüfen und zu beurteilen.

Für die Abschlussarbeit sind die folgenden Arbeitspakete geplant:

1. Einarbeitung und Vorarbeiten für optische Messverfahren in LiFePO₄-Zellen
 - Einarbeitung in Grundlagen und Effekte bei Lithium-Batterien
 - Inbetriebnahme der Mikroskopkamera mit modifizierter LED-Beleuchtung und des Spektrometers (inkl. Softwaresteuerung)
 - Entwurf und Modifikation von Matlab-Skripten für die Auswertung und Darstellung
 - Vorabmessungen, Platinenaufbau, Softwarekonfiguration und Inbetriebnahme des Zyklischerfassungssystems (Raspberry)

2. Herstellen des Messaufbaus

- Montage von Testzellen (Fa. El-Cell) mit Elektrodenproben
- Herstellen von Laboraufbauten jeweils zur Überwachung mittels Kamera und Spektrometer
- Betrieb der Messzellen mit dem Zyklrier- und Erfassungssystem (Raspberry)
- Vorbereitung des automatisierten Messbetriebs und der Datenerfassung (Probeläufe)
- Vorbereitung der Datenauswertung, z.B. zeitliche Zuordnung von Messdaten, grafische Darstellung
- Prüfung der Reproduzierbarkeit der Messungen, Kontrollmessungen mit Präzisionsmultimeter, ggfs. Korrekturwerte aufnehmen

3. Versuchsplanung und -durchführung

- Planung und Durchführung von umfangreichen Messreihen
- Aufnahme von elektrischen Werten wie Strom- und Spannungsverlauf sowie Berechnung des Ladungsumsatzes
- Korrektur von systematischen Messfehlern (z.B. stromrichtige Schaltung, Integrationsfehler) und ggf. weiteren Störeinflüssen (z.B. Umgebungstemperatur)
- Schrittweise Verbesserung und Anpassung des Messbetriebs nach Zwischenauswertungen

4. Messauswertung und Ergebnisanalyse

- Spektrumsauswertung, insb. Darstellung von Spektrumsveränderungen im Zyklrierbetrieb
- Darstellung der Änderung der spektralen Anteile bei verschiedenfarbigen Beleuchtungen im Zyklrierbetrieb
- Zuordnung der LED-Beleuchtungsfarben und der Kamerafarbkanäle zu spektralen Bereichen
- Auswertung und Zusammenführung der Messdaten und kombinierte graphische Darstellung (z.B. Beleuchtung und Spektrometer oder Farbkanal und Beleuchtung) durch eigene Softwarelösungen
- Option: Exemplarische Raman-Untersuchung an Testzellen im Zyklrierbetrieb

5. Einordnung, Bewertung und Ausblick

- Zusammenfassung der Ergebnisse, Beurteilung des Messergebnisses
- Beurteilung der Eignung für Nutzung als Referenz bzw. Korrekturwerte
- Bewertung der gewählten Konzepte und Lösungsvarianten
- Offene Punkte sowie einschränkende Erfahrungen und Beobachtungen

Dokumentation

Die Fachliteratur und die kommerziellen Unterlagen bzw. Datenblätter sind zielgerichtet zu recherchieren. Dabei sind insbesondere wichtige Grundlagen der Batterieeffekte näher zu betrachten. Die entwickelte Software, die gewählten Lösungen und die Funktionsweise sind gut nachvollziehbar und für die zukünftige Nutzung zu dokumentieren. Die Messergebnisse sind in aussagefähigem Umfang zu erfassen und auszuwerten. Die realisierten Lösungen und die Ergebnisse sind kritisch einordnend zu bewerten. Ansätze für Verbesserungen und weitere Arbeiten sind zu nennen.

Versicherung über die Selbstständigkeit

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

Hamburg, 24. November 2016

Ort, Datum

Unterschrift