

Bernhard Schober, BSc

Software zur Erstellung von Live NoDi*-Diagrammen bei
Teilentladungsmessungen mit Gleichspannung

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Elektrotechnik

eingereicht an der

Technischen Universität Graz

Betreuer:

Univ.-Prof. Dr.-Ing. Uwe Schichler

Dipl.-Ing. Alexander Pirker, BSc

Institut für

Hochspannungstechnik und Systemmanagement

Graz, Februar 2018

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Kurzfassung

Diese Arbeit beschäftigt sich mit Teilentladungs(TE-)messungen bei Gleichspannung. Genauer gesagt, die Verwendung neuer Darstellungsarten von Teilentladungsimpulssequenzen, deren Kenngrößen und daraus abgeleiteten Größen. Diese Darstellungen, darunter das NoDi*-Diagramm, sollen für Gleichspannung qualitative Aussagen darüber ergeben, welcher Defekt zugrundeliegt, ähnlich dem PRPD-Pattern bei Wechselspannung. Ziel der Arbeit war es, ein Programm zu entwickeln, welches, unter Verwendung des Teilentladungsmesssystems der Firma Omicron, die Messdaten über die zur Verfügung stehende COM-Schnittstelle bereits während einer Messung ausliest, weiterverarbeitet und in Diagrammen in Echtzeit darstellt. Die Arbeit gliedert sich in drei Abschnitte.

Im ersten Teil werden theoretische Grundlagen über TE-Messungen, wie normative Vorschriften, Erfassung der scheinbaren Ladung, sowie die Besonderheiten von TE-Messungen bei Gleichspannung erläutert. Ein weiteres Kapitel handelt von freien metallischen Partikeln in GIS- und GIL-Anlagen, deren Bewegungen unter der Wirkung eines elektrischen Felds und die im Betrieb auftretenden Erscheinungen, wie der Firefly-Effekt.

Im zweiten Teil werden bereits vorhandene Softwarelösungen zur graphischen Darstellung von TE-Impulsen beschrieben und ein in dieser Arbeit entwickeltes Programm *Online PDDC-Analyser* vorgestellt. Dazu werden verschiedene Dokumentationen bereitgestellt, welche die Funktionsweise und Bedienung der Software enthalten und Weiterentwicklungen ermöglichen sollen.

Der letzte Teil handelt von TE-Messungen von Partikel, darunter zwei ideale Partikel, wie eine Kugel, zwei Drahtstücke unterschiedlicher Länge und ein Span. Anhand von Hochgeschwindigkeitsaufnahmen und TE-Messungen wurden verschiedene Effekte, wie Firefly-Effekt und die Pseudoresonanz eines springenden Partikels beobachtet.

Schlüsselwörter:

- Teilentladung
- DC-Spannung
- Live NoDi*-Diagramm
- Metallische Partikel in GIS/GIL

Abstract

This thesis deals with partial discharge (PD) measurements at DC voltage. To be more precise, the usage of new methods of displaying PD impuls sequences, their parameters and derived values. These diagrams, such as NoDi* pattern, should give a qualitative knowledge about the underlying defect, similar to PRPD patterns at AC voltage. The goal of this thesis was to develop a program, which, by using the Omicron PD measurement system, reads the measured values through the available COM interface already during tests, processes and displays them live in diagrams. The thesis is separated into three parts.

The first part explains fundamental methods of PD measurements, like standardized regulations, measuring of the apparent charge, as well as the special attributes and differences of PD measurements at DC voltage. Another section deals with free metallic particles in GIS and GIL, their motions under the influence of an electric field and phenomena during operation, such as firefly effect.

The second part describes already existing software solutions for graphic representation of PD impulses and introduces a program developed in this thesis, *Online PDDC-Analyzer*. For this purpose, different documentations are provided, which contains the functionality and handling of the software and allow further developments.

The last part deals with PD measurements of particles, including two ideal particles, such as balls, two wires of different length and a spiral. Based on high speed recordings and PD measurements, different phenomena, like firefly effect and pseudo resonance of a bouncing particle were observed.

Keywords:

- Partial discharge
- DC voltage
- Live NoDi* pattern
- Metallic particles in GIS/GIL

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen der Teilentladungsmessung	2
2.1	Anforderungen an Prüfkreise und Messsysteme	2
2.2	Messsysteme für scheinbare Ladung	3
2.2.1	Allgemeines	3
2.2.2	Scheinbare Ladung	3
2.2.3	Integration im Frequenzbereich – Quasi-Integration	7
2.2.4	Integration im Zeitbereich – „echte“ Integration	10
2.2.5	Anforderungen an digitale TE-Messgeräte	12
2.2.6	IEC-Gewichtung	12
2.3	Darstellungsarten von Teilentladungen	13
2.3.1	Kenngößen	13
2.3.2	Wechselspannung	13
2.3.3	Gleichspannung	14
2.4	Defekte in GIS/GIL-Systemen	17
2.4.1	Allgemeines	17
2.4.2	Freie metallische Partikel in GIS und GIL	18
3	Software zur Verarbeitung von TE-Impulsen bei Gleichspannung	22
3.1	Einleitung	22
3.2	OMICRON Software for MDP and MI	23
3.2.1	TE-Messsystem der Firma Omicron	23
3.2.2	Software-Einstellungen	23
3.2.3	Darstellung der Teilentladungen	25
3.3	PDDC-Analyzer	28
3.4	Online PDDC-Analyzer	28
3.4.1	COM-Schnittstelle der OMICRON-Software	28
3.4.2	Graphische Benutzeroberfläche	32
3.4.3	Dokumentationen und Flussdiagramm	36
3.4.3.1	Programmiererdokumentation	36
3.4.3.2	Methodendokumentation	43
3.4.3.3	Benutzerdokumentation	47
3.4.3.4	Installationsdokumentation	51
3.4.4	Aufbau und Funktionsweise des Codes	51
3.4.5	Testdokumentation und Anwendungsbeispiele	54
3.4.6	Benutzerhandbuch	57
4	TE-Messungen an freien metallischen Partikeln in GIS und GIL	60
4.1	Messaufbau	60
4.2	Beschreibung der Defekte	62
4.3	Messprozedur	63
4.4	Ergebnisse bei Wechselspannung	65
4.5	Ergebnisse bei Gleichspannung	66
4.5.1	Allgemeines	66
4.5.2	Kugel	66

4.5.3	Draht	68
4.5.4	Span	71
5	Zusammenfassung	73
	Literaturverzeichnis	75
	Anhang	78
A	Bedienungsanleitung des Online PDDC-Analyzers	79
A.1	Voraussetzungen	79
A.2	Zugriff auf den Quellcode (falls erforderlich)	79
A.3	Durchführung von Messungen	79
A.4	Beschreibungen der GUI	81
B	Verzeichnisse	85

1 Einleitung

Der ständigen und unterbrechungsfreien Versorgung mit Elektrizität kommt in der heutigen Zeit eine besondere Bedeutung zu. Kaum ein industrieller Prozess kommt ohne elektrische Energie aus, aber auch im privaten Bereich stellt Elektrizität ein essenzielles Grundgut dar. Diverse Besonderheiten stellen die Energiewirtschaft vor einige Herausforderungen. Aufgrund der Leitungsgebundenheit benötigt es entsprechende Infrastruktur zur Übertragung der Energie, aber auch die Tatsache, dass Erzeugung und Verbrauch im ständigen Gleichgewicht sein müssen, bringt Schwierigkeiten mit sich. In den letzten Jahrzehnten ist auch eine Trendänderung von herkömmlichen hin zu erneuerbaren Energieträgern, wie beispielsweise Photovoltaik und Windkraft, zu erkennen. Diese Anlagen befinden sich aufgrund des großen Platzbedarfs oft dezentral und Offshore, wodurch die erzeugte Energie über weite Strecken hin zu Verbraucherschwerpunkten transportiert werden muss. Mit Wechselspannung ist dies nur in bestimmten Grenzen möglich, weshalb zunehmend auf Hochspannungsgleichstromübertragung (HGÜ) zurückgegriffen wird. Diese Anlagen bestehen aus für hohe Gleichspannungen ausgelegten Betriebsmitteln, welche Prüfungen unterzogen werden müssen, um einen sicheren und effizienten Betrieb zu gewährleisten. Eine der wichtigsten zerstörungsfreien Diagnosemethoden stellt die Teilentladung(TE-)messung dar, welche Alterungserscheinungen und Isolationsfehler aufzeigen kann.

Die Teilentladungsmessung hat sich seit den 70er Jahren international etabliert. Für Wechselspannungsmessungen gibt es spezielle Normen und es haben sich Auswerteverfahren, wie das phasenaufgelöste TE-Diagramm (PRPD-Pattern), durchgesetzt. Für Messungen mit hoher Gleichspannung ist die Norm IEC 60270 nach heutigem Stand nicht hilfreich, da Messverfahren für DC-Spannung nur kurz erwähnt werden [1]. Es müssen somit neue Auswertemethoden und Messprozeduren für derartige Messungen entwickelt werden. Ein solches Verfahren stellt das NoDi*-Diagramm dar. In einer vorangegangenen Bachelorarbeit wurde bereits ein Matlab-Programm mit graphischer Oberfläche entwickelt, welches die TE-Messdaten in verschiedenen Diagrammen, darunter das zuvor genannte NoDi*-Diagramm, darstellt [2]. Dies soll die Interpretation und Analyse der Messdaten, sowie daraus gezogene Schlüsse erleichtern. Es handelt sich bei dieser Art der Auswertung um ein „Postprocessing“, das bedeutet, dass erst nach der Messung ausgewertet wird. Das PRPD-Pattern hingegen kann in den meisten Messprogrammen bei AC-Prüfspannung bereits während einer Messung betrachtet werden.

Ziel dieser Arbeit ist es, die TE-Messdaten in Echtzeit während einer Messung mit hohen Gleichspannungen in verschiedenen Diagrammen darzustellen. Hierzu wurde ein eigenes Programm (*Online PDDC-Analyzer*) mit Matlab realisiert, welches aus zwei graphischen Benutzeroberflächen besteht. Die Messungen werden mit einem Teilentladungsmesssystem der Firma Omicron durchgeführt und die Messdaten anhand der COM-Schnittstelle ausgelesen. Anschließend werden die Messdaten in zeitbasierten Diagrammen und den zuvor genannten NoDi*-Diagrammen dargestellt. Das Programm enthält Fehlermeldungen, wodurch einer fehlerhaften Bedienung vorgebeugt werden soll, sowie Einstellungsmöglichkeiten zur Steuerung der Diagramme und Auswertung. Dieses neue Verfahren soll eine Analyse der Messdaten bereits während der Prüfung mit Gleichspannung zulassen.

2 Grundlagen der Teilentladungsmessung

2.1 Anforderungen an Prüfkreise und Messsysteme

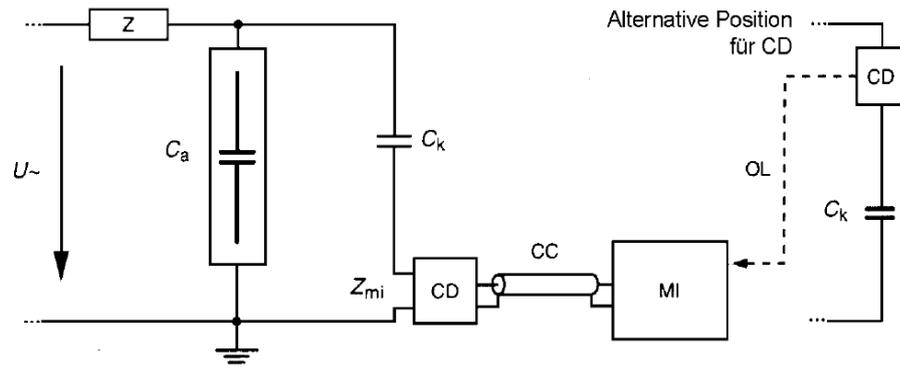
In diesem einleitenden Kapitel soll auf die normativen Anforderungen von Prüfkreisen und Messsystemen zur TE-Messung näher eingegangen werden. Diese Anforderungen sind in der Norm IEC 60270 in der Fassung 2015 niedergeschrieben und bei TE-Messungen zu beachten und einzuhalten [1].

Grundsätzlich besteht jeder Prüfkreis aus folgenden Komponenten:

- Prüfobjekt C_a
- Koppelkondensator C_k
- TE-Messsystem zur Detektion der Entladungen
- Hochspannungsquelle
- Verbindungselemente für die Hochspannung
- Gegebenenfalls eine Impedanz bzw. ein Filter.

Der prinzipielle Prüfkreis für TE-Messungen ist in Bild 1 mit den oben genannten Elementen dargestellt. Das Prüfobjekt stellt jenen Prüfling dar, dessen TE-Verhalten gemessen werden soll und kann in den meisten Fällen als Kapazität aufgefasst werden. Der Koppelkondensator C_k sollte niederinduktiv aufgebaut sein und bei der zu messenden Spannung möglichst keine oder nur geringe TE unterhalb eines festgelegten Pegels haben. In der Praxis bedeutet dies meist die Verwendung eines über große Spannungsbreiten hinweg möglichst TE-freien Koppelkondensators. TE-frei wiederum bedeutet, dass keine Werte oberhalb eines festgelegten Pegels gemessen werden. Anstatt eines Koppelkondensators kann auch ein zweiter Prüfling C_{a1} zum Einsatz kommen, welcher „ähnlich“ dem Prüfling C_a sein muss [1]. Die Hochspannungsquelle sowie die Verbindungselemente sollten einen nur geringen Grundstörpegel aufweisen, um einen TE-freien Prüfkreis zu gewährleisten. Der Prüfkreis muss laut IEC 60270 kalibriert werden und es wird empfohlen, die scheinbare Ladung als Messgröße heranzuziehen. Die Messung erfolgt über eine Koppeleinheit, welche einerseits im Erdfad oder andererseits hochspannungsseitig angeordnet werden kann. Die Platzierung der Koppeleinheit kann entweder in Serie zu C_k oder im Kreis des Prüflings erfolgen. Die erste Variante im Erdfad des Koppelkondensators hat zum Vorteil, dass Prüflinge gemessen werden können, welche einen Erdanschluss besitzen. Bei Auskopplung der TE-Impulse im Pfad des Prüflings sind besondere Schutzbeschaltungen vorzusehen, welche das empfindliche Messsystem beim Versagen des Prüflings schützen. Alternativ werden in IEC 60270 auch weitere Schaltungen, wie z. B. eine Brückenschaltung, die eine Kombination der beiden obigen Varianten darstellt, gezeigt, mit der eine gute Störunterdrückung erzielt werden kann [1].

Dem zu wählenden Teilverhältnis C_a/C_k kommt besondere Bedeutung zu, da die Messempfindlichkeit maßgeblich davon beeinflusst wird. Die beste Empfindlichkeit wird bei $C_k \gg C_a$ erreicht. In der Praxis ist diese Forderung, aufgrund der hohen



$U\sim\dots$	Hochspannungsversorgung	$C_a\dots$	Prüfling
$Z_{mi}\dots$	Eingangsimpedanz des TE-Messsystems	$C_k\dots$	Koppelkondensator
$CC\dots$	Verbindungsleitung des TE-Messsystems	$CD\dots$	Koppeleinheit
$Z\dots$	Filter	$MI\dots$	Messgerät

Bild 1: TE-Prüfkreis mit Koppeleinheit CD in Serie zum Koppelkondensator C_k [1]

Last, nur schwer zu erfüllen, wobei sich jedoch gezeigt hat, dass Kapazitätswerte um 1 nF eine ausreichende Genauigkeit erreichen.

Die Koppeleinheit CD, auch Ankoppelvierpol (AKV) genannt, stellt einen wichtigen Teil des Messsystems dar und ist im Allgemeinen auf die größtmögliche Empfindlichkeit ausgelegt. Es handelt sich um einen passiven oder aktiven Vierpol, welcher den impulsförmigen Eingangsstrom in eine Ausgangsspannung transferiert. Die Frequenzantwort wird so gewählt, dass die Eingangssignale um die Betriebsfrequenz, üblicherweise 50 Hz, und deren Harmonischen bereinigt wird [1].

2.2 Messsysteme für scheinbare Ladung

2.2.1 Allgemeines

In diesem Kapitel werden Messsysteme für scheinbare Ladung behandelt. Diese Messgeräte können je nach ihrer Funktionsweise in zwei Gruppen eingeteilt werden. Eine Gruppe nimmt eine Integration im Frequenzbereich vor und es kann zwischen schmal- und breitbandig unterschieden werden und eine zweite Gruppe von Messgeräten, welche eine Integration im Zeitbereich, eine „echte“ Integration, durchführen. Die Integration im Frequenzbereich wird auch als Quasi-Integration bezeichnet.

2.2.2 Scheinbare Ladung

Aufgrund des sehr komplexen Aufbaus von Isolierungen besteht keine Möglichkeit die an der Fehlerstelle **tatsächlich umgesetzte Ladung** zu messen, jedoch kann die nachgelieferte Ladung beim Auftreten von Teilentladungen an den Klemmen des Prüflings gemäß Gleichung 2.1 gemessen werden.

$$Q = C \cdot \Delta U \quad (2.1)$$

Dieser messbare Ladungswert wird **scheinbare Ladung** genannt, welche jedoch beachtlich kleiner ist als die in der Fehlerstelle tatsächlich umgesetzte Ladung. Oft wird das Wort scheinbar weggelassen, wenn von Teilentladungen gesprochen wird. Auch im weiteren Verlauf dieser Arbeit wird der Übersichtlichkeit wegen von Ladung gesprochen, auch wenn es sich genau genommen um die scheinbare Ladung handelt. Sie ist definiert als

„Scheinbare Ladung q eines Teilentladungsimpulses ist die Ladung, die, wenn sie innerhalb einer sehr kurzen Zeit zwischen den Anschlüssen des Prüfobjekts in einen festgelegten Prüfkreis eingespeist wird, dieselbe Anzeige auf dem Messgerät geben würde wie der TE-Impuls selbst. Die scheinbare Ladung wird gewöhnlich in Picocoulomb (pC) ausgedrückt.“
[1]

Obwohl es sich nicht um den wirklichen Ladungsumsatz handelt, hat sich die scheinbare Ladung international durchgesetzt und ist in nationalen sowie internationalen Prüfvorschriften definiert. Treten Teilentladungen auf, kommt es aufgrund der Ausgleichsvorgänge am Koppelkondensator C_k zu einem Ausgleichsstrom, welcher über den Ankoppelvierpol (AKV bzw. CD) eines TE-Messgeräts als Spannungsimpuls gemessen werden kann [3]:

$$q = \int_0^{\infty} i(t) dt = \frac{1}{Z_{CD}} \int_0^{\infty} u(t) dt . \quad (2.2)$$

TE-Impulse besitzen zwei wichtige Besonderheiten, welche sich die TE-Messtechnik zunutze macht. Einerseits ist die Impulsladung q invariant, d. h. dass selbst bei Verformung des Impulses durch einen Filter wie einen Tiefpass die Ladung gleich bleibt, obwohl sich die Kurvenform dabei ändert. Eine weitere wichtige Eigenschaft stellt die Amplitudendichte im komplexen Frequenzbereich dar. Allgemein kann ein Stromimpuls $i(t)$ im Frequenzspektrum als

$$F(j\omega) = F\{i(t)\} = \int_{-\infty}^{\infty} i(t) e^{-j\omega t} dt \quad (2.3)$$

dargestellt werden [4]. Werden die Grenzen nur über den positiven Zeitbereich mit $\omega = 2\pi f$ gewählt, ergibt sich vorheriges Integral mit Gleichung 2.2 zu

$$q = \int_0^{\infty} i(t) dt = |F(j\omega = 0)| = F(0) . \quad (2.4)$$

Somit ergibt sich die Ladung q aus dem Gleichanteil $F(0)$ der Fourier-Transformierten. Für einen idealen Dirac-Impuls würde das bedeuten, dass die Amplitudendichte über das gesamte Spektrum $F(0)$ gleich ist. Reale Teilentladungsimpulse besitzen jedoch eine endliche Anstiegszeit, sowie einen abflachenden Rücken. Ein allgemeiner Spannungs- oder Stromimpuls kann durch zwei Exponentialfunktionen wie folgt dargestellt werden:

$$i(t) = \hat{i}A \left(e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}} \right) , \text{ mit } \tau_1 < \tau_2 , \quad (2.5)$$

wobei A ein Skalar darstellt mit dem der Scheitelwert \hat{i} multipliziert werden muss, um zur Scheitelzeit den Scheitelwert zu erreichen (Bild 2a) [5]. Der Zusammenhang zwischen den Zeitparametern, Anstiegszeit t_p und Rückenhalbwertszeit t_2 eines

Impulses, und den Zeitkonstanten der Exponentialfunktionen wird mit heutiger Technik numerisch gelöst und erweist sich allgemein als eher schwierig und aufwendig, wobei hier auf einschlägige Literatur verwiesen sei [5]. Zur Berechnung des Frequenzspektrums eines Impulses muss eine Transformation vom Zeitbereich in den Bildbereich erfolgen. Dies kann anhand von Laplace- und Fourieranalyse durchgeführt werden. Die Laplace-Transformation einer einfachen Exponentialfunktion ergibt sich aus:

$$F(s) = \mathcal{L}\{i\}(s) = \int_0^{\infty} i(t)e^{-st} dt \quad (2.6)$$

$$\mathcal{L}\{e^{at}\} = \int_0^{\infty} e^{at} dt = \frac{1}{s-a} . \quad (2.7)$$

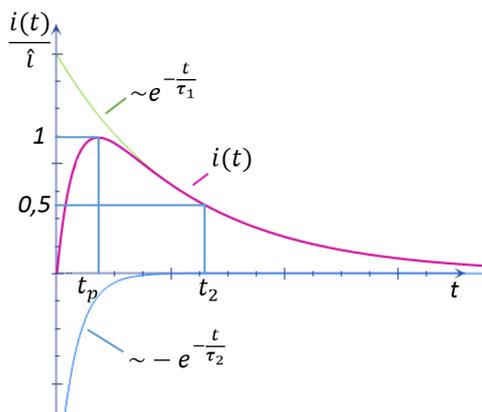
Daraus ergibt sich für die Laplace-Transformation von $i(t)$:

$$F(s) = \hat{i}A \left[\frac{1}{s + \frac{1}{\tau_1}} - \frac{1}{s + \frac{1}{\tau_2}} \right] . \quad (2.8)$$

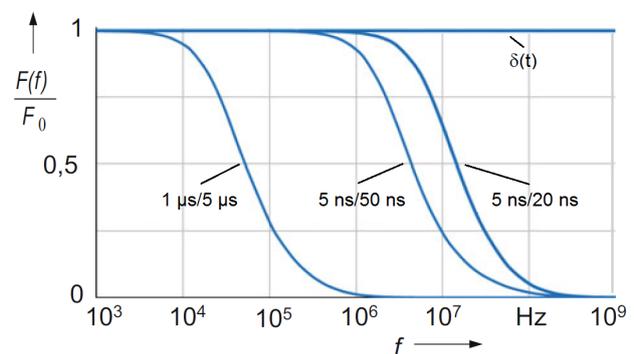
Mit $s = j\omega$ erfolgt ein Übergang in den Frequenzbereich. Die Amplitude kann aus dem Betrag von $|F(j\omega)|$ gewonnen werden, indem der Term in Real- und Imaginärteil aufgeteilt wird. Die Amplitudendichte ergibt sich aus dem Absolutbetrag bezogen auf $|F(\omega = 0)|$ gemäß Gleichung 2.9 zu:

$$F(\omega) = \left| \frac{F(j\omega)}{F(0)} \right| = \frac{1}{\tau_1 - \tau_2} \sqrt{\left[\frac{\tau_1}{1 + (\omega\tau_1)^2} - \frac{\tau_2}{1 + (\omega\tau_2)^2} \right]^2 + \left[\frac{\omega\tau_1^2}{1 + (\omega\tau_1)^2} - \frac{\omega\tau_2^2}{1 + (\omega\tau_2)^2} \right]^2} . \quad (2.9)$$

Wird die Amplitudendichte über die Frequenz aufgetragen, ergibt sich wie bei einem idealen Dirac-Impuls ein Bereich konstanter Amplitudendichte, wie in Bild 2b dargestellt [5, 6].



(a) Impulsform (in Anlehnung an [5])



(b) Amplitudendichte (in Anlehnung an [6])

Bild 2: Impulsform zweier überlagerter Exponentialfunktionen mit den Zeitkonstanten τ_1 und τ_2 im Zeitbereich sowie der Scheitel- t_p und der Rückenhalbwertszeit t_2 (a) und die Amplitudendichte verschiedener Impulse mit unterschiedlichen Zeitkonstanten im Frequenzbereich (b)

Es wurde gezeigt, dass die Herleitung für den Dirac-Impuls (Gleichung 2.4) auch für reale Impulse, zumindest im Bereich konstanter Amplitudendichte, welcher je nach Zeitkonstanten bis in den hohen Frequenzbereich (einige 100 kHz bis MHz) reichen kann, gilt. In diesem Bereich gilt $F(f) \approx F(0)$, woraus auch für reale Impulsformen folgt:

$$q = F(0) \approx F(f) . \quad (2.10)$$

Diese beiden Erkenntnisse stellen somit in beiden Fällen eine Integration dar und sind die Grundlage für heutige TE-Messsysteme. Es handelt sich einmal um eine „echte“ und einmal um eine sogenannten Quasi-Integration [6].

$$\text{Quasi-Integration: } q = F(0) \approx F(f) \quad (2.11)$$

$$\text{„echte“ Integration: } q = \int_0^{\infty} i(t)dt \quad (2.12)$$

Abgeleitete Größen

Der Vollständigkeit halber sei noch auf die von der scheinbaren Ladung abgeleiteten Größen eingegangen. In heutigen digitalen TE-Messgeräten kommen diese Größen, neben der scheinbaren Ladung, immer noch zum Einsatz, wodurch keine eigenen Messgeräte mehr dafür verwendet werden müssen, wie zu Anfangszeiten der TE-Diagnostik.

Impulswiederholrate n ist das Verhältnis aller gemessenen TE-Impulse zur Messdauer.

$$n = \frac{\text{Anzahl der Impulse}}{T} \quad (2.13)$$

Diese Messgeräte müssen eine ausreichend kurze Auflösung besitzen, um die höchste gewünschte Impulswiederholrate zu messen.

Mittlerer Entladungsstrom I ist das Verhältnis der Summe der Einzelladungsimpulse innerhalb des Zeitraums T zur gesamten Zeit.

$$I = \frac{1}{T}(|q_1| + |q_2| + |q_3| + \dots) \quad (2.14)$$

Entladungsleistung P ist die gemittelte Endladungsleistung im Zeitraum T . Das Prinzip beruht auf der Messung mit einem Oszilloskop.

$$I = \frac{1}{T}(q_1u_1 + q_2u_2 + q_3u_3 + \dots) \quad (2.15)$$

Quadratische Ladungsgröße D ist das Mittel der Summe der quadratischen Einzelladungswerte gemittelt über die Messdauer.

$$I = \frac{1}{T}(q_1^2 + q_2^2 + q_3^2 + \dots) \quad (2.16)$$

Funktstörspannung stellt eine Spezialform eines schmalbandigen Messgeräts mit Quasi-Integration dar, zeigt jedoch nicht direkt den Wert der scheinbaren Ladung an. Eine Umrechnung ist jedoch möglich.

Ultrabreitbandgeräte zur TE-Detektion werden in dieser Arbeit nicht näher behandelt.

Der mittlere Entladungsstrom, die Entladungsleistung und die quadratische Ladungsgröße können auch digital berechnet werden, was in aktuellen TE-Messgeräten zur Anwendung kommt [1, 6].

2.2.3 Integration im Frequenzbereich – Quasi-Integration

Wie bereits zuvor erwähnt, gibt es zwei verschiedene Arten nach denen TE-Messgeräte arbeiten. Hier werden Messgeräte behandelt die zur Ladungsbestimmung eine Quasi-Integration vornehmen, das bedeutet eine Integration im Frequenzbereich nach Gleichung 2.11. Bei diesem Verfahren muss zwischen schmal- und breitbandigen Messgeräten unterschieden werden. Die empfohlenen Bandbreiten und Mittenfrequenzen sind in der Norm IEC 60270 festgelegt [1].

- **schmalbandig:**

$$9 \text{ kHz} \leq \Delta f \leq 30 \text{ kHz}$$

$$50 \text{ kHz} \leq f_0 \leq 1 \text{ MHz}$$

- **breitbandig:**

$$30 \text{ kHz} \leq f_1 \leq 100 \text{ kHz}$$

$$f_2 \leq 1 \text{ MHz}$$

$$100 \text{ kHz} \leq \Delta f \leq 900 \text{ kHz}$$

Messgeräte, die eine Integration im Frequenzbereich vornehmen, müssen eine begrenzte Bandbreite aufweisen, welche sich innerhalb des Bereichs der konstanten Amplitudendichte befindet, um Gleichung 2.11 anwenden zu können. Ist dies nicht der Fall, muss mit einem Integrationsfehler gerechnet werden. Bei Messgeräten, die nach dem Prinzip der Quasi-Integration arbeiten, wird die Messung von $F(f)$ auf eine Messung im Zeitbereich zurückgeführt, jedoch wird die Konstanz der Amplitudendichte im Frequenzbereich zunutze gemacht.

Betrachtet wird ein idealer Bandpass mit einer Verstärkung von $A(f) = A_0$ innerhalb seines Durchlassbereichs zwischen den Grenzfrequenzen f_1 und f_2 und einer Mittenfrequenz von f_0 . Die Impulsantwort $u(t)$ dieses Bandpasses kann folgendermaßen beschrieben werden [4]:

$$u(t) = 2A_0F(f)\{f_2 \cdot \text{si}[2\pi f_2(t - t_0)] - f_1 \cdot \text{si}[2\pi f_1(t - t_0)]\}, \quad (2.17)$$

mit

$$\text{si } x = \frac{\sin x}{x}.$$

Die Zeit t_0 beschreibt die Impulslaufzeit im Messsystem. Das Maximum dieser Funktion ergibt sich nach [6] zu:

$$u_{max} = 2A_0F(f)\Delta f. \quad (2.18)$$

Das bedeutet, dass für einen Dirac-Impuls überall im Frequenzspektrum die Amplitude gleich dem Maximalwert u_{max} ist und mit Gleichung 2.11 folgt weiter, dass die Impulsladung diesem Wert proportional ist. Bei realen Impulsen ist

dies näherungsweise ebenfalls der Fall, wenn die Amplitudendichte bis zur oberen Grenzfrequenz gleich bleibt und es ergibt sich:

$$u_{max} \sim q . \quad (2.19)$$

In der Realität können geringe Abweichungen aufgrund der realen Bandpasscharakteristik vorkommen. Die Impulsantwort schaut je nach Filtercharakteristik, aber hauptsächlich in Abhängigkeit der Bandbreite, sehr unterschiedlich aus, was wiederum entscheidend für die TE-Messung ist [6].

Schmalbandige Quasi-Integration: Schmalbandige Messgeräte führen die Quasi-Integration der TE-Impulse mit einem schmalbandigen Bandpassfilter durch, wobei die Grenzfrequenzen so zu wählen sind, dass diese im Bereich konstanter Amplitudendichte liegen. Es sollte die Amplitude mindestens bis zur oberen Grenzfrequenz $f_2 = f_0 + \Delta f/2$ konstant bleiben. Der Ladungswert ergibt sich, wie zuvor gezeigt, aus dem Maximalwert der Impulsantwort $u_{max} \sim q$. Die Impulsantwort kann mit Gleichung 2.17 berechnet werden und schwingt um die Mittenfrequenz f_0 . Der Maximalwert ist im Vergleich zu breitbandigen Filtern sehr viel kleiner und die Dauer der Schwingung länger. Durch die starke Oszillation der Impulsantwort geht die Information der Polarität des ursprünglichen Impulses verloren. Es kann somit nicht mehr anhand der Einhüllenden der Kurvenform zwischen positiver und negativer Amplitude unterschieden werden. Die Halbwertszeit der Impulsantwort, d. h. die Zeit bis die Kurve auf die Hälfte des Maximalwerts zurückgegangen ist, ist durch die Verbreiterung der Filterantwort um einiges größer im Vergleich zur breitbandigen Messung. Diese liegt typischerweise im Bereich um $80 \mu s$ [1]. Daraus folgt, dass die Totzeit (engl. dead time) bzw. die Impulsauflösezeit, d. h. die Zeit in der das Messgeräte nicht zur Detektion weiterer Impulse zur Verfügung steht, größer gewählt werden muss. Wird die Totzeit zu klein gewählt, kann es zu einer Doppelerkennung eines einzelnen Impulses kommen.

Schmalbandige Messungen haben andererseits den Vorteil, dass das Frequenzband in dem gemessen wird, z. B. beim Auftreten von Störungen, verschoben werden kann [6].

Breitbandige Quasi-Integration: Breitbandige Messsysteme verwenden Bandpassfilter mit den oben angeführten Grenzfrequenzen, welche wieder im Bereich konstanter Amplituden liegen sollen. Die mit Gleichung 2.17 berechnete Impulsantwort weist im Vergleich zu schmalbandigen Bandpässen eine schmalere Kurvenform mit deutlichem Maximum im positiven oder negativen Bereich auf. Es handelt sich um eine stark gedämpfte Schwingung, wodurch wiederum die Totzeit kleiner gewählt werden kann und das Messgeräte schneller wieder zur Detektion weiterer Impulse bereit steht. Allgemein gilt, umso höher die Bandbreite, umso kleiner ist die Halbwertszeit. Die Impulsauflösezeit liegt typischerweise bei $5 \mu s$ bis $20 \mu s$ [1]. Der Ladungswert ist wiederum proportional zum Maximum der Antwort des Messgerätes. Dieses kann mit einem Spitzenwertdetektor oder einem Pulshöhenanalysator erkannt werden. Moderne Messgeräte verstärken und digitalisieren die Impulsantwort des Filters mit einem A/D-Wandler und werten die gespeicherten Datensätze über eine oder mehrere Perioden aus. Durch die Speicherung können Histogramme, wie das

PRPD-Pattern erstellt werden. Der Phasenbezug wird durch Abtastung des transformierten Spannungssignals mit einem zweiten A/D-Wandler hergestellt. Diese Art von Messgeräten verwenden zwar ein analoges Messprinzip mit Bandpass, gehören aber aufgrund der Digitalisierung der Daten zu den digitalen TE-Messgeräten [6].

Vergleich: Der Vergleich von breit- und schmalbandigen Messgeräten zeigt, dass beide Arten ihre Existenzberechtigung und ihre jeweiligen Vor- und Nachteile haben (Tabelle 1). Bild 3 zeigt die Impulsantworten im Vergleich. Der breitbandige Filter wurde mit einer Bandbreite von $f_1 = 10$ kHz bis $f_2 = 100$ kHz gewählt und der schmalbandige Bandpass mit einer Mittenfrequenz von $f_0 = 100$ kHz und einer Bandbreite von $\Delta f = 9$ kHz. Es wurde jeweils die Antwortfunktion $u(t)$ nach Gleichung 2.17 ausgewertet. Es ist deutlich zu erkennen, dass die Antwort eines breitbandigen Bandpasses auf einen Impuls deutlich höher ausfällt und die Halbwertsdauer sehr viel kürzer ist. Dadurch können TE-Impulse in kürzeren Abständen polaritätsrichtig erkannt werden. Dies erweist sich bei schmalbandigen Bandpässen schwieriger, da die Impulsantwort breiter und kleiner ist. Somit muss, wie bereits erwähnt, die Totzeit aufgrund der größeren Halbwertszeit länger gewählt werden.

Tabelle 1: Vor- und Nachteile von schmal- und breitbandigen Messgeräten

Filter	Vorteile	Nachteile
schmalbandig	<ul style="list-style-type: none"> • veränderbare Mittenfrequenz • störunempfindlich 	<ul style="list-style-type: none"> • keine Polarität • hohe Totzeit
breitbandig	<ul style="list-style-type: none"> • Polaritätsinformation • geringe Totzeit 	<ul style="list-style-type: none"> • störempfindlich

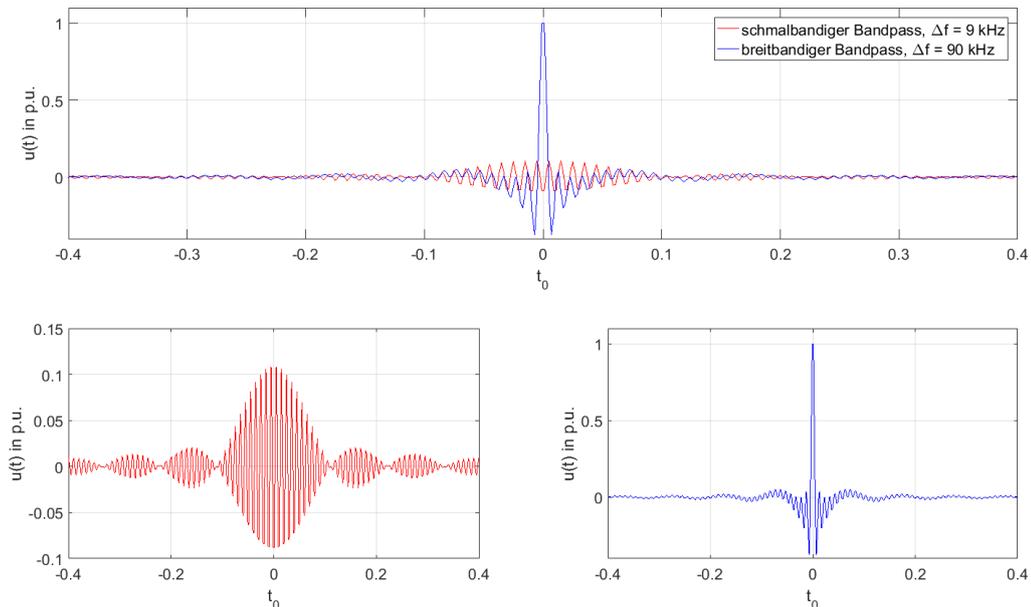


Bild 3: Impulsantwort schmal- und breitbandiger Bandpassfilter im Vergleich (oben) und Detailansichten (unten) normiert auf den Maximalwert von $u(t)$

2.2.4 Integration im Zeitbereich – „echte“ Integration

Im Vergleich zur Quasi-Integration arbeiten diese Messgeräte nach dem Prinzip der „echten“ Integration, wobei je nachdem wie die Integration durchgeführt wird, auch hier zwischen analogen und digitalen Messgeräten unterschieden werden muss. Die erste Gruppe von Messgeräten verwendet analoge Schaltungen mit Operationsverstärker. Diese Art kam jedoch nur sehr kurze Zeit zum Einsatz. Die zweite Gruppe führt eine Digitalisierung der Impulse und anschließende numerische Integration oder eine Quasi-Integration mit einem digitalen Filter durch.

Analoge TE-Messsysteme – Integrierende Verstärker: Dieses Messprinzip findet heutzutage in der TE-Diagnostik so gut wie keine Anwendung mehr, jedoch wird dieses Verfahren immer noch zur Kalibrierung von TE-Kalibratoren angewendet [6]. Die Messung der Impulse erfolgt hierbei durch einen integrierenden Operationsverstärker (OPV). Dabei kommt es durch kapazitive Rückkopplung zur zeitlichen Integration des am Eingang angelegten Stromes. Unter der Annahme eines idealen Verstärkers berechnet sich die Ausgangsspannung zu [7]:

$$u(t) = -\frac{1}{RC} \int_0^{\infty} i(t) dt . \quad (2.20)$$

Nach theoretisch unendlich langer Zeit stellt sich ein Endwert ein, welcher der Ladung q proportional ist (Gleichung 2.12). Wieder kann mit z. B. einem Spitzenwertdetektor oder durch Digitalisierung mit einem A/D-Wandler der Endwert detektiert werden. In der Praxis wird üblicherweise der OPV wieder auf Null zurückgesetzt, um zur Integration eines weiteren Impulses bereitzustehen. Ist dies nicht der Fall, entspricht der Endwert der Integration der Summenladung innerhalb der gewählten Zeitdauer. Bild 4 zeigt einen Integrierer an dessen Eingang ein Stromimpuls angelegt wird und die Ausgangsspannung, welche sich aus dem Integral des Eingangstromes berechnet.

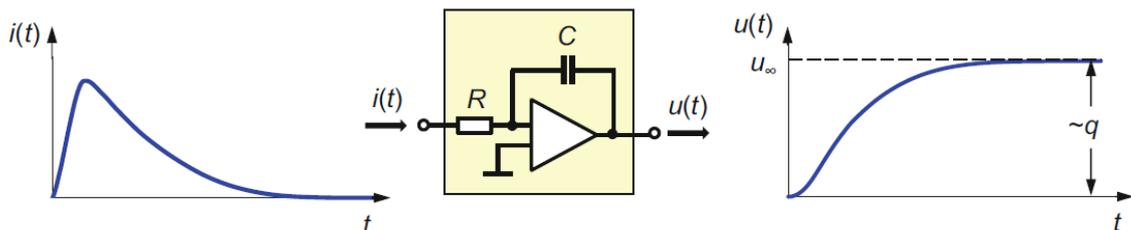


Bild 4: Integrierender Verstärker mit einem Stromimpuls $i(t)$ am Eingang und der Ausgangsspannung $u(t)$ [6]

Digitale TE-Messgeräte – numerische Integration: Aktuelle TE-Messgeräte arbeiten nach diesem Verfahren. Hierbei durchlaufen die TE-Impulse zunächst einen sehr breitbandigen Anti-Aliasing-Filter (typisch sind Grenzfrequenzen bis 20 MHz [6]). Zur korrekten Abtastung eines analogen Signals muss das Nyquist-Shannon-Abtasttheorem erfüllt sein, welches u. a. 1949 von Claude E. Shannon wie folgt beschrieben wurde:

„If a function $f(t)$ contains no frequencies higher than W cps [*cps* entspricht *cycles per second* und wurde vom SI-System durch Hertz abgelöst, Anmerkung des Verfassers], it is completely determined by giving its ordinates at a series of points spaced $1/2 W$ seconds apart.“ [8]

Praktisch bedeutet dies, dass im abgetasteten Signal keine Frequenzanteile größer der halben Abtastfrequenz f_a vorkommen dürfen [9]:

$$|Y(f)| = 0 \text{ für } |f| > \frac{f_a}{2} . \quad (2.21)$$

Ist diese Bedingung nicht erfüllt, kommt es zu spektralen Überlappungen, auch Aliasing genannt, wodurch eine genau Zuordnung der Anteile nicht gewährleistet ist. Ein Anti-Aliasing-Filter wirkt dabei wie ein Tiefpass der alle Spektralanteile bis zur halben Abtastfrequenz f_a störungsfrei durchlässt und alle Anteile darüber abschneidet. Das bandbegrenzte Signal wird anschließend verstärkt und durch einen A/D-Wandler digitalisiert. Anhand eines zweiten A/D-Wandlers wird die Spannung abgetastet und dadurch ein Phasenbezug der Teilentladungen hergestellt. Die beiden Signale werden nach der Abtastung einem FPGA (engl. **F**ield **P**rogrammable **G**ate **A**rray), genauer einer digitalen Logikschaltung, zugeführt und sehr schnell weiterverarbeitet [6].

Wie zuvor kurz erwähnt, gibt es zur Ladungsermittlung zwei verschiedene Varianten. Das erste Verfahren, Integration anhand numerischer Integration, arbeitet dabei wie eine näherungsweise „echte“ Integration nach Gleichung 2.12. Dieses Verfahren erweist sich jedoch als zeitaufwendiger im Vergleich zum zweiten, bei welchem die Integration der TE-Impulse wieder auf Quasi-Integration zurückzuführen ist (Gleichung 2.11). Im Gegensatz zur Quasi-Integration mit analogen, bandbegrenzenden Filtern, kommen bei dieser Methode digitale Filter zur Anwendung. Das grundsätzliche Prinzip bleibt aber das gleiche. Von großem Vorteil dieses Verfahrens stellt sich die Flexibilität heraus, da Mittenfrequenz und Bandbreite des Filters wählbar sind und somit etwaigen Störungen ausgewichen werden kann. Bild 5 zeigt das Blockschaltbild eines solchen TE-Messgerätes [6].

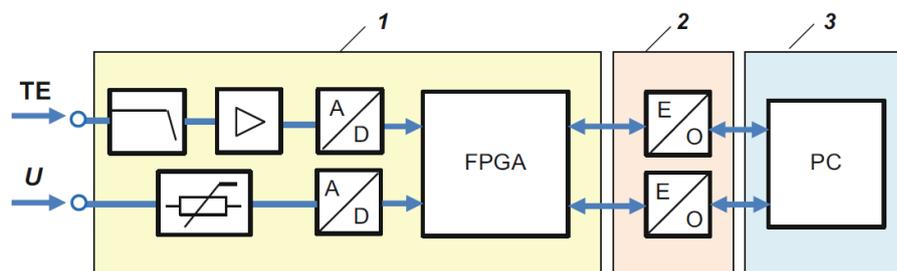


Bild 5: Blockschaltbild eines modernen digitalen TE-Messgerätes [6]

Moderne TE-Messsysteme, wie das in dieser Arbeit verwendete Gerät MPD 600 und dessen Vorgänger MPD 540 der Firma Omicron, arbeiten nach dem zweiten Prinzip mit digitalen Filtern [10].

2.2.5 Anforderungen an digitale TE-Messgeräte

Digitale TE-Messgeräte sollten folgende Anforderungen erfüllen [1]:

- Aufzeichnung der größten wiederholenden Stärke der TE-Impulse
- Aufzeichnung der scheinbaren Ladung q_i zur Zeit t_i
- Aufzeichnung des Momentanwerts der Spannung zur Zeit t_i eines Impulses q_i
- Aufzeichnung des Phasenwinkels eines Impulses q_i zur Zeit t_i

Die Aktualisierung der Anzeige sollte 1 s nicht überschreiten. Beim Auftreten von Grundrauschen oder verhältnismäßig vielen Teilentladungen, deren Wert klein in Bezug auf den größten zu messenden Wert ist, darf ein bipolarer Schwellwert (engl. Threshold) verwendet werden, welcher jedoch unbedingt aufgezeichnet werden soll.

2.2.6 IEC-Gewichtung

In den vorangegangenen Kapiteln wurde beschrieben, wie TE-Messgeräte den Wert der scheinbaren Ladung ermitteln und welche verschiedenen Messprinzipien dabei zum Einsatz kommen. Der von Messgeräten angezeigte Ladungswert (z. B. Q) entspricht nicht der tatsächlichen scheinbaren Ladung, sondern einem gewichteten Wert. Der Grund für die Gewichtung liegt darin, dass einzelne Störimpulse weniger stark in den Anzeigewert eingehen sollen als periodisch (z. B. mit 20 ms) oder häufiger auftretende Impulse. Die Gewichtung erfolgt anhand der Häufigkeit von TE-Impulsen innerhalb einer Sekunde. Dadurch wird eine konstante Impulssequenz mit z. B. einer Frequenz von $f = 1$ Hz mit 35 % gewichtet. Ist die Frequenz höher, so erhöht sich auch die prozentuale Gewichtung. Diese Gewichtungsfunktion ist in Tabelle 2 dargestellt. Grund für die Gewichtung, auch bei digitalen Messgeräten, welche den tatsächlichen scheinbaren Ladungswert anzeigen könnten, ist die Nachbildung eines analogen Zeigermessgeräts mit trägem Verhalten. Durch einen einzelnen Impuls würde der Ausschlag des Zeigers nur gering ausfallen, wohingegen sich konstante wiederholende Impulse den Zeiger je nach Ladungshöhe mehr zum Ausschlag bringen würden. Durch die Gewichtung kommt es zu einer gewissen Störunterdrückung oder Fehlerkorrektur. Die Anzeige des gewichteten Ladungswerts sollte dabei nicht länger als 0,44 s sein. Somit werden durch die Gewichtung alle in diesem Zeitfenster auftretenden Ladungswerte auf einen einzelnen Wert reduziert [1].

Tabelle 2: Impulsfolgeantwort von TE-Messgeräten – Gewichtungsfunktion nach [1]

N (1/s)	1	2	5	10	50	≥ 100
R_{\min} (%)	35	55	76	85	94	95
R_{\max} (%)	45	65	86	95	104	105

N ... Impulswiederholfrequenz

R ... Ablesewert

Die Gewichtungsfunktion ist auf Prüfungen mit Wechselspannung mit 50 Hz bzw. 60 Hz beschränkt und es wird in IEC 60270 darauf hingewiesen, dass sie **nicht für Prüfungen mit Gleichspannung** geeignet ist [1].

2.3 Darstellungsarten von Teilentladungen

2.3.1 Kenngrößen

Zur Interpretation von Teilentladungsereignissen kommen verschiedene Hilfsmittel zum Einsatz, wobei versucht wird, die je nach Spannungsart zur Verfügung stehenden Größen in Diagramme einzuzeichnen und darzustellen. Diese Kenngrößen können gemessen und weiterverarbeitet werden. Nicht nur gemessene Größen, sondern auch davon abgeleitete Größen, können in der TE-Diagnostik Rückschlüsse auf den zugrunde liegenden Fehler zulassen, wobei jedoch die physikalischen Effekte von Teilentladungen so komplex sind, dass eine eindeutige Aussage nur selten möglich ist.

Erschwerend kommt hinzu, dass bei Gleichspannung aufgrund des fehlenden Phasenbezugs, im Vergleich zu Wechselspannung, nicht dieselben bzw. nur teilweise dieselben Kenngrößen ausgewertet werden können (Bild 6). Tabelle 3 zeigt die verschiedenen Kennwerte für TE-Impulse bei AC- und DC-Spannung, wobei nur auf die Grundgrößen Bezug genommen wurde. Davon abgeleitete Größen, wie z. B. q oder Δt_i , wurden in der Tabelle nicht angeführt.

Tabelle 3: Kenngrößen von AC- und DC-TE-Messungen im Vergleich

Kenngrößen	
AC	DC
q, t, U_{\sim}, φ	$q, t, U_{=}$

q	...	scheinbare Ladung	q	...	scheinbare Ladung
t	...	Zeit	t	...	Zeit
U_{\sim}	...	Wert der Wechselspannung	$U_{=}$...	Wert der Gleichspannung
φ	...	Phasenlage			

Eines der heutzutage wichtigsten Interpretationsverfahren, das sogenannte PRPD-Pattern (**P**hase **R**esolved **P**artial **D**ischarge), kann aufgrund der Tatsache des fehlenden Phasenbezugs, bei TE-Messungen mit Gleichspannung, nicht angewendet werden. Deshalb mussten neue Interpretationsansätze gesucht werden, welche in diesem Kapitel erklärt werden.

2.3.2 Wechselspannung

Für die Interpretation von TE-Impulssequenzen bei Wechselspannung gibt es eine Vielzahl an verschiedenen Darstellungsarten, wobei die Behandlung all dieser Darstellungen den Rahmen dieser Arbeit sprengen würde. Es werden jedoch die wichtigsten und gebräuchlichsten Diagramme vorgestellt.

PRPD-Pattern: Das PRPD-Pattern, auch $H(q, \varphi)$ genannt, ist heutzutage eines der wichtigsten Methoden um Teilentladungen zu klassifizieren. Es werden die einzelnen Impulse mit ihrer scheinbaren Ladung über die Phasenlage der Spannung aufgetragen. Diese Darstellung kann noch erweitert werden, indem eine dritte Dimension für die Häufigkeit der Impulse hinzugefügt wird, um die Intensität farblich

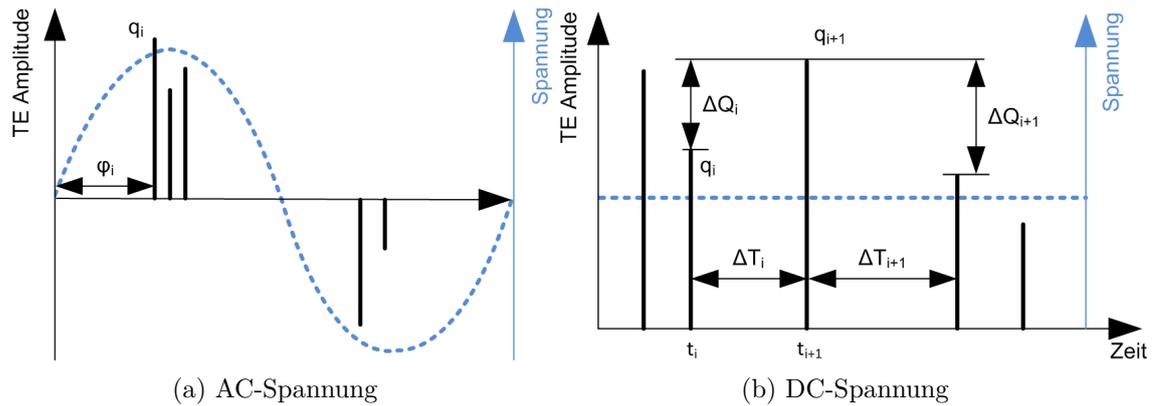


Bild 6: Kenngrößen von TE-Impulsen bei AC- (a) und DC-Spannung (b)

zu skalieren (Bild 7). Dies findet sich in der Literatur auch unter dem Namen φ -q-N-Diagramm. Das Diagramm kann als integrativ angesehen werden, da der Zeitbezug verloren geht und nur die Intensität der Impulse und deren Phasenlage gezeigt wird [11]. Manche modernen Messgeräte, wie das der Firma Omicron, bieten hierfür sogenannte Streaming-Files an, wodurch es möglich ist, auch nach einer Messung das Pattern und dadurch das zeitliche Auftreten einzelner Impulse nachzuvollziehen und zu analysieren [12].

Es hat sich gezeigt, dass anhand der Phasenlage auf verschiedene Fehler geschlossen werden kann, was das PRPD-Pattern zu einem mächtigen Interpretationstool macht [13]. Bei Spitze-Platte-Anordnungen kommt es beispielsweise zum Auftreten von Teilentladungen im Bereich der Spannungsmaxima (Bild 7).

Ellipsendarstellung: Die Ellipsendarstellung von Teilentladungsimpulsen beruht auf dem gleichen Prinzip wie das PRPD-Pattern. Anstatt jedoch den Phasenbezug auf der x-Achse darzustellen, dient die Ellipse zur Nachbildung der Phasenlage. Mit Indikatoren wie 0, +, 0, - oder die Angabe von Grad, wird die Phasenlage angezeigt und anhand der Höhe die scheinbare Ladung. Diese Darstellung wird auch noch bei digitalen Messgeräten angewendet und beruht auf dem gleichen Prinzip wie noch bei analogen Messgeräten. Der größte Unterschied zum PRPD-Pattern ist, dass diese Darstellung nur Augenblickswerte anzeigt [12, 13].

H(Q)-Verteilung: Dieses Diagramm ist eine Häufigkeitsverteilung der Ladungswerte. Angezeigt wird die Nummer von Teilentladungen mit einem bestimmten Ladungswert (Häufigkeit) in Bezug zur Ladung [12, 14].

Die genannten Beispiele sind nur ein kleiner Auszug, tatsächlich gibt es noch viel mehr Ansätze um Teilentladungen zu klassifizieren und charakterisieren. In [14] ist beispielsweise eine Auflistung von verschiedenen Darstellungen zu finden.

2.3.3 Gleichspannung

Die Interpretation bei Gleichspannung erweist sich als schwieriger, da es hier keine Standarddarstellung gibt, anhand welcher auf den Fehler geschlossen werden kann.

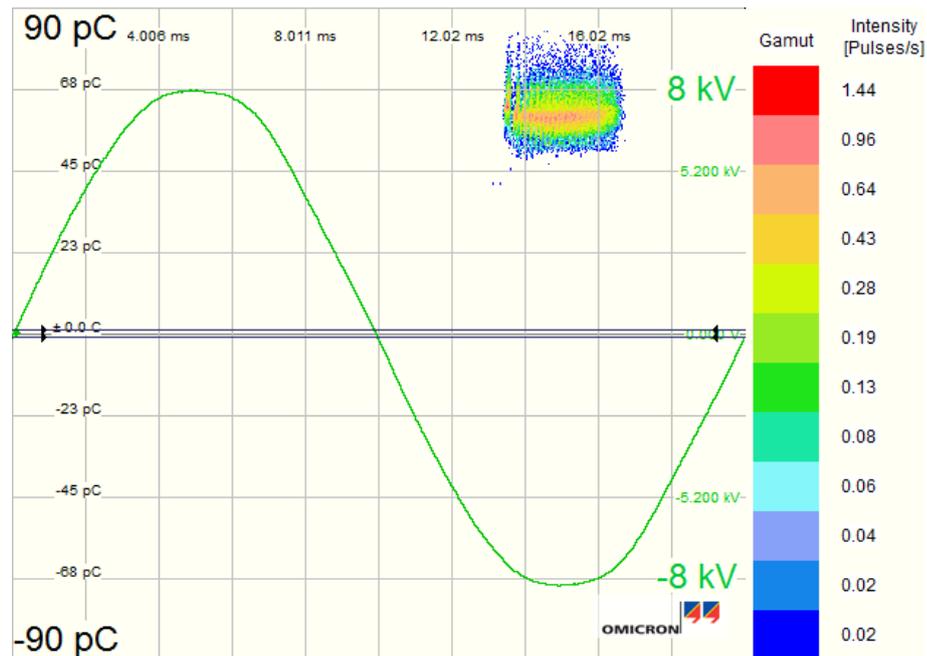


Bild 7: PRPD-Pattern einer Spitze-Platte-Anordnung in N_2 bei $p = 0,5$ MPa relativem Druck bei 1,1-facher TE-Einsatzspannung über eine Dauer von 1 min

Es gibt allerdings verschiedene Ansätze, die vielversprechend erscheinen, wie das im Folgenden beschriebene NoDi*-Diagramm [15]. Die größte Herausforderung besteht darin, anhand der in Tabelle 3 gezeigten Kenngrößen für Gleichspannung und davon abgeleiteten Größen, Darstellungen zu finden, die eine Analyse der Messdaten erleichtern.

In IEC 60270 sind Darstellungsarten, wie das $Q(t)$ -Diagramm, die akkumulierte Ladung und die dazugehörigen Balkendiagramme beschrieben [1]. Beispiele dafür werden in Kapitel 3.2.3 gezeigt.

$Q(t)$ -Diagramm: Da bei TE-DC-Messungen die Phase der Spannung nicht herangezogen werden kann, macht es Sinn, die Amplitude der scheinbaren Ladung über die Zeit aufzunehmen. Anhand von solchen Darstellungen können die Ladungshöhen und sogenannte Bänder verschiedener Ladungswerte erkannt werden. Wird dieses Diagramm erweitert, indem auch der zeitliche Abstand Δt zwischen zwei aufeinanderfolgenden Impulsen und die Spannung über die Zeit aufgetragen wird, so kann das TE-Verhalten auch mit der Spannung in Verbindung gebracht werden.

Akkumulierte Ladung: Hierbei handelt es sich im Grunde wie zuvor um eine Darstellung der TE-Impulse über die Zeit, jedoch wird auf der y-Achse die akkumulierte Ladung, d. h. die bis zum jeweiligen Zeitpunkt aufgetretene Summenladung dargestellt. Dieses Diagramm kann verwendet werden um sicherzustellen, dass die akkumulierte Ladung innerhalb eines bestimmten Zeitraums einen Grenzwert nicht überschreitet.

Balkendiagramme: Es werden in IEC 60270 auch zwei verschiedene Balkendiagramme angeführt, welche mehr Informationen über das Teilentladungsverhalten und die Überschreitung eines bestimmten Grenzwertes innerhalb einer vorgegebenen Zeit geben. Einerseits ein Balkendiagramm, welches die Anzahl der Impulse innerhalb der festgelegten Balkenbreite anzeigt und andererseits ein akkumuliertes Balkendiagramm [1].

NoDi*-Diagramm: Die grundsätzliche Idee von diesem Diagramm ist es, die Ladungs- und Zeitwerte eines jeden Impulses zu **normalisieren** und zu **differenzieren**, daher der Name NoDi [15,16]. Die Kenngrößen einer Teilentladung bei DC-Spannung sind u. a. der Ladungswert und die Zeit. Anstatt des Zeitwerts selbst, wird jedoch die Zeit zwischen zwei Impulsen Δt genommen. Die Normalisierung wird nur graphisch angewendet. Die Differenziation erfolgt durch die Differenzbildung zweier aufeinander folgender Werte und wird in Großbuchstaben angegeben:

$$Q'_i = \Delta q_i = |q_{i+1} - q_i| \quad (2.22)$$

$$T'_i = \Delta t_i = t_{i+1} - t_i . \quad (2.23)$$

Die Werte von Q'_i und T'_i können in verschiedenen Kombinationen gegeneinander aufgetragen werden, wodurch sich unterschiedliche Diagramme ergeben. Diese sind:

- NoDi*_Q-Diagramm: Q'_{i+1} über Q'_i
- NoDi*_T-Diagramm: T'_{i+1} über T'_i
- NoDi*_QT-Diagramm: Q'_i über T'_i .

Die Intensität kann wie beim PRPD-Pattern durch eine Rasterung des Diagramms und einer Farbskalierung veranschaulicht werden. Für unterschiedliche Fehler ergeben sich verschiedene Muster, was die Interpretation vereinfacht. Jedoch zeigt dieses Diagramm vor allem beim Auftreten von überlagerten Fehlern seine Schwächen, da hierbei typische Muster ineinander verschwimmen bzw. verloren gehen. Ein weiterer wichtiger Punkt, der unbedingt bei der Verwendung dieser Darstellung beachtet werden muss, ist es, dass die Aufnahme von Grundrauschen und Störimpulsen vermieden werden sollte. Die Gründe liegen in der Differenzbildung, welche durch Grundrauschen und mehreren gleichzeitig auftretenden Fehlern erheblich verändert wird. Ein Beispiel eines NoDi*-Diagramms ist in Bild 8 dargestellt. Es handelt sich abermals um eine Spitze-Platte-Anordnung in Stickstoff ($p = 0,5$ MPa, relativer Druck) bei Gleichspannung negativer Polarität.

Das NoDi*-Diagramm spielt eine zentrale Rolle in dieser Arbeit und kam im Rahmen einer Bachelorarbeit in einem Auswerteprogramm für TE-Messungen bei Gleichspannung bereits zur Anwendung [2].

Auch hier gibt es wieder weitaus mehr Diagramme, welche in [13,17] zu finden sind.

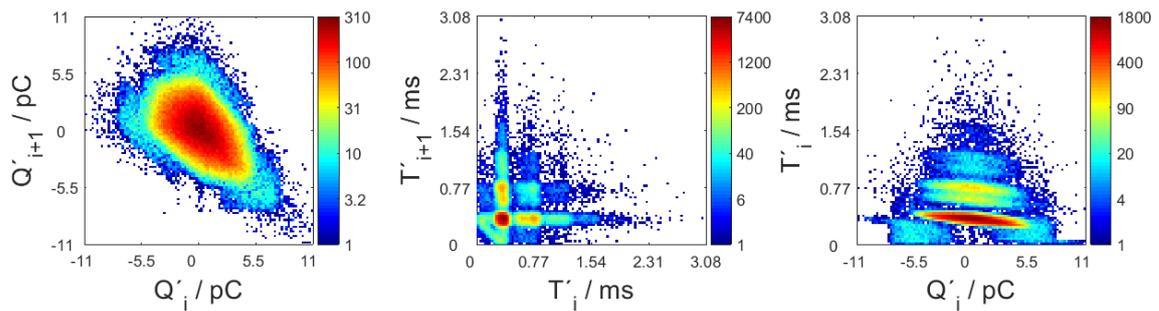


Bild 8: NoDi*-Diagramm einer Spitze-Platte-Anordnung in N_2 bei $p = 0,5$ MPa relativem Druck bei 1,1-facher TE-Einsatzspannung über eine Dauer von 1 min

2.4 Defekte in GIS/GIL-Systemen

2.4.1 Allgemeines

Bereits in den 60er Jahren des 20. Jahrhunderts wurden gasisolierte Schaltanlagen (GIS) entwickelt. Sie stehen Freiluftschaltanlagen und Innenraumschaltanlagen in offener Bauweise gegenüber, wobei GIS, aufgrund des verwendeten SF_6 als Isoliermedium kompakter aufgebaut werden können. Ein weiterer Vorteil besteht in der modularen Bauweise und dem geringen Wartungsaufwand. Gasisolierte Leitungen (GIL) sind aus der Technologie der GIS entstanden und besitzen die gleichen Vorteile. Im Laufe der Jahre wurde diese Technologie immer weiterentwickelt und verbessert [18]. Trotz des geringen Wartungsaufwandes sind auch GIS-Anlagen nicht vor Defekten und dadurch Feldstärkenüberhöhungen im Inneren geschützt. Durch Montagefehler, Herstellungsfehler oder auch unvollständige Reinigung der Teile vor der Inbetriebnahme kann es zu Störungen kommen, welche im Betrieb erhebliche Schäden und lange Wartungszeiten hervorrufen können. Typische Defekte sind:

- Hohlräume in den Isolatoren
- Spitzen auf Leitern und am Gehäuse
- Freie Potentiale
- Partikel auf Isolatoren
- Frei bewegliche, metallische Partikel

Diese Fehler können jedoch rechtzeitig vor der Inbetriebnahme und mit Monitoring-Systemen während des Betriebes erkannt werden. Eine der wichtigsten Diagnosemethoden zur Erkennung solcher Fehler stellt die Teilentladungsmessung dar. In den Bildern 9a und 9b sind der Aufbau einer GIS sowie mögliche Defekte dargestellt.

Am Institut für Hochspannungstechnik und Systemmanagement der TU Graz werden Untersuchungen zu diesen Defekten durchgeführt. Bereits im Vorfeld dieser Arbeit wurden Teilentladungen von typischen Defekten in Testzellen, unter Verwendung von SF_6 als Isoliergas, untersucht. Diese wurden mit Wechsel- sowie Gleichspannung unter der Variation einiger Parameter durchgeführt.

Die Messungen frei beweglicher Partikel wurden im Rahmen dieser Arbeit durchgeführt, weshalb hier nur auf diesen speziellen Defekt eingegangen wird.

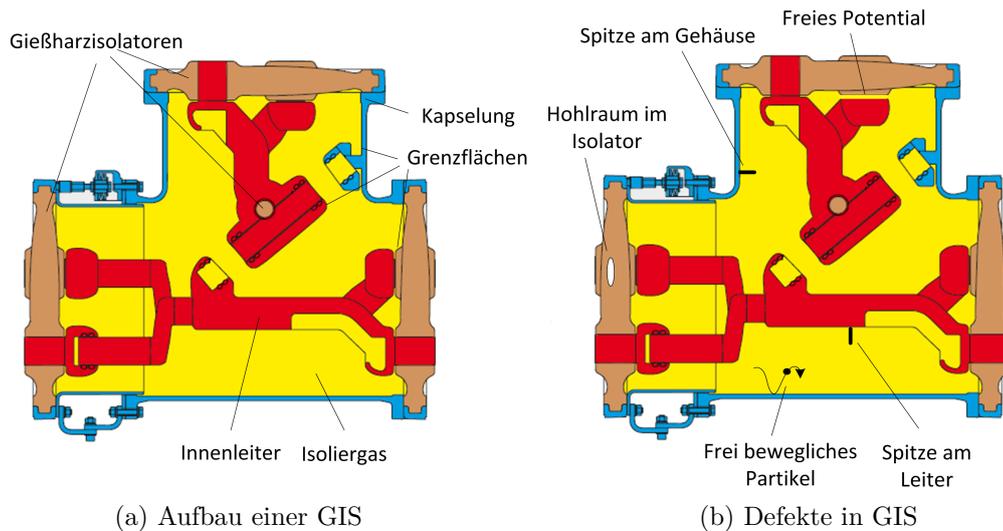


Bild 9: Aufbau einer modernen GIS (a) und mögliche Defekte (b) im gekapselten Aufbau am Beispiel eines Sammelschienenentrennschalters mit Erdungsschalter (Querschnittszeichnung entnommen aus [19])

2.4.2 Freie metallische Partikel in GIS und GIL

Eine Fehlerstatistik aus dem Jahr 1998 zeigt, dass mehr als 50 % aller Fehler in GIS auf elektrische Fehler zurückzuführen sind, die anhand von TE-Messungen und Online-Monitoring-Systemen erkannt werden können. Dabei fällt der größte Anteil von 20 % aller Fehler auf Partikel und Fremdkörper im gekapselten Aufbau. Bild 10 zeigt die statistische Aufteilung der verschiedenen Fehler [20].

Partikel oder Fremdkörper können durch Montagefehler, aufgrund von Produktionsfehlern, während des Transportes, unsauberes Arbeiten bei der Endreinigung oder durch Schalthandlungen auftreten. Die Durchschlagfeldstärke wird durch bewegliche metallische Partikel erheblich herabgesetzt, was zu Durchschlägen im SF_6 und Überschlügen an den Gießharzisolatoren führen kann, je nach verwendetem Druck und Größe des Partikels. Es zeigt sich jedoch, dass bei Stoßspannungen eine kritischere und dadurch niedrigere Feldstärke als bei Beanspruchungen mit reiner Wechselspannung von 50 Hz auftritt, wobei Blitzstoßspannungen mit einer Stirnzeit von $1,2 \mu\text{s}$ und einer Rückenhalbwertszeit von $50 \mu\text{s}$ ($1,2 \mu\text{s}/50 \mu\text{s}$) kritischer zu sehen sind als Schaltstoßspannungen ($250 \mu\text{s}/2500 \mu\text{s}$) [11, 21].

Partikelbewegung bei DC-Spannung: Gelangen metallische Fremdkörper in das gekapselte System, werden diese durch das elektrische Feld \vec{E} aufgeladen. Die Ladung eines metallischen Partikels kann anhand der Maxwell'schen Gleichungen, genauer mit dem Gauß'schen Gesetz, beschrieben werden:

$$Q_P = \oint_A \vec{D} \cdot d\vec{A} = \varepsilon_0 \cdot \oint_A \vec{E} \cdot d\vec{A}, \quad (2.24)$$

mit der elektrischen Flussdichte \vec{D}

$$\vec{D} = \varepsilon_0 \vec{E}. \quad (2.25)$$

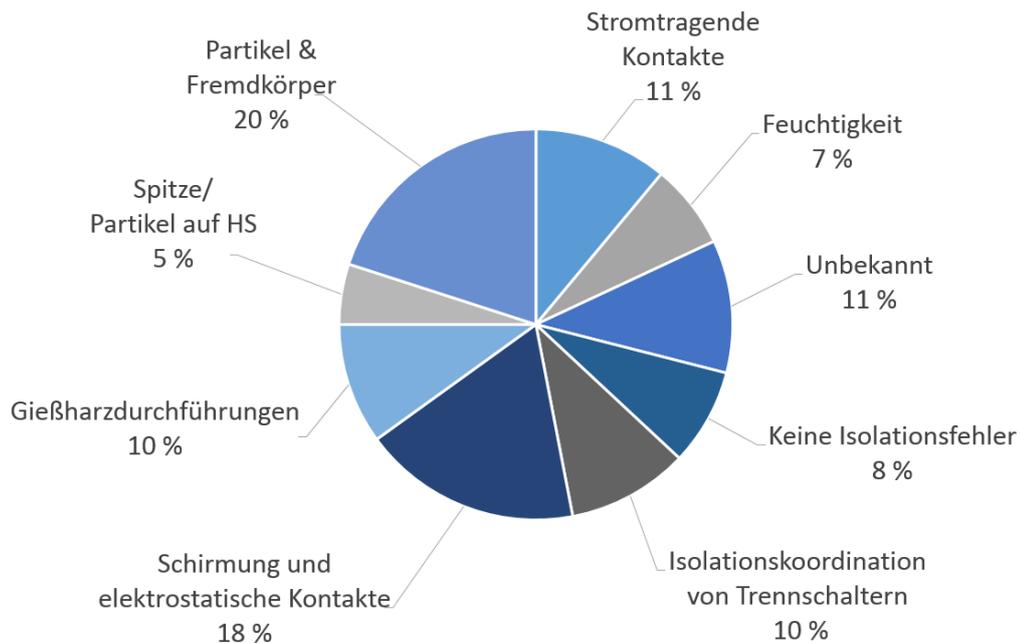


Bild 10: Fehlerursachen in GIS nach [20]

Durch die aufgenommene Ladung wird eine elektrische Kraft \vec{F}_Q auf das Partikel wirken, wodurch sich dieses beim Überwinden der Gravitationskraft und der Viskosität von SF_6 anfängt zu bewegen. Aus dem Coulomb'sches Gesetz

$$\vec{F} = Q \cdot |\vec{E}| \quad (2.26)$$

ergibt sich die auf das Partikel ausgeübte Kraft zu

$$\vec{F}_Q = \int_Q \vec{E} \cdot d\vec{q} . \quad (2.27)$$

Die Bewegung ergibt sich aus der Summe der auf das Partikel wirkenden Kräfte nach [22] zu

$$m \frac{d^2 \vec{r}}{dt^2} - \eta \frac{d\vec{r}}{dt} + q\vec{E} + m\vec{g} = 0 . \quad (2.28)$$

m...	Masse des Partikels	r...	Position des Partikels
η ...	Viskositätskoeffizient von SF_6	q...	Ladung des Partikels
\vec{E} ...	elektr. Feldstärke	g...	Gravitationskonstante

Die aufgenommene Ladung und dadurch die Abhebefeldstärke hängt stark von der Partikelgeometrie, der Form und den Abmessungen des Partikels, sowie von dessen Material ab. Die Abhebefeldstärke \vec{E}_L ist in Tabelle 4 für ausgewählte Partikel dargestellt. Wie zu erkennen ist, werden die Berechnungen für reale Geometrien, wie eines Spans, sehr viel aufwendiger. Wie weit das Partikel fliegt, ist u. a. abhängig von der Amplitude und Polarität der Gleichspannung sowie der Geometrie des Partikels.

Auch die Bewegung selbst ist recht komplex. Übersteigt die Kraft $q\vec{E}$ die Gravitationskraft $m\vec{g}$ fängt das Partikel an in Richtung Hochspannungselektrode abzuheben. Nach dem Ladungsaustausch an dieser, wird die Bewegung in Abwärtsrichtung fortgesetzt [22, 23].

Das Partikel kann jedoch auch frei schweben oder, was häufiger der Fall ist, eine Auf- und Abwärtsbewegung vollziehen, wie zuvor beschrieben. Kommt es in den feldschwachen Raum, ist die Chance, dass die Bewegung zum Stoppen kommt, recht groß. Driftet es aber in den feldstarken Raum und schließlich zum Innenleiter, kann es zum Gasdurchschlag kommen. Im günstigsten Fall wird das Partikel durch die Energie des Durchschlags zerstört. Während der Bewegung treten kleine Entladungen zwischen Partikel und den Elektroden auf, wobei elektrische und akustische Signale erzeugt werden. Diese können anhand von TE-Messungen und verschiedener Messmethoden (konventionell und nicht konventionell, wie akustisch und UHF) detektiert werden [23].

Tabelle 4: Elektrische Abhebefeldstärken verschiedener Partikelgeometrien [22]

Geometrie	elektrische Abhebefeldstärken
Kugel	$E_L = 0,49\sqrt{\frac{\rho g r}{\epsilon_0}}$
Zylinder (horizontal)	$E_L = \sqrt{\frac{\rho g r}{1,43\epsilon_0}}$
Zylinder (vertikal)	$E_L = (\ln(2L/r) - 1) \sqrt{\frac{2\rho r^2 g}{3\epsilon_0 L(\ln(L/r) - 0,5)}}$
Span (horizontal)	$E_L = d\sqrt{\frac{\rho g}{2,86\epsilon_0}} \sqrt{\left(\frac{\pi^2}{h}\right)^2 + \left(\frac{1}{D}\right)^2}$

ρ ...	Dichte des Partikels	d ...	Drahtdurchmesser der Spirale
L ...	Länge des Partikels	D ...	Durchmesser der Spirale
r ...	Radius des Partikels	h ...	Steigungswinkel der Spirale
g ...	Gravitationskonstante	ϵ_0 ...	Dielektrizitätskonstante

Firefly-Effekt: Der Firefly-Effekt ist ein Phänomen, welches bei Partikeln auftreten kann. Angenommen wird ein zylinderförmiges Partikel, eine positiv geladene Kapselung und ein negativ geladener Innenleiter. Ein positiv geladenes Partikel fliegt von der Kapselung in Richtung Innenleiter. Beim Kontakt mit diesem wird es seine Ladung an diesen abgeben und dabei negative Ladungen aufnehmen. Aufgrund der Geometrie des Partikels kommt es an der, der Kapselung zugewandten Seite zu einer Feldstärkenerhöhung, welche eine lokale Ionisation des SF_6 hervorruft. Die frei gewordenen Elektronen bewegen sich rasch zur äußeren positiven Kapselung. Aufgrund der negativen Polarität des Partikels wird auch dieses anfangen vom Innenleiter wegzudriften. Durch den unterbrochenen Kontakt werden keine negativen Ladungsträger mehr nachgeliefert. Die durch Ionisation entstandene positive Raumladungswolke, nach Wegdriften der Elektronen, ist nur träge und bewegt sich langsam in Richtung negativen Leiter. Kommt das negative Partikel in den Bereich

der Ladungswolke, kommt es wieder zu einem Ladungsaustausch, wodurch es positiv geladen wird. Dadurch kommt es abermals zu einer Bewegung in Richtung Leiter wie anfangs erklärt. Dieser Vorgang wiederholt sich immer wieder und wird Firefly-Effekt genannt. Es scheint als würde das Partikel auf der Innenelektrode „tanzen“. Ob und wie oft dieser Effekt in Erscheinung tritt, hängt davon ab wie das Partikel auf die Innenelektrode trifft, somit von der Kontaktstelle. Auch können Leuchterscheinungen aufgrund des Ladungstausches beobachtet werden.

Firefly-Effekt ist auch bei positivem Innenleiter zu finden und geht prinzipiell gleich vonstatten. Hier wird die träge Raumladungswolke durch Anlagerung von freien Elektronen an den SF₆-Molekülen erzeugt. Dieser Effekt tritt in SF₆ bei negativer und positiver Innenelektrode auf, wobei er aber bei negativer häufiger auftritt [24].

Partikelfallen: Trotz Vorsichtsmaßnahmen zur Vermeidung von leitfähigen Partikeln im Isoliersystem, können diese nie zur Gänze ausgeschlossen werden. Um die Auswirkung solcher Fehler zu verringern, werden in heutigen GIS-Anlagen sogenannte Partikelfallen angebracht. Es handelt sich dabei um feldschwache Räume, in denen sich Partikel absetzen und zum Liegen kommen. Dazu werden an der Hochspannungselektrode Schirmringe befestigt, welche lokal die Feldstärke soweit herabsetzen, wodurch Bewegungen und Firefly-Effekte gestoppt werden. Dadurch kommen die Partikel im feldschwachen Raum zum Erliegen und bleiben dort. Eine weitere Maßnahme ist die dielektrische Beschichtung der Außenleiter, um Aufladungen des Partikels zu vermeiden bzw. zu erschweren [25, 26].

3 Software zur Verarbeitung von TE-Impulsen bei Gleichspannung

3.1 Einleitung

Teilentladungsmessungen zählen nach aktuellem Stand der Technik zu einem der wichtigsten Diagnoseverfahren von Hochspannungsbetriebsmitteln. Da die TE-Messung eine sogenannte zerstörungsfreie Messung ist, d. h. dass der Prüfling während der Messung keine besonderen Schäden davonträgt, wird sie oft anderen Diagnoseverfahren vorgezogen. In den Anfängen der TE-Diagnostik musste noch auf konventionelle, analoge Messgeräte, wie Elektrodenstrahloszilloskope oder Schering-Messbrücken, zurückgegriffen werden (Bild 11). Mit Hilfe der digitalen Messtechnik und immer größer werdenden Rechenleistungen, stehen moderne, speziell für Teilentladungsmessungen entwickelte Messsysteme zur Verfügung. Durch Softwarelösungen gibt es viele Einstellungsmöglichkeiten, gute Störunterdrückungen und es kann eine digitale (Nach-)Bearbeitung der Messdaten vorgenommen werden [3].

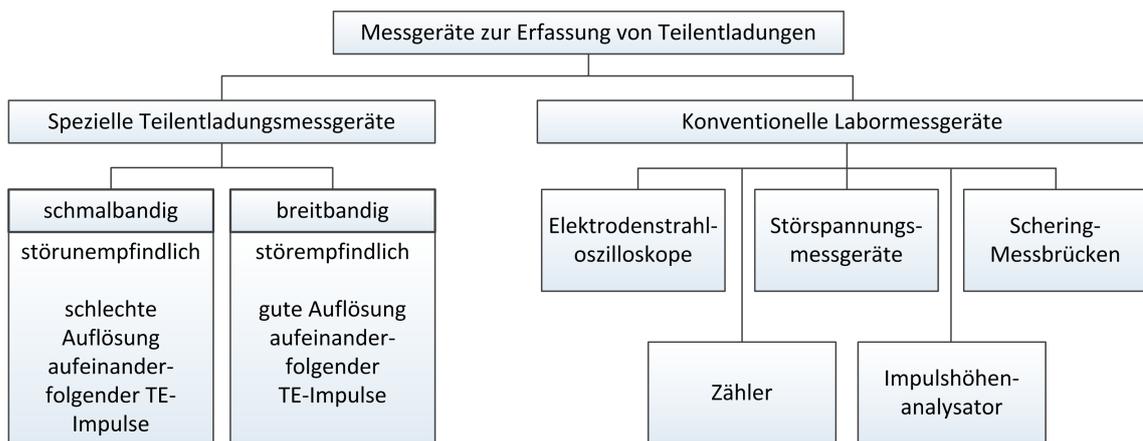


Bild 11: Messgeräte zur Erfassung von Teilentladungen nach [3]

Im Rahmen dieser Arbeit wurde, wie bereits oben erwähnt, eine Software zur Analyse von Teilentladungsmessdaten bei Messungen mit Gleichspannung entwickelt. Diese Software stellt die Schnittstelle zwischen der *OMICRON Software for MPD and MI* der Firma Omicron (im Folgenden auch kurz Omicron-Software genannt) und eines am Institut für Hochspannungstechnik und Systemmanagement der TU Graz im Rahmen einer Bachelorarbeit entwickelten Programms zur Nachbearbeitung und Analyse von TE-Messdaten (*PDDC-Analyzer*), dar (Bild 12). Hierbei sei angemerkt, dass diese Software seit der Bachelorarbeit stark weiterentwickelt wurde. Mit dieser ist es jedoch **nur nach einer Messung** möglich, Messdaten einzulesen und zu analysieren. Mit der neu entwickelten Software namens *Online PDDC-Analyzer* ist es nun auch möglich, Messdaten **während einer Messung** in Echtzeit in den für Gleichspannung favorisierten Diagrammen darzustellen und zu analysieren. Nachfolgend werden die einzelnen Programme kurz beschrieben, wobei das Hauptaugenmerk auf den *Online PDDC-Analyzer* gelegt wurde.

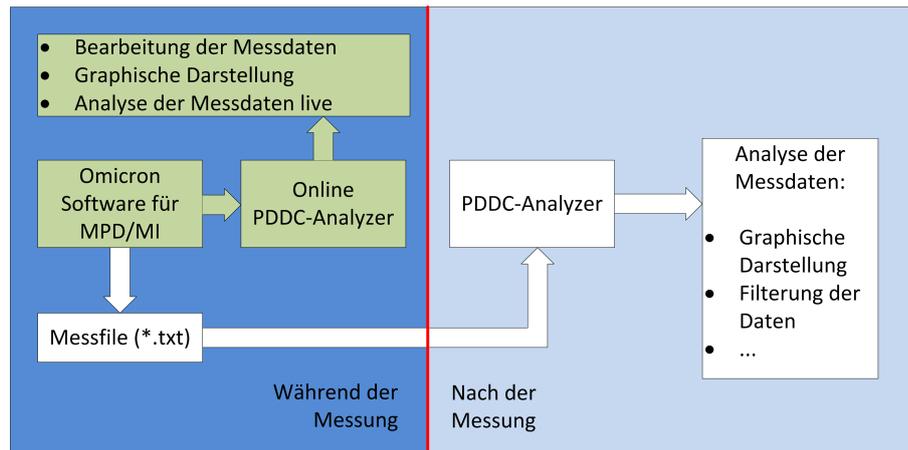


Bild 12: Vergleich zwischen *Online PDDC-Analyzer* (grüner Pfad) und *PDDC-Analyzer* (weißer Pfad)

3.2 OMICRON Software for MDP and MI

3.2.1 TE-Messsystem der Firma Omicron

Das Teilentladungsmesssystem der Firma Omicron besteht aus Soft- sowie Hardware. Dieses teilt sich hardwareseitig in Ankoppelvierpol, Messeinheit MPD und Kontrolleinheit MCU auf, sowie softwaretechnisch auf die *OMICRON Software for MPD and MI* auf. Die Auskopplung der TE-Impulse erfolgt über den Ankoppelvierpol CPL 542, die Digitalisierung und Filterung der Messdaten in der Messeinheit MPD 600 und die Übertragung an einen PC mit der Software und der Kontrolleinheit MCU über Lichtwellenleiter, um Störungen während der Übertragung zu vermeiden. Die jeweiligen Einheiten und das Anschlussschema sind in Bild 13 dargestellt, angelehnt an den bereits bekannten Prüfkreis nach IEC 60270. In Bild 14 ist die Softwareoberfläche dargestellt, mit den sogenannten *Scope*-Ansichten links und den Einstellungen rechts. Im folgenden Kapitel wird die Software für TE-Messungen bei Gleichspannung und die darin benötigten Einstellungen beschrieben.

3.2.2 Software-Einstellungen

Die Software der Firma Omicron bietet im sogenannten *Expert-Mode* sehr viele Einstellungsmöglichkeiten. Hier sollen nur die für TE-Messungen bei Gleichspannung relevanten Einstellungen diskutiert werden, für alle anderen Einstellungen und Möglichkeiten der Software sei an dieser Stelle auf das Benutzerhandbuch verwiesen [12].

Zunächst ist es wichtig, sich mit dem Aufbau der Software vertraut zu machen. Links befinden sich die sogenannten *Scope*-Ansichten, welche sich in große *Scope*-Ansicht (Bild 14 rechts oben) und kleine *Scope*-Ansicht (Bild 14 rechts unten) aufteilen. Hier werden die Teilentladungsimpulse auf verschiedenste Weise dargestellt, wie z. B. das PRPD-Pattern, die Ellipsen-Darstellung und diverse DC-Diagramme. Rechts davon befinden sich die Bedienfelder mit allen relevanten Einstellungen, die getätigt werden können. Die in dieser Arbeit als am wichtigsten betrachteten Registerblätter sind *V-Tab*, *PD-Tab*, *Q-Tab* und *DC-Tab*. Die meisten dieser Einstellungen werden vom

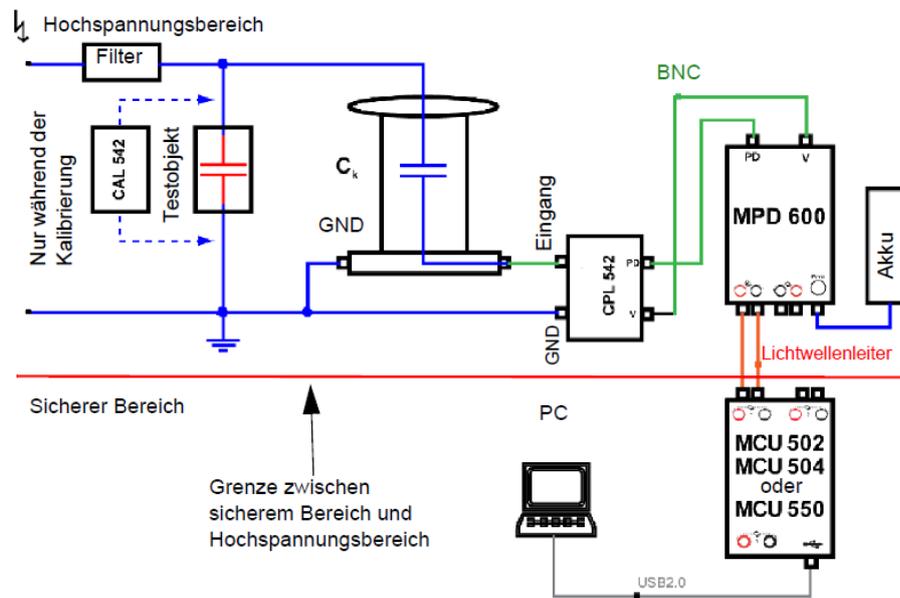


Bild 13: Anschluss der Messeinheiten an einen AC-Prüfkreis [12]

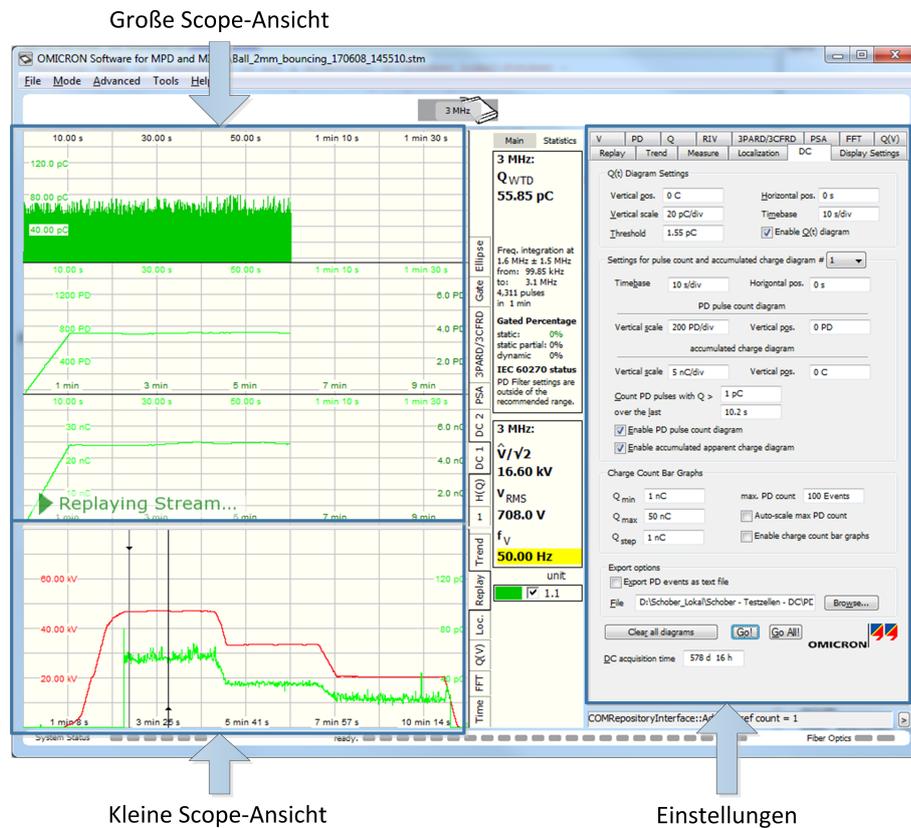


Bild 14: Softwareoberfläche der *Omicron Software for MPD and MI*

Online PDDC-Analyzer automatisch gesetzt, wie in Kapitel 3.4 beschrieben:

- a) **V-Tab:** Hier befinden sich Einstellungen für die Spannung. Einerseits können Displayeinstellungen zur Darstellung der Spannung, Einstellungen zur Triggierung der Spannung und andererseits die Kalibrierung vorgenommen werden. Für DC-Messungen muss die **Dominant Voltage** auf **DC** eingestellt, sowie vor jeder Messung eine **Kalibrierung** durchgeführt werden. Dafür wird die Spannung auf ein praktikables Niveau erhöht, am Prüfling gemessen und dieser Wert im Feld **V_{DC}(target value)** eingetragen, wie im Benutzerhandbuch des MPD 600 beschrieben wird [12]. Der *Divider Factor* wird nach einem Klick auf *Compute* von der Software selbst berechnet.
- b) **PD-Tab:** Dieses Registerblatt steuert die Erfassung und Anzeige der am PD-Eingang des MPD aufgenommenen TE-Impulse. Relevant für DC-Messung ist hier lediglich die **Verstärkereinstellung (Gain)**. Da es bei Gleichspannung vorkommt, dass die Impulswiederholrate nur sehr gering im Vergleich zu Prüfungen mit Wechselspannung ist und dadurch Impulse nicht kontinuierlich auftreten, sollte **kein Auto Gain** verwendet werden, um ständiges Umschalten der Verstärkung beim Auftreten von einzelnen TE-Impulsen zu vermeiden. Somit muss die Verstärkung selbst geregelt werden. Als Anzeige, ob die aktuelle Verstärkung richtig gewählt wurde oder der Eingang übersteuert wird, dient die Level-Skala. Der *Gain* sollte händisch so gesetzt werden, dass die **Level-Anzeige grün** ist und somit die erwarteten TE-Impulse innerhalb des Messbereichs liegen und der Eingang des Messgeräts nicht übersteuert.
- c) **Q-Tab:** Dieser Tab enthält, vom *DC-Tab* abgesehen, die wichtigsten Einstellungen für Teilentladungsmessungen. Es können Einstellungen zur Anzeige und Detektion sowie Auswertungen gemacht werden. Zur Detektion der Impulse können die Filtereinstellungen, die TE-Triggerschwelle (TE-Threshold) und die Totzeit (dead time) eingestellt werden. Der **TE-Threshold sollte über dem Level des Grundrauschens** gewählt werden, die **dead time** abhängig von den Filtereinstellungen und der Impulsform. Darüber hinaus kann hier die TE-Kalibrierung vorgenommen werden. Zur Kalibrierung sei auf die Norm IEC 60270 verwiesen [1]. Die Kalibration muss parallel an den Anschlüssen des Prüflings erfolgen. Der am Kalibrator eingestellte Wert wird in **Q_{IEC}(target)** eingetragen und mit *Compute* der Q_{IEC}-Wert kalibriert. Unter anderem befindet sich hier auch der Button für die Streamaufzeichnung.
- d) **DC-Tab:** enthält alle relevanten Einstellungen für die DC-Diagramme. Die Messung kann mit **Klick auf Go** gestartet, sowie die Speicherung eines Textfiles mit den Messdaten in den **Export options** eingestellt werden. Die Einstellungen sind größtenteils selbst erklärend und steuern die Achsenskalierungen und Ausgaben in den DC-Diagrammen, welche nachfolgend beschrieben werden.

3.2.3 Darstellung der Teilentladungen

Wie bereits in Kapitel 2.3 erklärt, kann die phasenaufgelöste Darstellung der Teilentladungen bei Gleichspannung, aufgrund des fehlenden Phasenbezugs, nicht verwendet werden. Die anderen Kenngrößen eines TE-Impulses müssen somit herangezogen

werden. In der Omicron-Software stehen insgesamt fünf verschiedene Diagramme zur Bewertung der Messdaten zur Verfügung. Diese befinden sich in den bereits erwähnten sogenannten großen *Scope*-Ansichten in den Reitern bzw. Ansichten DC1 und DC2 [12].

DC1: Zeitbasierte Diagramme

Q(t)-Diagramm stellt die Amplituden der TE-Impulse über die Zeit dar, wobei nur jene Impulse eingezeichnet werden, die oberhalb eines definierten, einstellbaren Pegels liegen. Die beiden Achsen können im Tab *DC* eingestellt werden. Tritt ein Impuls auf, wird dieser nicht als Punkt, sondern, abhängig von der Skalierung der Zeitachse, als senkrechte Linie dargestellt (Bild 15a oben).

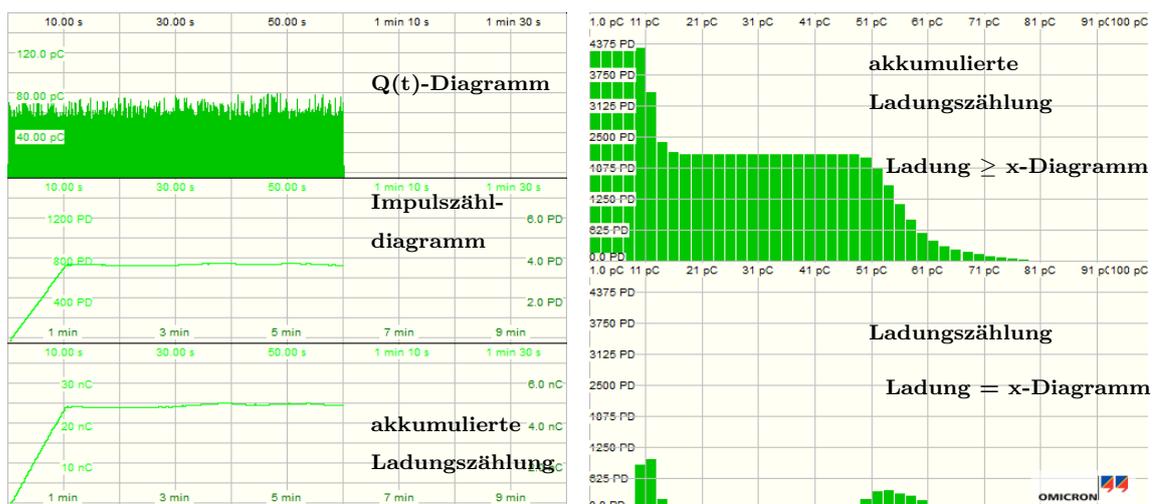
Impulszähl-diagramm zeigt an, wie viele Impulse innerhalb eines eingestellten Zeitfensters oberhalb eines definierten Pegels liegen. Diese Darstellungsart ist vorteilhaft bei der Ermittlung der TE-Einsatzspannung (z. B. drei Impulse größer 1 pC innerhalb von 30 s). Es lässt sich ebenfalls die TE-Aktivität über die Messdauer betrachten (Bild 15a mittig).

Akkumulierte Ladungszählung zeigt die akkumulierten Ladungswerte der Impulse im gleichen Zeitfenster wie zuvor, wenn die Impulse oberhalb des zuvor genannten Grenzwertes liegen (Bild 15a unten).

DC2: Balkendiagramme

Ladungszählung zeigt die Anzahl der Ladungen, welche innerhalb eines bestimmten Ladungsbereichs liegen (Bild 15b unten).

akkumulierte Ladungszählung zeigt die akkumulierte Anzahl der Ladungen, d. h. die Anzahl der Ladungen, welche innerhalb oder über dem auf der x-Achse dargestellten Ladungsbereich liegen (Bild 15b oben).



(a) Ansicht DC1 – Zeitbasierte Diagramme

(b) Ansicht DC2 – Balkendiagramme

Bild 15: Große *Scope*-Ansichten der beiden DC-Ansichten in der Omicron-Software

Obwohl fünf Diagramme zur Interpretation der Teilentladungen bei Gleichspannung zur Verfügung stehen, wurden diese als nicht ausreichend betrachtet, um aufgrund dieser Darstellungen auf den vorliegenden Defekt schließen zu können. Deshalb wurden für den *Online PDDC-Analyzer* weitere Darstellungsarten gewählt, welche bereits im *PDDC-Analyzer* zum Einsatz kommen.

Ein weiterer Nachteil ist es, dass die Messzeit bei DC-Messungen nur anhand der Diagramme ersichtlich ist. Treten keine Teilentladungen auf, so ist es nur schwer möglich auf die aktuelle Messdauer zu schließen. Da die Achsenskalierungen selbst eingestellt werden müssen, kommt dies beim Abschätzen der bereits gemessenen Zeit erschwerend hinzu. Die Teilentladungsimpulse werden im $Q(t)$ -Diagramm als senkrechter Strich über die dazugehörige Zeit eingezeichnet. Dadurch können TE-Impulse kleiner Amplitude, bei gleichzeitig vielen großen Impulsen, nicht mehr im Diagramm erkannt werden, vor allem, wenn die Zeitachse nicht ausreichend klein skaliert wurde. Auch die Polarität der Teilentladungen ist aus dem vorhandenen $Q(t)$ -Diagramm nicht ersichtlich, da es sich um eine unipolare Darstellung handelt. In Bild 15a ist, obwohl 10 Sekunden pro Division eingestellt wurden, nicht zu erkennen, ob auch Teilentladungen mit kleinerem Ladungswert als ca. 70 pC aufgetreten sind oder mit welcher Polarität diese gemessen wurden. In Bild 16 ist ein Vergleich zwischen *Online PDDC-Analyzers* und Omicron-Software visualisiert. In der Ansicht oben ist das $Q(t)$ -Diagramm des *Online PDDC-Analyzers* dargestellt und darunter das Diagramm der Omicron-Software. Wie zu erkennen ist, treten auch Teilentladungen unterhalb von 70 pC auf. Ersichtlich ist mit der modifizierten Darstellung auch die Polarität der Impulse.

Weitere Nachteile ergeben sich bei der Erstellung und Auswertung von Messfiles bei DC-Spannung. Einerseits enthalten die Dateinamen keinen eindeutigen Stempel wie Datums- und Uhrzeitangaben. Dadurch kann es passieren, dass ein bereits vorhandenes Textfile überschrieben wird. Andererseits enthalten diese Textfiles Informationen über den Ladungswert eines Impulses, den Zeitpunkt und die Spannung, wobei die Daten nur beim Auftreten von TE-Impulsen in das Textfile geschrieben werden. Eine Information der Spannung zwischen den TE-Impulsen ist nicht vorhanden, was vor allem problematisch sein kann, wenn die einzelnen Impulse zeitlich weit auseinander liegen. Der *Online PDDC-Analyzer* wurde unter Berücksichtigung dieser Nachteile entwickelt und es wurde versucht diese Schwächen zu beseitigen.

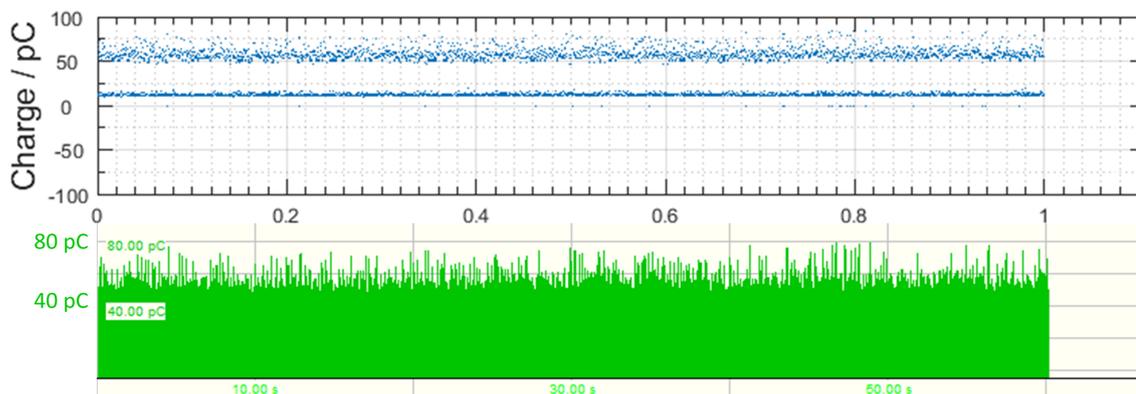


Bild 16: Vergleich des $Q(t)$ -Diagramms der beiden Programme *Online PDDC-Analyzer* (oben) und Omicron-Software (unten)

3.3 PDDC-Analyzer

Der *PDDC-Analyzer* wurde im Rahmen einer Bachelorarbeit am Institut für Hochspannungstechnik und Systemmanagement an der TU Graz entwickelt und fortlaufend verbessert. Der Autor dieser Arbeit hat an der Weiterentwicklung des Programms mitgewirkt und auch neue Inhalte (Features) dazu beigetragen. Dies ermöglichte einen genaueren Einblick in die Software und stellte die Grundlage für den in Kapitel 3.4 beschriebenen *Online PDDC-Analyzer* dar.

Wie bereits im einleitenden Kapitel 3.1 ausgeführt, ist der *PDDC-Analyzer* als „post-processing“ Programm ausgelegt, d. h. dass die analytische Auswertung und graphische Darstellung der Messdaten erst nach (lat. post) einer Messung erfolgt. Dazu können die Messdaten in verschiedenen Diagrammen ausgegeben und miteinander verglichen werden. Einige der Diagramme sind bereits in der zuvor beschriebenen Software von Omicron enthalten, wie das $Q(t)$ -Diagramm und ein Balkendiagramm. Für eine genauere Beschreibung der einzelnen Diagramme sei auf die Bachelorarbeit verwiesen [2]. Erwähnt seien hier lediglich die Time-Plots ($Q(t)$ -, $\Delta t(t)$ - und $V(t)$ -Diagramme) und NoDi*-Diagramme, da diese aufgrund von Erfahrungswerten als besonders wichtig zur Interpretation von Teilentladungen erachtet und im *Online PDDC-Analyzer* integriert wurden.

3.4 Online PDDC-Analyzer

3.4.1 COM-Schnittstelle der OMICRON Software for MPD and MI

Die **C**omponent **O**bject **M**odel (COM) Schnittstelle ist eine 1992 von Microsoft entwickelte und eingeführte Schnittstelle und dient zur Kommunikation zwischen zwei Prozessen.

„COM is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft’s OLE (compound documents) and ActiveX (Internet-enabled components) technologies.“ [27]

Sie dient der Interprozesskommunikation in vielen bekannten Programmiersprachen, darunter auch C++, und ist plattform- und sprachenunabhängig. Die Kommunikation erfolgt dabei nach dem Client-Server-Prinzip, wobei ein Server COM-Komponenten anbietet, ein Client ein Objekt einer solchen COM-Komponente erzeugt und die Funktionen selbiger benutzt. Der Server kann dabei eine dynamische Programmbibliothek (engl. dynamic link library, kurz .dll) oder ein ausführbares Programm in Form einer .exe-Datei darstellen. Letztgenannte werden als *Local Server* bezeichnet. Die Kommunikation selbst erfolgt über die COM-Schnittstelle, welche vorgegebene Interfaces wie IUnknown und optional auch spezielle Interfaces (vgl. *IRepository2* unten) besitzt [28].

Die *OMICRON Software for MPD and MI* bietet ein solches COM-Interface, das bedeutet eine Schnittstelle über die andere Programme mit der Software respektive Hardware kommunizieren können. Im Falle von Omicron handelt es sich um einen *Local Server*, wie oben beschrieben, was bedeutet, dass bei der Erzeugung einer entsprechenden COM-Komponente das Programm gestartet wird, sofern es nicht

bereits zuvor ausgeführt wurde. Um die COM-Schnittstelle der MCU verwenden zu können, benötigt es eine speziell dafür vorgesehene Lizenz von Omicron. Zur Verwendung der Schnittstelle muss diese Lizenz auf der verwendeten Kontrolleinheit MCU vorhanden sein bzw. aktualisiert werden [29]. Wird versucht auf Elemente der Software über die Schnittstelle zuzugreifen und ist die Lizenz nicht vorhanden, dann kommt es zu einer Fehlermeldung.

Die Verbindung über die COM-Schnittstelle ist eine bidirektionale Verbindung, das bedeutet, es können Daten, z. B. Messdaten, von der Software gelesen, gleichzeitig aber auch Daten an das Messsystem übertragen werden, z. B. zum Setzen diverser Einstellungen. Grundsätzlich lässt sich auf nahezu jedes Element, welches auch in der Benutzeroberfläche der Software zu finden ist, zugreifen. Je nach Berechtigung für das jeweilige Element kann dessen Wert nur gelesen oder auch neu gesetzt werden. Bei reinen Messanzeigen macht es keinen Sinn deren Wert überschreiben zu wollen, wohingegen es bei Messeinstellungen zweckmäßig ist, diese steuern und verändern zu können.

Die COM-Schnittstelle besteht aus vier Interfaces, zwei davon welche bereits in der Software implementiert sind (*IRepository* und *IRepository2*) und zwei weitere die von einem anderen Programm implementiert werden können (*IConsumer* und *IConsumer2*). Bei *IRepository* und *IConsumer* handelt es sich um alte Versionen, welche von den jeweiligen Interfaces mit der Nummer 2 abgelöst wurden. Die alten Versionen sollten bei neu geschriebenem Code nicht mehr verwendet werden.

Es gibt eine Vielzahl an Variablen, auf die zugegriffen werden kann. Nahezu jedes Bedienelement der Software kann mit mindestens einer Variable in Verbindung gebracht werden. Um Programmierern die Verwendung der COM-Schnittstelle zu erleichtern, gibt es eine Dokumentation aller Variablen [30]. Zu jeder Variable werden wichtige Informationen, wie der Name der Variable, der Typ, der Wertebereich usw. angegeben. Ein Beispiel für die Auflistung der Informationen, wie sie in der Dokumentation verwendet wird, ist in Tabelle 5 zu sehen.

Die Informationen einer Variable lauten wie folgt:

- **Name der Variable** über den anhand der COM-Schnittstelle auf die Variable zugegriffen werden kann. Der Name ist *case-sensitive*, d. h. das Groß- bzw. Kleinschreibung beachtet werden muss.
- **Qualifier** wird benötigt um die Variable eindeutig zu halten, da es vorkommen kann, dass es zwei Variablen mit dem selben Namen gibt, z. B. wenn mehrere MPD an eine MCU angeschlossen werden.
- **Einheit** gibt an, mit welcher Einheit der Wert in Verbindung steht.
- **Variablentyp** definiert den Typ einer Variable, wie z. B. *string* oder *double*.
- **Verwendungstyp** gibt an wie auf die Variable zugegriffen werden kann. „I“ und „M“ sind *read-only*, das bedeutet auf diese Werte kann nur lesend zugegriffen werden, während „S“ und „U“ auch Schreibzugriff ermöglichen.
- **Standard-, Minimal- und Maximalwert** geben an, in welchem Wertebereich der Wert liegen kann.

- **Anzeige-Genauigkeit** beschreibt mit wie vielen signifikanten Stellen ein Wert in der Software dargestellt wird.
- **Lebensdauer-Flag** bestimmt, ob der Wert einer Variablen nach dem Neustart des Programms Bestand hat oder zurückgesetzt wird.
- **Einheitenpräfix-Flag** gibt an, ob der Wert mit einem Präfix wie kilo oder pico angezeigt wird.
- **Version** gibt Information darüber, ab welcher Softwareversion die Variable existiert.

Tabelle 5: Variablendokumentation – allgemein (oben) und am Beispiel für den Q_{IEC} -Wert (unten) [30]

Name der Variable		Qualifier		Einheit	Variablentyp	Verwendungstyp
Standardwert	Minimalwert	Maximalwert	Anzeige-genauigkeit	Lebensdauer-Flag	Einheitenpräfix-Flag	Version
Beschreibung und Verwendung der Variable						

qIEC		stationX.		C	double	M
0	-1.00E+302	1.00E+302	4	no	yes	
<p>For the unit identified by the qualifier, this variable is set to the IEC 60270-compliant apparent charge value. The voltageRun. qualifier is valid only during a voltage run that being is performed in cable mode (uiMode = 2), and in that case corresponds to the currently selected unit.</p> <p>Note: Since version 1.6 of the software measurements that are performed outside of the parameters defined by the IEC 60270 are no longer displayed as QIEC but as QWTD (for "weighted"). However those values are still saved in the qIec variable (there is no "qWtd"variable).</p>						

Um auf die einzelnen Software-Elemente zugreifen zu können, benötigt es eine Programmierumgebung, sowie eine geeignete Programmiersprache. Wie zuvor erklärt, befinden sich alle bereits implementierten Funktionen der COM-Schnittstelle im sogenannten *IRepository2*-Interface. Dieses beinhaltet unter anderem vorgefertigte Funktionen, um Variablen zu manipulieren. Die wichtigsten davon sind `getHandle`, `getVariant` sowie `setVariant` [31].

Wie diese Funktionen angewendet werden können, wozu diese verwendet und welche Parameter übergeben werden müssen zeigt nachfolgendes Listing 1. `getVariant` und `setVariant` benötigen als Übergabeparameter jeweils das sogenannte `handle` einer Variable, welche die Variable eindeutig identifiziert. Dieses `handle` wird mit der Funktion `getHandle` abgerufen. Die beiden anderen Funktionen werden verwendet um Werte einer Variable zu setzen oder abzurufen.

Listing 1: Prototypen der Funktionen `getHandle`, `getVariant` und `setVariant` [31]

<pre> getHandle DWORD mtronixApp.getHandle(const char *qualifiedVar, const char *domain); // qualifiedVar: den zuvor genannten Qualifier und der Name der // Variable, z. B. "station1.qIEC". // domain: Domain, welcher die Variable zugeordnet ist. // z. B. "MPD540". // Beschreibung: Diese Funktion liefert das Handle einer Variable // als 32-bit Integer ohne Vorzeichen zurück, welches für andere // Funktionen benötigt wird.</pre>
<pre> getVariant VARIANT mtronixApp.getVariant(const DWORD handle); // handle: Rückgabewert der Funktion getHandle. // Beschreibung: Liefert den aktuellen Wert, "verpackt" in einem // Variant, zu der in Handle definierten Variable zurück.</pre>
<pre> setVariant void mtronixApp.setVariant(const DWORD handle, const VARIANT value); // handle: Rückgabewert der Funktion getHandle. // value: Wert auf den die Variable mit dem Handle handle gesetzt // werden soll, "verpackt" als Variant. // Beschreibung: Setzt den Wert, "verpackt" in einem Variant, der // in handle definierten Variable.</pre>

Es stehen verschiedene Programmiersprachen zur Verfügung, in denen das COM-Interface eingebunden werden kann. Eine Beschreibung wie das Interface eingebunden wird und kleine Programmierbeispiele für bekannte Sprachen, wie

- VisualBasic 6 und Microsoft Office VBA,
- Microsoft .Net und
- C++

sind in [31] zu finden. In dieser Arbeit wurde die Sprache C++ gewählt, da es sich um eine hardwarenahe und dadurch im Vergleich zu anderen höheren Programmiersprachen (sogenannte *high-level programming languages*), um eine relativ schnelle Sprache handelt.

Nachfolgend ist ein Beispiel eines Programms zu finden (Listing 2), welches in C++ geschrieben wurde. Es soll dieses Kapitel zusammenfassen und die Anwendung der COM-Schnittstelle verdeutlichen. Das Programm liest den Q_{IEC} -Wert aus der Omicron-Software aus und schreibt ihn anschließend in ein Konsolenfenster, welches

in Bild 17 zu sehen ist. Verwendet wurde dazu ein Streamfile (.stm) einer zuvor durchgeführten Messung eines *floating potentials*, welches im Bereich um $q = 25$ nC gemessen wurde.

Listing 2: Programm zum Lesen des Q_{IEC} -Wertes einer MPD und Ausgabe in der Console mit C++

```
1 // Import-Direktive zum Einbinden des COM-Interfaces
2 #import "C:\\Program Files\\OMICRON\\Software for MPD and MI\\
   mpd_mi.exe" no_namespace
3
4 int main(int argc, char *argv[])
5 {
6     // Pointer auf das Interface IRepository2
7     IRepository2Ptr rep3(__uuidof(mtronixMeasurementApp));
8
9     // Deklaration eines Varianten mit Typ double (VT_R8)
10    VARIANT qiec;
11    qiec.vt = VT_R8;
12
13    // Lesen der Variable "qIec" der ersten MPD am Channel 1 der
       MCU (station1) in der Domain "MPD540"
14    qiec = rep3->getVariant(rep3->getHandle("station1.qIec", "MPD540
       "));
15
16    // Ausgabe des Werts in pC in der Konsole
17    printf("Q-IEC: %g pC", (qiec.dblVal)*(1e12));
18 }
```

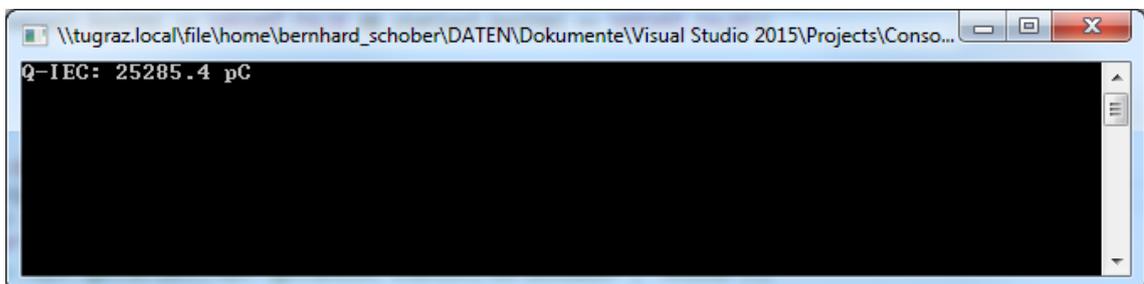


Bild 17: Konsolenausgabe von Listing 2

3.4.2 Graphische Benutzeroberfläche

In der heutigen Zeit ist es nicht mehr zeitgemäß nur anhand von Konsoleneingaben Programme und Maschinen zu steuern, sondern möglichst viel Feedback zurückzugeben. Moderne, benutzerfreundliche Programme sollten nach Möglichkeit über eine graphische Benutzeroberfläche, eine sogenannte GUI (engl. **G**raphical **U**ser **I**nterface) verfügen, welche verständlich, einfach gehalten und leicht zu bedienen ist. Im vorherigen Kapitel wurde beschrieben, wie mit Hilfe einer geeigneten Programmiersprache die COM-Schnittstelle der Omicron-Software verwendet werden kann. C++ bietet in seiner Grundform keine Bedienelemente oder Benutzeroberflächen, lediglich ein Konsolenfenster als Mensch-Maschine-Schnittstelle. Dadurch benötigte

es eine weitere Programmiersprache, welche genau diese Anforderung erfüllt und bei welcher es zusätzlich möglich ist, relativ einfach graphische Oberflächen zu erstellen. Außerdem sollte diese Sprache in der Lage sein, C++-Code lesen bzw. interpretieren zu können. Die Wahl fiel dabei auf Matlab, was mehrere Gründe hatte.

Einerseits wurde der zuvor beschriebene *PDDC-Analyzer* bereits in Matlab realisiert und da dieses neue Programm darin implementiert werden sollte, wurde Matlab als die einfachste Lösung angesehen. Andererseits wurde der *Online PDDC-Analyzer* als Messprogramm ausgelegt, was bedeutet, dass Messdaten nicht nur angezeigt, sondern auch analysiert und Berechnungen mit diesen durchgeführt werden müssen. Matlab stellt in dieser Hinsicht ein leistungsfähiges Tool dar, hat jedoch auch seine Grenzen und ist vorrangig nicht als Anzeige von Echtzeitdaten ausgelegt, wie sich im Laufe dieser Arbeit gezeigt hat. Ein weiteres Kriterium war, dass es möglich sein sollte C++-Code zu implementieren. Matlab bietet hierfür sogenannte MEX-Funktionen (**M**ATLAB **e**xecutable) [32]. MEX-Funktionen werden in Kapitel 3.4.4 näher behandelt. Ein weiteres Kriterium, weshalb Matlab als weitere Programmiersprache gewählt wurde, ist, dass im Gegensatz zu C++, Matlab die Möglichkeit bietet eine GUI erstellen zu können. In Matlab können einfache, graphische Benutzeroberflächen recht schnell erstellt und anschließend programmiert werden. Komplexere GUI hingegen, wie sie im Rahmen dieser Arbeit erstellt wurden, erfordern einiges mehr an Vorkenntnissen und Zeit bei der Erstellung.

Im Vorfeld musste geklärt werden, wie das Programm grundsätzlich aussehen soll. Die Grundidee bestand darin, zwei GUI zu erstellen, eine welche die Verbindung zur Omicron-Software steuert und eine weitere, in der die Messdaten ausgegeben werden. Der Grundgedanke einer graphischen Oberfläche ist es, die Bedienung zu erleichtern und intuitiv zu gestalten, ohne lange Einarbeitungszeit. Dieser Gedanke wurde bei der Erstellung und im Hinblick auf Benutzerfreundlichkeit verfolgt, wobei es aus Sicht des Entwicklers schwer sein kann objektiv zu beurteilen. Vieles erscheint aus Sicht des Programmierers leicht und selbsterklärend. Diese Aussage kann jedoch nicht ohne weiteres auch auf Benutzer umgelegt werden.

In Matlab können graphische Oberflächen auf mehrere Arten erzeugt werden. Einerseits steht die Möglichkeit zur Verfügung die GUI, alle Elemente darin, deren Positionen in der GUI, Callbacks usw. selbst zu definieren und programmieren (programmatischer Ansatz). Andererseits kann auch das in Matlab integrierte Tool namens *GUIDE* (**G**raphical **U**ser **I**nterface **D**esign **E**nvironment) zur Erstellung von GUI verwendet werden. Dieses Tool bietet nicht ganz so viele Möglichkeiten wie die programmiertechnische Lösung, jedoch können die beiden Varianten kombiniert werden, was in dieser Arbeit gemacht wurde. Als weitere Variante sei der sogenannte *App Designer* genannt, welcher in Version R2016a von Matlab eingeführt wurde. Dieser wurde jedoch aufgrund seiner erst kürzlichen Einführung und aufgrund von fehlender Abwärtskompatibilität zu alten Versionen außer Acht gelassen. Der grundlegende Aufbau und das Aussehen der Oberflächen wurde mit dem Tool *GUIDE* erstellt und festgelegt. Alle Funktionen, die dadurch nicht abgedeckt werden konnten, wie z. B. die Implementierung eines *Gain*-Sliders zur Veränderung des Verstärkerlevels, wurden programmatisch realisiert und hinzugefügt. Hierzu musste teils auf die Sprache JAVA und deren Elemente zurückgegriffen werden, da Matlab als Grundgerüst auf dieser Programmiersprache aufbaut [33].

In Bild 18 ist die *GUIDE*-Oberfläche der Main-GUI visualisiert. In der Liste links befinden sich vordefinierte Bedienelemente, wie verschiedene Arten von Buttons, Tabellen, Checkboxes usw. Es können auch Menü- und Toolbars hinzugefügt und mit eigenen Elementen versehen werden. Ist die GUI mit den vorhandenen Elementen erstellt, muss diese abgespeichert und ausgeführt werden. Matlab erzeugt dabei selbstständig ein sogenanntes *.m*-File. Darin enthalten sind die Hauptfunktionen der GUI, welche beim Öffnen und Schließen aufgerufen werden, sowie Callbacks zu allen Bedienelementen. Bei Callbacks handelt es sich um herkömmliche Funktionen, die beim Betätigen oder Verändern eines Elements der GUI, wie das Drücken eines Buttons, ausgeführt werden. Es können auch komplexe Callbacks programmiert werden, z. B. ein Callback sobald sich die Maus über einem Element befindet. Das erzeugte *.m*-File hat bereits eine vorgegebene Struktur, die bei Änderungen eingehalten werden muss. Sollten andere Bausteine, als die in *GUIDE* angebotenen, verwendet werden, so müssen diese programmiert werden. Der entsprechende Code wurde dabei in die *OpeningFcn*-Funktion geschrieben. Es handelt sich um jene Funktion, die bei jedem Programmstart ausgeführt wird, noch bevor die graphische Oberfläche am Bildschirm erscheint. Da der Benutzer nur die fertig initialisierte GUI mit allen Komponenten dargestellt bekommen soll, war diese Funktion die beste Möglichkeit hierzu. Der zuvor genannte *Gain*-Slider beispielsweise, wurde in dieser *OpeningFcn*-Funktion implementiert, dessen Position innerhalb der GUI festgelegt und seine Callback-Funktion definiert.

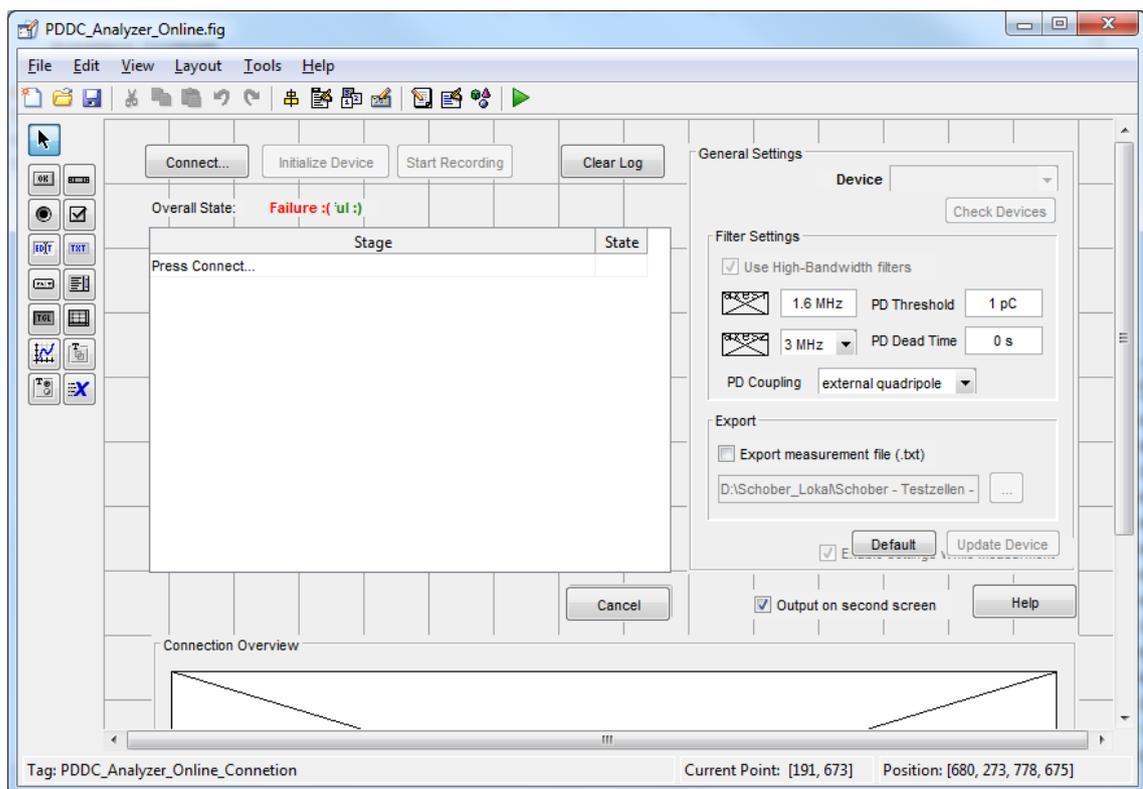


Bild 18: *GUIDE*-Oberfläche der Main-GUI [34]

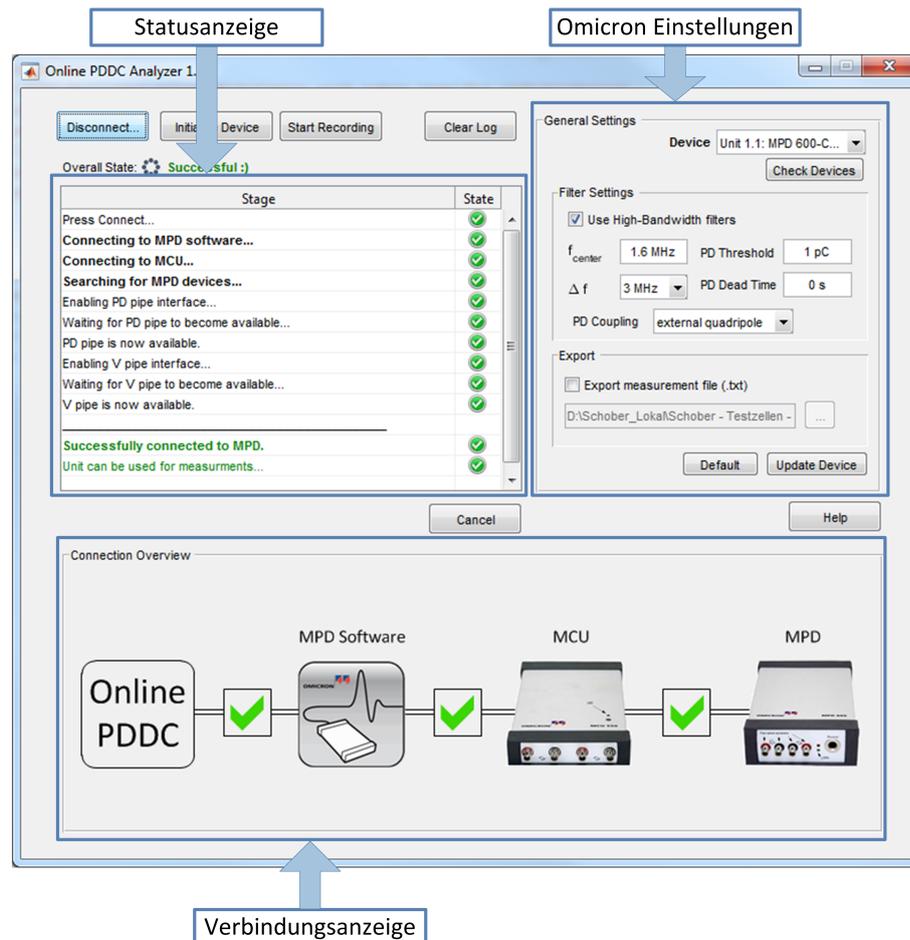


Bild 19: Main-GUI zur Verbindung mit der Omicron-Software

Nach diesem Schema wurden zwei komplexe GUI mit einer teils beachtlichen Anzahl an Elementen erstellt. In Bild 19 ist die Main-GUI dargestellt, welche beim Öffnen des Programms erscheint. Sie dient dem Verbindungsaufbau mit der *Omicron Software for MPD and MI*, besitzt eine Statusanzeige und eine graphische Darstellung der Verbindung (Verbindungsanzeige), sowie einige wichtige Einstellungen der Omicron-Software, wie Filter- und TE-Detektionseinstellungen. Unter *Device* im rechten oberen Eck kann bei mehreren angeschlossenen Messeinheiten zwischen diesen umgeschaltet werden. Mit einem Klick auf den Button *Connect...*, versucht das Programm sich mit der Messsoftware von Omicron zu verbinden. Ist dies erfolgt, kann mit *Initialize Device* die Omicron-Software auf DC-Messungen angepasst werden. Alle relevanten Einstellungen werden dabei automatisch gesetzt. *Start Recording* öffnet eine weitere GUI, wie sie in Bild 20 dargestellt ist, und startet einen Livestream der Echtzeitdaten aus der Omicron-Software. Die Messdaten werden verarbeitet und in den Diagrammen angezeigt. Diese GUI besitzt einerseits die zeitbasierten Diagramme oben (Time Functions) und darunter die NoDi*-Diagramme. In den Tabs links können diverse Einstellungen vorgenommen werden. Nach der Erstellung der beiden Oberflächen mussten die zu den GUI-Elementen zugehörigen Callbacks und alle weiteren benötigten Funktionen programmiert und deren Funktionsweise festgelegt werden.

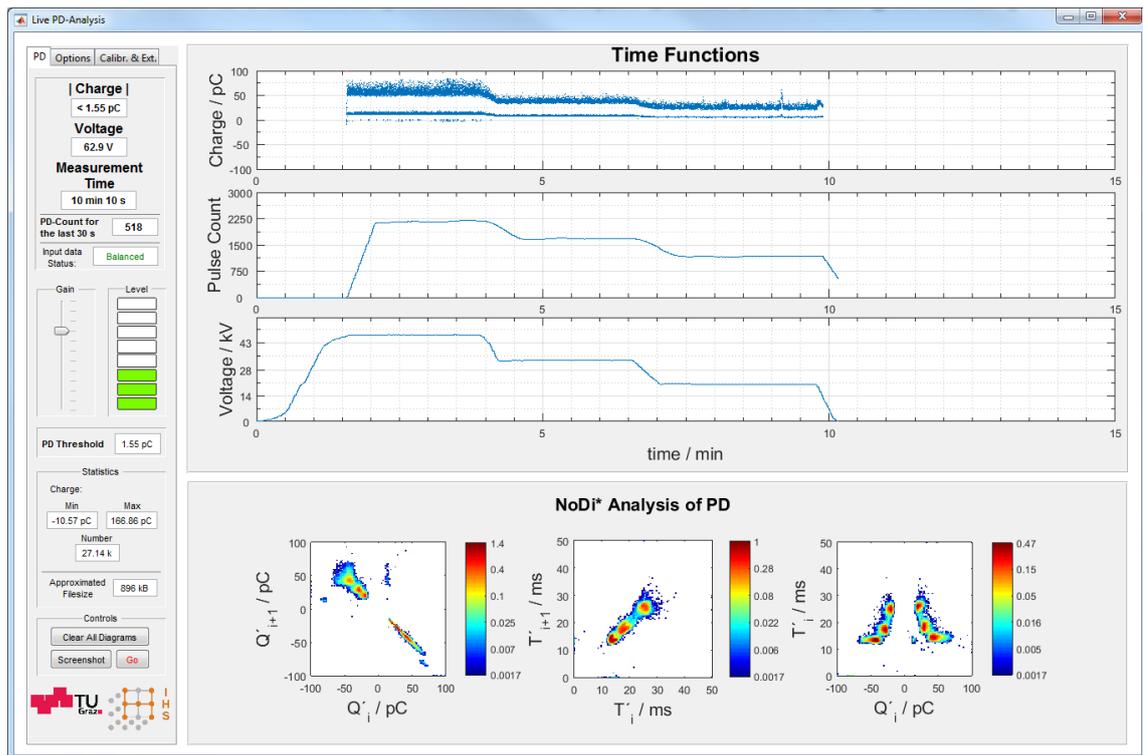


Bild 20: Stream-GUI zur Darstellung der Messdaten in Echtzeit in verschiedenen Diagrammen und Einstellungsmöglichkeiten

3.4.3 Dokumentationen und Flussdiagramm

3.4.3.1 Programmiererdokumentation

Diese Dokumentation dient zur Beschreibung des Quellcodes und soll eine Weiterentwicklung des Programms durch Andere erleichtern. Grundsätzlich ist sie für den Endbenutzer nicht von Bedeutung [35].

Heutige Softwaredokumentationen verfolgen den Ansatz, den Quellcode selbsterklärend zu gestalten und Variablenamen so zu wählen, dass diese intuitiv sind. Auch soll diese Art der Dokumentation in den Quellcode eingearbeitet sein. Diese Ansätze wurden bei der Erstellung der Software berücksichtigt. Der Quellcode selbst ist auch als Dokumentation zu sehen. Es wurden ausreichend Kommentare hinzugefügt, damit schnell ersichtlich ist, was im jeweiligen Codeabschnitt passiert. Bei den selbstgeschriebenen Funktionen werden zusätzlich am Anfang der Datei die Ein- und Ausgabeparameter der jeweiligen Funktion erklärt. Diese Dokumentation soll als Zusatz zum Quellcode angesehen werden, da an dieser Stelle nur grundlegende Programmstrukturen beschrieben werden.

Wie die Erstellung der GUI durchgeführt wurde, wurde bereits im vorhergehenden Kapitel beschrieben. Wie erklärt, erstellt Matlab beim Speichern der GUI zwei Files, ein .fig- und ein .m-File. In der .fig-Datei ist das Layout der GUI enthalten und kann dort auch nachträglich verändert werden, im .m-File wiederum befinden sich alle notwendigen Funktionen der GUI. Diese Funktionen bzw. Callbacks können je nach gewünschter Funktionsweise mit dem benötigten Code beschrieben werden. Das nachfolgende Listing 3 zeigt das Beispiel eines solchen Callbacks. Die Eingabe-

parameter `hObject`, `eventdata` und `handles` werden beim Aufrufen eines Callbacks (Event) unbedingt benötigt und automatisch von Matlab mitgesendet.

hObject enthält das Objekt, welches das Event ausgelöst hat, üblicherweise ist dies das Objekt selbst. Es kann auch über `handles.objectName` darauf zugegriffen werden, da jedes Element der GUI auch im Handles-Struct abgespeichert ist.

eventdata enthält Informationen darüber, wie das Event getriggert wurde und Angaben über betätigte Maus- oder Tastaturaktionen. Damit kann z. B. überprüft werden, welche Tasten gedrückt wurden. Dadurch können bestimmten Tastenkombinationen Aktionen zugewiesen werden.

handles ist ein Struct, welches alle wichtigen Informationen und Elemente der GUI und deren Parameter enthält. Es können auch eigene Variablen in diese Struktur geschrieben werden, um diese Daten zwischen den Callbacks verfügbar zu machen. Anstatt globale Variablen zu verwenden, können dadurch Informationen zwischen verschiedenen Funktionen und allen Callbacks ausgetauscht werden. Mit den Matlab-Funktionen `set()` und `get()` können die Eigenschaften eines GUI-Elements abgerufen bzw. neu beschrieben werden. In neueren Matlab Versionen (ab Version R2014b) kann hierfür auch die aus vielen Programmiersprachen verwendete Punktnotation verwendet werden, die alte Methode funktioniert aber auch weiterhin. Als Beispiel dient eine Checkbox `checkBox`, dessen Wert gleich 1 im angeklickten Zustand und sonst 0 ist. Dieser Wert ist in der Eigenschaft `'Value'` des Elements gespeichert. Wie bereits erwähnt, ist jedes Element über dessen Namen (in Matlab `'Tag'`) aufrufbar.

```
% Vor und nach Matlab Version R2014b
get(handles.checkBox, 'Value'); % Wert der Checkbox lesen
set(handles.checkBox, 'Value', 1); % Neu setzen des Wertes

% Nach Matlab Version R2014b
handles.checkBox.Value; % Wert des Checkbox lesen
handles.checkBox.Value = 1; % Neu setzen des Wertes
```

Das Code-Beispiel in Listing 3 soll die Verwendung des Handles-Structs und der Callbacks verdeutlichen. Es zeigt die Callbacks zweier Elemente, einer Checkbox und eines Push-Buttons, welche beim Drücken des Push-Buttons bzw. beim Ändern der Checkbox automatisch ausgeführt werden. Beim Klick auf den Push-Button wird eine Multiplikation von `a*2` durchgeführt. Die Variable `a` ist im Handles-Struct abgespeichert und wird über dieses Struct abgerufen. Das Ergebnis der Multiplikation wird in der Variable `b` in den Handles gespeichert. Beim Ändern der Checkbox wird dessen Callback `checkbox_callback` ausgeführt. Durch die Checkbox kann der Wert von `a` verändert werden.

Ist die Checkbox angeklickt, so ist das Ergebnis der Multiplikation `b = 4`. Ist sie jedoch nicht angeklickt, dann ist `b = 8`. So können Daten und Variablen leicht zwischen verschiedenen unabhängigen Funktionen innerhalb der GUI ausgetauscht werden. `hObject` ist in beiden Fällen das Element selbst. Wenn die Checkbox geändert und der Callback ausgeführt wird, so ist `hObject` das Element `checkbox`. Im Callback des Push-Buttons `function pushButton_Callback()` könnte ebenfalls über

`handles.checkbox` auf die Checkbox und ihre Eigenschaften zugegriffen werden.

Listing 3: Beispiel zweier Callback-Funktionen und Austausch von Variablen zwischen diesen

```
%% Callback der Checkbox
function checkbox_Callback(hObject, eventdata, handles)
if get(hObject, 'Value') % falls Checkbox angeklickt
    a = 2;
else % sonst
    a = 4;
end

% Speichern von a in handles.a
handles.a = a;

% Zurückspeichern des Handles-Structs
guidata(hObject, handles);

%% Callback des Push-Buttons
function pushButton_Callback(hObject, eventdata, handles)
a = handles.a;

b = a*2;

% Speichern von b in handles.b
handles.b = b;

% Zurückspeichern des Handles-Structs
guidata(hObject, handles);
```

Ein Großteil der programmierten Funktionen, die im Programm verwendet werden, arbeiten nach diesem Schema und sind auf diese Weise aufgebaut. Bei den MEX-Funktionen wurde darauf geachtet, dass bestimmte Konventionen bei der Dokumentation eingehalten wurden, da sich die Notation in C++ sehr von der in Matlab unterscheidet. Die grundsätzliche Struktur dieser Funktionen wird in Kapitel 3.4.4 beschrieben. Es sei vorweggenommen, dass es sich bei MEX-Funktionen, bis auf einige kleinere Änderungen, um C++-Code und -Notation handelt.

Quellcode: Der Quellcode ist über ein Subversion-Projekt (SVN) an der TU Graz verfügbar. Es benötigt hierzu den TUGRAZonline Username und das Passwort als Zugangsdaten, sowie eine Zugriffsberechtigung.

URL: <https://svn.tugraz.at/svn/PDDC-Analyzer-live>

Flussdiagramm: Zum leichteren Verständnis wurde ein Flussdiagramm des Programms angefertigt. Dieses soll veranschaulichen, welche Strukturen das Programm bei der Ausführung und beim Betätigen bestimmter Aktionen durchläuft. Das Flussdiagramm kann keinesfalls jede einzelne Zeile des Programms widerspiegeln, soll jedoch einen Überblick darüber liefern wie Funktionen und Parameter verknüpft sind. Die Symbole wurden nach DIN 66001:1983-12 gewählt. Zur leichteren Übersicht im Code, wurden die Funktionsnamen ebenso ins Diagramm übernommen.

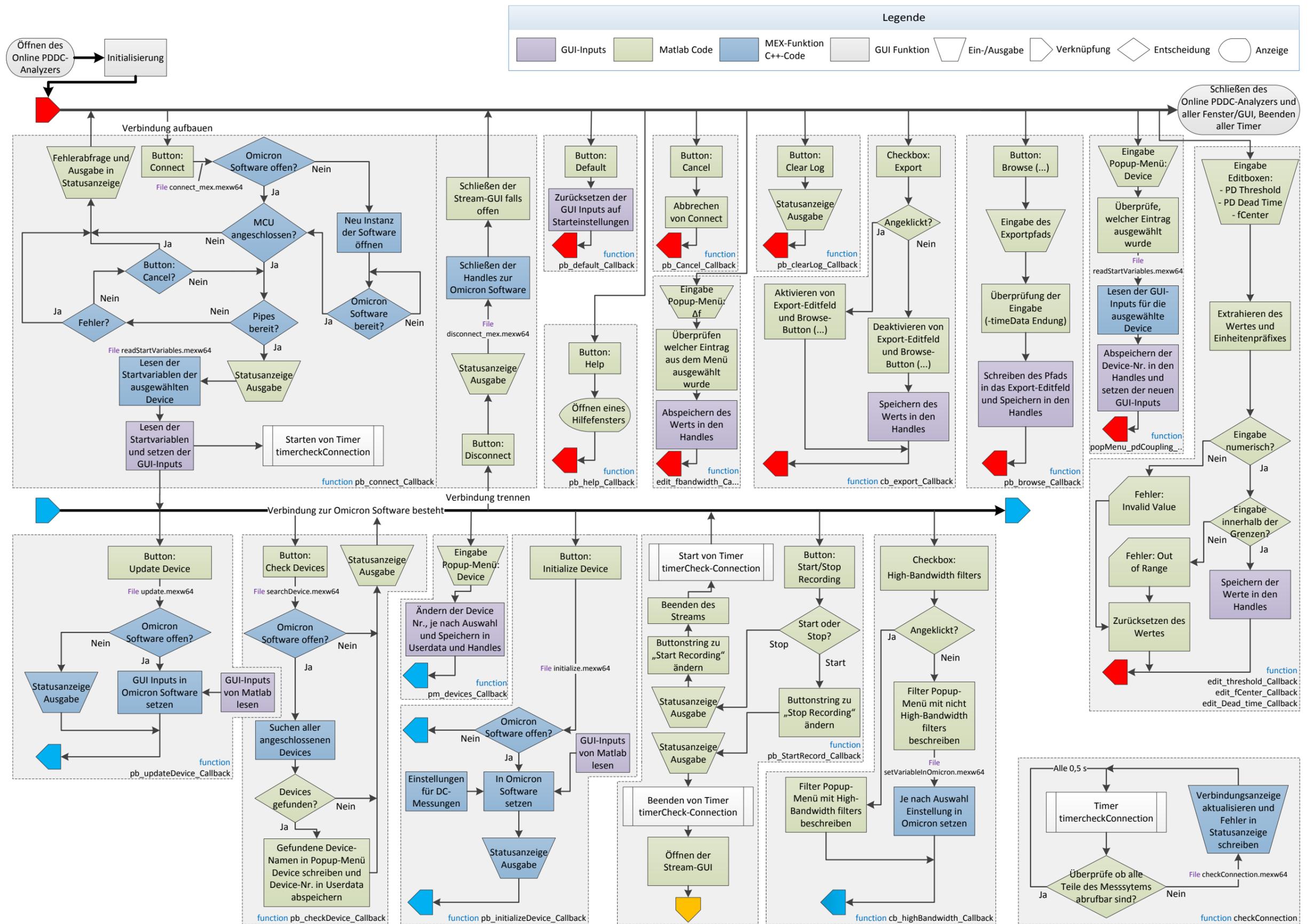


Bild 21: Flussdiagramm der Main-GUI

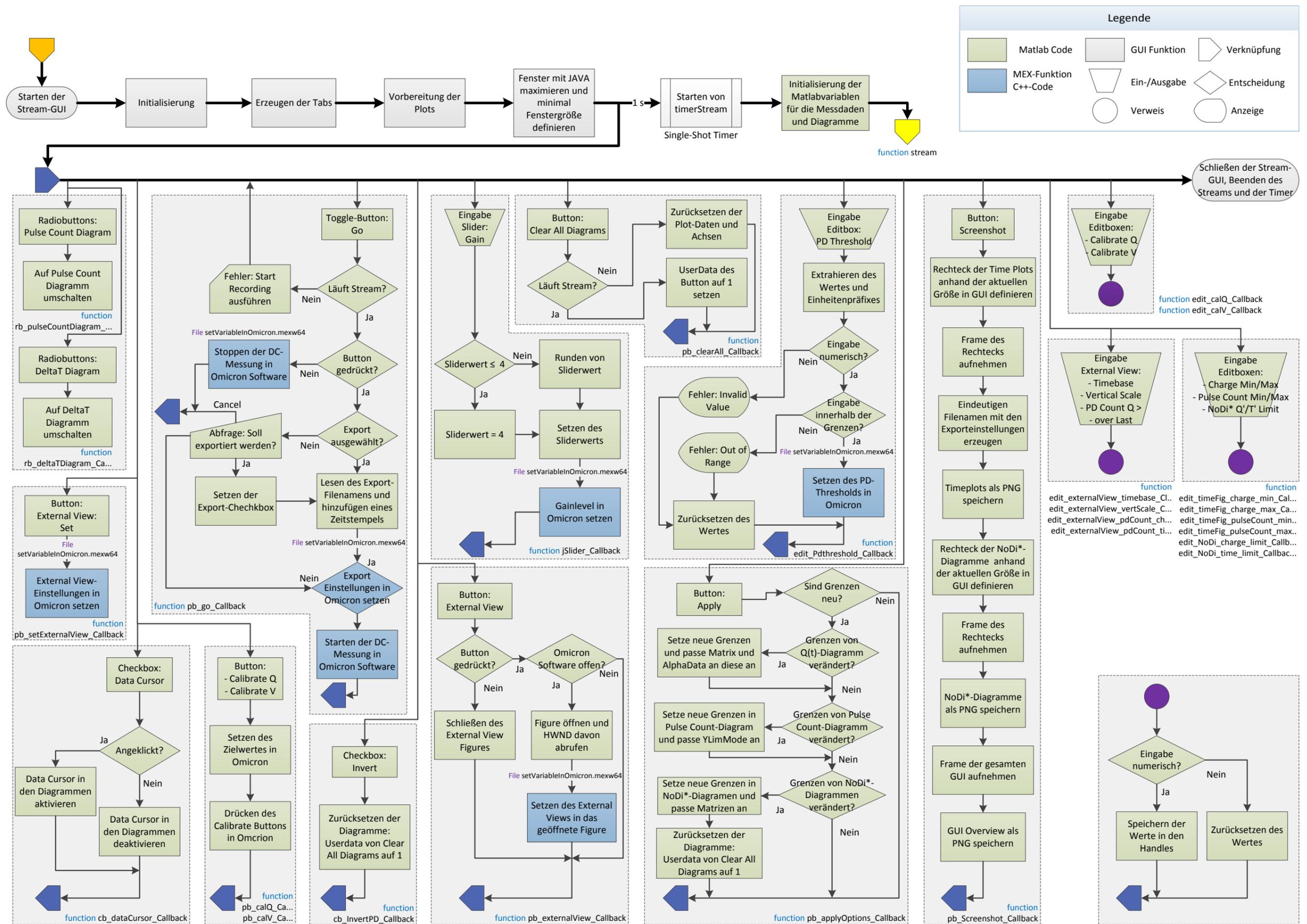


Bild 22: Flussdiagramm der Stream-GUI

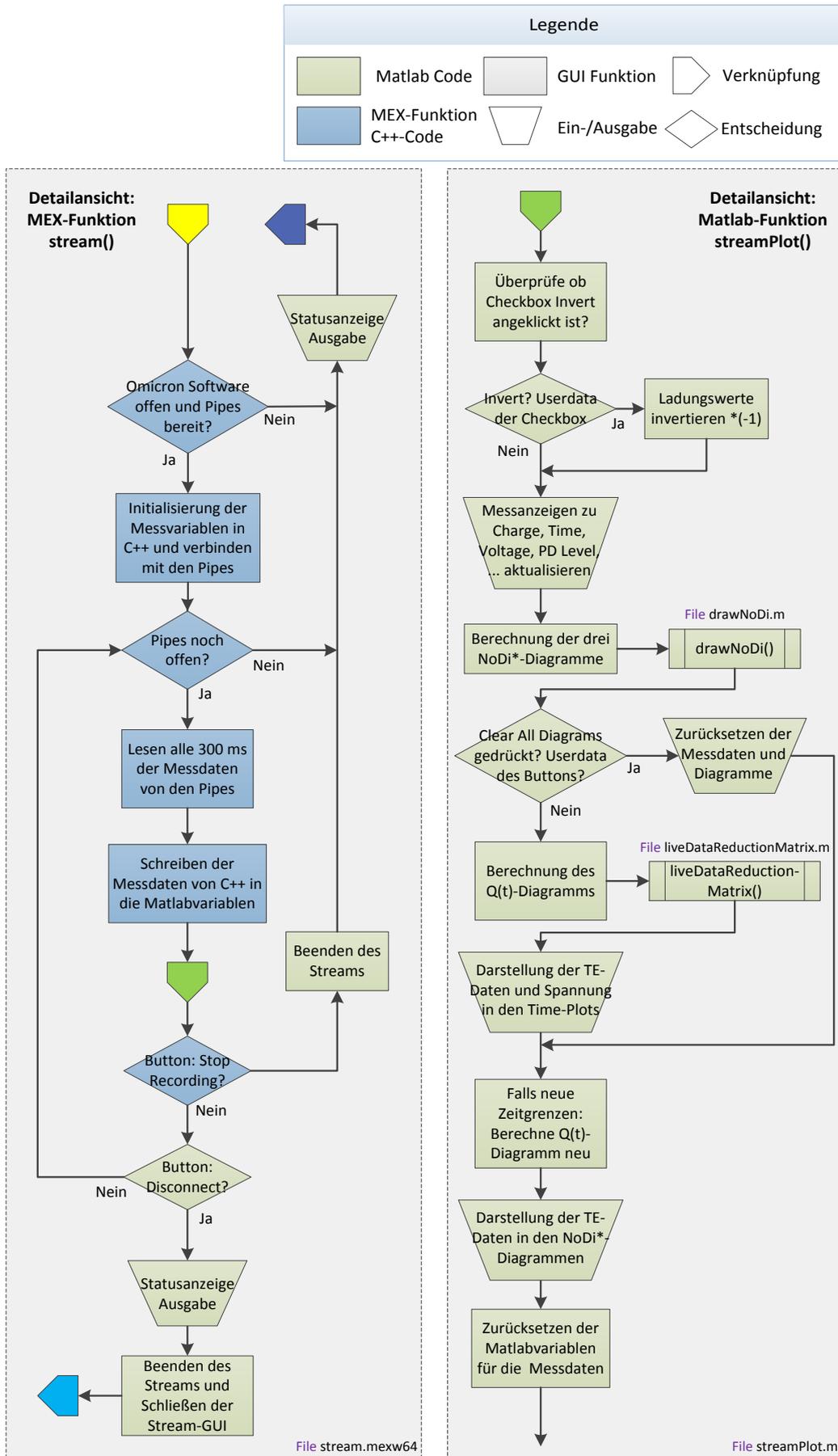


Bild 23: Detailansichten der Flussdiagramme (links die MEX-Funktion stream() und rechts die Matlab-Funktion streamPlot())

Datei-Übersicht: Um den Code übersichtlich zu halten, wurden einige Funktionen in separate Files ausgelagert, oder aus Notwendigkeit, da MEX-Funktionen eigene Dateien sind. Die nachfolgende Tabelle 6 zeigt alle benötigten Files und die jeweiligen Ordner an. Unterordner werden beim Öffnen des Programms automatisch zum Suchpfad von Matlab hinzugefügt. Insgesamt benötigt das Programm 32 einzelne Dateien zur Ausführung. Jedes MEX-File enthält zusätzlich auch ein .cpp-File.

Tabelle 6: File- und Ordnerübersicht

Nr.	Filename	Filetyp	Legende
Ordner: Programmordner			MEX-Funktion
1	checkConnection	mex	Matlab Figure
2	connect_mex	mex	Matlab m-File
3	Connection_error	png	PNG/JPG-Bild
4	Connection_loading	png	GIF-Bild
5	Connection_success	gif	
6	connectOverview_load	gif	
7	disconnect_mex	mex	
8	drawNoDi	m	
9	getVariableInOmicron	mex	
10	gui_stream	fig	
11	gui_stream	m	
12	initialize	mex	
13	liveDataReductionMatrix	m	
14	PDDC_Analyzer_Online	fig	
15	PDDC_Analyzer_Online	m	
16	readStartVariables	mex	
17	Runden	m	
18	searchDevice	mex	
19	setVariableInOmicron	mex	
20	SI_units	m	
21	stopConnection	m	
22	stream	mex	
23	streamPlot	m	
24	update	mex	
Ordner: /Figures			
25	connectOverview	png	
26	connectOverview_check	png	
27	connectOverview_fail	png	
28	Help_Connection_of_the_ ...	png	
29	IHSklein	jpg	
30	PDDC_Analyzer_Online_Help	fig	
31	PDDC_Analyzer_Online_Help	m	
32	TULogo	png	

3.4.3.2 Methodendokumentation

Die Methodendokumentation dient zur Beschreibung der Grundlagen, wie beispielsweise mathematische Rechenoperationen und technisch-wissenschaftliche Verfahren des Programms. Dieser Art der Dokumentation kann einem Anwender zur Verfügung gestellt werden, um die Hintergründe des Programms besser zu verstehen [35].

Herausforderungen der Echtzeit-Darstellung: Die Anzeige der Messdaten in den Diagrammen hat einige Herausforderungen, verglichen mit dem bereits vorhandenen Programm *PDDC-Analyzer*, mit sich gebracht. In diesem werden die gesamten Messdaten ins Programm eingelesen und die Darstellungen bzw. Diagramme können von der Software so eingestellt werden, dass alle Datenpunkte innerhalb der Grenzen liegen und angezeigt werden. Bei Echtzeitdaten ist dies nicht der Fall:

- Da die Messdaten im Vorfeld nicht bekannt sind, müssen die Achsengrenzen fest eingestellt werden.
- Die Grenzen der Matrizen zur Berechnung des $Q(t)$ -Diagramms und der drei NoDi*-Diagramme müssen festgelegt werden.
- Graphische Anzeigen, wie Farbskalen, müssen dynamisch gehalten und in Abhängigkeit der Messdaten angepasst werden.

Diese Vorgaben bedeuten jedoch nicht, dass die Grenzen während einer Messung nicht verändert werden können, lediglich, dass die Anzeige vom Benutzer händisch auf das bestmögliche Ergebnis eingestellt werden muss und nicht automatisch erfolgen kann.

Kalibrierung der Ladung und Spannung: Der *Online PDDC-Analyzer* besitzt eine Funktion den Ladungswert und die Spannung der Messeinheit MPD 600 zu kalibrieren. Zwar können über die COM-Schnittstelle die Zielwerte der Kalibrierung in der Omicron-Software gesetzt werden, jedoch ist der Button, welcher die Kalibrierung durchführt, nicht über diese ausführbar. Es musste somit ein Weg gefunden werden, dennoch den Button mit Matlab betätigen zu können. Dazu wurde der Weg über sogenannte Windows Handles `HWND` gewählt, welche eindeutige Referenzwerte darstellen. Mit Hilfe der Windows API und der Funktion `SendMessage()` lassen sich programmatisch Events bestimmter Elemente oder Objekte ausführen. So kann mit `SendMessage(hWnd, BM_CLICK, 0, 0)`; das Klicken des Buttons mit dem Handle `hWnd` durchgeführt werden. Um die Buttons zur Ladungs- und Spannungskalibration anwählen zu können, mussten diese zunächst im System gesucht werden:

1. Suchen der Prozess-ID der Omicron-Software aus den vorhandenen Prozessen mit `MyGetProcessId(TEXT("mpd_mi.exe"))`;
2. Suchen des Handles `HWND` der Omicron-Software anhand der Prozess-ID mit der Funktion `GetWindowThreadProcessId()`;
3. Suchen des `HWND` der `SysTabControl32` mit dem `HWND` der Omicron-Software.
4. Den entsprechenden Tab der `SysTabControl32` in der Omicron-Software sichtbar machen: *Q*-Tab für Ladungskalibrierung und *V*-Tab für die Spannung.

5. Suchen des *Compute*-Buttons in der Software anhand der ID-Nummer 184 und der Eigenschaft `WS_VISIBLE`, da es in der Software mehrere Buttons mit der gleichen ID-Nummer 184 gibt, jedoch immer nur einer sichtbar ist.

Da auf diesem Weg die Buttons eindeutig identifiziert werden konnten, war es möglich, einen Buttonklick durchzuführen, um die Messeinheit zu kalibrieren.

Datenreduzierung: Eines der Hauptprobleme während der Entwicklung des Programms war die Verarbeitung von zu großen Datenmengen innerhalb kürzester Zeit, d. h. im Bereich von Millisekunden während einer Messung. TE-Messsysteme reagieren äußerst empfindlich und haben eine hohe Zeitauflösung. Wie bereits in Kapitel 2.3.3 beschrieben wurde, sollte Grundrauschen vermieden werden, da ansonsten die NoDi*-Diagramme verfälscht werden. Es kann jedoch dennoch passieren, dass Grundrauschen nicht immer vermieden werden kann. Zum Beispiel kann sich der Grundstörpegel während einer Messung ändern, oder dass die Verstärkung angepasst werden muss, wodurch sich abermals der Grundstörpegel ändert. Auch zur erstmaligen Einstellung des TE-Thresholds, muss Rauschen aufgenommen werden. Abgesehen von Rauschen, gibt es Defekte, bei denen im Bereich von Millisekunden sehr viele Teilentladungen, einige tausende bis zehntausende, auftreten. Solche Defekte können Spitze-Platte-Anordnungen oder freie metallische Partikel sein.

Es ergaben sich folgende Ansätze, die Anzeige der Messdaten zu realisieren:

- Die Diagramme und aktuellen Messdaten nur für eine bestimmte Zeit am Bildschirm anzeigen (ähnlich einem Oszilloskop).
- Die Diagramme und aktuellen Messdaten nach bestimmten Zeitabständen, beispielsweise in Abhängigkeit der Performance oder in festen Zeitintervallen zurückzusetzen.
- Die gesamten Messdaten in den Diagrammen anzeigen, jedoch bei schlechter Performance, aufgrund sehr vieler Messdaten innerhalb kurzer Zeit, auf die erstgenannte, oszillographische Methode umschalten.
- Die gesamten Messdaten in den Diagrammen anzeigen und die Zeitachsen zyklisch erweitern.

Die ersten beiden Möglichkeiten stellten keine Option dar, da die Messdaten über die gesamte Zeit betrachtet werden sollen. Durch diese Forderung wurde die letztgenannte Option gewählt und Lösungsansätze gesucht, dies zu bewerkstelligen.

Kommt es während einer Messung zu sehr vielen Messdaten, müssen diese verarbeitet und angezeigt werden. Eine Anforderung des Programms war es, dass in solch einem Fall, weder das Programm Daten verlieren sollte, noch dass aufgrund der immensen Datenmenge das Programm zeitlich nicht mehr folgen kann und die Verzögerung vom Auftreten der Messdaten und der Anzeige am Bildschirm immer größer wird. Zunächst wurde versucht eine Lösung zu finden, alle Messdaten in den Diagrammen anzuzeigen. Schnell hat sich herausgestellt, dass selbst mit verschiedensten Funktionen und Lösungsansätzen, die Anzeige von sehr vielen Daten

am Bildschirm, welche sich regelmäßig erweitern und mehr werden, nicht zu den Stärken des mathematischen Tools gehören. Ein weiterer Punkt der sich während verschiedener Tests und Performance-Optimierungen herausgestellt hat, war es, dass nicht die Verarbeitung, zum Beispiel die Berechnung der NoDi*-Diagramme, das Problem darstellt, sondern in erster Linie die Anzeige und Aktualisierung der Time-Plots. Die Anzeige der NoDi*-Diagramme stellte kein Problem dar, da es sich immer um eine Matrix fixer Größe handelt, wodurch nie mehr Speicher als die Größe der Matrix verwendet wird.

Der Grund für die schlechte Performance der Time-Plots besteht in der Arbeitsweise von Matlab. Wird ein Diagramm erstellt, reserviert Matlab im Arbeitsspeicher des Rechners einen entsprechenden Block. Das Programm versucht stets für ein Diagramm, aber auch für Vektoren und Matrizen einen durchgehenden Block zu belegen. Wird das Diagramm erweitert, indem Punkte angehängt werden, versucht Matlab diesen Block zu erweitern. Da andere Programme auch Arbeitsspeicher verwenden, kommt es häufig vor, dass Blöcke nicht einfach erweitert werden können. Um wieder einen durchgehenden Block zu bekommen, wird der alte kopiert, ein neuer Bereich im Speicher gesucht, welcher groß genug ist, dieser reserviert und die Daten verschoben. Dieses Prozedere ist sehr performance-intensiv und benötigt viel Zeit. Andere Lösungsansätze waren die Verwendung von sogenannten `Cells`, wodurch teilweise die Aufspaltung der Daten im Speicher und somit der Kopiervorgang verhindert werden, oder die Verwendung von sogenannten Low-Level-Functions wie `line()`, da diese schneller arbeiten als andere Befehle, wie beispielsweise `plot()`. Keiner dieser Ansätze hat jedoch zum gewünschten Erfolg geführt.

Ein weiterer Ansatz war es, die Stärken von Matlab zu verwenden, um die Schwächen auszugleichen. Er beruht im Grunde auf einer Reduzierung der Messdaten, auch *thinning* oder *decimation* genannt [33]. Werden viele Impulse in kurzer Zeit detektiert und über die Zeit in einem Diagramm aufgetragen, ergibt sich aufgrund der Auflösung des Bildschirms auch eine Auflösung der x- und y-Achsen. Würden alle Punkte eingezeichnet werden, kann es vorkommen, dass mehrere Punkte auf dasselbe Pixel fallen, welche Speicher benötigen, jedoch keinen Informationsgehalt haben. Die gleiche Information würde auch einer dieser Punkte beinhalten. Aufgrund dieser Tatsache wurde ein Algorithmus geschrieben, welcher die Daten sinnvoll reduzieren und gleich viel Informationsgehalt wie die ursprünglichen Daten beinhalten soll, jedoch sehr viel schneller bei der Darstellung ist. Dieser Algorithmus wird im Folgenden beschrieben.

Für die Datenreduzierung wurden das PRPD- und NoDi*-Diagramm als Idee herangezogen. Wie zuvor kurz erwähnt, handelt es sich bei beiden Darstellungen um Matrizen fixer Größe, welche einmal im Speicher reserviert werden müssen. Deren Speicherbedarf ändert sich danach nicht mehr. Die Idee bestand darin, die Pixel der Time-Plots am Bildschirm in x- und y-Richtung als Matrix anzusehen, die Messdaten zu verarbeiten und in diese Matrix zu projizieren. Fallen mehrere Punkte auf die gleiche Position in der Matrix, äußert sich dies lediglich in einer Erhöhung der Zahl in der Matrix, nicht jedoch in der Größe und damit der Speichermenge. Im PRPD-Pattern wird dies durch die Intensität (Farbskalierung) dargestellt, was jedoch bei dieser Darstellung über die Zeit nicht von Interesse ist. Im Q(t)-Diagramm interessiert lediglich, ob TE-Impulse auftreten, nicht wie viele auf jeden Pixel fallen.

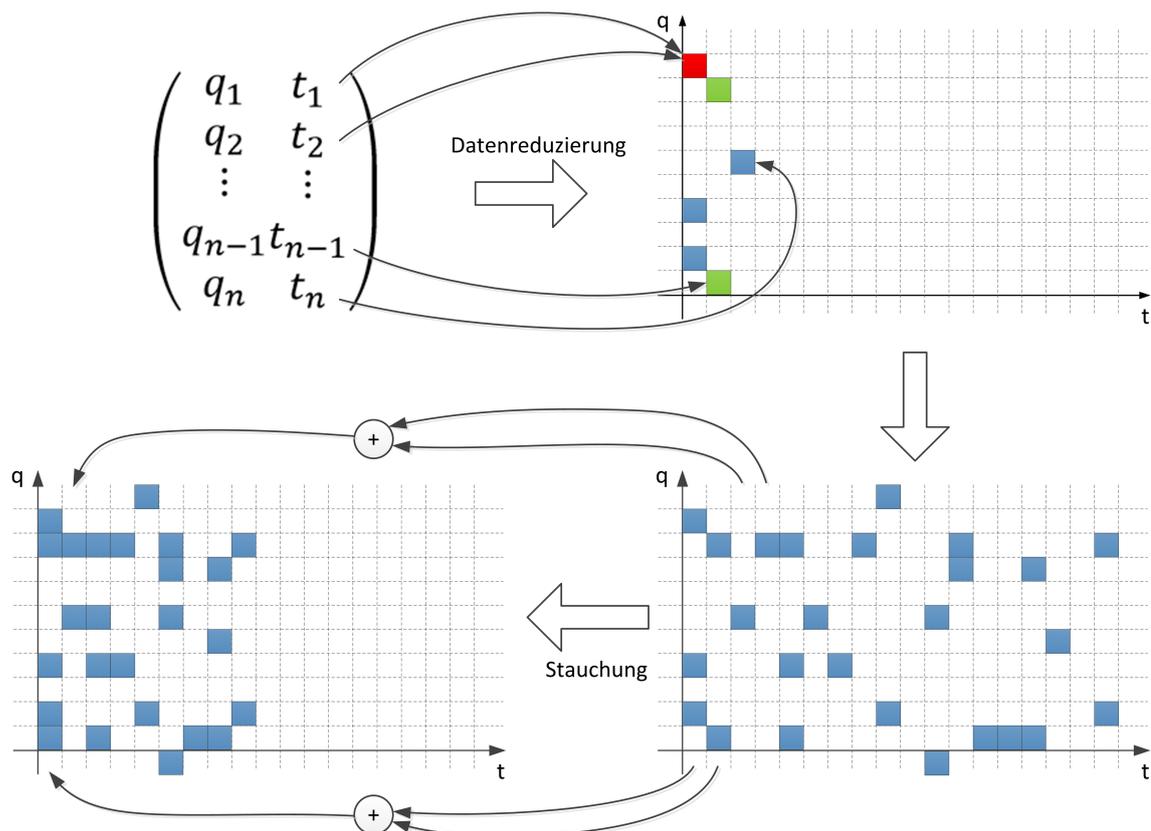


Bild 24: Funktionsweise des Datenreduzierungsalgorithmus

In Bild 24 ist die Funktionsweise der Datenreduzierung dargestellt. Zunächst werden die Daten anhand eines Algorithmus in eine Matrix projiziert. Fallen mehrere Punkte in das gleiche Raster, so würde sich die Intensität erhöhen. Dies ist jedoch nicht erwünscht, weshalb alles einheitlich skaliert wird. Die Größe der Matrix entspricht dabei genau den Pixeln in x- und y-Richtung der Plotachsen am Bildschirm. Fallen Punkte außerhalb der Grenzen, werden diese rot am Rand eingezeichnet, um kenntlich zu machen, dass TE-Impulse außerhalb der dargestellten Grenzen liegen. Umso weiter die Messzeit steigt, umso voller wird die Matrix besetzt, vorausgesetzt es treten TE-Impulse auf. Steigt die Messzeit über die Zeitskalierung der Plots, wird die aktuelle Matrix auf die Hälfte gestaucht und in einer weiteren Matrix mit der ursprünglichen Größe eingezeichnet. Die Stauchung selbst erfolgt indem immer zwei nebeneinanderliegende Spalten miteinander addiert werden. Spalte 1 und Spalte 2 werden somit zu Spalte 1 der neuen Matrix, Spalte 3 und Spalte 4 zu Spalte 2. Durch diese Stauchung ist sichergestellt, dass die Größe der Matrix konstant bleibt und somit die Speichernutzung nicht stetig steigt. Ein weiterer Vorteil durch die Addition besteht darin, dass durch die Stauchung keine Daten bzw. Informationen darüber verloren gehen, in welches Raster die TE-Impulse gefallen sind. Ein Nachteil ergibt sich jedoch durch die Reduzierung. Werden die Grenzen des $Q(t)$ -Plots verändert, so ist keine Information mehr außerhalb der Grenzen vorhanden. Ein freies Zoomen der Achsen ist dadurch nicht mehr gegeben. Anstatt beim Verändern der Grenzen alle Daten aus den Plots zu löschen, werden im $Q(t)$ -Diagramm die vorhandenen Daten transparent dargestellt, um ersichtlich zu machen,

dass die alten Daten nicht mehr mit den dargestellten neuen Grenzen übereinstimmen. Im Vergleich dazu, beim PRPD-Pattern werden die Daten im Histogramm gelöscht, wenn die Grenzen verändert werden. Um freies Zoomen zu ermöglichen, müssten alle bisherigen Daten gespeichert, diese behalten und beim Verändern der Diagrammgrenzen die Matrix mit allen bisherigen Daten neu berechnet werden. Dies würde einen enormen Rechenaufwand und Speicherbedarf bei zunehmender Messzeit bedeuten.

Der große Vorteil dieses Algorithmus ist der große Geschwindigkeitsgewinn. Auch beim Auftreten großer Datenmengen innerhalb kurzer Zeit und bei steigender Messzeit, werden diese immer auf die gleiche Größe reduziert.

3.4.3.3 Benutzerdokumentation

Diese Dokumentation ist für den Endbenutzer die wichtigste Art der Dokumentation, da hier die Benützung des Programms und die Benutzerschnittstelle (z. B. GUI) dokumentiert werden [35].

Das Programm *Online PDDC-Analyzer* wurde so ausgelegt, dass die Bedienung mit einigen Vorkenntnissen der Software von Omicron, aber auch ohne diese, größtenteils selbsterklärend ist. Treten Fehler durch Falschbedienung oder sonstiges auf, so werden in der Regel Fehlermeldungen ausgegeben, welche Details zu dem jeweiligen Fehler enthalten. Dieser Teil der Dokumentation soll als Gebrauchsanleitung für den Endbenutzer dienen und die grundsätzliche Funktionsweise des Programms aufzeigen, ohne Kenntnis des im Hintergrund ausgeführten Codes zu haben. Das Programm selbst umfasst ebenfalls eine rudimentäre Hilfefunktion, welche eine Schritt für Schritt Anleitung enthält (Button *Help*).

Beim Starten der Software erscheint die in Bild 19 gezeigte GUI.

Schritt 1: Verbinden

Mit Klick auf *Connect...* wird das Programm versuchen eine Verbindung zur Omicron-Software aufzubauen. Ist bereits eine Instanz der Software offen (das kann auch ein Streamfile im Offline-Modus darstellen), dann wird zu dieser Instanz verbunden. Falls noch keine Instanz geöffnet wurde, öffnet das Programm selbstständig eine neue und verbindet sich zu dieser. Der Verbindungsaufbau wird einerseits in der Statusanzeige als Tabelle angezeigt, aber auch graphisch in der Verbindungsanzeige (Bild 19). Schlägt die Verbindung fehl, ist somit auf einen Blick erkennbar zu welcher Einheit des Messsystems die Verbindung unterbrochen wurde. Auch in der Statusanzeige werden genauere Details dazu angezeigt. Bei erfolgreicher Verbindung wird diese auch weiterhin stetig kontrolliert. Einige Buttons der GUI werden erst aktiviert, nachdem eine Verbindung aufgebaut wurde. Dauert der Verbindungsaufbau zu lange, kann dieser mit dem Button *Cancel* unterbrochen werden.

Schritt 2: Device auswählen

In den *General Settings* rechts oben, kann das Device eingestellt werden, mit dem die Messung vorgenommen werden soll. Dies ist nur von Relevanz, falls mehrere MPD angeschlossen werden. Ist nur eine MPD mit der MCU verbunden, so wird auf diese zugegriffen. Mit Klick auf *Check Devices* kann auch bei bestehender Verbindung auf neu angeschlossene Messeinheiten überprüft und diese verwendet werden.

Schritt 3: Initialisierung

Erst nachdem die Verbindung aufgebaut wurde, können Buttons, wie *Initialize Device*, angeklickt werden. Mit diesem werden alle relevanten Einstellungen in der Omicron-Software automatisch gesetzt und diese für Messungen bei Gleichspannung angepasst. Auch die Filtereinstellungen in den *General Settings* werden direkt aus der GUI gelesen und in der Messsoftware gesetzt. Diese können, wie in Schritt 4 erklärt wird, auch nachträglich noch verändert werden. In der Statusanzeige werden die in der Omicron-Software veränderten Parameter aufgelistet.

Schritt 4: Filter und Export

Die in Schritt 3 erklärten Einstellungen sind hauptsächlich dafür gedacht, die Omicron-Software auf DC-Messungen einzustellen. Werden in der GUI Änderungen in den Feldern *Export* oder den *Filter Settings* vorgenommen, dann können diese auch mit *Update Device* aktualisiert werden. Hierbei werden lediglich diese Einstellungen neu gesetzt. Wichtig ist jedoch, dass diese erst nach Klicken des Buttons übernommen werden. Mit Klick auf *Default* werden die Einstellungen verworfen und die Filter-, Export- und Detektionseinstellungen auf die Starteinstellungen zurückgesetzt. In den *Filter Settings* können Einstellungen, die die TE-Detektion betreffen, verändert werden. Soll zusätzlich auch ein Messfile in Form eines Textfiles während einer Messung abgespeichert werden, kann im Feld *Export* der Pfad gewählt werden, wobei die dazugehörige Checkbox ausgewählt sein muss.

Schritt 5: Starten der Messung

Sind alle Einstellungen gesetzt und wurde das Messsystem richtig konfiguriert, erfolgt mit Klick auf *Start Recording* das Lesen und Anzeigen der Messdaten in Echtzeit. Dies geschieht in einer separaten GUI, welche, falls sie noch nicht offen ist, geöffnet und initialisiert wird (Bild 20). Sobald dies abgeschlossen ist, erfolgt das stetige Lesen und die Anzeige der Messdaten mit einer Aktualisierungsrate von 300 ms. Die graphische Oberfläche besteht aus zeitbasierten Diagrammen oben, drei NoDi*-Diagrammen unten und links aus Tabs (Bild 25). Die zeitbasierten Diagramme bestehen von oben nach unten aus dem Q(t)-Diagramm, mittig dem Impulszähldiagramm und dem V(t)-Plot. Das Spannungs-Zeit-Diagramm zeigt den Mittelwert der Spannung von 300 ms Zeitfenstern an. Im Tab *Options* kann unter *Display Options* das mittlere Diagramm durch die Radiobuttons geändert werden. Standardmäßig wird das Impulszähldiagramm dargestellt, wahlweise kann aber auch auf das $\Delta t(t)$ -Diagramm, welches den Mittelwert von Δt in 300-ms-Abständen anzeigt, umgeschaltet werden. Die einzelnen Tabs auf der linken Seite enthalten weitere für die Messung relevante Einstellungen. Die Einstellung TE-Threshold sollte so eingestellt werden, dass kein Grundrauschen zu sehen ist. Grund dafür ist, dass aufgenommene Messfiles sehr groß werden und dadurch nicht mehr verarbeitet werden können. Die Datenmenge wurde auf 100.000 Impulse innerhalb von 300 ms beschränkt, was ein mittleres Δt zwischen den Impulsen von $3 \mu\text{s}$ bedeutet. Es können auch aufeinander folgende Impulse kleiner $3 \mu\text{s}$ ausgewertet werden, so lange die mittlere Zeitdifferenz $3 \mu\text{s}$ nicht übersteigt. Die Anzeige *Input Data Status* gibt an wie viele Daten aktuell auftreten.

Balanced bedeutet, dass die zuvor genannte Datenmenge nicht überschritten wird. *Overdriven* bedeutet hingegen, dass zu viele bzw. *No Input Data*, dass weder TE- noch Spannungsdaten aufgenommen werden.

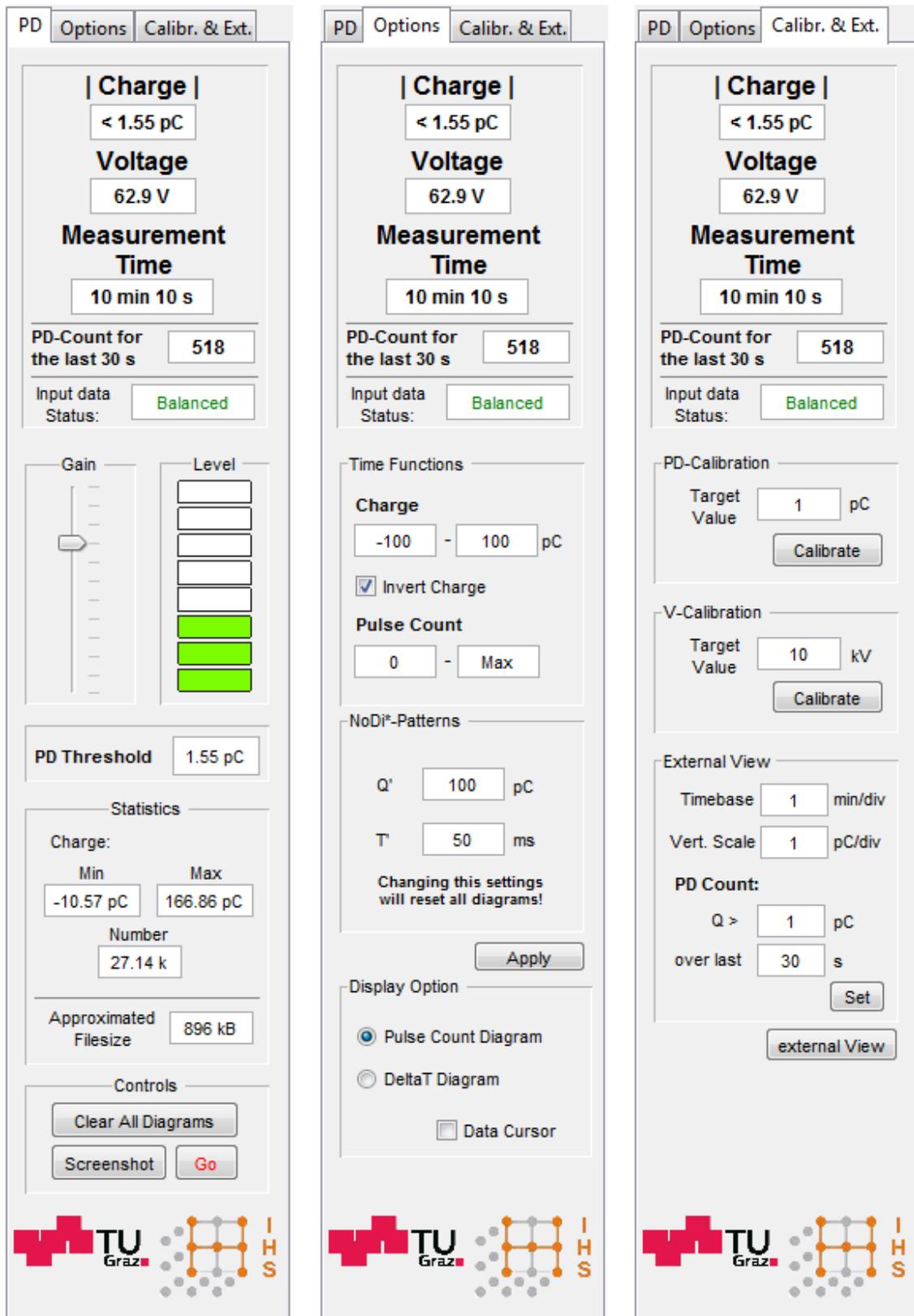
Links oben in den Tabs werden immer die aktuelle Ladung, die Spannung und die Messzeit angezeigt, welche alle 300 ms aktualisiert werden. Es handelt sich hierbei um gemittelte Werte für die Ladung und die Spannung, wobei sich der Ladungswert aus der Mittelung des Absolutbetrags ergibt. Die Anzeige darunter *PD Counts for the last 30 s* zählt die Impulse in den letzten 30 Sekunden. Die Einstellung innerhalb welchen Zeitraums Impulse oberhalb eines bestimmten Wertes gezählt werden sollen, kann im Tab *Calibr. & Ext.* unter *External View - PD-Count* verändert werden. Der Wert des Impulzählendiagramms wird alle 300 ms direkt aus der Omicron-Software gelesen und eingezeichnet. Die Anzeigen für *Gain* und *Level* im *PD*-Tab dienen der Verstärkereinstellung. Falls die Farbskala des *Levels* rot wird, sollte *Gain* verändert werden, bis der Level wieder grün anzeigt. Das Panel darunter enthält eine Statistik über die maximale und minimale Ladung, die Anzahl der Impulse sowie die geschätzte Filegröße. Die Statistik bezieht sich auf die in den Diagrammen dargestellten Impulse. Mit *Clear All Diagrams* werden die Diagramme gelöscht und die Statistik zurückgesetzt. Mit *Screenshot* können Bilder der Time-Funtions, NoDi*-Diagramme und ein Overview der gesamten GUI im Ordner, welcher in den Exporteinstellungen gewählt wurde, abgespeichert werden. Der Tab *Options* dient der Achseneinstellungen der verschiedenen Diagramme. Falls die NoDi*-Einstellungen verändert werden, werden die Diagramme zurückgesetzt. Mit Klick auf *Apply* werden die vorgenommenen Änderungen übernommen. Unter *Calibr. & Ext.* kann ein sogenannter *External View* aktiviert werden. Das bedeutet, dass in einem neuen Matlabfenster die Diagramme aus Ansicht DC1 der Omicron-Software direkt angezeigt werden. Das hat den Vorteil, dass nicht ständig zwischen den Programmen *Online PDDC-Analyzer* und der Omicron-Software gewechselt werden muss. Die Einstellungen in *External View* dienen zur Achseneinstellung der Diagramme in der Omicron-Software.

Schritt 6: Aufnahme und Speichern von Messdaten

Mit Klick auf *Go* wird in der Omicron-Software eine neue DC-Messung gestartet. Dazu werden die Diagramme zurückgesetzt. Mit nochmaligem Klick auf den Button wird die DC-Messung beendet und ein Textfile mit den Messdaten sowie ein *-vData.txt*-File mit den genauen Spannungsdaten in 300 ms Abständen abgespeichert. Der Filename der Textfiles ist so gestaltet, dass dieser das aktuelle Datum und die Uhrzeit enthält, um eindeutige Dateinamen zu generieren und einem Überschreiben bereits vorhandener Files vorzubeugen. Der Speicherort des Files ist der in den Exporteinstellungen gesetzte Pfad. Standardmäßig ist das der Pfad der *.exe*-Datei bzw. der Matlabfiles.

Schritt 7: Beenden des Streams und Schließen

Wird in der Main-GUI *Stop Recording* gedrückt, so bleibt die Stream-GUI offen, es werden jedoch keine neuen Messdaten mehr angezeigt. Gleiches passiert, wenn die Stream-GUI während des Streams geschlossen oder die Verbindung zur Omicron-Software unterbrochen wird. Die Verbindung kann auch manuell mit Klick auf *Disconnect* oder schließen der Main-GUI getrennt werden.



(a) Tab PD

(b) Tab Options

(c) Tab Calibr. & Ext.

Bild 25: Einzelne Tabs der Stream-GUI mit den Messanzeigen und Einstellungsmöglichkeiten

3.4.3.4 Installationsdokumentation

Die Installationsdokumentation richtet sich in erster Linie an Administratoren und dient zur Erklärung, welche Hard- und Software, Betriebssysteme und Software-Umgebungen benötigt werden [35].

Das Programm wurde mit der Software-Umgebung Matlab entwickelt. Deshalb muss auch eine gültige Version von Matlab zur Ausführung des Codes am jeweiligen Rechner vorhanden sein. Damit alle Funktionen einwandfrei funktionieren, vor allem in Bezug auf MEX-Funktionen, aber auch des Matlab-Codes selbst, sollte bestenfalls die Version von Matlab verwendet werden, auf der das Programm entwickelt wurde. Es handelt sich hierbei um die **Matlab Version R2016a (9.0.0.341360)**. Das Programm kann dem Endnutzer in zwei Varianten zur Verfügung stehen. Dies können entweder alle vom Programm benötigten Dateien mit der Dateiendung „.p“ sein oder die einfachere Variante eine .exe-Datei, welche alle benötigten Funktionen und Files bereits beinhaltet. Bei den sogenannten .p-Files handelt es sich um geschützte Dateien (protected), um Änderungen im Code durch den Benutzer zu verhindern. Zur Verwendung solcher Dateien muss jedoch eine kostenpflichtige Vollversion von Matlab vorhanden sein. Die zweite, unkomplizierte Variante ist es, das Programm in Form einer ausführbaren Datei (.exe-Datei) bereitzustellen. Diese benötigt zum Ausführen keine Vollversion von Matlab, sondern kann mit der kostenfreien Matlab-Runtime verwendet werden. Diese muss zunächst auf dem Rechner installiert werden. Danach kann das Programm ausgeführt werden. Wichtig ist auch hier, dass dieselbe Version der Runtime benutzt wird, auf welche das Programm kompiliert wurde. Es handelt sich um die **Runtime Version R2016a (9.0.1)**. Diese kann von [36] heruntergeladen werden.

Anmerkung zu den MEX-Files: Anhand der Dateierweiterung ist erkennbar, auf welchem System das MEX-File erzeugt wurde. Für 64-bit Versionen von Windows ist das **mexw64**. Wichtig ist, dass 32-bit-Versionen auf 64-bit-Systemen nicht verwendet werden können. MEX-Files sind üblicherweise **rückwärtskompatibel**, das bedeutet, dass mit einer neueren Version von Matlab erzeugte MEX-Files auch auf älteren Versionen funktionieren. Falls dies nicht der Fall ist, sollte der C++-Code erneut für die Zielplattform kompiliert werden. Es ist auch möglich, dass eine Vorwärtskompatibilität auftritt und somit ältere Funktionen auf neueren Matlabversionen einwandfrei funktionieren, was jedoch nicht offiziell seitens Matlab unterstützt wird. Der Grund für die Kompatibilität liegt in den *runtime libraries*, welche von MEX-Funktionen verwendet werden.

3.4.4 Aufbau und Funktionsweise des Codes

Dieses Kapitel dient keinesfalls dazu den gesamten geschriebenen Code zu beschreiben. Es soll aber die Funktionsweise und den Aufbau des Programms etwas näher erklären. Grundsätzlich teilt sich der Code in drei verschiedene Teile auf:

- Matlab-Code mit GUI
- C++-Code
- MEX-Funktionen

In Bild 26 sind die einzelnen Programmteile und ihr Zusammenspiel dargestellt. Der dunkelblaue Pfad stellt dabei den Pfad dar, der während des Ausführens des Programms zum Einsatz kommt. Der hellblaue Pfad wiederum, ist nur während der Entwicklung von Bedeutung. Es wird kurz auf die einzelnen Blöcke eingegangen.

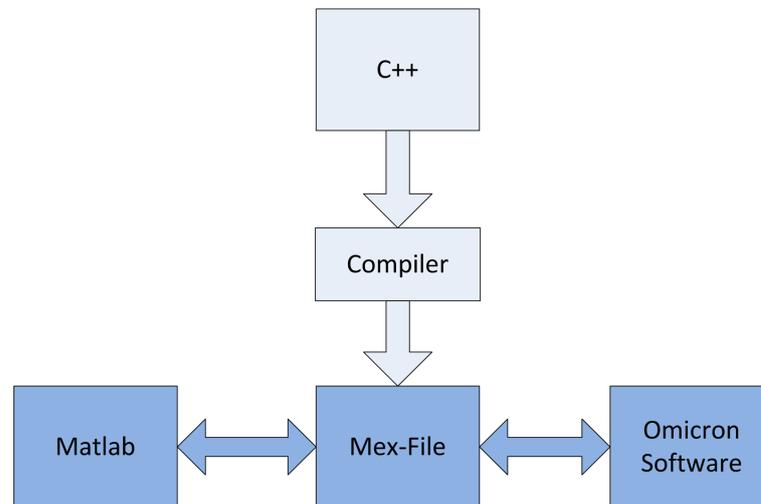


Bild 26: Blockdiagramm zur Funktionsweise des Programms *Online PDDC-Analyzer*

Matlab-Code mit GUI: Die GUI wurde, wie bereits erwähnt, mit Matlab erzeugt. Nachdem die GUI mit ihren Grundelementen erstellt wurde, wurden die einzelnen Elemente programmiert. Die GUI an sich ist eventorientiert, das heißt, dass beim Ausführen des Programms auf eine Aktion des Benutzers gewartet wird. Das wird auch eventorientierte oder ereignisorientierte Programmierung genannt. Im Gegensatz dazu gibt es auch prozedurale Programmierung, bei dem der Code immer nach einem bestimmten Schema und in einer vordefinierten Reihenfolge ausgeführt wird. Kommt es zur Interaktion mit einem Element der GUI, stellt diese Aktion ein Event dar und der entsprechende Codeabschnitt wird ausgeführt. Dieser Block wird dann jedoch wieder prozedural in einer bestimmten Reihenfolge abgearbeitet.

In Matlab kann mit der Eigenschaft `'Interruptible'` definiert werden, was passieren soll, wenn gerade Codeabschnitte ausgeführt werden und ein anderes Event währenddessen triggert. Die Eigenschaft kann entweder `'on'`, der Code darf von einem anderen Event unterbrochen werden, oder `'off'` sein, d. h. der Code darf nicht unterbrochen werden. Mit der Eigenschaft `'BusyAction'` kann definiert werden, was passieren soll, wenn der Code nicht unterbrochen werden darf. Das unterbrechende Event kann dabei entweder in eine Warteschlange gereiht oder ignoriert werden. Der Matlab-Code wird, im großen Unterschied zu C++-Code, während der Laufzeit interpretiert und ausgeführt. Das heißt eine Kompilierung des Matlab-Teils des Codes ist nicht notwendig.

Die Hauptaufgabe des Matlab-Codes und der GUI ist es, dem Benutzer graphisches Feedback und Einstellungsmöglichkeiten bereitzustellen, sowie die Messdaten zu verarbeiten und in den Diagrammen anzuzeigen. Der Matlab-Code dient als Basis des Programms, von welcher aus alle anderen Programmteile gesteuert werden.

C++-Code: C++ ist eine objektorientierte Programmiersprache und hat sich 1979 als Erweiterung aus der Sprache C entwickelt. Einer der größten Vorteile dieser Programmiersprachen ist es, dass diese sehr hardwarenah sind im Gegensatz zu höheren Sprachen wie Matlab oder VBA. Durch diese Maschinennähe sind diese Programme sehr schnell, enthalten jedoch auch viel Fehlerpotential, da auf den physikalischen Speicher (RAM) direkt zugegriffen werden kann und dieser selbst reserviert werden muss. Wird mit einem Zeiger auf nicht belegte Adressen im Speicher (Nullzeiger) oder auf von anderen Programmen belegte Speicheradressen zugegriffen, kann es zum Absturz des Programms kommen. Im Gegensatz dazu wird die Speicherreservierung des zuvor genannten Matlab-Codes von Matlab selbst durchgeführt, wodurch sich der Anwender in den meisten Fällen keine Gedanken darüber machen muss. Wie zuvor erklärt, benötigt es zur Ausführung von C++ eine Kompilierung des Programmcodes. Dadurch entstehen eigenständige ausführbare Dateien bzw. Programme. Im einfachsten Fall handelt es sich dabei um eine .exe-Datei. Um die Weiterverarbeitung durch Matlab zu erleichtern, wurden in dieser Arbeit ausführbare Dateien, welche Matlab interpretieren kann, erzeugt, sogenannte MEX-Funktionen. Der C++-Code kann, wie anfangs bei der Entwicklung der Software gemacht, in Microsoft Visual Studio geschrieben werden. Jedoch muss hierfür der Compiler so umgeändert werden, dass eine MEX- anstatt einer .exe-Datei erzeugt wird. Dies ist zwar grundsätzlich möglich, stellt sich aber als sehr umständlich heraus. Die zweite, einfachere Möglichkeit ein MEX-File zu erzeugen ist es, den Code in Matlab zu schreiben und Matlab selbst das File erzeugen zu lassen. Dazu muss ein Compiler am benutzten PC vorhanden sein. Mit einer einfachen Befehlszeileneingabe `mex Filename.cpp` wird anschließend automatisch eine Datei mit der richtigen Dateierweiterung erzeugt. Je nach verwendetem Betriebssystem sind das .mexw64- oder .mexw32-Dateien, abhängig davon, ob es sich um ein 32-Bit oder 64-Bit System handelt.

Der primäre Zweck des C++-Codes ist es, die Kommunikation über die COM-Schnittstelle mit der Omicron-Software herzustellen, Parameter in der Software zu setzen und auszulesen sowie während einer Messung die Messdaten über zwei verschiedene Pipes zu lesen und an Matlab zu übergeben. Wie in Bild 26 dargestellt, wird durch einen Compiler ein MEX-File erzeugt, welches die Kommunikation zwischen Matlab und der Messsoftware herstellt. Theoretisch könnte aus dem C++-Code auch ein eigenständiges Programm geschrieben werden, welches die Kommunikation durchführt.

MEX-Funktionen: Diese sind ausführbare Dateien, kompiliert aus C++-Code, und entsprechen den unter Windows bekannten .dll-Dateien (auch dynamisch verlinkte Subroutinen oder Bibliotheken). Der große Vorteil von MEX-Funktionen ist es, dass diese wie herkömmliche Matlabfunktionen benutzt und aufgerufen werden können. Auch die Parameterübergabe und -ausgabe kann, wie sonst auch, in einfachen Klammern erfolgen, `Ausgabe = Filename(Parameter1, Parameter2, ...)`. Es handelt sich um C++-Code den Matlab zu einer MEX-Funktion umwandelt und interpretieren kann. Viele Operationen und Algorithmen, welche in Matlab viel Rechenleistung und Zeit in Anspruch nehmen würden, können durch die Hardwarenähe von C++ um einiges beschleunigt werden.

Damit aus C++ eine solche Datei erzeugt werden kann, sind jedoch einige kleine Änderungen im Code selbst vorzunehmen:

- Die aus C++ bekannte Main-Funktion (vgl. Listing 2) muss durch `void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray* prhs[])` ersetzt werden. `nlhs` und `nrhs` sind ganzzahlige Zahlen, welche die Anzahl der Ein- (`nlhs`; l steht für links) und Ausgabeparameter (`nrhs`; r steht für rechts) angibt. Die Parameter selbst bzw. Pointer oder Zeiger auf das erste Element in den Arrays, sind in den sogenannten `mxArrays* plhs[]` und `prhs[]` enthalten.
- Preprozessor Direktive `#include "mex.h"` zur Einbindung der MEX-Bibliothek. Damit werden alle Funktionen zur Manipulation von `mxArrays` in das Programm integriert.
- Von Anfang bis Ende der Hauptfunktion `void mexFunction(...)`, im Code mit `{}` erkennbar, kann mit normaler C++-Syntax gearbeitet werden.

Nachteil solcher Funktionen ist es, dass die Fehlersuche im Code nicht möglich ist, da es keinen, wie aus Matlab oder Visual Studio bekannten Debug-Modus gibt. Aufgrund dessen ist sorgfältiges Programmieren sehr wichtig, da es beim Ausführen fehlerhafter MEX-Funktionen schnell zum Absturz des gesamten Matlab-Programms kommt und die Fehlersuche nur über Umwege und sehr zeitaufwendig durchgeführt werden kann. Genau genommen gibt es einen offiziellen Weg, MEX-Files zu Debuggen. Es wird dafür aber eine Version von Microsoft Visual Studio vorausgesetzt. Das Debuggen des Codes kann jedoch mit der `mexPrintf()`- oder durch vorzeitiges Einfügen der `return()`-Funktion durchgeführt werden, um den fehlerbehafteten Codeabschnitt einzugrenzen. Ein weiterer Nachteil ist es, dass schnelle Änderungen des Codes, wie bei reinem Matlab-Code, nicht möglich ist. Es bedarf immer einer Kompilierung um Änderungen zu übernehmen.

Die Funktion der MEX-Files stellt die Verbindung zwischen Matlab und C++ respektive der Omicron-Software dar und ist ein zentrales Element zum Empfangen der Messdaten.

3.4.5 Testdokumentation und Anwendungsbeispiele

Testdokumentation

Das Programm hat sich im Rahmen dieser Masterarbeit entwickelt und ist an Umfang sowie Funktionsweise über die Zeit angewachsen. Die Software, sowie alle darin enthaltenen Funktionen wurden auf ihre fehlerfreie Funktionsweise getestet. Diese Prozedur gestaltete sich so, dass alle Bedienelemente, welche direkt über die COM-Schnittstelle mit der Omicron-Software kommunizieren, durch die Eingabe im *Online PDDC-Analyzer* und der korrekten Übernahme in der Messsoftware von Omicron getestet wurden. Alle Einstellungsmöglichkeiten im *Online PDDC-Analyzer* selbst, wie z. B. Diagrammeinstellungen, wurden ebenfalls auf ihre korrekte Funktion hin überprüft. Dazu wurden Einstellungen gesetzt und getestet, ob diese korrekt übernommen wurden. Die nachfolgenden Tabellen 7 und 8 listen die wichtigsten getesteten Funktionen und das dazugehörige Testergebnis auf.

Tabelle 7: Beschreibung und Ergebnis der durchgeführten Tests der Main-GUI

Main-GUI	
Beschreibung	Ergebnis
Verbinden mit der Omicron-Software	✓
Erkennung falls Teile des Messsystem nicht verfügbar sind	✓
Korrekte Anzeige der Verbindungsanzeige	✓
Einlesen der Startparameter aus der Omicron-Software	✓
Korrekte Anzeigen der Statusanzeige	✓
Löschen der Statusanzeige	✓
Korrekte Initialisierung der Omicron-Software	✓
Umschaltung zwischen verschiedenen Devices	✓
Übernahme der Filter- und TE-Detektionseinstellungen	✓
Zurücksetzen der Filter- und TE-Detektionseinstellungen	✓
Test der Exportfunktion	✓
Korrektes Schließen der GUI und Beenden des Programms	✓
Umschaltfunktion des Buttons Start/Stop Recording	✓

Tabelle 8: Beschreibung und Ergebnis der durchgeführten Tests der Stream-GUI

Stream-GUI	
Beschreibung	Ergebnis
Lesen und korrekte Anzeige der Messdaten	✓
Korrekte Anzeige der Diagramme	✓
Korrekte Anzeige der Mess- und Statistikanzeigen	✓
Korrekte Anzeige des Verstärkerlevels (Level)	✓
Setzen einer neuen Verstärkerstufe (Gain)	✓
Korrekte Eingabe und Übernahme des PD-Thresholds	✓
Löschen der Diagramme	✓
Speichern von Screenshots im Exportordner	✓
Starten, beenden und Speichern eines Messfiles	✓
Speichern eines Spannungsfiles	✓
Achseinstellungen der Time-Functions	✓
Achseinstellungen der NoDi*-Diagramme	✓
Umschaltung zwischen Pulse-Count- und Δt -Diagramm	✓
Umschaltung des Datacursors	✓
Kalibration der Spannung	✓
Kalibration der Teilentladungen	✓
Umschaltung von External View	✓
Achseinstellungen von External View	✓
Achseinstellungen des Pulse-Count-Diagramms	✓
Beenden des Streams	✓
Korrektes Schließen der GUI	✓

Anwendungsbeispiele

Im Rahmen der Testdokumentation wurden die Funktionen auf ihre Korrektheit hin überprüft. Dieses Kapitel soll das Programm anhand konkreter Messungen als gesamtes in Bezug auf Funktionsweise und Benutzerfreundlichkeit hin betrachten. Hierzu wurden im Labor Messungen verschiedener, ausgewählter Defekte durchgeführt, welche als Anwendungsbeispiele dienen sollen.

Grundrauschen: Grundrauschen mit einer Rate von ca. 300 Impulsen/s wurde über 63 Stunden gemessen, wobei es zu keinen merklichen Performanceverlusten kam. Wird die Messzeit deutlich erhöht, kann es aufgrund der großen Datenmengen von Pulse-Count-, Δt - und Spannungsdiagramm zu Performanceeinbußen kommen. Dies kann jedoch durch Zurücksetzen der Diagramme, durch Klicken des Buttons *Clear All Diagrams*, umgangen werden. Grundsätzlich sollte aber die Aufnahme von Grundrauschen vermieden werden, vor allem da hier sehr große Impulswiederholraten auftreten.

Innere Teilentladungen: Diese Fehlerart stellt aufgrund der geringen Impulswiederholrate üblicherweise kein Problem dar. Es kommt bei derartigen Defekten typischerweise zu einer Impulswiederholrate von ca. 250 Impulse/Stunde, bei Polaritätswechseln können jedoch entsprechend mehr Impulse auftreten.

Spitze-Platte-Anordnung: Wird die Spannung sehr weit über die TE-Einsatzspannung erhöht, kommt es zu hoher TE-Aktivität an der Spitze. Dieser Effekt ist bei Gleichspannung negativer Polarität sehr viel ausgeprägter. Hierbei treten sehr viele TE-Impulse innerhalb kürzester Zeit auf. Da das Programm 100.000 Impulse in 30 ms bearbeiten kann, kommt es bei Überschreitung dieser Grenze zu einer Anzeigeverzögerung. Hier sei jedoch angemerkt, dass solche Messfiles schnell eine Dateigröße von einigen hundert MB haben, deshalb nur mehr schwer weiterverarbeitet werden können und somit längere Messungen dieser Art nicht sinnvoll sind. Vor allem, da auch die Software von Omicron, Probleme mit sehr riesigen Datenmengen bei DC-Messungen hat, wodurch diese regelmäßig einfriert und nicht reagiert.

Frei bewegliches, metallisches Partikel: Dieser Defekt wird ausführlicher in Kapitel 4 behandelt. Aufgrund der Bewegung des Partikels können hohe Impulswiederholraten auftreten, welche jedoch abhängig von der Art und hauptsächlich der Geometrie des Partikels sind. Im Rahmen der Messungen hat sich gezeigt, dass ein zylinderförmiges Partikel in Form eines Drahtstücks, aufgrund der scharfen Kanten, im Vergleich zu anderen Geometrien mehr Teilentladungen produziert. Hier wurden mittlere zeitliche Abstände zwischen den Impulsen Δt von ca. 4 μs bis 18 μs beobachtet, was im Schnitt ca. 16.000 bis 80.000 Impulse in 300 ms bedeutet. Dieser Defekt sollte somit grundsätzlich kein Problem für den *Online PDDC-Analyser* darstellen, wobei auch hier die Omicron-Software teilweise Probleme mit so hohen Impulswiederholrate hatte.

Freies Potential: Bei freien Potentialen wurden mittlere Δt von Millisekunden (beispielweise in SF₆ und Gleichspannung positiver Polarität) bis Sekunden beobachtet, was keine Performanceprobleme bedeutet.

3.4.6 Benutzerhandbuch

Dieses Kapitel soll als Benutzerhandbuch dienen und eine schnelle und einfache Übersicht darüber geben, wie Messungen mit der Software durchgeführt werden können.

Voraussetzungen

- Omicron Hardware und Software in der Version 1.6.6.25
- MCU mit den Lizenzen für:
 - COM-Interface
 - Expert Mode
 - DC-Measurements
- Matlab Vollversion in der Version R2016a für .p-Files
- Matlab Runtime in der Version R2016a für *PDDC_Analyzer_Online.exe*-Datei

Durchführung von Messungen

- Starten der Software

Die Software kann entweder durch Ausführen der .exe-Datei oder in Matlab durch Rechtsklick auf die Datei *PDDC_Analyzer_Online.p* und Auswahl von *Run* aus dem Kontextmenü gestartet werden. Beim Starten der Software erscheint die in Bild 19 gezeigte GUI.

- **Verbinden mit Omicron:** Mit Klick auf *Connect...* wird das Programm versuchen, eine Verbindung zur Omicron-Software aufzubauen. Ist bereits eine Instanz der Software offen (das kann auch ein Streamfile im Offline-Modus darstellen), dann wird zu dieser Instanz verbunden. Falls noch keine Instanz geöffnet wurde, öffnet das Programm selbstständig eine und verbindet sich.

- **Device auswählen:** In den *General Settings* rechts oben, kann das Device ausgewählt werden. Dies ist nur von Relevanz, falls mehrere MPD angeschlossen werden.

- **Initialisierung der Omicron-Software:** Mit Klick auf *Initialize Device* wird die Omicron-Software auf Messungen für Gleichspannung angepasst.

- **Filter- und Exporteinstellungen:** Die gewünschten Filtereinstellungen zur Detektion von Teilentladungen können rechts in der GUI angepasst werden. Der Ordner sowie der Projektname zum Speichern von Messfiles und Screenshots kann durch Anklicken der Checkbox im Panel *Export* eingestellt werden. Mit Klick auf *Update Device* werden die Einstellungen in der Omicron-Software übernommen.

- **Starten der Messung:** Durch Klicken des Buttons *Start Recording* öffnet sich nach Initialisierung eine weitere GUI mit den Diagrammen, Messanzeigen und weiteren Einstellungen.

- **Kalibration der Ladung und Spannung:** Im Tab *Calibr. & Ext.* kann der Ladungs- bzw. Spannungswert, auf den kalibriert werden soll, eingestellt werden. Mit Klick auf *Calibrate* erfolgt die Kalibrierung.
- **Anpassen von Verstärkung und TE-Threshold:** Die Verstärkerstufe im Tab *PD* kann mit dem Regler *Gain* verändert werden und ist so zu wählen, dass die Anzeige *Level* bei den zu erwartenden TE-Impulshöhen im grünen Bereich liegt. Der TE-Threshold kann im selben Tab angepasst werden und ist ein Schwellpegel für den Ladungswert unter dem keine TE-Impulse erfasst werden. Er ist so zu wählen, dass kein Grundrauschen aufgenommen wird und wird bei Änderungen direkt in der Omicron-Software übernommen. Die Höhe des Grundstörpegel kann sich beim Verändern der Verstärkerstufe ändern, wodurch der TE-Threshold gegebenenfalls angepasst werden muss.
- **Anpassen der Diagramme:** Die Zeitachse der Diagramme wird automatisch gesetzt. Die Grenzen des $Q(t)$ -Diagramms, des *Pulse Count*-Diagramms und der NoDi*-Diagramme können im Tab *Options* eingestellt werden. Zusätzlich können, je nachdem wie gemessen wird, die Ladungswerte invertiert werden. Die Einstellungen werden mit Klick auf *Apply* übernommen. Im Panel *Display Options* kann das mittlere Diagramm umgeschaltet werden. Die Einstellungen der Zeit und des Thresholds des *Pulse Count*-Diagramms werden im Tab *Calibr. & Ext.* und *External View-PD Count* und mit Klick auf *Set* vorgenommen.
- **Messanzeigen:** Die Messanzeigen oben in den Tabs für den Ladungswert, der Spannung und Messzeit, sowie die Diagramme werden alle 300 ms aktualisiert. Es können höchstens 100.000 Daten in 300 ms erfasst werden. Wird diese Grenze überschritten, ändert sich der *Input Data Status* von *Balanced* zu *Overdriven*. Das Spannungs- und $\Delta t(t)$ -Diagramm zeigen Mittelwerte von 300 ms an. Im Tab *PD* gibt es zusätzlich eine Statistik der erfassten Impulse. Die aufgezeichneten Daten können mit Klick auf *Clear All Diagrams* zurückgesetzt werden.
- **Screenshots aufnehmen:** Durch Klicken des Buttons *Screenshot* im *PD*-Tab können Screenshots erstellt werden. Es werden die Time-Plots, NoDi*-Diagramme, sowie ein Bild der gesamten Programmoberfläche gespeichert. Der Pfad ist dabei derjenige, welcher in den Exporteinstellungen gewählt wurde.
- **Aufnahmen von Messfiles:** Soll ein Textfile zur späteren Auswertung aufgenommen werden, kann die Aufzeichnung mit Klick auf *Go* gestartet werden. Dazu werden die Diagramme zurückgesetzt. Bei nochmaligem Drücken des Buttons stoppt die Aufnahme des Messfiles und dieses sowie eine *-vData.txt*-File mit den Spannungsdaten in 300 ms Abständen wird abgespeichert.
- **Beenden und Schließen:** Wird in der Main-GUI *Stop Recording* gedrückt, so bleibt die Stream-GUI offen, es werden jedoch keine neuen Messdaten mehr angezeigt. Gleiches passiert, wenn die Stream-GUI während des Streams geschlossen oder die Verbindung zur Omicron-Software unterbrochen wird. Die Verbindung kann auch manuell mit Klick auf *Disconnect* oder durch Schließen der Main-GUI getrennt werden.

In Bild 27 ist der Programmablaufplan zur Durchführung von Messungen dargestellt.

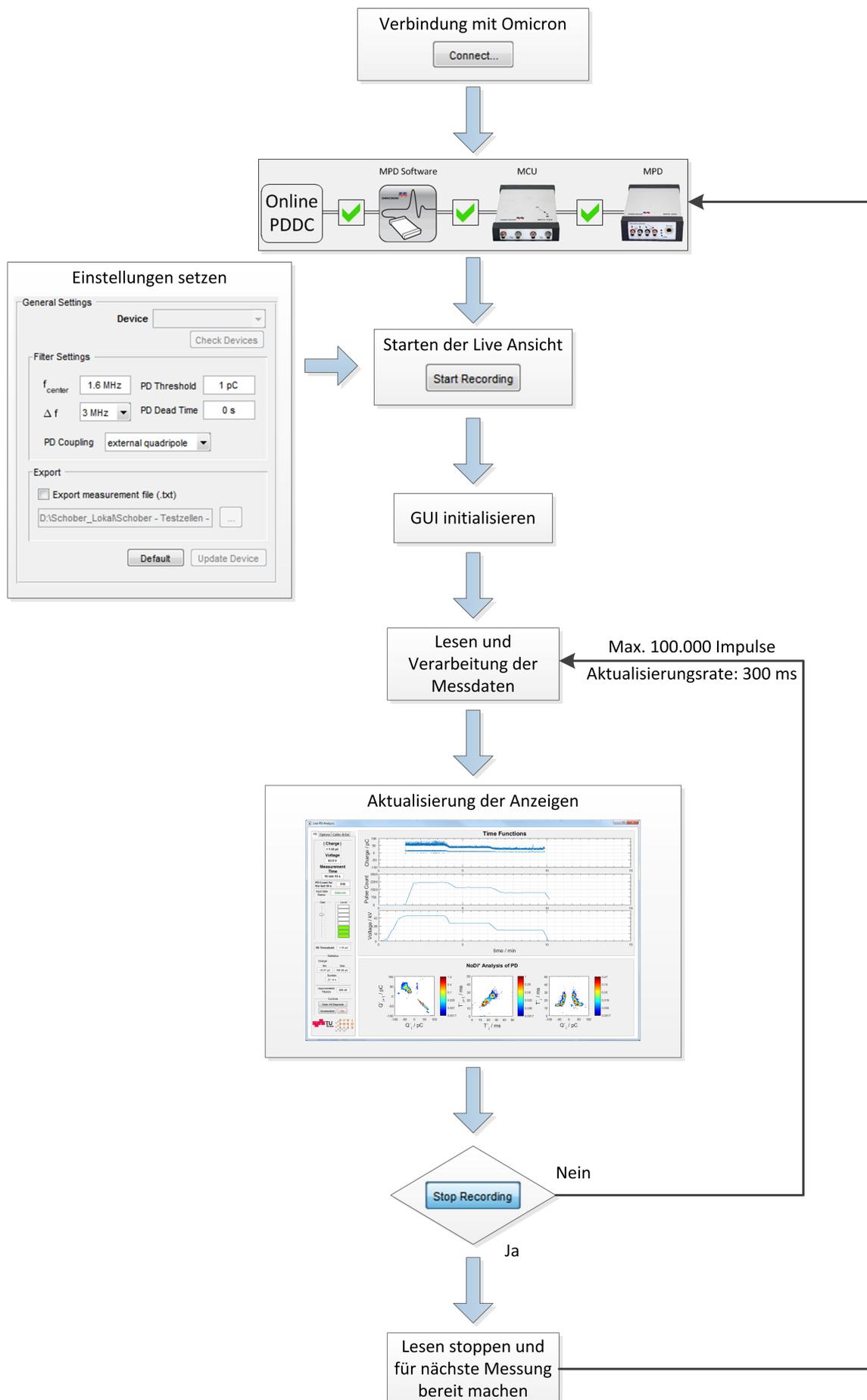


Bild 27: Grundsätzlicher Programmablaufplan

4 TE-Messungen an freien metallischen Partikeln in GIS und GIL

4.1 Messaufbau

Die TE-Messungen der verschiedenen Partikel wurden bei Gleich- und Wechselspannung durchgeführt. Bei dem TE-Messsystem handelte es sich um ein System der Firma Omicron mit AKV CPL 542, MCU 502 und MPD 600, sowie das Programm *Omicron Software for MPD and MI* (Version 1.6.6.25). Der Messaufbau der AC-Messungen bestand aus Transformator (MWB TEO 100/10), einem 50-k Ω -Widerstand zur Strombegrenzung und einem Koppelkondensator mit $C = 1,2$ nF. Die Auskopplung der TE-Impulse erfolgte im Pfad des Koppelkondensators. Über diesen Kondensator wurde, nach Kalibration der Spannung in der Messsoftware, die Prüfspannung gemessen. Der Prüfling in Form einer Testzelle mit Partikel befand sich im Parallelzweig (Bild 28).

Der Aufbau für die DC-Messungen ist in den Bildern 29 und 30 dargestellt. Nach dem Transformator wurde zur Erzeugung der Gleichspannung ein Einweggleichrichter geschaltet. Dieser bestand aus einer Diode, welche einen Serienwiderstand von 100 k Ω besaß, und einem Glättungskondensator mit $C = 10$ nF. Je nach gewünschter Polarität der Spannung wurde die Diode im Aufbau umgedreht. Ein Polaritätswechsel (polarity reversal) wurde während der Messungen nicht durchgeführt. Anschließend an den Gleichrichter war ein Serienwiderstand mit $R = 10$ M Ω geschaltet, um im Fall eines Durchschlags den Kurzschlussstrom zu begrenzen. Die Spannungsmessung erfolgte über einen parallel geschalteten ohmschen Spannungsteiler. Dieses Spannungssignal wurde direkt an den V-Eingang des MPD 600 geführt, wobei das Teilverhältnis von 140 M Ω /214 k Ω so gewählt wurde, dass die maximale Spannung des Eingangs von 60 V_{eff} nicht überschritten wurde. Die Auskopplung der Impulse wurde abermals mit einem Kondensator mit $C = 1,2$ nF parallel zum Prüfling durchgeführt.

Zusätzlich wurden bei jeder Messung Videos mit einer High-Speed-Kamera aufgenommen, um die Partikelbewegungen zu dokumentieren. Bei der Kamera handelte es sich um das Model *EXILIM EX-F1* der Firma CASIO. Die Kamera wurde mit einem Stativ oberhalb der Testzelle angebracht. Im Deckel der Testzelle befand sich ein Sichtfenster, durch welches gefilmt werden konnte (Bild 30a). Die Testzelle sowie die Elektrodenanordnung sind in Bild 31b dargestellt. Die Testzelle wurde im weiteren Verlauf der Messungen in einem Winkel von 20 Grad gegenüber der Horizontalen platziert, um die Bewegungen des Partikels durch das Fenster im Deckel der Testzelle besser filmen zu können. Das Stativ wurde zusätzlich geerdet, um unerwünschte Teilentladungen zu vermeiden. Die Framerate der Videoaufnahmen wurden mit 300 fps gewählt, was in einer sehr kurzen Belichtungszeit resultiert. Zur Bereitstellung von genügend Licht, wurde unter der Kamera ein Flutlichtstrahler angebracht und während der Messungen eingeschaltet.

Die Omicron-Software wurde, wie in Kapitel 3.2.2 beschrieben, auf DC-Messungen eingestellt. Kalibriert wurde der Prüfaufbau vor jeder Messung parallel zum Prüfling

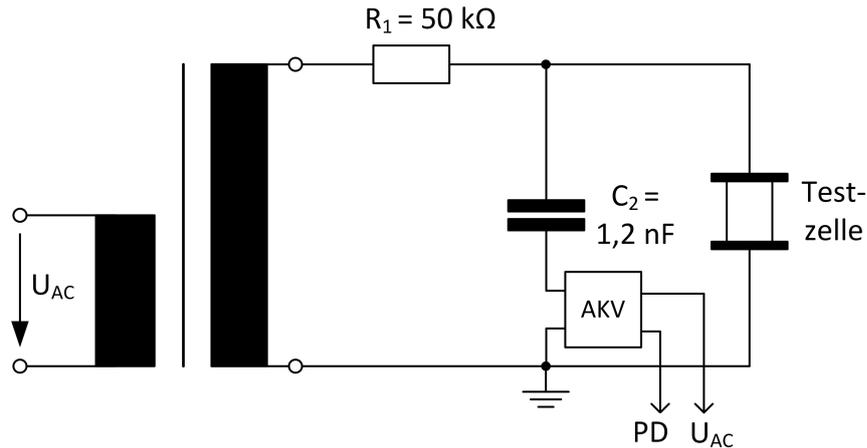


Bild 28: Messschaltung bei Wechselspannung

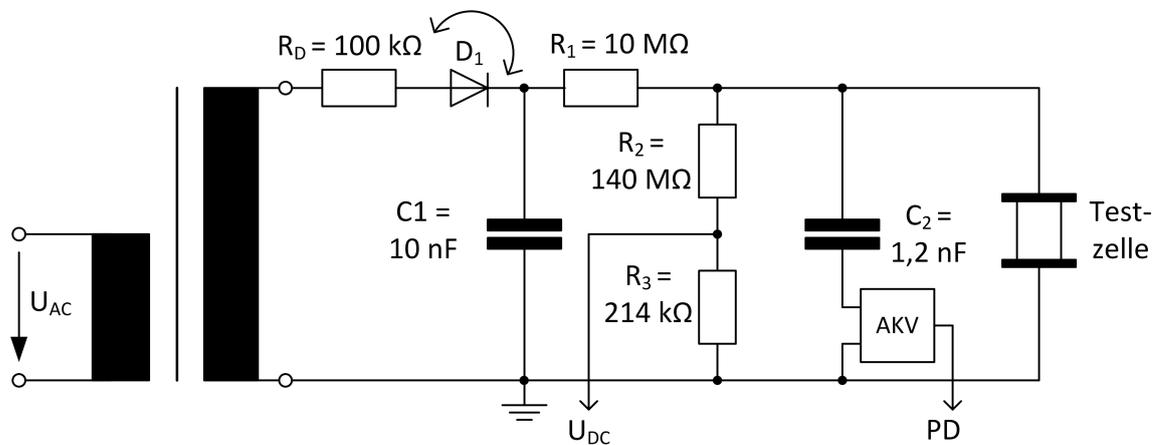


Bild 29: Messschaltung bei Gleichspannung

mit einem TE-Kalibrator der Firma Power Diagnostix vom Typ CAL1A/DP. Der TE-Threshold wurde auf 1 pC gesetzt. Die Filtereinstellungen wurden breitbandig mit einer Mittenfrequenz von $f_m = 1,6$ MHz und einer Bandbreite von $\Delta f = 3$ MHz gewählt, um eine richtige Polaritätserkennung der TE-Impulse zu gewährleisten. Das Grundrauschen lag mit ca. $Q = \text{ca. } 70$ fC im Labor sehr niedrig.

Die Verstärkereinstellung wurde, je nach Impulshöhen, manuell gesetzt, sodass der Verstärker nicht übersteuert wurde. Beim Verstellen der Verstärkerstufe ist darauf zu achten, dass sich dadurch auch die Ladungshöhe des Grundrauschens verändert, weshalb gegebenenfalls der Wert des TE-Thresholds angepasst werden musste. Jedenfalls wurde darauf geachtet, die Aufnahme von Grundrauschen zu vermeiden, damit die Datensätze nicht zu groß ausfielen.

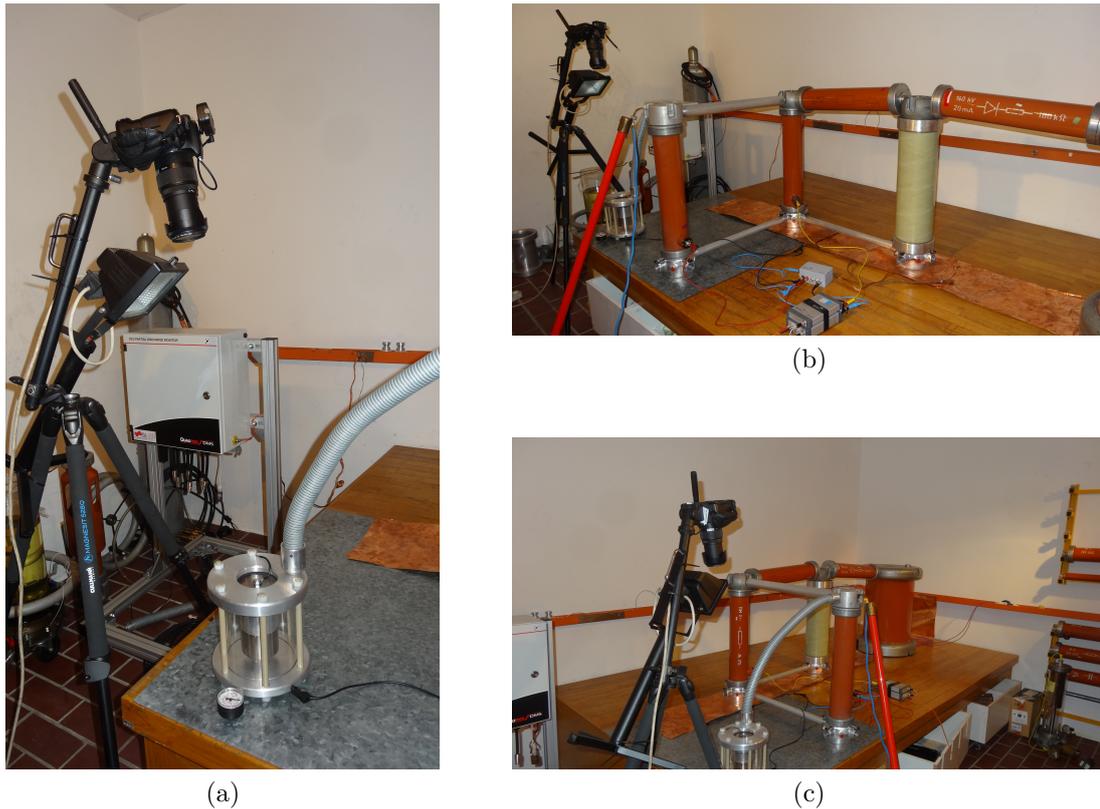


Bild 30: Messaufbau zur TE-Messung bei Gleichspannung:

- (a) High-Speed-Kamera mit Testzelle
- (b) Messaufbau von rechts nach links: Gleichrichter, ohmscher Spannungsteiler, Koppelkondensator und dahinter die Testzelle, mittig das TE-Messsystem
- (c) Messaufbau von der anderen Seite

4.2 Beschreibung der Defekte

Die Defekte wurden anhand einer Testzelle, in welcher metallische Partikel frei beweglich platziert wurden, erzeugt. Insgesamt wurden Messungen an fünf unterschiedlichen Partikeln durchgeführt:

- Kugel aus Stahl mit einem Durchmesser von $d = 2$ mm,
- Kugel aus Stahl mit einem Durchmesser von $d = 2,5$ mm
- Zylinderförmiges Partikel aus Stahl mit einer Länge von $l = 4,44$ mm und einem Durchmesser von $d = 0,5$ mm
- Zylinderförmiges Partikel aus Stahl mit einer Länge von $l = 1,58$ mm und einem Durchmesser von $d = 0,5$ mm
- Span aus Stahl mit einer Länge von $l = \text{ca. } 2$ mm

Der Span wurde gewählt, um auch in der Realität vorkommende Partikel, z. B. durch Montagefehler einer GIS, zu vermessen. Die drei anderen Defekte sind eher als ideale Labordefekte anzusehen und kommen real nur selten vor. Die Bezeichnungen sowie Beschreibungen und Gewichtsangaben sind in Tabelle 9 zu finden. Wie zu

Tabelle 9: Partikel an denen TE-Messungen durchgeführt wurden

Bezeichnung	Beschreibung	Gewicht in mg
S1	Kugel ($d = 2,5$ mm)	64,5
S2	Kugel ($d = 2$ mm)	32,8
W1	Draht ($d = 0,5$ mm, $l = 4,44$ mm)	3,8
W2	Draht ($d = 0,5$ mm, $l = 1,58$ mm)	2,4
SP	Stahlspan ($l = 2$ mm)	0,5

erkennen ist, hat der Stahlspan das geringste Gewicht und die Kugeln mit größerem Durchmesser das höchste Gewicht. Das Gewicht ist, wie in Kapitel 2.4 beschrieben wurde, für die Bewegung und die Abhebefeldstärke von Bedeutung (Tabelle 4).

Die Partikel wurden bei Wechsellspannung, sowie mit Gleichspannung positiver und negativer Polarität unter Variation des Drucks und der Spannungsamplitude gemessen. Bei allen Messungen wurde die Testzelle mit SF_6 gefüllt. Der Druck wurde mit $p = 0,2$ MPa und $p = 0,6$ MPa (absoluter Druck) gewählt. Die Messmatrix ist in Tabelle 10 zu finden und dient als Übersicht zu den durchgeführten Messungen. In Bild 31a sind die Mikroskopaufnahmen der Partikel und ein Größenvergleich dargestellt.

Tabelle 10: Messmatrix zu den durchgeführten Messungen

Defekt	AC		DC+		DC-	
	SF_6		SF_6		SF_6	
	0,2 MPa	0,6 MPa	0,2 MPa	0,6 MPa	0,2 MPa	0,6 MPa
S1	✓	✓	✓	✓	✓	✓
S2	✓	✓	✓	✓	✓	✓
W1	✓	✓	✓	✓	✓	✓
W2	✓	✓	✓	✓	✓	✓
SP	✓	✓	✓	✓	✓	✓

4.3 Messprozedur

Bei jeder Messung wurde die Testzelle mit Isopropanol gereinigt, neu verschraubt, mit Hilfe einer Vakuumpumpe evakuiert und anschließend zunächst mit einem Druck von bis zu $p = 0,2$ MPa SF_6 und danach bis zu einem Druck von $p = 0,6$ MPa (absolut) gefüllt. Die Partikel wurden jedes Mal in der Mitte der unteren Elektrode platziert, um Vergleiche anstellen zu können und eine definierte Ausgangssituation zu erhalten. Vor jeder Messung wurde der Prüfaufbau auf 100 pC kalibriert. Das Grundrauschen lag im fC-Bereich, weshalb ein TE-Threshold von 1 pC, teils 2 pC, in Abhängigkeit der Verstärkerstufe, eingestellt wurde. Bei den Messungen mit Wechsellspannung wurde der Verstärker automatisch von der Software geregelt, bei Gleichspannung jedoch händisch eingestellt. Der Messaufbau wurde zu Beginn im

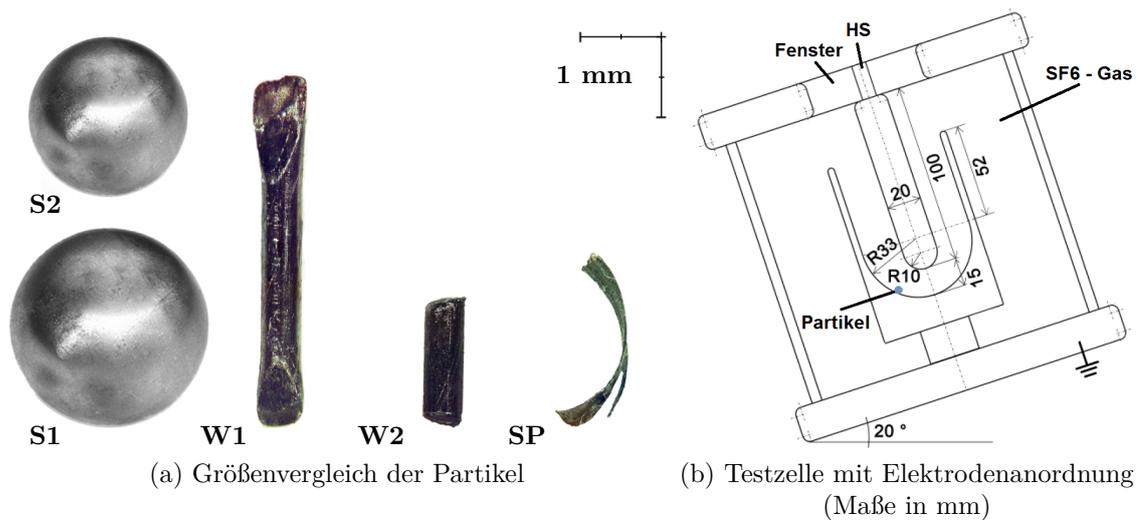


Bild 31: Größenvergleich der Partikel an denen die Messungen durchgeführt wurden (a) und verwendete Testzelle (b) [37]

Leerlauf ohne Testzelle auf TE-Freiheit untersucht. Dabei traten bis 60 kV keine Teilentladungen über dem eingestellten TE-Threshold von 1 pC auf, was als sehr gut betrachtet wurde.

Zur Ermittlung der Teilentladungseinsetzspannung U_E wurde die Spannung langsam erhöht, bis sich wiederholende Teilentladungen auftraten. Das war dann der Fall, wenn die Coulomb'sche Kraft auf das Partikel die Gravitationskraft überwunden hat und das Partikel abgehoben ist. Die Einsetzspannung wurde zum nächsten Vielfachen von fünf aufgerundet und danach in 5-kV-Schritten erhöht. Die Teilentladungen und Videos wurden bei jeder Spannungsstufe für 30 Sekunden bei Gleichspannung und für 1 Minute bei Wechselspannung aufgezeichnet. Diese Dauer wurde als ausreichend erachtet, da die Partikel durch die Bewegung im eingeschwungenen Zustand ausreichend viele Teilentladungen hervorgerufen haben. Vorausgesetzt, dass das Partikel während einer Messung nicht bereits zuvor aus der Elektrodenanordnung gesprungen ist, wurde die Spannung bei Wechselspannung bis $U_{eff} = 55$ kV und bei Gleichspannung bis $U_{DC} = \pm 55$ kV stufenweise erhöht und anschließend bis zum Aussetzen der Teilentladungen wieder gesenkt. Die Spannungsfahrt ist in Bild 32 dargestellt. Ist das Partikel während einer Messung herausgesprungen, wurde die Messung abgebrochen und entschieden, ob genug Daten aufgenommen wurden oder die Messung wiederholt werden sollte. Der Ablauf wurde bei Wechsel- sowie Gleichspannung durchgeführt. Für jede Spannungsstufe wurde ein eigenes Messfile, sowie ein eigenes Video abgespeichert. Nach den Messungen wurden die Videoaufnahmen und die TE-Messfiles betrachtet und ausgewertet.

Beispielbilder dieser Videoaufnahmen sind in den Bildern 33 und 34 dargestellt. Die Partikel wurden farblich hervorgehoben. Es sind jeweils aufeinanderfolgende Einzelbilder dargestellt. Das Bild 33 zeigt die Hin-und-Her-Bewegung zwischen den Elektroden der Kugel S2 und das Bild 34 den Firefly-Effekt von Partikel W1 zum Zeitpunkt des Abhebens von der Hochspannungselektrode.

Um die Videos und die Messfiles zu synchronisieren, wurde die High-Speed-Kamera mit einer Software der Firma CASIO vom PC aus gesteuert. Es wurde darauf geachtet, beide Aufnahmen gleichzeitig zu starten, was sich jedoch aufgrund von Verzögerungen nicht zu 100 % realisieren lies. Wenn die Aufnahme der Kamera nicht aktiviert war, konnte in der Software am Bildschirm ein Livebild der High-Speed-Kamera betrachtet werden.

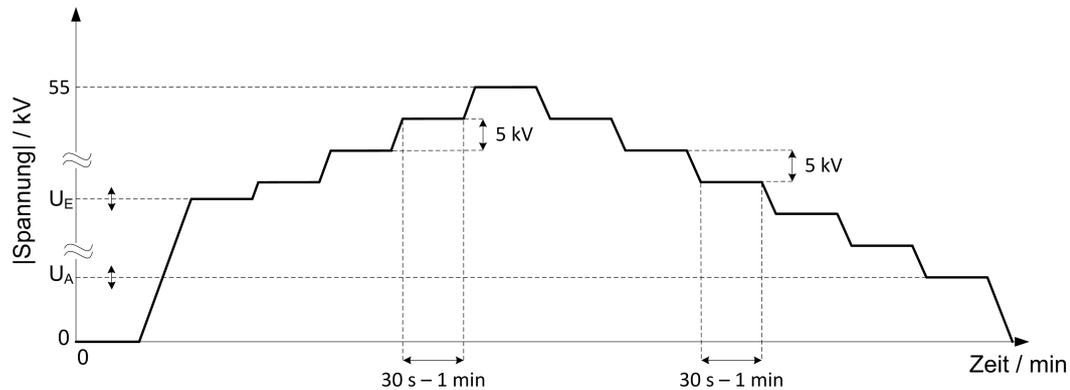


Bild 32: Spannungsfahrt der TE-Messungen, angefangen bei der Einsetzspannung U_E , bis maximal ± 55 kV und zur Aussetzspannung U_A mit einer Stufenhöhe von 5 kV und einer Stufenbreite von 30 s bei Gleichspannung bzw. 1 min bei Wechselspannung

4.4 Ergebnisse bei Wechselspannung

Diese Messungen wurden als Referenz für die Messungen bei Gleichspannung durchgeführt. Wurde die Spannung ausreichend erhöht, so erfuhr das Partikel aufgrund der Feldstärke eine Kraft und fing an sich zu bewegen. Die Kugeln S1 und S2 haben über die gesamte Spannungsfahrt eine Auf-und-Ab-Bewegung zwischen Hochspannungselektrode und Erdpotential vollzogen. Typische PRPD-Patterns sind in Bild 35 für verschiedene Drücke jeweils bei den TE-Einsetzspannungen dargestellt. Die Drahtstücke W1 und W2 zeigten bei geringen Spannungen ein ähnliches Muster dem der Kugeln. Bei höheren Spannungen wurden jedoch weitaus höhere Entladungen gemessen, wie Bild 36 zeigt. Diese ruhen daher, dass das Partikel an der HS-Elektrode wie eine Spitze wirkt und TE-Impulse sehr hoher Amplitude im Bereich einiger hundert pC erzeugt, bis es wieder in Richtung Erdpotential driftet. Im PRPD-Pattern kommt es durch dieses Phänomen zu einer Überlagerung der Muster einer Spitze auf HS (TE im Spannungsmaximum, Bild 7) und das eines idealen Partikels (Bild 35). Der Stahlspan (Bild 37) zeigt ein ähnliches Verhalten, bei niedrigen Feldstärken und einem Druck von $p = 0,6$ MPa annähernd das TE-Muster einer Kugel und darüber ähnlich dem der Drahtstücke, wobei die Entladungen in der negativen Halbwelle größer als in der positiven sind. Bei Drücken von $p = 0,2$ MPa hingegen ergab sich bereits ab der TE-Einsetzspannung ein Muster mit höheren Teilentladungsimpulsen (Bild 37b). Durch die geringere Isolationsfestigkeit des SF₆ bei geringerem Druck, wirkt der Span bereits bei niedrigeren Feldstärken wie eine Spitze auf Hochspannung.

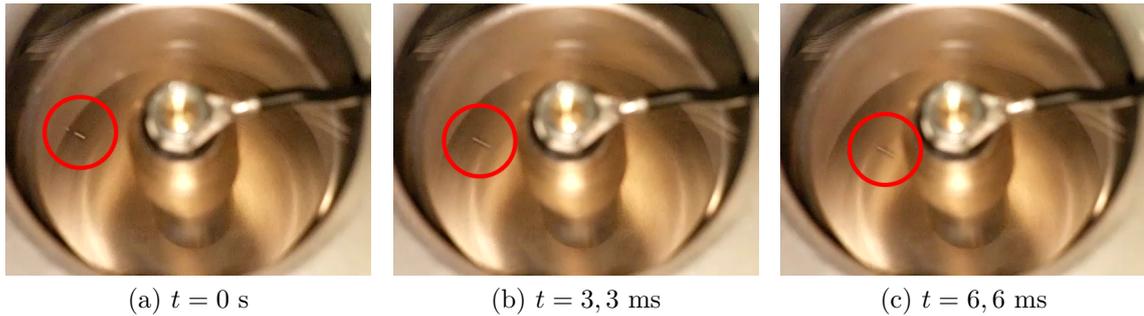


Bild 33: Momentaufnahmen aus einem High-Speed-Video von Partikel S2 zu verschiedenen Zeitpunkten

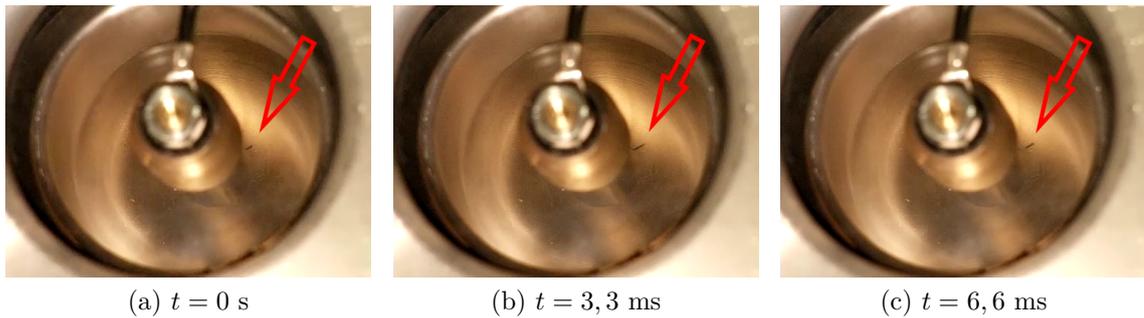


Bild 34: Momentaufnahmen aus einem High-Speed-Video von Partikel W1 zu verschiedenen Zeitpunkten

4.5 Ergebnisse bei Gleichspannung

4.5.1 Allgemeines

Die durchgeführten TE-Messungen in Kombination mit den Hochgeschwindigkeitsaufnahmen der High-Speed-Kamera haben zwei Effekte gezeigt. Einerseits den in Kapitel 2.4.2 bereits beschriebenen Firefly-Effekt und andererseits eine konstante Hin-und-Her-Bewegung des Partikels zwischen Hochspannungselektrode und Erdpotential. Dieser zweite Effekt wurde bei den Partikeln in Form einer Kugel, S1 und S2, festgestellt. Der Firefly-Effekt wiederum bei den Partikeln mit scharfen Kanten, W1, W2 und SP.

Bei allen Partikeln wurde eine Zunahme der Amplitude der Teilentladungen bei gleichzeitig kürzeren Δt zwischen den Impulsen bis zur maximalen Spannung von $U_{DC} = \pm 55 \text{ kV}$ festgestellt. Bei Verringerung der Spannung bis zur Aussetzspannung hat sich wiederum ein umgekehrtes Verhalten ergeben, wie in Bild 38 dargestellt.

4.5.2 Kugel

Partikel S1 und S2: Die beiden Kugeln mit den Durchmessern von 2 mm und 2,5 mm haben über die gesamte Spannungsfahrt das gleiche Verhalten gezeigt. Ab der Abhebefeldstärke bei der sich die Kugel zu bewegen begonnen hat, hat sich eine homogene Bewegung zwischen den beiden Elektroden ergeben. Diese Bewegung hat annähernd immer an den gleichen Stellen stattgefunden. Bis zur Aussetzspannung hat sich dieses Verhalten nicht geändert. Durch die konstante Bewegung kam es zu ei-

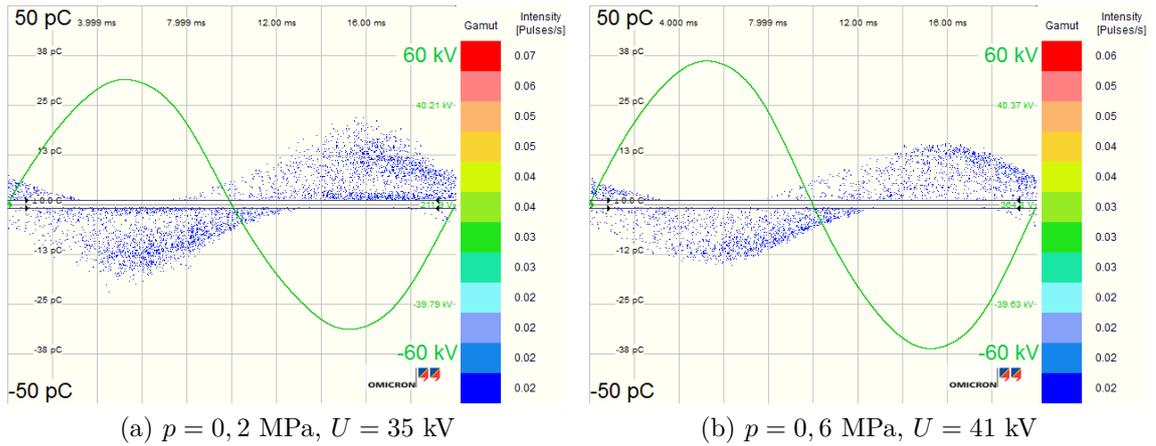


Bild 35: PRPD-Pattern der Kugel S2 über eine Dauer von 1 min jeweils bei der TE-Einsatzspannung für die Drücke $p = 0,2 \text{ MPa}$ (a) und $p = 0,6 \text{ MPa}$ (b)

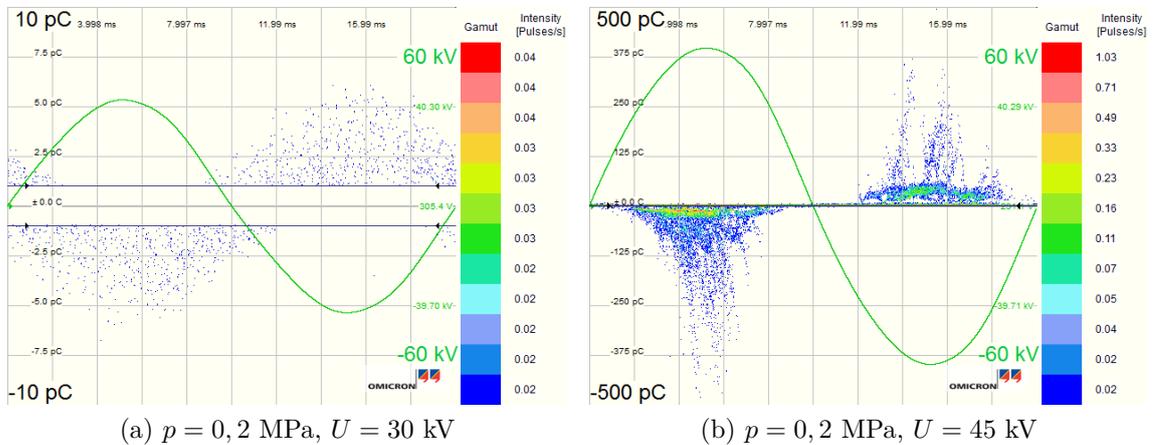


Bild 36: PRPD-Pattern des Partikels W2 über eine Dauer von 1 min bei der TE-Einsatzspannung (a) und einer Spannung von $U = 45 \text{ kV}$ für einen Druck von $p = 0,2 \text{ MPa}$ (b)

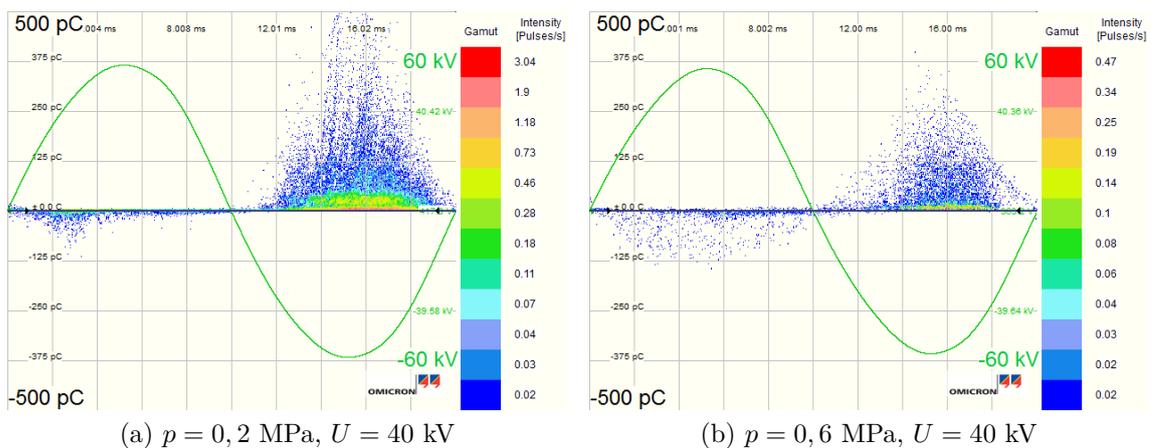


Bild 37: PRPD-Pattern des Stahlspans SP über eine Dauer von 1 min jeweils bei einer Spannung von $U = 40 \text{ kV}$ für die Drücke $p = 0,2 \text{ MPa}$ (a) und $0,6 \text{ MPa}$ (b)

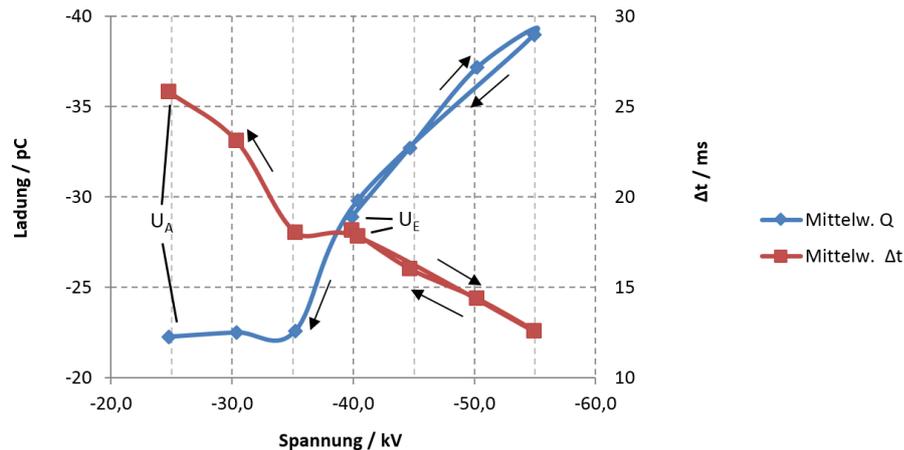


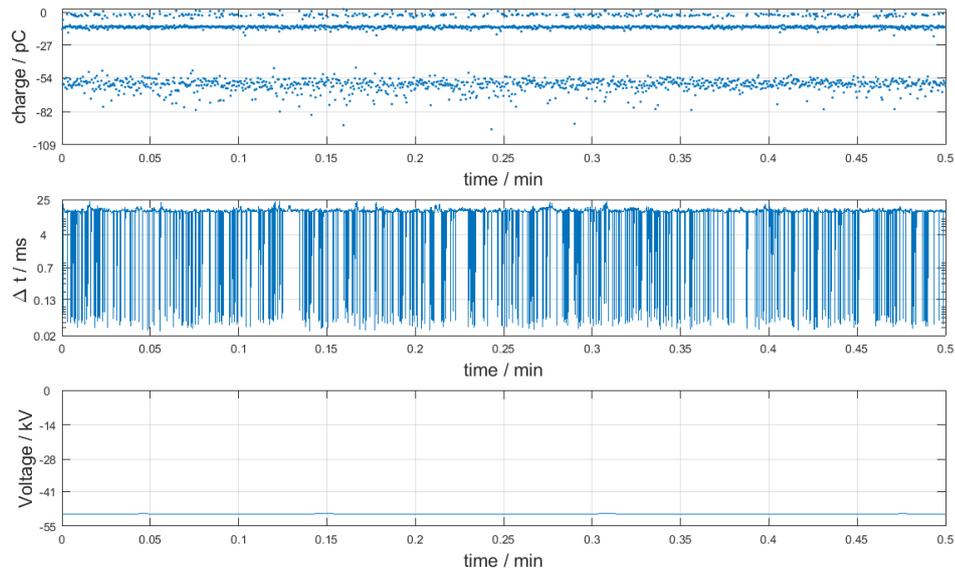
Bild 38: Mittelwerte der TE-Amplitude und Δt einer Spannungsfahrt von Partikel S2 bei einem Druck von $p = 0,6$ MPa und Gleichspannung negativer Polarität

ner Pseudoresonanz mit einer Frequenz von $f = 83$ Hz ($T = 12$ ms) bei $U_{DC} = -55$ kV bis $f = 37$ Hz ($T = 27$ ms) bei $U_{DC} = -15$ kV kurz vor der Aussetzspannung. Diese Pseudoresonanz ist als Häufungspunkt in den NoDi^*_{T} - und $\text{NoDi}^*_{\text{QT}}$ -Diagrammen erkennbar. In den Bildern 39a und 39b sind die Diagramme einer Spannungsstufe bei Gleichspannung negativer Polarität dargestellt. Im $Q(t)$ -Diagramm kommt es durch den Ladungsaustausch des Partikels mit den Elektroden zu unterschiedlichen Amplitudenhöhen. Bei Gleichspannung positiver Polarität haben sich die gleichen Effekte wie bei negativer Polarität der Gleichspannung gezeigt. Wie aus Bild 39c zu erkennen ist, dreht sich das NoDi^*_Q -Diagramm aufgrund der umgekehrten Polarität der Teilentladungen um 180 Grad um.

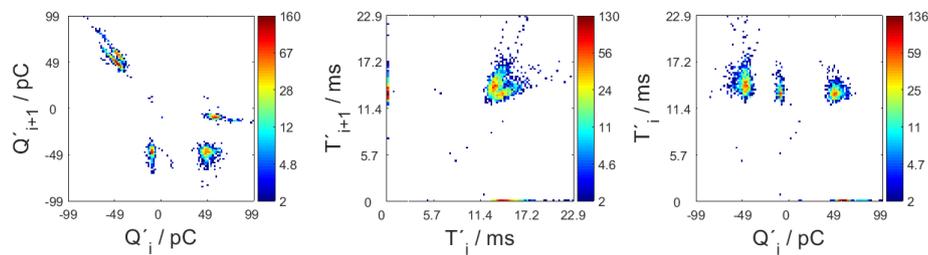
4.5.3 Draht

Bei den zwei gemessenen Drahtstücken wurden je nach Geometrie des Partikels und verwendetem Gasdruck beide oben genannten Effekte beobachtet.

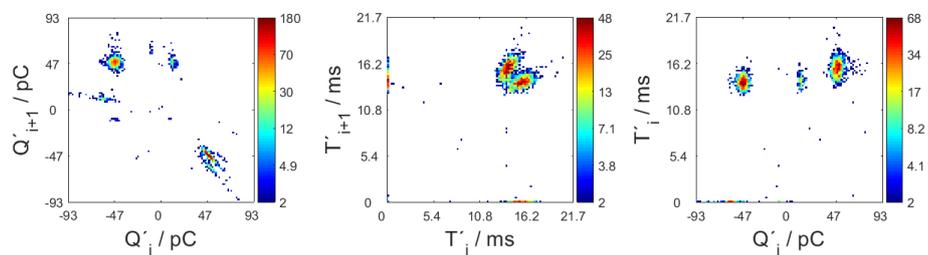
Partikel W1: Bei einem absoluten Druck von $p = 0,2$ MPa und Gleichspannung negativer Polarität konnte ab der Teilentladungseinsetzspannung, während der gesamten Spannungsfahrt ein Firefly-Effekt des Partikels betrachtet werden. Die Bewegungen wurden bei Erhöhung der Spannung schneller. Durch den konstanten Firefly-Effekt war die Anzahl der Teilentladungen so groß, dass das verwendete Messsystem der Firma Omicron Probleme bei der Verarbeitung der großen Datenmenge hatte, wodurch die Software regelmäßig einfro. Die höchste Anzahl an Teilentladungen wurde bei $U_{DC} = -50$ kV mit über 5 Millionen Impulsen in ca. 30 sec aufgezeichnet. In Bild 40a ist das $Q(t)$ -Diagramm für den Firefly-Effekt dargestellt. Es zeigt sich eine charakteristische Impulssequenz durch die „tanzenden“ Bewegungen des Partikels. Durch den Firefly-Effekt kommt es, je nach Position des Partikels, im NoDi^* -Diagramm (Bild 40b), zur Überlagerung der Muster eines freien Potentials (dreiseitiger Stern [15]) und das einer Spitze-Platte-Anordnung. Das typische Muster einer Spitze-Platte-Anordnung wurde bereits in Bild 8 gezeigt. Bei einem Druck von $p = 0,2$ MPa kam es zu keinem Firefly-Effekt. Die Bewegungen



(a) Zeitbasierte Diagramme bei Gleichspannung negativer Polarität



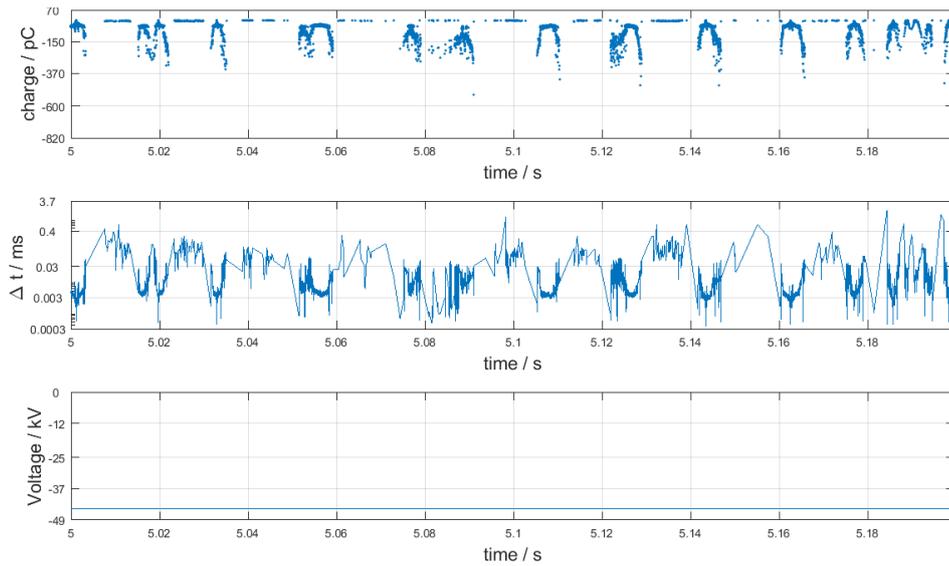
(b) NoDi*-Diagramme bei Gleichspannung negativer Polarität



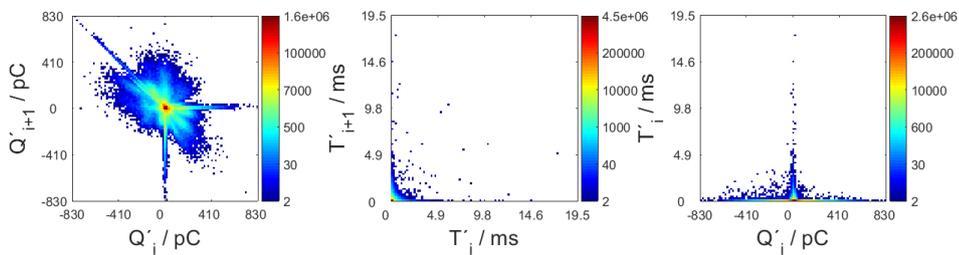
(c) NoDi*-Diagramme bei Gleichspannung positiver Polarität

Bild 39: Diagramme von Partikel S2 bei einem Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -50$ kV (a, b) sowie das NoDi*-Diagramm bei einer Spannung von $U_{DC} = 50$ kV (c) jeweils über eine Dauer von 30 s

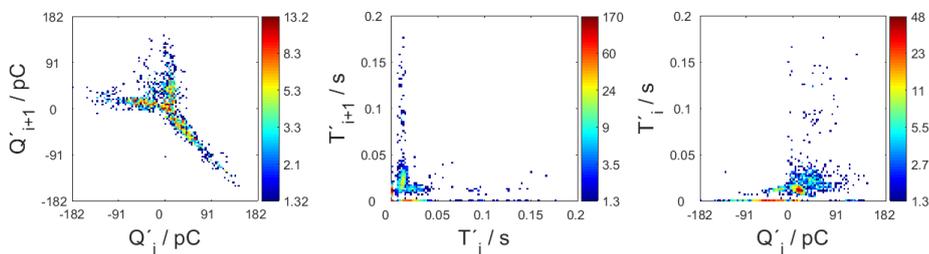
zwischen den Elektroden waren in der verwendeten Anordnung nicht wie zuvor annähernd homogen, sondern eher diffusen Charakters. Aus diesem Grund sprang das Partikel, auch nach wiederholten Versuchen, bereits nach einigen Sekunden aus der Anordnung. Bei Gleichspannung positiver Polarität kam es in seltenen Fällen kurzzeitig zum Firefly-Effekt an der Erdelektrode, kurz bevor das Partikel aus der Anordnung sprang. Abgesehen davon, wurden unabhängig vom Druck, nur springende Bewegungen zwischen den Elektroden festgestellt (Bild 40c). Auch hier konnten wieder nur einige Sekunden an Daten aufgezeichnet werden.



(a) Zeitbasierte Diagramme bei Gleichspannung negativer Polarität



(b) NoDi*-Diagramme bei Gleichspannung negativer Polarität

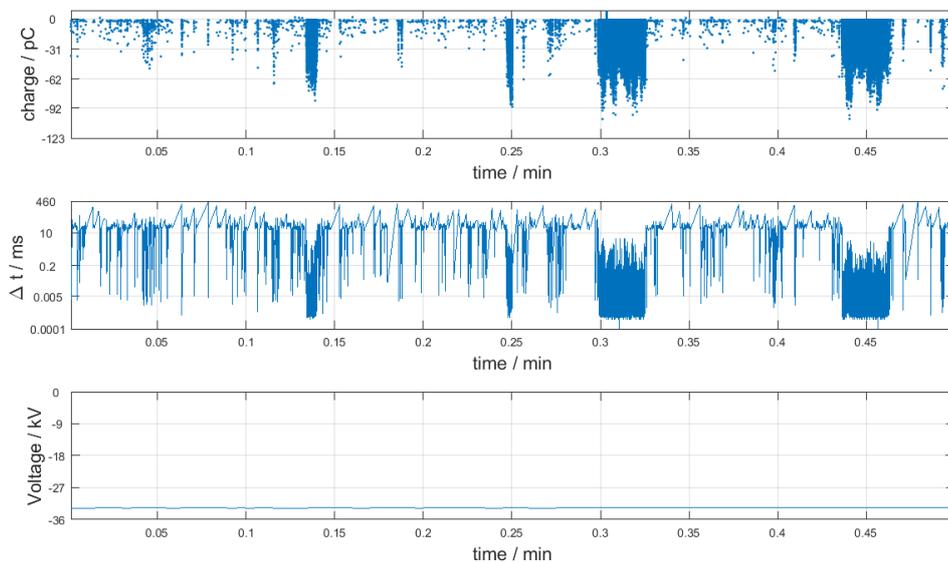


(c) NoDi*-Diagramme bei Gleichspannung positiver Polarität

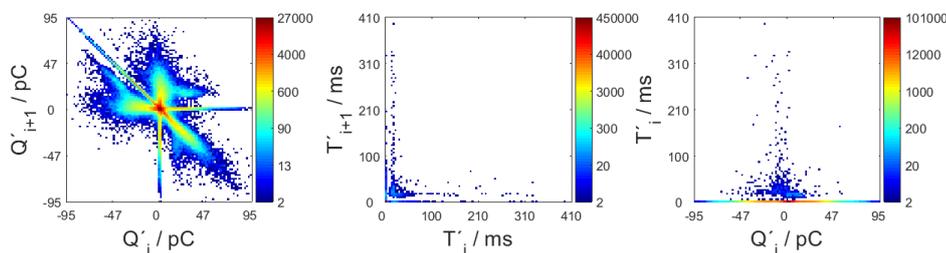
Bild 40: Diagramme von Partikel W1 mit Firefly-Effekt bei einem absoluten Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -45$ kV (a, b) sowie das NoDi*-Diagramm bei einer Spannung von $U_{DC} = 41$ kV (c)

Partikel W2: Das Partikel W2 war kürzer als W1 und im Gegensatz zu diesem zeigte sich bei Gleichspannung mit negativer Polarität und einem absoluten Druck von $p = 0,2$ MPa, eine Kombination aus Firefly-Effekt und willkürlichen Bewegungen. Es wurde ein ständiges Wechseln der beiden Effekte beobachtet. Dies ist im $Q(t)$ -Diagramm durch die unterschiedlichen Impulswiederholraten zu erkennen (Bild 41a). Jene Bereiche hoher Impulswiederholraten können darauf zurückgeführt werden, dass hier das Partikel auf der Elektrode „tanzt“. Das NoDi*-Diagramm zeigt das gleiche Muster wie ein reiner Firefly-Effekt. Grund dafür ist, dass in der Zeit in der der Firefly-Effekt auftritt, sehr viele Impulse gemessen werden, welche

aufgrund der hohen Impulswiederholraten das Muster der NoDi*-Diagramme sehr stark beeinflussen. Zu welchem Effekt es kommt, ist vermutlich aufgrund der scharfen Kanten des Partikels davon abhängig, in welchem Winkel und wie das Partikel auf die negative Elektrode auftrifft. Bei erhöhtem Druck wurde dieser Wechsel nicht mehr beobachtet. Es kam abermals zu einem springenden Partikel, ohne Firefly-Effekt. Bei Gleichspannung positiver Polarität hat sich das gleiche Verhalten wie bei dem vorhergehenden Drahtstück gezeigt.



(a) Zeitbasierte Diagramme bei Gleichspannung negativer Polarität



(b) NoDi*-Diagramme bei Gleichspannung negativer Polarität

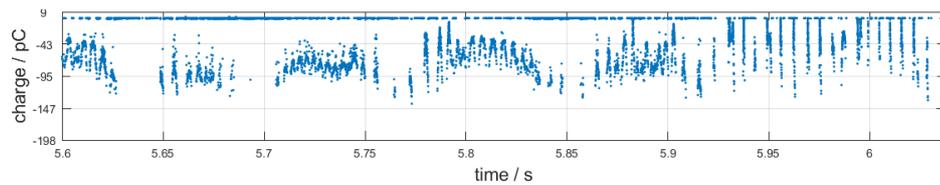
Bild 41: Zeitbasierte Diagramme (a) und NoDi*-Diagramme (b) von Partikel W2 mit Wechsel zwischen Hin-und-Her-Bewegung und Firefly-Effekt bei einem Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -23,8$ kV

4.5.4 Span

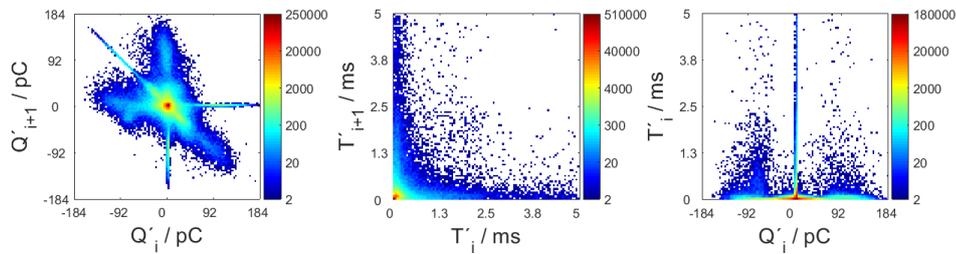
Der Span stellt aufgrund seiner realen Geometrie das interessanteste Partikel dar. Zunächst wird, wie zuvor, das Verhalten bei Gleichspannung negativer Polarität mit einem Druck von $p = 0,2$ MPa besprochen. Wurde die Gravitationskraft durch Erhöhung der Spannung überwunden ($U_{E,DC} = -19$ kV), so stellte sich eine Kraft und eine daraus resultierende Bewegung des Partikels ein. In den ersten Sekunden zeigte sich ein Wechseln zwischen Firefly-Effekt und eines diffusen Springens des Spans. Bei Erhöhung der Spannung kam es bis $U_{DC} = -35$ kV zu einem konstanten Firefly-Effekt, wie im $Q(t)$ -Diagramm (Bild 42a) und in den NoDi*-Diagrammen

(Bild 42b), ersichtlich ist. Bei weiterer Erhöhung konnte ab einer Spannung von $U_{DC} = -40$ kV ein Wechselspiel zwischen den beiden Effekten betrachtet werden. Bei einem Druck von $p = 0,6$ MPa stellte sich, wie zuvor, kein Firefly-Effekt ein, jedoch war die Bewegung zwischen den Elektroden konstanter und glich eher der, der Kugeln als der Drahtstücke (Bild 42c). Bei Gleichspannung positiver Polarität wurden dieselben Effekte wie zuvor, eine Hin-und-Her-Bewegung unabhängig vom Gasdruck, beobachtet.

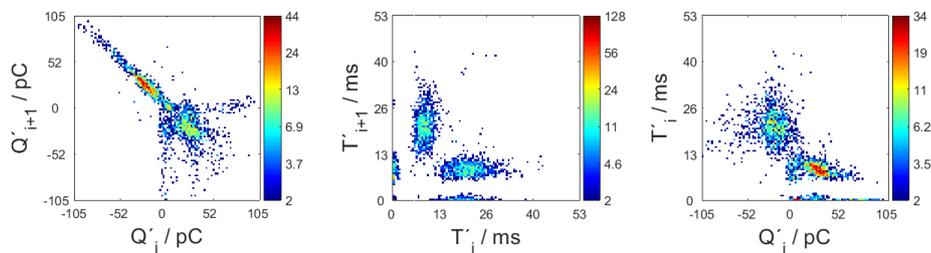
Der Span aus Stahl kann aufgrund der Hochgeschwindigkeitsaufnahmen in Hinblick auf die Bewegungen als Kombination von Kugel und Drahtstück gesehen werden.



(a) $Q(t)$ -Diagramme bei Gleichspannung negativer Polarität und einem Druck von $p = 0,2$ MPa



(b) NoDi*-Diagramme bei Gleichspannung negativer Polarität und einem Druck von $p = 0,2$ MPa



(c) NoDi*-Diagramme bei Gleichspannung negativer Polarität und einem Druck von $p = 0,6$ MPa

Bild 42: Diagramme von Partikel SP mit Wechsel zwischen Hin-und-Her-Bewegung und Firefly-Effekt bei einem Druck von $p = 0,2$ MPa (a, b) und springendes Partikel bei einem Druck von $p = 0,6$ MPa (c) bei jeweils einer Spannung von $U_{DC} = -30$ kV

5 Zusammenfassung

Durch die zunehmend größere Nachfrage elektrische Energie über weite Strecken zu transportieren, bekommen HGÜ-Anlagen und dadurch Betriebsmittel für Gleichspannung eine große Bedeutung. Um diese möglichst betriebsnahen Messungen zu unterziehen, benötigt es neue, innovative Messmethoden und Auswerteverfahren für Teilentladungsmessungen bei Gleichspannung.

Teilentladungsmessungen bei Wechselspannung und Gleichspannung sowie verschiedene Messverfahren werden in IEC 60270 beschrieben, wobei sich je nach verwendeter Spannung und Messmethode unterschiedliche Vor- und Nachteile ergeben:

1. Für Messungen bei Wechselspannung sind die Anforderungen an Prüfkreise und Messsysteme, sowie die Auswertung der Messdaten normativ in großem Maß geregelt.
2. Für Messungen bei Gleichspannung sind die normativen Vorgaben nach aktuellem Stand unzureichend.
3. Schmalbandige Messsysteme haben durch die veränderbare Mittenfrequenz eine sehr gute Störuneempfindlichkeit, zeigen jedoch Schwächen bei der Impulserkennung, besonders in Bezug auf die Impulsauflösezeit und der Polaritätserkennung.
4. Breitbandige Messsysteme hingegen besitzen eine gute Polaritätserkennung und kurze Impulsauflösezeiten, sind aber anfälliger für Störimpulse.

Der Interpretation der Messergebnisse kommt in der heutigen Teilentladungsdiagnostik eine entscheidende Rolle zu. Aufgrund teils unterschiedlicher Kenngrößen von TE-Impulsen bei Wechsel- und Gleichspannung müssen in Abhängigkeit der Spannung andere Darstellungsarten gewählt werden:

5. Im Bereich der Wechselspannungsprüfungen hat sich bereits seit Jahrzehnten das PRPD-Pattern etabliert.
6. Bei Gleichspannung haben vorangegangene Untersuchungen gezeigt, dass das NoDi*-Diagramm eine vielversprechende Alternative darstellt. Aufgrund verschiedener Muster kann auf den zugrunde liegenden Defekt geschlossen werden.

Damit Defekte frühzeitig erkannt werden können, benötigt es Diagnostik und Messsysteme, welche eine Analyse der Messdaten bereits während einer Messung zulassen. Es konnten folgende Erkenntnisse ermittelt werden:

7. Vorhandene Softwarelösungen der am Markt erhältlichen Messsysteme wurden für Prüfungen mit Gleichspannung als nicht ausreichend und unpraktikabel betrachtet, wodurch eine Interpretation der Defekte nicht möglich ist.
8. Beim Omicron MPD 600 hat sich gezeigt, dass das $Q(t)$ -Diagramm in der vorhandenen Form nicht sehr praktikabel erscheint, da nicht alle TE-Impulse unterschiedlicher Amplituden innerhalb desselben Zeitintervalls abgelesen werden können. Durch Modifikationen konnten diese Nachteile beseitigt werden.

9. Anhand des bereits vorhandenen Programms (*PDDC-Analyzer*) konnten die Messdaten erst nach einer Messung eingelesen und weiterverarbeitet werden. Das NoDi*-Diagramm konnte bisher nur im „Postprocessing“ analysiert werden.
10. Die zur Verfügung stehende COM-Schnittstelle der *Omicron Software for MPD and MI* bietet die Möglichkeit die Messdaten bereits während einer Messung auszulesen, weiterzuverarbeiten und zu analysieren. Das entstandene Programm (*Online PDDC-Analyzer*) stellt eine Kombination aus der Programmiersprache C++ und dem mathematischen Programm Matlab dar und ermöglicht es die Messdaten über Pipes auszulesen, an Matlab zu übergeben, in Echtzeit zu verarbeiten und dann in Diagrammen anzuzeigen.
11. Die Kommunikation und der Datenaustausch zwischen den beiden Programmen wurde mit Hilfe von MEX-Funktionen realisiert. Diese ermöglichen Matlab eine Interpretation von C++-Code. MEX-Funktionen sind durch C++ sehr schnell, jedoch wird bei Änderungen des Codes eine neuerliche Kompilierung notwendig.
12. Um die Anzeigeverzögerung des *Online PDDC-Analyzers* bei sehr vielen Daten kurz zu halten, wurde ein Algorithmus zur Datenreduzierung entwickelt.
13. Durch den *Online PDDC-Analyzer* ergibt sich die weltweit erstmalige Möglichkeit die NoDi*-Diagramme in Echtzeit zu analysieren und Aussagen zum Defekttyp zu treffen.

Metallische Partikel innerhalb einer GIS-Anlage können die Durchschlagfestigkeit erheblich herabsetzen und sollten bereits frühzeitig erkannt werden. Die durchgeführten TE-Messungen bei Gleichspannung in Kombination mit Hochgeschwindigkeitsaufnahmen von fünf unterschiedlichen Partikeln haben gezeigt, dass sowohl im $Q(t)$ -Diagramm sowie den NoDi*-Diagrammen eindeutige Muster erkennbar sind.

14. Die beiden Kugeln S1 und S2 haben eine ausgeprägte Pseudoresonanz in Abhängigkeit der Amplitude der Spannung gezeigt. Diese Pseudoresonanz konnte in den NoDi*-Diagrammen als Häufungspunkte erkannt werden.
15. Bei dem Drahtstück W1 konnte ein Firefly-Effekt beobachtet werden. Dieser Effekt stellt im NoDi*_Q-Diagramm eine Überlagerung zwischen einem freien Potential und einer Spitze auf Hochspannung dar.
16. Das kürzere Drahtstück W2 zeigte eine Kombination aus Firefly-Effekt und springendem Partikel.
17. Der Span hat einen konstanten Firefly-Effekt nur bei bestimmten Spannungshöhen negativer Polarität gezeigt.

Durch die entwickelte Software (*Online PDDC-Analyzer*) und den durchgeführten TE-Messungen, soll es zukünftig leichter werden, weitere Forschung auf dem Gebiet der TE-Messung bei Gleichspannung zu betreiben und die Aussagekraft des NoDi*-Diagramms weiter stärken.

Literaturverzeichnis

- [1] DIN EN 60270: *Hochspannungs-Prüftechnik – Teilentladungsmessungen*. VDE, November 2016
- [2] Rothleitner: *Grafische Auswertung von Teilentladungen bei hoher Gleichspannung*. Bachelorarbeit, Institut für Hochspannungstechnik und Systemmanagement, Technische Universität Graz, 2015
- [3] Schwab: *Hochspannungsmesstechnik - Messgeräte und Messverfahren*. Springer-Verlag, 2. Auflage, 1981; Nachdruck 2011
- [4] Küpfmüller: *Die Systemtheorie der elektrischen Nachrichtenübertragung*. S. Hirzel Verlag, 1974
- [5] Schon: *Stoßspannungs- und Stoßstrommesstechnik - Grundlagen - Messgeräte - Messverfahren*. Springer Verlag, 2010
- [6] Schon: *Hochspannungsmesstechnik - Grundladeden - Messgeräte - Messverfahren*. Springer-Verlag, 2016
- [7] Lerch: *Elektrische Messtechnik - Analoge, digitale und computergestützte Verfahren*. Springer Verlag, 6. Auflage, 2012
- [8] Shannon: *Communication in the Presence of Noise*. IEEE Proceedings of the IRE, Vol. 37, Issue 1, pp. 10 – 21, 1949
- [9] Kiencke, Eger: *Messtechnik - Systemtheorie für Elektrotechniker*. Springer Verlag, 6. Auflage, 2005
- [10] Rethmeier: *Neue Auskoppelverfahren und Sensoren zur Vor-Ort-Teilentladungsmessung an Hochspannungs-Kabelanlagen*. Dissertation, Technische Universität Berlin, 2006
- [11] CIGRE WG D1.03: *Risk Assessment on Defects in GIS based on PD Diagnostics*. CIGRE Technical Brochure 525, 2013
- [12] Omicron Electronics GmbH: *MPD600 User Manual*, 2014
- [13] Kreuger: *Industrial High DC Voltage*. Delft University Press, 1995
- [14] CIGRE WG 15.03: *CIGRE Data Format for GIS Partial Discharge Software Application*. Electra No. 177, 1998
- [15] Pirker, Schichler: *Partial Discharges at DC Voltage - Measurement and Pattern Recognition*. International Conference on Condition Monitoring and Diagnosis, Xi'an, China, 2016
- [16] Seo, Koo, Lee et al.: *Identification of Insulation Defects by modified Chaotic Analysis of Partial Discharge under DC Stress*. CIGRE, Report D1-310, 2012
- [17] Fromm: *Partial Discharge and Breakdown Testing at High DC Voltage*. Dissertation, Delft University of Technology, 1995

- [18] Oeding, Oswald: *Elektrische Kraftwerke und Netze*. Springer Verlag, 7. Auflage, 2011
- [19] ABB: *Gas-insulated Switchgear ELK-04 - Modular System up to 170 kV, 4000 A, 63 kA*. Product Brochure, 2009
- [20] CIGRE JWG 33/23.12: *Insulation Co-ordination of GIS: Return of Experience - On Site Tests and Diagnostic Techniques*. Electra No. 176, 1998
- [21] Qi, Li, Xing, Wei: *Partial Discharge Initiated by Free Moving Metallic Particles on GIS Insulator Surface: Severity Diagnosis and Assessment*. IEEE Transactions on Dielectrics and Electrical Insulation, Vol. 21, Issue 2, pp. 766 – 774, 2014
- [22] CIGRE WG D1.03: *Gas Insulated System for HVDC: DC Stress at DC and AC Systems*. CIGRE Technical Brochure 506, 2012
- [23] CIGRE WG D1.25: *UHF Partial Discharge Detection System for GIS: Application Guide for Sensitivity Verification*. CIGRE Technical Brochure 654, 2016
- [24] Diessner, Trump: *Free Conducting Particles in a Coaxial Compressed-Gas-Insulated System*. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-89, Issue 8, pp. 1970 – 1978, 1970
- [25] Hasegawa, Yamaji, Hatano, Endo, Rokunohe, Yamagiwa: *Development of Insulation Structure and Enhancement of Insulation Reliability of 500 kV DC GIS*. IEEE Transactions on Power Delivery, Vol. 12, Issue 1, pp. 194 – 202, 1997
- [26] Küchler: *Hochspannungstechnik - Grundlagen - Technologie - Anwendungen*. Springer-Verlag, 3. Auflage, 2009
- [27] Microsoft: *Component Object Model (COM)*, [https://msdn.microsoft.com/de-de/library/windows/desktop/ms680573\(v=vs.85\).aspx](https://msdn.microsoft.com/de-de/library/windows/desktop/ms680573(v=vs.85).aspx), abgerufen am 27.06.2017
- [28] Wikipedia: *Component Object Model*, https://de.wikipedia.org/wiki/Component_Object_Model, abgerufen am 27.06.2017
- [29] Omicron Electronics GmbH: *Upgrade license on MCU 502 / 504*, 2015
- [30] Omicron Electronics GmbH: *OMICRON Software for MPD and MI - COM Interface Documentation*, 2014
- [31] Omicron Electronics GmbH: *OMICRON Software for MPD and MI - COM Interface Documentation - Introduction and programming overview*, 2014
- [32] The MathWorks, Inc.: *Introducing MEX Files*, https://de.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html, abgerufen am 22.06.2017
- [33] Altman: *Accelerating MATLAB Performance - 1001 tips to speed up MATLAB programs*. CRC-Press, 2014

-
- [34] The MathWorks, Inc.: *GUIDE Templates*, https://de.mathworks.com/help/matlab/creating_guis/selecting-a-gui-template.html, abgerufen am 23.06.2017
- [35] Lehner: *Software-Dokumentation und Messung der Dokumentationsqualität*. Hanser, 1994
- [36] The MathWorks, Inc.: *MATLAB Runtime*, <https://www.mathworks.com/products/compiler/mcr.html>, abgerufen am 19.10.2017
- [37] Pirker, Schichler: *Partial Discharge Measurement at DC Voltage - Evaluation and Characterization by NoDi* Pattern*. IEEE Transactions on Dielectrics and Electrical Insulation, Vol. 25, Issue 1, 2018

Anhang

A Bedienungsanleitung des Online PDDC-Analyzers

Diese Bedienungsanleitung soll eine schnelle und einfache Übersicht darüber geben, wie Messungen mit der Software durchgeführt werden können und die Software zu bedienen ist. Vorausgesetzt werden Kenntnisse des Benutzers bei der Bedienung der *Omicron Software for MPD and MI*. Diese Software wird im Benutzerhandbuch des MPD 600 beschrieben [12].

A.1 Voraussetzungen

- Omicron Hardware und Software in der Version 1.6.6.25
- MCU mit den Lizenzen für:
 - COM-Interface
 - Expert Mode
 - DC-Measurements
- Matlab Vollversion in der Version R2016a für .p-Files
- Matlab Runtime in der Version R2016a für *PDDC_Analyzer_Online.exe*-Datei

A.2 Zugriff auf den Quellcode (falls erforderlich)

Der Quellcode ist über ein Subversion-Projekt (SVN) an der TU Graz verfügbar. Es benötigt hierzu den TUGRAZonline Username und das Passwort als Zugangsdaten, sowie eine Zugriffsberechtigung.

URL: <https://svn.tugraz.at/svn/PDDC-Analyzer-live>

A.3 Durchführung von Messungen

- Starten der Software

Die Software kann entweder durch Ausführen der .exe-Datei oder in Matlab durch Rechtsklick auf die Datei *PDDC_Analyzer_Online.p* und Auswahl von *Run* aus dem Kontextmenü gestartet werden. Beim Starten der Software erscheint die in Bild 43 gezeigte GUI.

- **Verbinden mit Omicron:** Mit Klick auf *Connect...* wird das Programm versuchen, eine Verbindung zur Omicron-Software aufzubauen. Ist bereits eine Instanz der Software offen (das kann auch ein Streamfile im Offline-Modus darstellen), dann wird zu dieser Instanz verbunden. Falls noch keine Instanz geöffnet wurde, öffnet das Programm selbstständig eine und verbindet sich.

- **Device auswählen:** In den *General Settings* rechts oben, kann das Device ausgewählt werden. Dies ist nur von Relevanz, falls mehrere MPD angeschlossen werden.

- **Initialisierung der Omicron-Software:** Mit Klick auf *Initialize Device* wird die Omicron-Software auf Messungen für Gleichspannung angepasst.

- **Filter- und Exporteinstellungen:** Die gewünschten Filtereinstellungen zur Detektion von Teilentladungen können rechts in der GUI angepasst werden. Der Ordner sowie der Projektname zum Speichern von Messfiles und Screenshots kann durch Anklicken der Checkbox im Panel *Export* eingestellt werden. Mit Klick auf *Update Device* werden die Einstellungen in der Omicron-Software übernommen.
- **Starten der Messung:** Durch Klicken des Buttons *Start Recording* öffnet sich nach Initialisierung eine weitere GUI mit den Diagrammen, Messanzeigen und weiteren Einstellungen (Bild 44).
- **Kalibration der Ladung und Spannung:** Im Tab *Calibr. & Ext.* kann der Ladungs- bzw. Spannungswert, auf den kalibriert werden soll, eingestellt werden. Mit Klick auf *Calibrate* erfolgt die Kalibrierung.
- **Anpassen von Verstärkung und TE-Threshold:** Die Verstärkerstufe im Tab *PD* kann mit dem Regler *Gain* verändert werden und ist so zu wählen, dass die Anzeige *Level* bei den zu erwartenden TE-Impulshöhen im grünen Bereich liegt. Der TE-Threshold kann im selben Tab angepasst werden und ist ein Schwellpegel für den Ladungswert unter dem keine TE-Impulse erfasst werden. Er ist so zu wählen, dass kein Grundrauschen aufgenommen wird und wird bei Änderungen direkt in der Omicron-Software übernommen. Die Höhe des Grundstörpegels kann sich beim Verändern der Verstärkerstufe ändern, wodurch der TE-Threshold gegebenenfalls angepasst werden muss.
- **Anpassen der Diagramme:** Die Zeitachse der Diagramme wird automatisch gesetzt. Die Grenzen des $Q(t)$ -Diagramms, des *Pulse Count*-Diagramms und der NoDi*-Diagramme können im Tab *Options* eingestellt werden. Zusätzlich können, je nachdem wie gemessen wird, die Ladungswerte invertiert werden. Die Einstellungen werden mit Klick auf *Apply* übernommen. Im Panel *Display Options* kann das mittlere Diagramm umgeschaltet werden. Die Einstellungen der Zeit und des Thresholds des *Pulse Count*-Diagramms werden im Tab *Calibr. & Ext.* und *External View-PD Count* und mit Klick auf *Set* vorgenommen.
- **Messanzeigen:** Die Messanzeigen oben in den Tabs für den Ladungswert, der Spannung und Messzeit, sowie die Diagramme werden alle 300 ms aktualisiert. Es können höchstens 100.000 Daten in 300 ms erfasst werden. Wird diese Grenze überschritten, ändert sich der *Input Data Status* von *Balanced* zu *Overdriven*. Das Spannungs- und $\Delta t(t)$ -Diagramm zeigen Mittelwerte von 300 ms an. Im Tab *PD* gibt es zusätzlich eine Statistik der erfassten Impulse. Die aufgezeichneten Daten können mit Klick auf *Clear All Diagrams* zurückgesetzt werden.
- **Screenshots aufnehmen:** Durch Klicken des Buttons *Screenshot* im *PD*-Tab können Screenshots erstellt werden. Es werden die Time-Plots, NoDi*-Diagramme, sowie ein Bild der gesamten Programmoberfläche gespeichert. Der Pfad ist dabei derjenige, welcher in den Exporteinstellungen gewählt wurde.
- **Aufnahmen von Messfiles:** Soll ein Textfile zur späteren Auswertung aufgenommen werden, kann die Aufzeichnung mit Klick auf *Go* gestartet werden.

Dazu werden die Diagramme zurückgesetzt. Bei nochmaligem Drücken des Buttons stoppt die Aufnahme des Messfiles und dieses sowie eine *-vData.txt*-File mit den Spannungsdaten in 300 ms Abständen wird abgespeichert.

- **Beenden und Schließen:** Wird in der Main-GUI *Stop Recording* gedrückt, so bleibt die Stream-GUI offen, es werden jedoch keine neuen Messdaten mehr angezeigt. Gleiches passiert, wenn die Stream-GUI während des Streams geschlossen oder die Verbindung zur Omicron-Software unterbrochen wird. Die Verbindung kann auch manuell mit Klick auf *Disconnect* oder durch Schließen der Main-GUI getrennt werden.

In Bild 46 ist der Programmablaufplan zur Durchführung von Messungen dargestellt.

A.4 Beschreibungen der GUI

Main-GUI

Sobald das Programm gestartet und initialisiert wurde wird die in Bild 43 dargestellte graphische Oberfläche angezeigt. Diese Oberfläche steuert die Verbindung zur Omicron-Software und enthält Messeinstellungen, welche vor einer Messungen vorgenommen werden sollen. Links oben befindet sich die Statusanzeige, in welcher Statusupdates, wie gesetzte Einstellungen oder Fehler chronologisch angezeigt werden. Mit *Clear Log* kann diese zurückgesetzt werden. Rechts davon befinden sich Einstellungsmöglichkeiten zur TE-Detektion und den Export-Einstellungen von Messfiles. Sind mehrere MPD 600 and die MCU angeschlossen, kann im Popup-Menu die gewünschte Device ausgewählt werden. Die Einstellungen werden mit dem Button *Update Device* übernommen. Darunter befindet sich eine graphische Übersicht der Verbindung dieser Software zur MPD 600. Sollte die Verbindung fehlschlagen oder unterbrochen werden, wird hier auf einen Blick dargestellt, zu welcher Einheit keine Verbindung hergestellt werden konnte. Dauert der Verbindungsaufbau zu lange, kann dieser mit dem Button *Cancel* unterbrochen werden. Wird diese GUI vom Benutzer geschlossen werden alle zum Programm zugehörigen Fenster ebenfalls beendet, die Verbindung zur Omicron-Software unterbrochen und das Programm ordnungsgemäß beendet.

Stream-GUI

Sobald der Datenstream (Button *Start Recording*) gestartet wurde, wird die GUI in Bild 44 am Bildschirm angezeigt. Solange diese GUI geöffnet ist **und** der Datenstream läuft, wird diese GUI in Zeitabständen von 300 ms aktualisiert. Links in der Oberfläche befinden sich die Tabs *PD*, *Options* und *Calibr. & Ext.* (Bild 45), welche verschiedene Einstellungsmöglichkeiten bieten, wie im Teil A.3 „Durchführung von Messungen“ beschrieben wird. Oben in den Tabs befinden sich Messanzeigen, welche immer den aktuelle Ladungs-, Spannungs-, Zeitwert usw. anzeigen. Oben in der GUI befinden sich die zeitbasierten Diagramme (Time Functions) und darunter die drei NoDi*-Diagramme. Die Diagrammeinstellungen können im Tab *Options* gesetzt werden. Wird die GUI geschlossen, wird der Datenstream beendet.

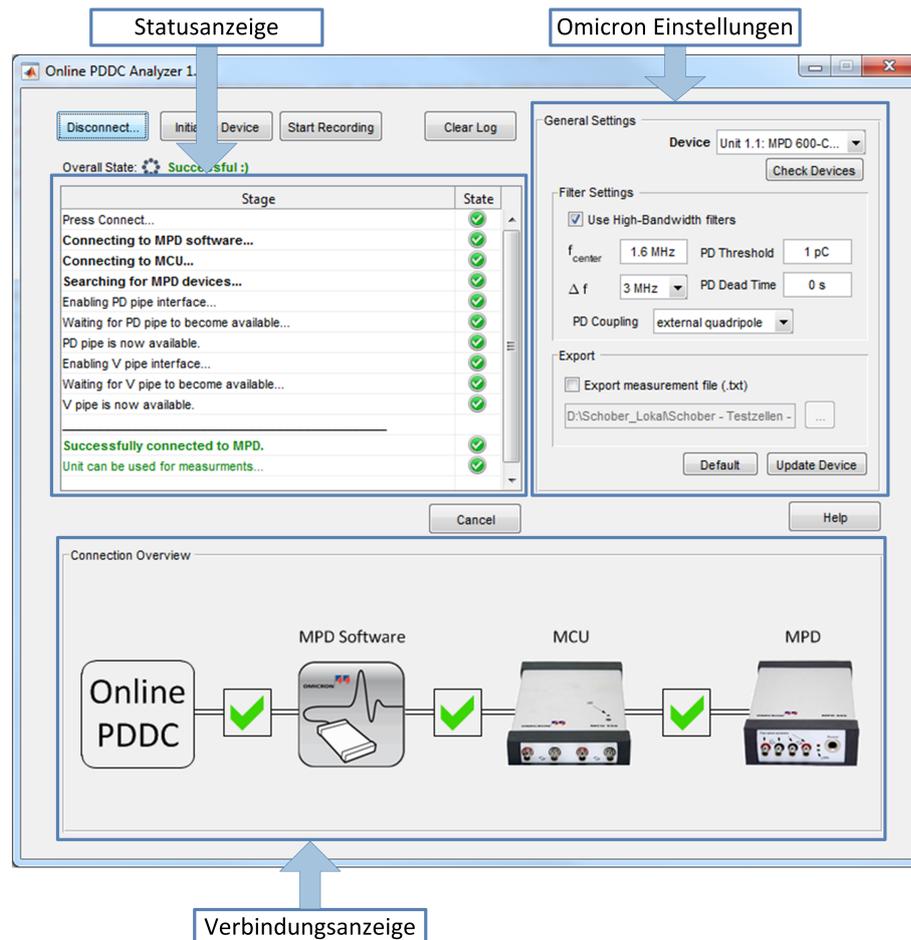


Bild 43: Main-GUI zur Verbindung mit der Omicron-Software

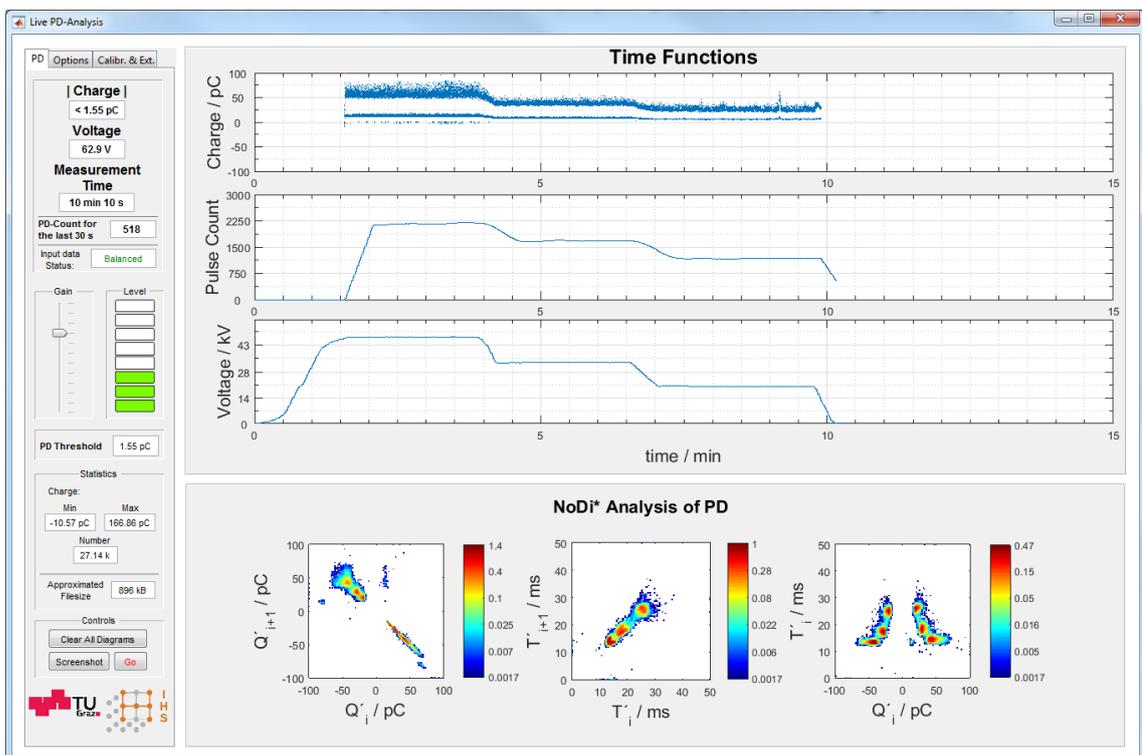
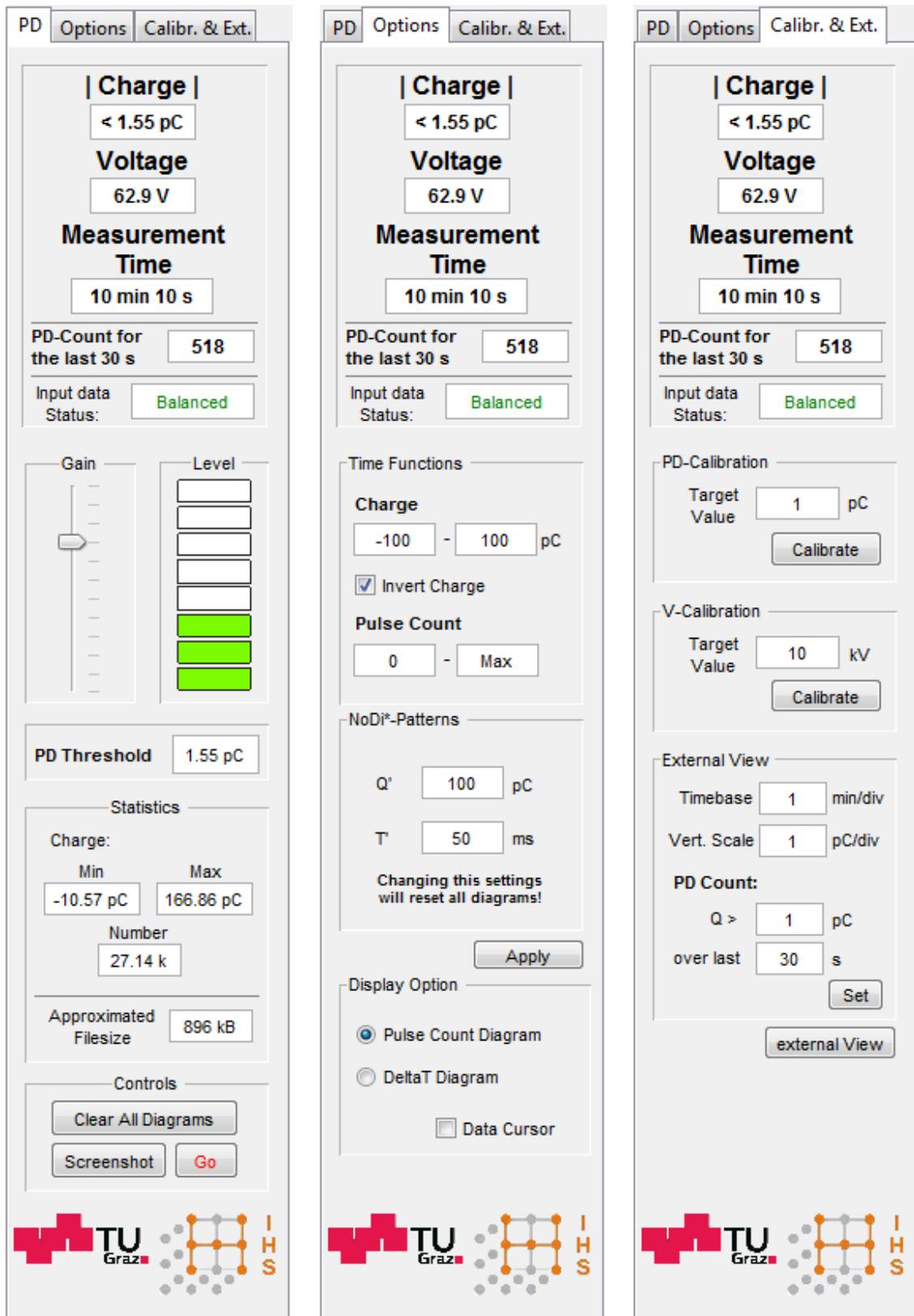


Bild 44: Stream-GUI zur Darstellung der Messdaten in Echtzeit in verschiedenen Diagrammen und Einstellungsmöglichkeiten



(a) Tab PD

(b) Tab Options

(c) Tab Calibr. & Ext.

Bild 45: Einzelne Tabs der Stream-GUI mit den Messanzeigen und Einstellungsmöglichkeiten

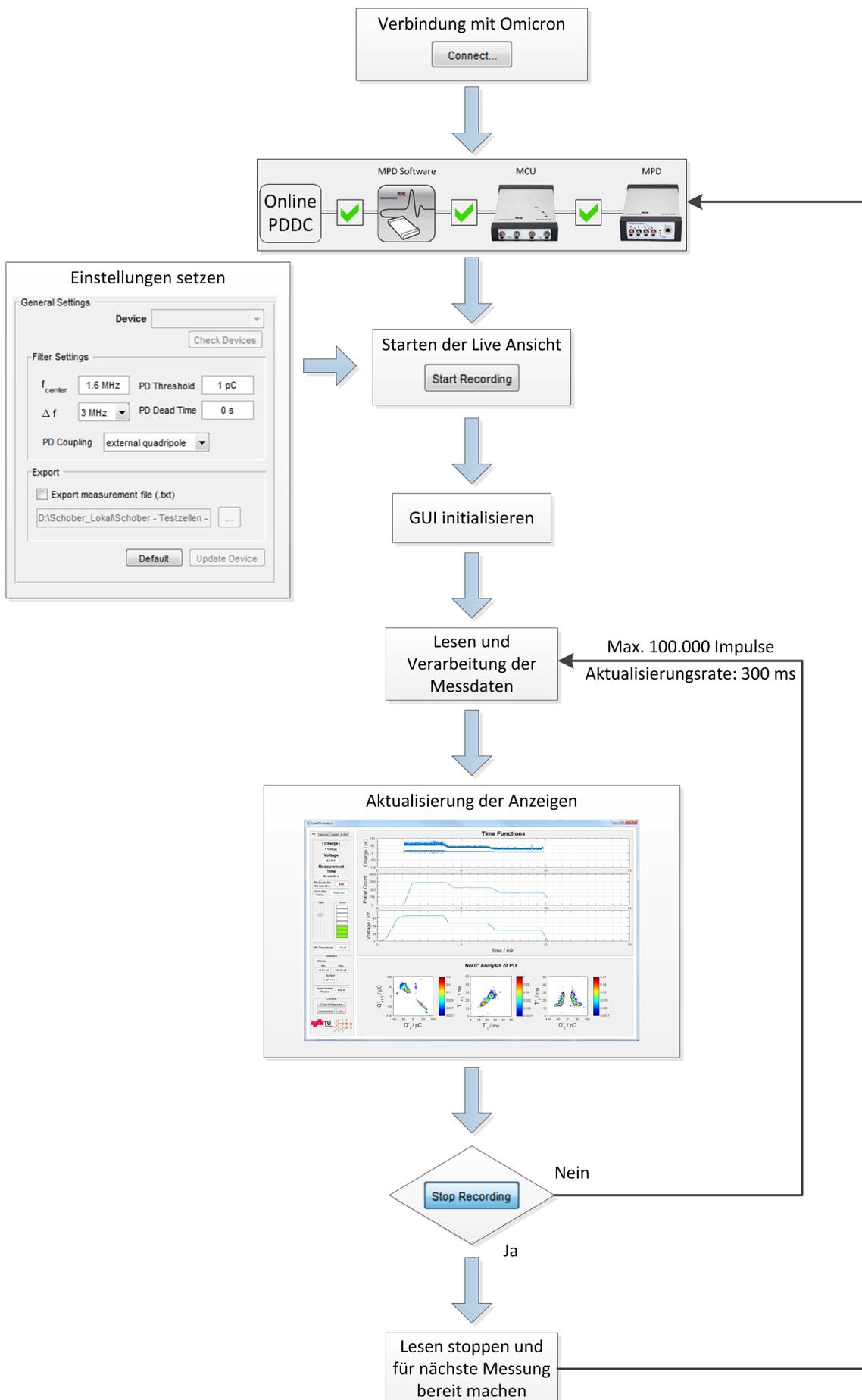


Bild 46: Grundsätzlicher Programmablaufplan

B Verzeichnisse

Abkürzungsverzeichnis

A/D	Analog-Digital
AC	Alternating Current (Wechselspannung)
AKV	Ankoppelvierpol
CD	Coupling Device
COM	Component Object Model
DC	Direct Current (Gleichspannung)
FPGA	Field Programmable Gate Array
GIL	Gasisolierte Leitung
GIS	Gasisolierte Schaltanlage
GUI	Graphical User Interface
GUIDE	Graphical User Interface Design Environment
HGÜ	Hochspannungsgleichstromübertragung
MEX	MATLAB executable
NoDi	Normalisiert Differenziert
OPV	Operationsverstärker
PD	Partial Discharge
PDDC	Partial Discharge Direct Current
PRPD	Phase Resolved Partial Discharge
TE	Teilentladung(en)
UHF	Ultra High Frequency

Bildverzeichnis

1	TE-Prüfkreis mit Koppeleinheit CD in Serie zum Koppelkondensator C_k [1]	3
2	Impulsform zweier überlagerter Exponentialfunktionen mit den Zeitkonstanten τ_1 und τ_2 im Zeitbereich sowie der Scheitel- t_p und der Rückenhalbwertszeit t_2 (a) und die Amplitudendichte verschiedener Impulse mit unterschiedlichen Zeitkonstanten im Frequenzbereich (b)	5
3	Impulsantwort schmal- und breitbandiger Bandpassfilter im Vergleich (oben) und Detailansichten (unten) normiert auf den Maximalwert von $u(t)$	9
4	Integrierender Verstärker mit einem Stromimpuls $i(t)$ am Eingang und der Ausgangsspannung $u(t)$ [6]	10
5	Blockschaltbild eines modernen digitalen TE-Messgerätes [6]	11
6	Kenngrößen von TE-Impulsen bei AC- (a) und DC-Spannung (b)	14
7	PRPD-Pattern einer Spitze-Platte-Anordnung in N_2 bei $p = 0,5$ MPa relativem Druck bei 1,1-facher TE-Einsatzspannung über eine Dauer von 1 min	15
8	NoDi*-Diagramm einer Spitze-Platte-Anordnung in N_2 bei $p = 0,5$ MPa relativem Druck bei 1,1-facher TE-Einsatzspannung über eine Dauer von 1 min	17
9	Aufbau einer modernen GIS (a) und mögliche Defekte (b) im gekapselten Aufbau am Beispiel eines Sammelschienentrennschalters mit Erdungsschalter (Querschnittzeichnung entnommen aus [19])	18
10	Fehlerursachen in GIS nach [20]	19
11	Messgeräte zur Erfassung von Teilentladungen nach [3]	22
12	Vergleich zwischen <i>Online PDDC-Analyzer</i> (grüner Pfad) und <i>PDDC-Analyzer</i> (weißer Pfad)	23
13	Anschluss der Messeinheiten an einen AC-Prüfkreis [12]	24
14	Softwareoberfläche der <i>Omicron Software for MPD and MI</i>	24
15	Große <i>Scope</i> -Ansichten der beiden DC-Ansichten in der Omicron-Software	26
16	Vergleich des $Q(t)$ -Diagramms der beiden Programme <i>Online PDDC-Analyzer</i> (oben) und Omicron-Software (unten)	27
17	Konsolenausgabe von Listing 2	32
18	<i>GUIDE</i> -Oberfläche der Main-GUI [34]	34
19	Main-GUI zur Verbindung mit der Omicron-Software	35
20	Stream-GUI zur Darstellung der Messdaten in Echtzeit in verschiedenen Diagrammen und Einstellungsmöglichkeiten	36
21	Flussdiagramm der Main-GUI	39
22	Flussdiagramm der Stream-GUI	40
23	Detailansichten der Flussdiagramme (links die MEX-Funktion <code>stream()</code> und rechts die Matlab-Funktion <code>streamPlot()</code>)	41
24	Funktionsweise des Datenreduzierungsalgorithmus	46
25	Einzelne Tabs der Stream-GUI mit den Messanzeigen und Einstellungsmöglichkeiten	50

26	Blockdiagramm zur Funktionsweise des Programms <i>Online PDDC-Analyzer</i>	52
27	Grundsätzlicher Programmablaufplan	59
28	Messschaltung bei Wechselspannung	61
29	Messschaltung bei Gleichspannung	61
30	Messaufbau zur TE-Messung bei Gleichspannung:	62
31	Größenvergleich der Partikel an denen die Messungen durchgeführt wurden (a) und verwendete Testzelle (b) [37]	64
32	Spannungsfahrt der TE-Messungen, angefangen bei der Einsatzspannung U_E , bis maximal ± 55 kV und zur Aussetzspannung U_A mit einer Stufenhöhe von 5 kV und einer Stufenbreite von 30 s bei Gleichspannung bzw. 1 min bei Wechselspannung	65
33	Momentaufnahmen aus einem High-Speed-Video von Partikel S2 zu verschiedenen Zeitpunkten	66
34	Momentaufnahmen aus einem High-Speed-Video von Partikel W1 zu verschiedenen Zeitpunkten	66
35	PRPD-Pattern der Kugel S2 über eine Dauer von 1 min jeweils bei der TE-Einsatzspannung für die Drücke $p = 0,2$ MPa (a) und $p = 0,6$ MPa (b)	67
36	PRPD-Pattern des Partikels W2 über eine Dauer von 1 min bei der TE-Einsatzspannung (a) und einer Spannung von $U = 45$ kV für einen Druck von $p = 0,2$ MPa (b)	67
37	PRPD-Pattern des Stahlspans SP über eine Dauer von 1 min jeweils bei einer Spannung von $U = 40$ kV für die Drücke $p = 0,2$ MPa (a) und $0,6$ MPa (b)	67
38	Mittelwerte der TE-Amplitude und Δt einer Spannungsfahrt von Partikel S2 bei einem Druck von $p = 0,6$ MPa und Gleichspannung negativer Polarität	68
39	Diagramme von Partikel S2 bei einem Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -50$ kV (a, b) sowie das NoDi*-Diagramm bei einer Spannung von $U_{DC} = 50$ kV (c) jeweils über eine Dauer von 30 s	69
40	Diagramme von Partikel W1 mit Firefly-Effekt bei einem absoluten Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -45$ kV (a, b) sowie das NoDi*-Diagramm bei einer Spannung von $U_{DC} = 41$ kV (c)	70
41	Zeitbasierte Diagramme (a) und NoDi*-Diagramme (b) von Partikel W2 mit Wechsel zwischen Hin-und-Her-Bewegung und Firefly-Effekt bei einem Druck von $p = 0,2$ MPa und einer Spannung von $U_{DC} = -23,8$ kV	71
42	Diagramme von Partikel SP mit Wechsel zwischen Hin-und-Her-Bewegung und Firefly-Effekt bei einem Druck von $p = 0,2$ MPa (a, b) und springendes Partikel bei einem Druck von $p = 0,6$ MPa (c) bei jeweils einer Spannung von $U_{DC} = -30$ kV	72
43	Main-GUI zur Verbindung mit der Omicron-Software	82
44	Stream-GUI zur Darstellung der Messdaten in Echtzeit in verschiedenen Diagrammen und Einstellungsmöglichkeiten	82

45	Einzelne Tabs der Stream-GUI mit den Messanzeigen und Einstellungsmöglichkeiten	83
46	Grundsätzlicher Programmablaufplan	84

Tabellenverzeichnis

1	Vor- und Nachteile von schmal- und breitbandigen Messgeräten . . .	9
2	Impulsfolgeantwort von TE-Messgeräten – Gewichtungsfunktion nach [1]	12
3	Kenngrößen von AC- und DC-TE-Messungen im Vergleich	13
4	Elektrische Abhebefeldstärken verschiedener Partikelgeometrien [22] .	20
5	Variablendokumentation – allgemein (oben) und am Beispiel für den Q_{IEC} -Wert (unten) [30]	30
6	File- und Ordnerübersicht	42
7	Beschreibung und Ergebnis der durchgeführten Tests der Main-GUI .	55
8	Beschreibung und Ergebnis der durchgeführten Tests der Stream-GUI	55
9	Partikel an denen TE-Messungen durchgeführt wurden	63
10	Messmatrix zu den durchgeführten Messungen	63

Listings

1	Prototypen der Funktionen <code>getHandle</code> , <code>getVariant</code> und <code>setVariant</code> [31]	31
2	Programm zum Lesen des Q_{IEC} -Wertes einer MPD und Ausgabe in der Console mit C++	32
3	Beispiel zweier Callback-Funktionen und Austausch von Variablen zwischen diesen	38