Zentralinstitut für Engineering, Elektronik und
Analytik (ZEA) · Systeme der Elektronik (ZEA-2)

Automatisierte Erkennung des Coulomb-Oszillationsbereichs in Sensorpunkt-Daten zur Kalibrierung bei Quantencomputern

Nils Wehrmeyer

Jül-4445

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Zentralinstitut für Engineering, Elektronik und Analytik (ZEA) · Systeme der Elektronik (ZEA-2)

Automatisierte Erkennung des Coulomb-Oszillationsbereichs in Sensorpunkt-Daten zur Kalibrierung bei Quantencomputern

Nils Wehrmeyer

Berichte des Forschungszentrums Jülich
Jül-4445 · ISSN 0944-2952
Zentralinstitut für Engineering, Elektronik und
Analytik (ZEA) · Systeme der Elektronik (ZEA-2)

DE-A96 (Bachelor, FH Aachen, 2023)

Vollständig frei verfügbar über das Publikations-
portal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess

Forschungszentrum Jülich GmbH · 52425 Jülich
Zentralbibliothek, Verlag
Tel.: 02461 61-5220 · Fax: 02461 61-6103
zb-publikation@fz-juelich.de
www.fz-juelich.de/zb

This is an Open Access publication distributed under the terms of the **Creative Commons Attribution License 4.0**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Erklärung

Diese Arbeit ist von mir selbständig angefertigt und verfasst. Es sind keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden.

Name: _____

Jülich, den _____

_____

Unterschrift der Studentin / des Studenten

Diese Arbeit wurde betreut von:

1. Prüfer: Prof. Dr. rer. nat. Matthias Grajewski
2. Prüfer: M. Sc. Sarah Fleitmann

Kurzfassung

Sensorpunkte werden bei der Kalibrierung von Doppelquantenpunkten, welche eine Basis für Quantenbits darstellen können, genutzt. Um die Sensorpunkte verwenden zu können, muss zunächst die Leitfähigkeit der Sensorpunkte kalibriert werden. Diese wird hauptsächlich über die Barrierenspannung gesteuert. Gesucht ist der Spannungsbereich, in dem der Sensor möglichst sensitiv auf seine Umgebung reagiert. Um diesen Bereich zu finden, wird zunächst der Coulomb-Oszillationsbereich gesucht. Dazu werden zur Zeit sogenannte „Wide Scans“ durchgeführt, die die Leitfähigkeit des Sensorpunkts in Abhängigkeit von den Barrierenspannungen messen. Ziel ist es aber mit kleineren Scans den Oszillationsbereich zu finden, um die Zeit für die Datenaufnahme zu verringern.
Diese Arbeit beschäftigt sich mit Verfahren für die möglichst performante Suche der linken unteren Ecke dieses Bereichs und dem Vergleich der Qualität der Ergebnisse. Hierbei sollen nur kleine Scans aufgenommen und analysiert werden, um die zeitintensiven Wide Scans zu vermeiden und mit möglichst wenig Aufwand die linke untere Ecke des Coulomb-Oszillationsbereichs zu finden. Hierfür wurde ein Algorithmus entwickelt, der mithilfe von Dispersionsmaßen Kanten erkennen kann, den nichtleitenden vom leitenden Bereich unterscheiden kann und den Oszillationsbereich durch das Entwickeln eines Templates und Pattern Matching detektieren kann. Je nachdem, wo der Narrow Scan sich im Wide Scan befindet, wird der Narrow Scan in eine gewisse Richtung verschoben. So soll die linke untere Ecke des Coulomb-Oszillationsbereich gefunden werden.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Zentralinstitut für Engineering, Elektronik und Analytik . . . 1
- 1.2 JARA-Institut für Quanten Information . . . 1
- 1.3 Motivation . . . 2

2 Grundlagen des Quantencomputings **4**
- 2.1 Quantenbits . . . 4
- 2.2 Nutzung von Quantenpunkten als Qubits . . . 5
- 2.3 Kalibrierung des Sensorpunktes . . . 5

3 Suche nach dem Coulomb-Oszillationsbereich **8**
- 3.1 Datensätze . . . 8
- 3.2 Beschreibung unterschiedlicher Merkmale . . . 9
- 3.3 Algorithmus . . . 10
- 3.4 Normalisierung . . . 12
- 3.5 Kantenerkennung mithilfe von Dispersionsmaßen . . . 15
- 3.6 Unterscheidung von leitendem und nichtleitendem Bereich . . . 16
- 3.7 Pattern-Matching . . . 17
  - 3.7.1 Template basierend auf Sinusfunktion . . . 20
  - 3.7.2 Template basierend auf Gaußfunktionen . . . 21
- 3.8 Schrittweite . . . 22

4 Ergebnisse **23**
- 4.1 Labels . . . 23
- 4.2 Bewertungsmetrik . . . 27
- 4.3 Bewertung der Normalisierungsverfahren . . . 27
- 4.4 Bewertung der Dispersionsmaße zur Kantenerkennung . . . 29
- 4.5 Bewertung des Mittelwerts zur Unterscheidung von leitendem und nichtleitendem Bereich . . . 32
- 4.6 Bewertung des Pattern-Matching-Verfahrens . . . 32
- 4.7 Bewertung des Gesamtergebnisses vom Algorithmus . . . 36

5 Fazit und Ausblick **38**

Abbildungsverzeichnis

1.1 Wide Scan mit finalem Narrow Scan . . . 3
2.1 Gate Layout . . . 6
2.2 Experimenteller Wide Scan . . . 7
3.1 Zeile aus Narrow Scan . . . 10
3.2 Flussdiagramm des Algorithmus . . . 11
3.3 Sigmoid-Funktionen mit verschieden Faktoren für p . . . 13
3.4 Beispielbild für `LogisticScaler` . . . 14
3.5 Beispielbild für `SignScaler` . . . 15
3.6 Narrow Scan und angepasste Sinus-Modellfunktion . . . 20
3.7 Narrow Scan und angepasste Gauß-Modellfunktion . . . 21
4.1 Beispielbild zum Vorgehen bei der Bestimmung des Labels für die linke untere Ecke des Oszillationsbereichs . . . 24
4.2 Wide Scans ohne klare Ecke . . . 25
4.3 Beispiele für Narrow Scans pro gelabelter Kategorie . . . 26
4.4 `SignScaler`-Beispiele für verschiedene Parameterwerte . . . 28
4.5 `LogisticScaler`-Beispiele für verschiedene Parameterwerte . . . 29
4.6 Dispersionsmaße für unterschiedliche Kategorien von Narrow Scans . . . 30
4.7 45°-Kante mit Oszillationen . . . 33
4.8 Original-Daten und generiertes Template, welches stark von den Original-Daten abweicht (Sinus) . . . 34
4.9 Höhenunterschiede der Peaks im 1d Scan in der markierten Zeile des Narrow Scans . . . 34
4.10 Original-Daten und generiertes Template, welches stark von den Original-Daten abweicht (Gauß) . . . 35
4.11 Gefundene Oszillationen, welche weit entfernt von der linken unteren Ecke des Oszillationsbereichs liegen . . . 35
4.12 Ergebnis mit guter Pinch-Off Messung und geringer Abweichung der Kantenwinkel . . . 36

1 Einleitung

Diese Arbeit wurde im Institutsbereich Systeme der Elektronik des Zentralinstituts für Engineering, Elektronik und Analytik (siehe Abschnitt 1.1) des Forschungszentrums Jülich angefertigt. In Zusammenarbeit mit dem JARA Institut für Quanten Information (siehe Abschnitt 1.2) der Rheinisch-Westfälischen Technischen Hochschule Aachen (RWTH Aachen) und des PGI-11 forscht das ZEA-2 unter anderem an der Realisierung von skalierbaren Quantencomputern.

1.1 Zentralinstitut für Engineering, Elektronik und Analytik

Das Zentralinstitut für Engineering, Elektronik und Analytik (ZEA) ist ein Teil der Forschungszentrum Jülich GmbH. Zusammen mit Wissenschaftler*innen aus anderen Jülicher Instituten werden im ZEA Geräte, Experimente, Prozesse, Analyseverfahren, Mess-, Analyse- und Regelungsanlagen sowie Detektorsysteme entwickelt. Das ZEA arbeitet mit Universitäten und wissenschaftlichen Einrichtungen weltweit zusammen. Das Institut teilt sich in drei Institutsbereiche auf [1]:

1. Engineering und Technologie (ZEA-1)
2. Systeme der Elektronik (ZEA-2)
3. Analytik (ZEA-3)

Das ZEA-2 entwickelt komplexe elektronische und informationstechnische Systeme für wissenschaftliche Zwecke. Die Systeme reichen von Sensoren und Detektoren über Signal- und Datenverarbeitung bis hin zu Informationsextraktions- und Visualisierungsverfahren [2].
Dabei fokussiert sich das ZEA-2 unter anderem auf den Bereich Quantencomputing und hierbei im Speziellen auf die skalierbare Kontroll- und Auslese-Elektronik für Temperaturen nahe des absoluten Nullpunkts.

1.2 JARA-Institut für Quanten Information

Das JARA-Institut für Quanten Information (QI) gehört zur Jülich Aachen Research Alliance (JARA) und vereint theoretische und experimentelle Forschung in der Quanteninformatik, um Beiträge zur Realisierung von hochskalierbaren Quantenrechnern zu liefern. In diesem Institut wird die Zusammenarbeit der RWTH und des FZJ deutlich.

Insbesondere mit der Quantum Technology Group [3] arbeitet das ZEA-2 eng zusammen. Schwerpunkte der Forschung des JARA-Instituts sind Halbleiter-Spin-Quantenbits, Quantenfehlerkorrektur und die Implementierung von Quantencomputing in verrauschten Systemen [4].

1.3 Motivation

Diese Arbeit beschäftigt sich mit Gate-definierten Halbleiter-Spin-Quantenbits. Um diese zu kalibrieren und auszulesen, werden Sensorpunkte verwendet. Die Funktionsweise der Quantenbits (Qubits), sowie die Funktionsweise des Sensors wird in Abschnitt 2.2 erklärt. Damit der Sensorpunkt möglichst sensitiv auf die Änderungen im Quantenbit reagiert, muss er selbst auch kalibriert werden (siehe Abschnitt 2.3). Für die Kalibrierung muss der Startpunkt des Coulomb-Oszillationsbereichs detektiert werden. Hierfür können sogenannte Wide Scans durchgeführt werden, die die Leitfähigkeit des Sensorpunktes in Abhängigkeit von den Barrierenspannungen messen. Allerdings beanspruchen Wide Scans viel Zeit für die Messung. Aus diesem Grund liegt der Fokus dieser Arbeit auf Verfahren, die ausschließlich Narrow Scans erfordern, welche einen deutlich kleineren Spannungsbereich abdecken und dadurch schneller aufgenommen werden können. Zur Verdeutlichung ist in Abbildung 1.1 ein Wide Scan gezeigt, wobei der für weitere Kalibrierungsschritte benötigte Narrow Scan als roter Kasten dargestellt ist. Innerhalb des Narrow Scans befindet sich der Startpunkt des Coulomb-Oszillationsbereichs, welcher mit einem roten Punkt markiert ist.
Um einen Algorithmus zur möglichst genauen Detektion des Coulomb-Oszillationsbereichs zu entwickeln, muss sich zunächst mit Merkmalen auseinandergesetzt werden, die ausschlaggebend für das Detektieren des Startpunktes des Coulomb-Oszillationsbereichs sind. Darauf wird in Abschnitt 3.2 eingegangen. Mit diesen Merkmalen wird im Folgenden ein Verfahren vorgestellt, was durch geschickte Verschiebungen eines Narrow Scans diesen Startpunkt sucht. Dieses Verfahren wird in Abschnitt 3.3 genauer erklärt und besteht aus verschiedenen Schritten: Dazu gehört das Erkennen von Bereichen ohne Oszillationen, Bereichen mit Kanten und dem Oszillationsbereichs selbst. Das Ziel ist es, einen erfolgreichen und genauen Algorithmus mit möglichst geringer Laufzeit zu entwickeln. Daher werden für die einzelnen Schritte des Algorithmus verschiedene Varianten getestet und anschließend bewertet.

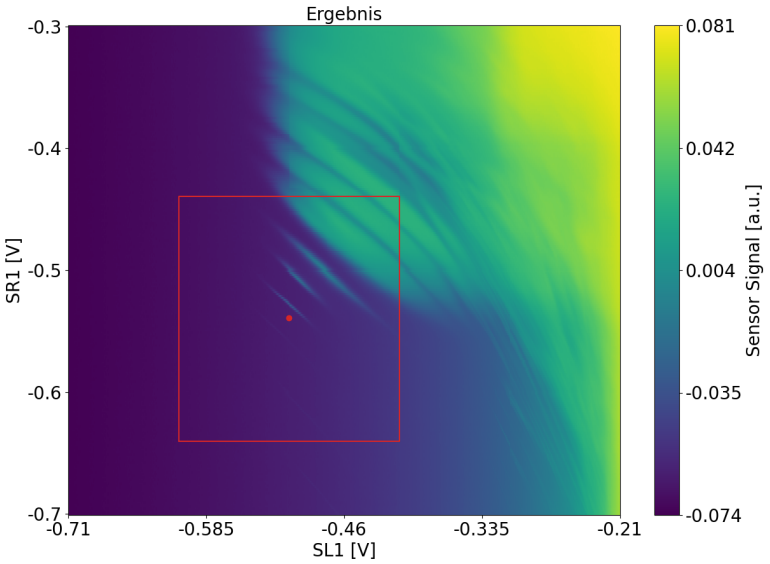


Abbildung 1.1: Finaler Narrow Scan im Coulomb-Oszillationsbereich als roter Kasten markiert in einem Wide Scan

2 Grundlagen des Quantencomputings

Dieses Kapitel vermittelt Grundlagen über Quantencomputing. Dazu werden zuerst die Besonderheiten eines Quantenbits im Vergleich zu einem klassischen Bit erklärt (siehe Abschnitt 2.1).
Eine Möglichkeit, ein Quantenbit zu realisieren, ist die Verwendung von Quantenpunkten (siehe Abschnitt 2.2). Dazu müssen diese Quantenpunkte zunächst kalibriert werden. Für die Kalibrierung der Quantenpunkte wird wiederum ein Sensorpunkt benötigt, der ebenfalls zunächst kalibriert werden muss. Ein Schritt dieser Kalibrierung ist die Suche des Coulomb-Oszillationsbereiches, worauf auch der Fokus dieser Arbeit liegt. Die dafür genutzten Datensätze und deren Besonderheiten werden in Abschnitt 2.3 erklärt.

2.1 Quantenbits

Während klassische Computer mit Bits arbeiten, die entweder die Werte 0 oder 1 annehmen können, arbeiten Quantencomputer mit Qubits. Im Gegensatz zu herkömmlichen Bits können diese sich in einer Superposition aus diesen beiden Werten befinden. Der Zustand eines Qubits wird in der Ket-Schreibweise als „$|\Psi\rangle$“ angegeben:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{mit } \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1. \tag{2.1}$$

Dabei sind $|0\rangle$ und $|1\rangle$ die orthogonalen Basiszustände, aus denen der Gesamtzustand eines Qubits linear kombiniert wird. Der Zustand eines Qubits ist also eine Überlagerung der Basiszustände mit komplexen Vorfaktoren. Die Länge des Zustandsvektors ist auf 1 normiert. Die Wahrscheinlichkeit, dass sich eine Qubit im Zustand $|1\rangle$ sich befindet, wird durch $|\alpha|^2$ beschrieben und die Wahrscheinlichkeit, dass es sich im Zustand $|1\rangle$ befindet, durch $|\beta|^2$.
Eine weitere Eigenschaft von Qubits ist die sogenannte Verschränkung. Durch die Verschränkung von Qubits wächst die Anzahl darstellbarer Zustände exponentiell mit der Anzahl der Qubits. Genau diese Eigenschaft sorgt für einen Vorteil eines Quantencomputers gegenüber einem klassischen Computer beim Lösen bestimmter Probleme, wie dem Springer-Problem [5].
Eine von vielen Herausforderungen, die zum erfolgreichen Einsatz eines Quantencomputers gelöst werden müssen, ist die Quantenkorrelation, welche bei verschränkten Systemen auftritt. Dies bedeutet, ein Qubit kann sich theoretisch in allen Zuständen aufhalten, aber durch Korrelationen zwischen den Qubits sind einige Kombinationen von Zuständen niemals erreichbar. Dafür ist es ermöglicht, Rückschlüsse von einem Zustand auf andere Zustände zu ziehen.
Für ausführlichere Erklärungen zur Quantenmechanik wird auf [6] verwiesen.

2.2 Nutzung von Quantenpunkten als Qubits

Bei dem in Abschnitt 2.1 beschriebenen Qubit handelt es sich zunächst um ein theoretisches Konstrukt, das physikalisch auf verschiedene Weisen realisiert werden kann. Eine mögliche Realisierung ist die Verwendung des Spinzustands von Elektronen. Der Spin ist eine quantenmechanische Eigenschaft, die man sich vereinfacht als Drehung um die eigene Achse vorstellen kann. Da sich ein Elektron in einer Superposition aus den Basiszuständen Spin-Up und Spin-Down befinden kann, eignet es sich gut für die Verwendung als Quantenbit [7].
Um den Spinzustand eines Elektrons messen zu können, wird es zunächst separiert. Dazu kann ein sogenannter Quantenpunkt genutzt werden. Dies ist eine künstliche Struktur, welche mit einer beliebigen Anzahl an Elektronen gefüllt werden kann. Dazu werden Spannungen an verschiedenen Gates angelegt. Diese werden im betrachteten Fall so kalibriert, dass sich genau ein Elektron im Quantenpunkt befindet [8].
In Abbildung 2.1 ist das Layout zweier Doppelquantenpunkte zu sehen. Die Regionen, in denen sich die Quantenpunkte befindet, sind hier mit roten Kreisen markiert. Die Elektronen, die von den Quantenpunkten gefangen werden sollen, befinden sich in einer Schicht aus 2-dimensionalem Elektronengas, welches in einer Gallium-Arsenid/Aluminium-Gallium-Arsenid (GaAs/AlGaAs) Heterostruktur erzeugt wird. Durch Veränderung der Spannung an den Gate-Elektroden kann man die Elektronen in den Quantenpunkten isolieren. Die Spannung an P1 beeinflusst hierbei hauptsächlich den linken Quantenpunkt und P2 den rechten Quantenpunkt.
Oberhalb der Doppelquantenpunkte liegen die Sensorpunkte SD_L und SD_R. In der Arbeit wird nur der linke Sensorpunkt betrachtet, die vorgestellten Verfahren sind aber genauso auch für den rechten Sensorpunkt anwendbar. Die Leitfähigkeit eines Sensorpunktes ist proportional zu den elektrostatischen Ladungen in seiner nahen Umgebung. Er kann daher genutzt werden, um Informationen aus dem Doppelquantenpunkt auszulesen. Für die Kalibrierung der Quantenpunkte kann beispielsweise das Eintreten und Austreten von Elektronen in den Quantenpunkt beobachtet werden. Nun ist es möglich zunächst alle Elektronen aus dem Quantenpunkt zu entfernen und dann die gewünschte Anzahl hinzuzufügen. Bei der späteren Nutzung der Doppelquantenpunkte als Qubit werden auch weitere Informationen, wie der Spin der Elektronen, über den Sensorpunkt ausgelesen [9].

2.3 Kalibrierung des Sensorpunktes

Damit der Sensorpunkt optimal auf seine elektrostatische Umgebung reagiert, muss er zunächst kalibriert werden. Die Leitfähigkeit wird hauptsächlich über die Spannung an den Barrieren-Gates des Sensors (SL und SR) gesteuert. Um einen Bereich zu finden, in dem der Sensor möglichst sensitiv auf seine Umgebung reagiert, wird der Leitwert in Abhängigkeit dieser Barrierenspannungen gemessen. Abbildung 2.2 zeigt eine solche

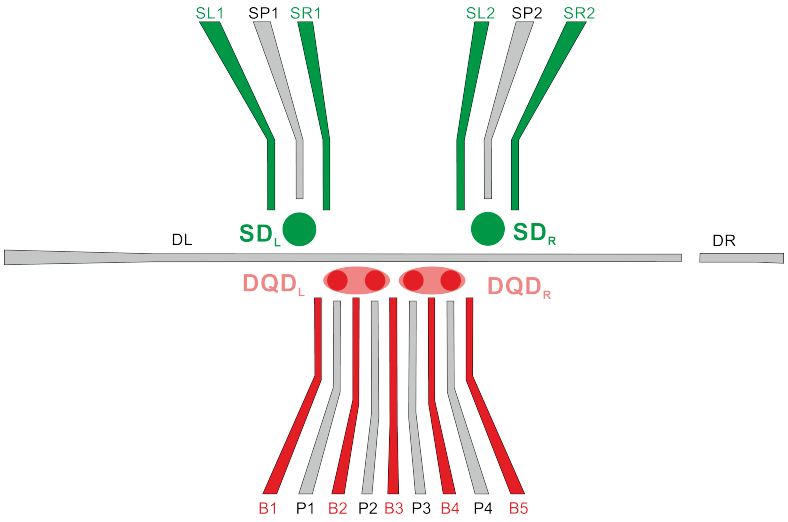


Abbildung 2.1: Gate Layout für zwei Doppelquantenpunkte [10]

Messung. Da diese Messung einen großen Spannungsbereich abdeckt, wird diese auch Wide Scan genannt. Diese Scans umfassen in der Regel einen großen Spannungsbereich von 400 bis 600 Millivolt (mV), um eine umfassende Charakterisierung der elektronischen Eigenschaften zu ermöglichen. Allerdings erfordern Wide Scans einen erheblichen Zeit- und Ressourcenaufwand, da sie zahlreiche Spannungspunkte abdecken müssen. Angesichts der Tatsache, dass das Interesse oft auf einen spezifischen Bereich gerichtet ist, in dem Fall die linke untere Ecke des Coulomb-Oszillationsbereiches (mit rotem Kreuz in Abbildung 2.2 markiert), wird nach effizienteren Methoden gesucht, um diesen Bereich zu lokalisieren. Der Coulomb-Oszillationsbereich ist an den diagonal im Bild gelb-grünlich verlaufenden Peaks zu erkennen, die bei höheren Spannungen beginnen ineinander zu fließen. Rechts oben über dem Oszillationsbereich folgt ein kontinuierlicher elektrischer Stromdurchfluss durch den Sensorpunkt. Hier wird im verwendeten Messaufbau ein Messwert um 0,05 a.u.[1] erwartet. Dort sind keine getrennten Peaks mehr zu erkennen. Unter und links neben dem Oszillationsbereich fehlt jeglicher Stromfluss. Dieser Bereich wird daher auch als nichtleitender Bereich bezeichnet. Hier wird im verwendeten Messaufbau ein Messwert von -0,05 a.u. erwartet.

Ein erster Ansatz, um sich dem Coulomb-Oszillationsbereich ohne vollständigen Wide Scan zu nähern, sind sogenannte Pinch-Off-Messungen. Hierbei wird jeweils nur eines der Barrieren-Gates durchfahren, um zu ermitteln, wo sich die Leitfähigkeit des Sensorpunktes ändert. Durch das Erkennen dieser Übergangspunkte pro Gate kann ein Punkt in der zwei-dimensionalen Messung bestimmt werden, der den möglichen Beginn des Coulomb-Oszillationsbereichs markiert. An diesem Punkt kann dann ein kleiner, hochaufgelöster Scan, auch Narrow Scan genannt, durchgeführt werden, um den spezifischen Bereich genauer zu untersuchen.

[1] arbitrary unit

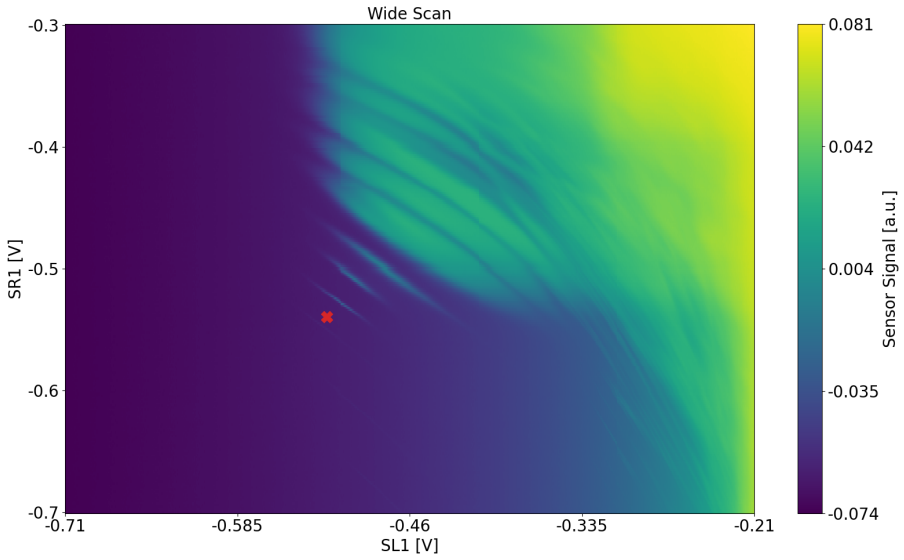


Abbildung 2.2: Experimenteller Wide Scan. Untere linke Ecke des Coulomb-Oszillationsbereiches bei ungefähr bei SL1 = -0,52V, SR1= -0,54V, markiert als rotes Kreuz

Narrow Scans bieten den Vorteil einer kürzeren Messdauer im Vergleich zu Wide Scans. Diese Messungen beschränken sich auf einen kleineren Spannungsbereich von, in der Arbeit gewählten, 200mV, wodurch die Datenerfassung deutlich beschleunigt wird. Sie können, ohne höheren Zeitaufwand, entlang eines Gates höher aufgelöst werden, wodurch eine detaillierte Charakterisierung der einzelnen Coulomb-Oszillationen erreicht werden kann. Dies ist insbesondere wichtig, um einen geeigneten Bereich auszuwählen, in dem der Sensor besonders sensitiv auf seine Umgebung reagiert.
Für die Kalibrierung des Sensorpunktes muss allerdings zunächst die linke untere Ecke des Coulomb-Oszillationsbereiches genauer bestimmt werden. Da das Ergebnis der Pinch-Off-Messungen in der Regel nur einen groben Anhaltspunkt liefert, wird in dieser Arbeit ein Algorithmus untersucht, der sich der linken unteren Ecke des Coulomb-Oszillationsbereichs mithilfe einer schrittweisen Verschiebung des Narrow Scans nähert. Um die Richtung dieser Verschiebung zu bestimmen, werden in Abschnitt 3.2 verschiedene Merkmale der Narrow Scans untersucht.

3 Suche nach dem Coulomb-Oszillationsbereich

In diesem Kapitel wird zunächst erklärt, welche Daten für diese Arbeit vorliegen und wie diese entstehen. Nachfolgend wird die Funktionsweise des Algorithmus für die Suche nach der linken unteren Ecke des Coulomb-Oszillationsbereichs erklärt. Zu diesem Zweck werden Merkmale und bekannte Werte dargelegt und dann die einzelnen Schritte des Algorithmus abschnittsweise genauer erklärt. Während dieses Kapitel nur der Vorstellung möglicher Verfahren für die einzelnen Schritte des Algorithmus dient, werden in Kapitel 4 die verschieden Verfahren bewertet und entsprechend für den Algorithmus ausgewählt.

3.1 Datensätze

Zunächst wird zwischen zwei verschieden Datensätzen unterschieden. Einer dieser Datensätze besteht aus experimentellen Daten, während der andere aus simulierten Daten besteht. Damit eine Durchschreitung des Spannungsraums mithilfe von Narrow Scans simuliert werden kann, werden Wide Scans verwendet, aus denen während der Ausführung des Algorithmus Narrow Scans ausgeschnitten werden.
Die experimentellen Wide Scans, die in der Arbeit verwendet werden, wurden aus einzelnen kleineren Scans zusammen geschnitten und liegen in Matlab-Dateien vor. Diese werden mithilfe eines Python-Scripts ausgelesen, bearbeitet und dargestellt. Es stehen ungefähr 40 Wide Scans dieser Art zur Verfügung.
Die simulierten Daten basieren auf einer Implementierung von Paul Surrey, Doktorand der Quantum Technology Group, welche zum JARA-Institut für QI gehört (siehe Abschnitt 1.2). Für die Simulation werden zunächst die Gates initialisiert, die die Position des Oszillationsbereichs am stärksten beeinflussen, darunter auch SL1 und SR1. Mit den Gates wird ein `SimpleCorner`-Objekt erzeugt, was einen Wide Scan simuliert. Hierbei kann der Nutzer einen Startwert übergeben, mit dem ein Zufallszahlengenerator initialisiert wird. Die `SimpleCorner`-Klasse funktioniert wie folgt: Mithilfe der Gates wird der Startpunkt des Oszillationsbereichs und dadurch auch das Label ausgelesen und eine `cross_talk_matrix` erzeugt. Diese `cross_talk_matrix` gibt an, wie sehr sich die Effekte der Gates gegenseitig beeinflussen, was z.B. einen Einfluss darauf hat, wie die Winkel der Kanten im simulierten Scan aussehen. Ebenfalls werden gewisse Oszillationseinstellungen getroffen, die über den Zufallszahlengenerator initialisiert werden. Diese Einstellungen verändern den Coulomb-Oszillationsbereich, hinsichtlich der Funktion mit denen die Peaks erzeugt werden und deren Parameter, die die Abstände und Breite der Peaks beeinflussen. Mit diesem Setup werden daraufhin zwei Kanten erzeugt und mit einer Fermi-Dirac-Statistik geglättet [11]. Die Oszillationen werden dann mit den Parametern aus den Oszillationseinstellungen als eine Sinus-Funktion bzw. mehrere Gauß-Funktionen erstellt und in den simulierten Scan eingefügt. Als letztes wird ein weißes Rauschen auf den simulierten Wide Scan gelegt und die simulierten Daten werden zurückgegeben. Die

Stärke des Rauschens kann der Nutzer verändern oder sogar abstellen. Für das Erzeugen des Rauschens wurde eine Wert zwischen 0,01 und 0,05 verwendet. Die experimentellen Daten sind auch noch mit pinkem Rauschen kontaminiert, was in den simulierten Daten nicht beachtet wird. Die simulierten Daten decken außerdem einen wesentlich größeren Spannungsbereich von 10V für die Gates SL1 und SR1 ab, während die experimentellen Daten einen Bereich von maximal 0,5V abdecken. Dies führt unter anderem dazu, dass mehr Coulomb-Peaks in den simulierten Daten sichtbar sind.
Es wurde 50 simulierte Scans erzeugt und als eigener Datensatz gespeichert.

3.2 Beschreibung unterschiedlicher Merkmale

In diesen Abschnitt werden bestimmte Merkmale der Sensor-Scans und bekannte Werte für den Algorithmus aufgezeigt. Mit diesen Merkmalen kann überprüft werden, ob sich der Oszillationsbereich in einem Narrow Scan befindet.
Ausgangspunkt des Algorithmus sind die Ergebnisse der Pinch-Off-Messungen. Bei der Pinch-Off-Messung wird nur eines der Gates SL1 und SR1 durchfahren, während das andere auf einen festen Wert gesetzt wird. In den Messungen kann dann der Wechsel von leitendem zu nichtleitendem Bereich gesucht werden, welcher für eine Schätzung des Startpunktes des Oszillationsbereichs genutzt wird.
Außerdem sind der minimale und maximale Leitwert im Wide Scan bekannt. Dies sind die Messwerte, die im nichtleitenden Bereich bzw. im leitenden Bereich zu erwarten sind. Im realen Experiment müssen die Werte für den minimalen und maximalen Leitwert einmalig berechnet werden und sind anschließend für weitere Messungen bekannt. Da die berechneten Werte des minimalen und maximalen Leitwerts für diese Arbeit jedoch nicht vorliegen, werden sie alternativ über das Minimum und Maximum der experimentellen Wide Scans bestimmt. Diese Werte dienen im entwickelten Algorithmus sowohl der Normalisierung der Narrow Scans als auch der Entscheidung darüber, ob sich der Narrow Scan im leitenden oder nichtleitenden Bereich befindet.
Ein weiteres Merkmal sind die vertikale und horizontale Kante, die vom Oszillationsbereich ausgehen und den nichtleitenden Bereich vom leitenden Bereich abgrenzen. Auch wenn hier von Kanten gesprochen wird, ist für die Detektion dieser wichtig, dass es sich eher um Verläufe als um tatsächliche harte Kanten handelt. Bewegt man sich entlang dieser Kanten nach unten bzw. links, so gelangt man zum Beginn des Coulomb-Oszillationsbereiches.
Charakteristisch für den Coulomb-Oszillationsbereich sind die Coulomb-Peaks. Sie haben durchschnittlich einen Abstand von ca. 100mV und sind ca. 20mV breit. Diese Durchschnittswerte wurden aus 40 experimentellen Wide Scans extrahiert. Mithilfe dieser Merkmale der Oszillationen kann ein Template zur Anwendung von Pattern-Matching auf die Narrow Scans erzeugt werden, was genauer in Abschnitt 3.7 erklärt wird. Bei der Anzahl der Peaks gibt es Unterschiede zwischen den simulierten und experimentellen Daten. Da die simulierten Daten einen wesentlich größeren Spannungsbereich für die Gates

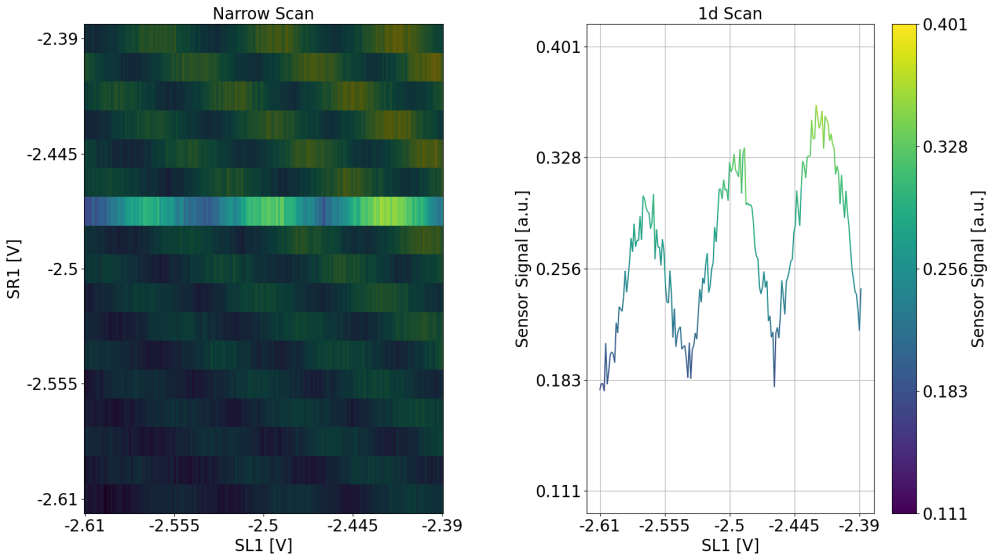


Abbildung 3.1: Eine Zeile aus einem Narrow Scan mit Oszillationen. Links ist der Narrow Scan, rechts die ausgewählte Zeile zu sehen

SL1 und SR1 abdecken, können auch bis zu 100 Peaks im Coulomb-Oszillationsbereich gefunden werden. Bei den experimentellen Daten findet man maximal 7 Peaks.

3.3 Algorithmus

Nachdem die Merkmale der Datensätze, die dem Algorithmus für die Suche nach dem Coulomb-Oszillationsbereich zur Orientierung dienen können, analysiert wurden, wird im folgenden Abschnitt der Ablauf des Algorithmus erklärt. Abbildung 3.2 zeigt den Ablaufplan des Algorithmus. Der Algorithmus bekommt zunächst die Ergebnisse der Pinch-Off-Messungen, um daraus einen Startpunkt für die Aufnahme eines Narrow Scans zu berechnen. Der aufgenommene Narrow Scan wird dann mit einem Normalisierungsverfahren bearbeitet, um die enthaltenen Werte vergleichbar zu machen. Die im Rahmen dieser Arbeit untersuchten Normalisierungsverfahren sind in Abschnitt 3.4 genauer erklärt.
Anschließend wird mithilfe eines Dispersionsmaßes überprüft, ob der Oszillationsbereich bzw. eine Kante vorliegt. Die dazu untersuchten Dispersionsmaße werden in Abschnitt 3.5 vorgestellt.
Wenn Oszillationen erkannt wurden, wird ein Template aus dem Oszillationsbereich erzeugt und mit Pattern-Matching wird bestätigt bzw. widerlegt, dass es sich um den Beginn des Oszillationsbereiches handelt. Dieser Schritt wird in Abschnitt 3.7 erklärt.

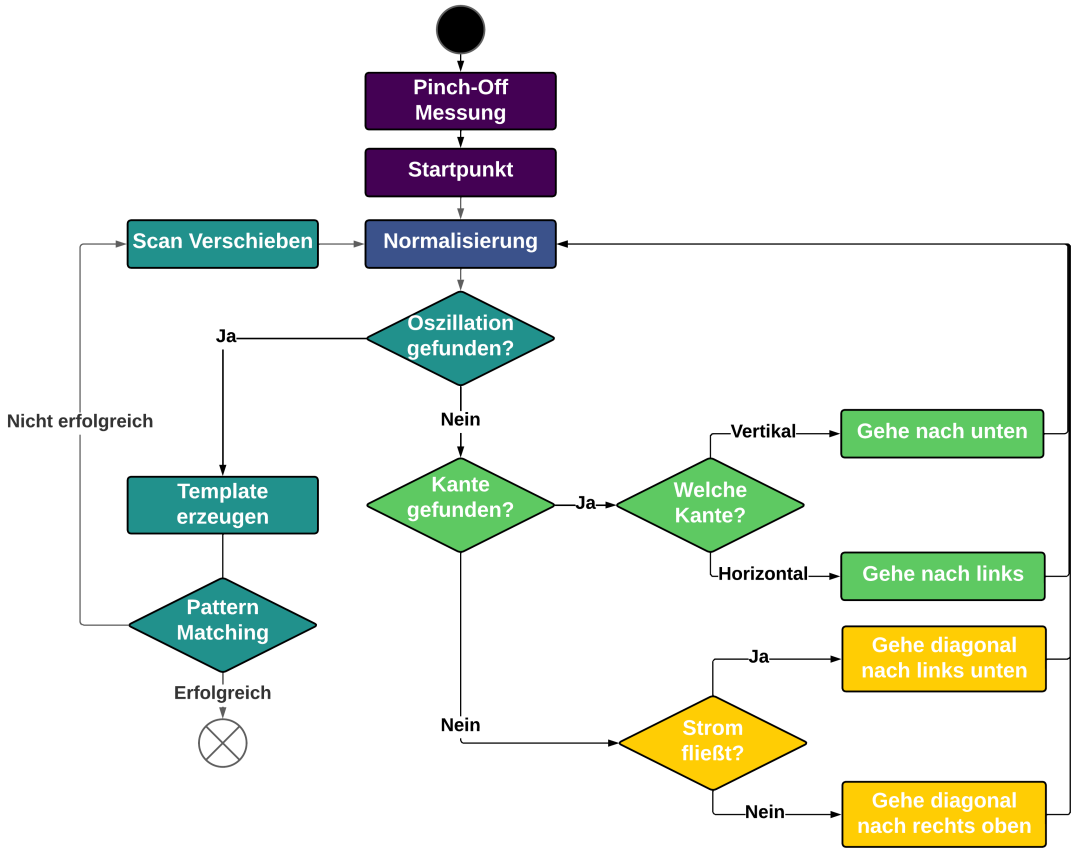


Abbildung 3.2: Flussdiagramm des Algorithmus

Wurde der Oszillationsbereich bestätigt, ist der Algorithmus beendet.
Wenn keine Oszillationen erkannt wurden, muss der Narrow Scan verschoben werden. Hierfür wird über ein Dispersionsmaß überprüft, ob eine horizontale oder vertikale Kante vorliegt. Wenn es sich um eine horizontale Kante handelt, wird der Scan nach links verschoben, wenn es sich um eine vertikale Kante handelt, nach unten. Wenn keine Kante im Narrow Scan vorhanden ist, dann befindet man sich entweder im leitenden oder nichtleitenden Bereich. Welcher der Bereiche sichtbar ist, wird mithilfe des Mittelwertes über den Narrow Scan untersucht. Je nach Bereich wird der Scan diagonal nach links unten oder rechts oben verschoben. Dies wird in Abschnitt 3.6 erklärt. Wie die Schrittweite für das Verschieben der Scans gewählt wird, wird in Abschnitt 3.8 genauer erklärt. Nach dem Verschiebungsschritt wird ein neuer Narrow Scan erstellt, woraufhin wieder mit dem Normalisierungsschritt begonnen wird. Diese Schritte werden wiederholt, bis der Algorithmus den Oszillationsbereich erfolgreich detektiert hat und terminiert.

Bevor die Schritte des Algorithmus einzeln erklärt werden können, müssen zunächst einige Variablen definiert werden: Die Matrix $X \in \mathbb{R}^{N \times M}$ mit $N, M \in \mathbb{N}$ enthält die

Leitwerte des Wide Scans. N und M geben hierbei die Auflösung in Zeilen und Spalten des Wide Scans an. Davon sind, wie im vorherigen Abschnitt erklärt, nur das Minimum $\min(X)$ und das Maximum $\max(X)$ zur Laufzeit des Algorithmus bekannt. Mit $x \in \mathbb{R}^{n \times m}$ mit $n, m \in \mathbb{N}$ werden die Messwerte des aktuellen Narrow Scans bezeichnet, diese sind dem Algorithmus alle bekannt. Hierbei gibt n und m die Auflösung in Zeilen und Spalten des Narrow Scans an.

3.4 Normalisierung

Um vergleichbare Ergebnisse für die folgenden Schritte des Algorithmus zu erhalten, müssen die Daten als erstes normalisiert werden. Da für den Algorithmus nur die Narrow Scans bekannt sind, werden die Transformationen auch nur auf den Narrow Scans durchgeführt. Bei den in diesem Abschnitt zur Veranschaulichung verwendeten Beispielbildern, Abbildung 3.4 und Abbildung 3.5, handelt es sich um Wide Scans. Da die Transformationsverfahren mit den gleichen Parametern angewandt werden, macht es keinen Unterschied, ob die Transformation auf einem Narrow oder Wide Scan durchgeführt wird. Es wurden folgende vier Verfahren implementiert:

1. `MinMaxScaler` [12]
2. `LogisticScaler` [13]
3. `SignScaler` [14]
4. `ZScoreScaler` [15]

Hierbei ist der `MinMaxScaler` die wichtigste Transformation, da diese die Daten eines Scans mithilfe von $\min(X)$ und $\max(X)$ auf eine Skala von 0 bis 1 skaliert und somit die Messwerte für weitere Berechnungen vergleichbar macht. Mathematisch sieht diese Skalierung wie folgt aus:

$$x_{ij,\text{norm}} = \frac{x_{ij} - \min(X)}{\max(X) - \min(X)} \quad \forall i = 1, \ldots, n \; j = 1, \ldots, m \tag{3.1}$$

Formel (3.1) führt die Normalisierung von x mithilfe des minimalen und maximalen Wertes von X durch. Die Variable x_{norm} hat nur noch Werte von 0 bis 1.
Anschließend können die Werte auch linear auf eine andere Skala transformiert werden, was insbesondere für nachfolgende Transformationsschritte relevant ist.

$$x_{ij,\text{scl}} = x_{ij,\text{norm}} \cdot (\max - \min) + \min \quad \forall i = 1, \ldots, n \; j = 1, \ldots, m \tag{3.2}$$

Die Formel (3.2) skaliert die normalisierten x-Werte linear auf eine andere Skala. Hierbei ist min der Anfangspunkt und max der Endpunkt der Skala.

Der `LogisticScaler` sollte nach dem `MinMaxScaler`, mit min $= -1$ und max $= 1$, angewendet werden. Mathematisch ist der `LogisticScaler` die Sigmoid-Funktion angewendet auf alle x-Werte:

$$\sigma\left(x_{ij}\right) = \frac{1}{1 + e^{-px_{ij}}} \quad \forall i = 1, \ldots, n \; j = 1, \ldots, m \tag{3.3}$$

Die Funktion kann eine Kante weicher oder härter machen, je nachdem wie der Parameter p gewählt wird. Um den gewünschten Effekt zu erreichen, muss die relevante Kante nach der Anwendung des `MinMaxScaler`s einen Vorzeichenwechsel mit ähnlichem Absolutwert vor und nach der Kante haben. Die relevante Kante ist im beschriebenen Anwendungsfall der Übergang zwischen nichtleitendem und leitendem Bereich. Kanten ohne Vorzeichenwechsel werden automatisch unterdrückt, was auch zu einer Unterdrückung von einem Großteil des Rauschens führt.
In Abbildung 3.3 sind Sigmoid-Funktionen mit verschiedenen Parametern dargestellt. Dort ist zu erkennen, dass, durch die Wahl eines hohen Wertes für $p > 0$, der Gradient von Kanten mit Vorzeichenwechsel erhöht wird, während er für Kanten außerhalb dieses Bereichs auf einen Wert nahe 0 gesenkt wird. Dadurch kann eine höhere Streuung an den

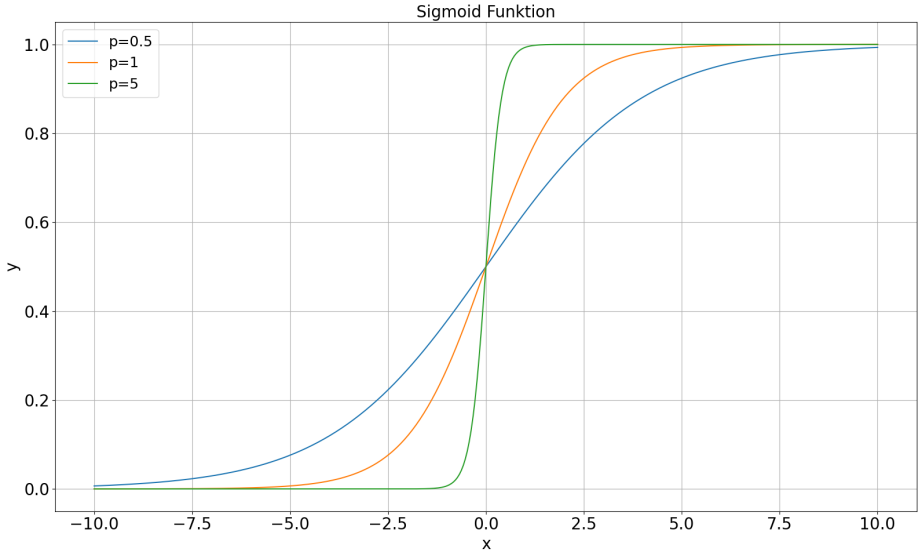


Abbildung 3.3: Sigmoid-Funktionen mit verschieden Faktoren für p

Kanten erzeugt werden und somit im nächsten Schritt besser zwischen Narrow Scans im nichtleitenden Bereich bzw. leitenden Bereich und Narrow Scans mit einer bzw. mehreren Kanten zwischen diesen Bereichen unterschieden werden. In Abbildung 3.4 wurde auf dem linken Bild nur der `MinMaxScaler` mit min $= 0$ und max $= 1$ angewandt, während auf dem rechten Bild der `MinMaxScaler` mit der angepassten Skala von -1 zu 1 und danach der `LogisticScaler` mit Parameter $p = 5$ angewandt wurde. Hier ist eine

Kontrastveränderung im Scan zu sehen. Die Werte, die um oder nahe der weichen Kante liegen, werden durch den Faktor erhöht bzw. gesenkt und die Kante wird so deutlicher.

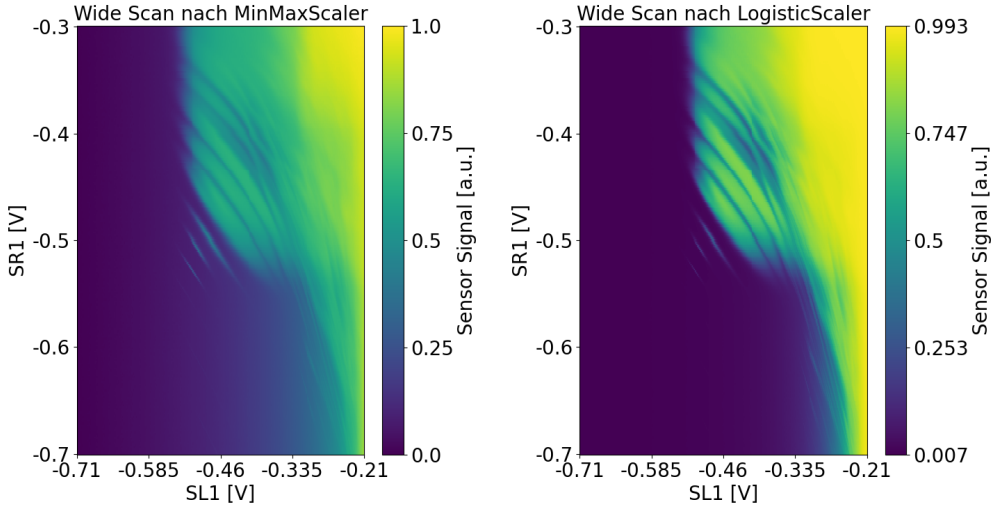


Abbildung 3.4: Beispielbild für `LogisticScaler`

Während der `LogisticScaler` einen weichen Schwellenwert erzeugt, wird über den `SignScaler` ein harter Schwellenwert erzeugt. Er erzeugt nur die zwei Werte -1 oder 1. Der Schwellenwert für die Sign-Funktion ist eigentlich 0, kann aber mithilfe des Parameters k verschoben werden, wie in Formel (3.4) gezeigt.

$$s(x_{ij}, k) = \text{sign}(x_{ij} - k) \quad \forall i = 1, \ldots, n \; j = 1, \ldots, m \tag{3.4}$$

In Abbildung 3.5 ist ein Beispiel für die Anwendung des `SignScaler`s gezeigt. Dort wird auch deutlich, dass durch die Anwendung des harten Schwellenwerts wichtige Informationen, wie ein Teil der Coulomb-Peaks, verloren gehen können.
Als letztes wurde noch der `ZScoreScaler` implementiert, welcher eine Standardisierung der Messwerte bewirkt, das heißt der Mittelwert und die Varianz eines Wide Scan ändern sich. Der Mittelwert wird zum Wert 0 und die Varianz zum Wert 1 nach der Anwendung dieser Transformation. Somit wird auch die Varianz in den Narrow Scans verändert. Im leitenden oder nichtleitenden Bereich sollte die Varianz um 0 liegen und an einer Kante oder Oszillation bei ungefähr 1. Durch die Standardisierung wird eine einheitliche Varianz bei Kanten, Oszillationen und leitendem, nichtleitendem Bereich erzeugt. Dadurch kann der Algorithmus diese Bereiche mit hoher Sicherheit unterscheiden. Die Berechnung dieses Normalisierungsverfahrens erfolgt über Formel (3.5).

$$z = \frac{x - \mu_X}{\sigma_X} \tag{3.5}$$

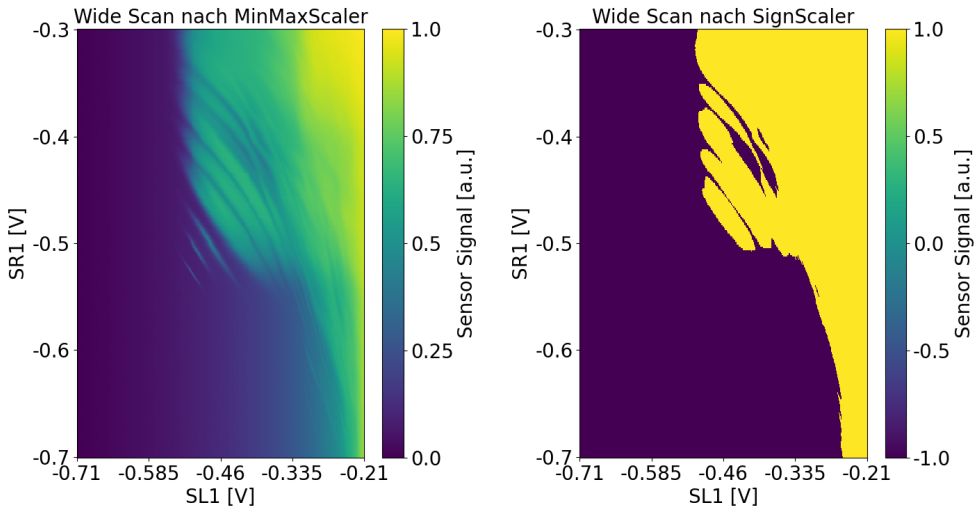


Abbildung 3.5: Beispielbild für `SignScaler`

Hierbei ist μ_X der Mittelwert und σ_X die Standardabweichung der Leitwerte des Wide Scans.

3.5 Kantenerkennung mithilfe von Dispersionsmaßen

Nach dem Normalisieren können Dispersionsmaße verwendet werden, um zu prüfen, ob sich Kanten bzw. Oszillationen im Scan befinden. Hierfür wird das 75%-Quantil der Dispersionswerte der Zeilen und Spalten berechnet. Wenn das 75%-Quantil sowohl in den Zeilen als auch in den Spalten einen gewissen Schwellenwert überschreitet, wird entweder eine 45°-Kante oder eine Oszillation vermutet. Die Überprüfung, dass es sich hier um einen Oszillationsbereich handelt, wird in Abschnitt 3.7 beschrieben. Wenn das nicht der Fall ist, wird überprüft, ob eine Überschreitung des Schwellenwerts nur in den Spalten bzw. nur in den Zeilen stattfindet. Dies lässt auf eine horizontale bzw. vertikale Kante schließen.
Die Schwellenwerte der einzelnen Dispersionmaße wurden empirisch, anhand von Beispielen, bestimmt, worauf genauer in Abschnitt 4.4 eingegangen wird. Es wurden folgende Dispersionsmaße implementiert [16]:

1. Varianz
2. Spannweite
3. Interquartilsabstand

4. Median der absoluten Abweichungen vom Median [17]

5. Mittlere absolute Abweichung vom Mittelwert [18]

Die Berechnung der Dispersionsmaße ist im Folgenden immer zeilenweise angegeben. Eine Berechnung für die Spalten des Scans funktioniert analog. Die empirische Varianz $\mathrm{Var}(x_i)$ der i-te Zeile bzw. empirische Standardabweichung σ_i der i-te Zeile ist in Formel (3.6) beschrieben, wobei μ_i das arithmetische Mittel der jeweiligen Zeile ist.

$$\mathrm{Var}(x_i) = \sigma_i^2 = \frac{1}{m} \sum_{j=1}^{m} (x_{ij} - \mu_i)^2 \tag{3.6}$$

Die Spannweite ist die Differenz zwischen minimalem und maximalem Wert der jeweiligen Zeile i des Scans:

$$\mathrm{R}(x_i) = \max(x_i) - \min(x_i) \tag{3.7}$$

Ähnlich ist auch der Interquartilsabstand mit dem Unterschied, dass nicht die Differenz zwischen kleinstem und größtem Wert, sondern zwischen oberem und unterem Quartil der i-te Zeile berechnet wird:

$$\mathrm{IQR}(x_i) = x_{i,0,75} - x_{i,0,25} \tag{3.8}$$

Der Median der absoluten Abweichungen vom Median, kurz MAD, ist für die Zeile i wie folgt definiert:

$$\mathrm{MAD}(x_i) = \mathrm{median}(|x_i - \mathrm{median}(x_i)|) \tag{3.9}$$

Für die mittlere absolute Abweichung vom Mittelwert wird der Median der Abweichungen durch das arithmetische Mittel ersetzt:

$$\mathrm{AAD}(x_i) = \frac{1}{m} \sum_{j=0}^{m} |x_{ij} - \mathrm{mean}(x_i)| \tag{3.10}$$

Hierbei ist mean das arithmetische Mittel.

3.6 Unterscheidung von leitendem und nichtleitendem Bereich

Wurde mithilfe der obigen Maße festgestellt, dass sich keine Kante im Scan befindet, so muss entschieden werden, ob der nichtleitende oder leitende Bereich aufgenommen

wurde, um darüber zu entscheiden, in welcher Richtung der nächste Scan aufgenommen wird. Da der minimale und maximale Leitwert bekannt sind, kann mit Hilfe eines Mittelwertes über den Narrow Scan entschieden werden, zu welchem der beiden Werte ein geringerer Abstand besteht. Wenn der Mittelwert des Narrow Scans kleiner ist als der Mittelwert zwischen $\min(X)$ und $\max(X)$, liegt er näher am Leitwert, der für den nichtleitenden Bereich zu erwarten ist, wie in Formel (3.11) gezeigt.

$$\begin{aligned} m(x) &< \frac{\min(X) + \max(X)}{2} \Rightarrow \text{nichtleitender Bereich} \\ m(x) &\geq \frac{\min(X) + \max(X)}{2} \Rightarrow \text{leitender Bereich} \end{aligned} \tag{3.11}$$

Hierbei ist $m(x)$ ein Mittelwert über den Narrow Scan. Als Mittelwert wurde das arithmetische Mittel und der Median betrachtet. Da es effizienter zu berechnen ist und ähnliche Abstände zum Minimum und Maximum erzeugt, wird im Folgenden das arithmetische Mittel verwendet.
Im Algorithmus wird vor der Bestimmung, ob es sich um den leitenden oder nichtleitenden Bereich handelt, ein Normalisierungsverfahren angewandt. Je nach angewandtem Normalisierungsverfahren müssen statt $\min(X)$ bzw. $\max(X)$ die für die Skalierung verwendeten Werte min und max verwendet werden.

3.7 Pattern-Matching

Wurde im Schritt „Oszillation gefunden?“ festgestellt, dass eine 45°-Kante vorliegt, so muss anschließend geprüft werden, ob es sich dabei um den Oszillationsbereich handelt oder nicht und ob es sich um den Anfangspunkt des Oszillationsbereich handelt. Dies wird mithilfe von Pattern-Matching geprüft. Pattern-Matching bezieht sich auf Verfahren, bei denen diskrete Strukturen oder Teilmengen davon, wie der Startpunkt des Coulomb-Oszillationsbereichs, anhand eines vordefinierten Musters (Pattern) identifiziert werden [19]. Hierfür enthält die Bibliothek OpenCV [20] die Methode `matchTemplate` mit verschiedenen Vergleichsmethoden. Unter den Vergleichsmethoden sind Verfahren wie die Kreuzkorrelation, der Korrelationskoeffizient oder die Summe der quadratischen Differenzen sowie die dazugehörigen normierten Verfahren [21].
Um das Pattern-Matching anwenden zu können, muss zunächst eine Oszillation im aufgenommenen Scan vorliegen. Wenn das gewählte Dispersionsmaß den gewählten Schwellenwert in den Spalten und in den Zeilen überschreitet, dann wurde entweder eine diagonale Kante oder der Oszillationsbereich gefunden. Um diese Fälle zu unterscheiden, wird zunächst geprüft, wie viele Coulomb-Peaks im Scan vorhanden sind. Dazu werden mithilfe der Methode `signal.find_peaks` [22] des `scipy`-Moduls Peaks gesucht, wobei die in Abschnitt 3.2 bestimmten Abstände und Breiten der Peaks als Merkmale mitgegeben werden. Wenn die Anzahl der Peaks größer gleich 2 ist, wird dies als gefundener Oszillationsbereich gekennzeichnet. Wenn dies nicht der Fall ist, dann hat der Algorithmus eine diagonale Kante gefunden und bricht hier ab.

Um anschließend zu prüfen, ob sich auch die untere linke Ecke des Oszillationsbereichs im Narrow Scan befindet, werden zunächst Templates für das Pattern-Matching bestimmt. Hierfür wurden zwei Ansätze betrachtet:

1. Ein vorgefertigtes Template strecken, drehen und verschieben

2. Ein Template aus den Daten der Scans erzeugen

Für die erste Idee müsste man sich aus den bekannten Merkmalen der Coulomb-Oszillationen zunächst ein Template vorfertigen. Das Template muss anschließend so gedreht werden, dass es entlang der Oszillationen ausgerichtet ist. Dazu müsste zunächst bestimmt werden, welche Peaks jeweils zusammen zu einer Diagonalen gehören, sodass man darüber einen durchschnittlichen Winkel für die diagonalen Wellenfronten bestimmen kann. Außerdem müsste es so gestreckt und verschoben werden, dass die Positionen der Peaks mit den Positionen der Peaks im Scan übereinstimmen. Dieses Problem wäre aber leicht lösbar, wenn die Positionen der Peaks bekannt sind. Ein weiteres Problem sind die Unterschiede in den Höhen der Peaks. Die Höhe der Peaks steigt stetig. Hier müsste zuerst ein Verfahren entwickelt werden, um ein vorgefertigtes Template, nicht nur zu strecken, drehen und verschieben, sondern auch die unterschiedlichen Höhen der Peaks einzubauen und dadurch eine Gruppe von Datenpunkten im Template zu verändern.
Die zweite Idee hat den Vorteil, dass die Positionen der Peaks im Scan direkt verwendet werden können, um ein passend ausgerichtetes Template zu erzeugen. Je nach verwendeter Modellfunktion für die Templates, können auch die unterschiedlichen Höhen der Peaks sofort in das Template eingebaut werden. Aufgrund dieser Vorteile wurde nur das zweite Verfahren im Algorithmus umgesetzt.
Da die Scans in den Zeilen eine höhere Auflösung haben als in den Spalten, wurde mithilfe von `scipy.optimize.curve_fit` zunächst eine 1D-Regressionsanalyse pro Zeile durchgeführt [23]. Diese Methode versucht eine vordefinierte Funktion an die Daten der Zeile anzupassen. Hierfür kann man Startwerte für die Parameter der Funktion mitgeben. Diese Startwerte werden mithilfe des Levenberg-Marquardt-Verfahrens angepasst, sodass der Fehler nach der Methode der kleinste Quadraten minimal wird [24].
Für die 1D-Regressionsanalyse wurden zwei Modellfunktionen getestet, eine Sinuskurve, in Abschnitt 3.7.1 beschrieben, und eine Summe von Gauß-Glockenfunktionen, in Abschnitt 3.7.2 beschrieben. Die Sinuskurve wurde gewählt, da die Peaks periodisch vorkommen. Hier wurden aber weder die unterschiedlichen Höhen der Peaks noch der leitende und nichtleitende Bereich beachtet, was zu zusätzlichen Oszillationen am Rand führen kann, wie in Abschnitt 4.6 erklärt.
Um diese Probleme zu umgehen, wurde zusätzlich eine Summe von Gauß-Glockenfunktionen zur Template-Erzeugung implementiert. Diese beachtet die Höhenunterschiede der Peaks und da nur gefundene Peaks mit der Funktion nachgestellt werden, gibt es auch keine Probleme mit zusätzlichen Oszillationen am Rand. Die Gauß-Glockenfunktion beschreibt die Peaks somit insgesamt besser als die Sinusfunktion, was sich auch in Abschnitt 4.6 zeigt.

Da die Daten ein periodisches Verhalten zeigen und die Sinusfunktion ebenfalls periodisch ist, gibt es hier unendlich viele Möglichkeiten, die Funktion an die Daten anzulegen. Dadurch gibt es auch unendlich viele lokale Minima, in die die Funktion für die Regressionanalyse geraten kann, weshalb die Startwerte vor allem für die Sinusfunktion eine hohe Bedeutung haben. Ein weiterer Grund für die Wahl der Startwerte ist die Laufzeit. Das Optimierungsverfahren passt die Startwerte iterativ an. Je besser die Startwert gewählt wurden, desto weniger Iterationen benötigt das Verfahren, wodurch die Laufzeit niedriger ist. Um das Ziel einer möglichst geringen Laufzeit für die Kalibrierung der Sensorpunkte zu erreichen, hat die Wahl der Startwerte also eine hohe Bedeutung.
Um herauszufinden, ob die linke untere Ecke des Oszillationsbereichs gefunden wurde, wird der Narrow Scan zunächst in 2x2 kleinere Scans unterteilt. Da oft nur in der rechten oberen Ecke des Narrow Scans Coulomb-Oszillationen sichtbar sind, wird nur aus dem rechten oberen Teilstück ein Template generiert. Die Teilstücke werden anschließend mithilfe der Funktion `matchTemplate` [21] mit dem erzeugten Template verglichen. Von den durch OpenCV zur Verfügung gestellten Vergleichsmethoden hat sich für die Arbeit `TM_CCOEFF` als geeignetste Methode herausgestellt. Diese berechnet den Korrelationskoeffizenten zwischen dem Template und den gemessenen Daten aus. Wenn der Absolutwert von diesem hoch ist, dann sind das Template und die Daten sehr ähnlich. Wenn aber dieser Wert gegen 0 strebt, haben die Daten mit dem Template wenig bis keine Ähnlichkeit.
Wenn nur das obere rechte Teilstück gut korreliert und die anderen Teilstücke wesentlich weniger Korrelation zeigen, liegt der gesuchte Punkt mit hohe Wahrscheinlichkeit in der Mitte des Narrow Scans. Hierbei wird die Korrelation des oberen rechten Teilstücks R_{OR} als Vergleichswert zur Korrelationen des Templates mit den anderen Teilstücken $R_k$verwendet. Der absolute Quotient zwischen den Korrelationen R_k und R_{OR} wird anschließend als Entscheidungsmerkmal für eine Verschiebung des Narrow Scans genutzt, wie in Formel (3.12) beschrieben.

$$\left|\frac{R_k(x)}{R_{\mathrm{OR}}(x)}\right| > 0,2 \Rightarrow \text{Verschieben} \qquad k \in \{\mathrm{UR}, \mathrm{UL}, \mathrm{OL}\} \tag{3.12}$$

Wenn für eines der drei Teilstücke, die nicht in die Erzeugung des Templates eingeflossen sind, der Schwellenwert von 0,2 überschritten wird, dann wird der Scan in die Richtung des entsprechenden Teilstücks verschoben, da in dieser Richtung weitere ähnliche Oszillationen liegen und man somit vermutlich dem Startpunkt des Oszillationsbereichs näher kommt. Dementsprechend wird auch der Anfang des Oszillationsbereichs in dieser Richtung erwartet.
Wenn es mehr als ein Teilstück gibt, was diesen Wert überschreitet, wird in die Richtung verschoben mit dem größten absoluten Quotienten der Korrelationen. Grund hierfür ist, dass je höher der Quotient ist, desto ähnlicher sind sich die beiden Teilstücke. Daher verschiebt der Algorithmus immer in die Richtung des maximalen Quotienten. Im Fall, dass keiner der Teilstücke den Wert überschreitet, liegt die linke untere Ecke genau zwischen den 4 Teilstücken bzw. im Mittelpunkt des Narrow Scan. Somit kann der Al-

gorithmus abbrechen und den Mittelpunkt als linke untere Ecke des Oszillationsbereichs zurückgeben.

3.7.1 Template basierend auf Sinusfunktion

Die Sinus-Modellfunktion zur Erzeugung des Templates sieht wie folgt aus:

$$f(x) = a\sin(bx + c) + d \tag{3.13}$$

Hierbei ist a die Skalierung der Amplitude, b die Frequenz und c und d Verschiebungen. Um geeignete Startwerte für die Kurvenanpassung zu erhalten, werden diese Parameter mit folgenden Verfahren geschätzt: a wird mit der Standardabweichung geschätzt, da diese eine Streuung der Funktionswerte beschreibt und über einen höheren Skalierungsfaktor auch eine höhere Streuung erreicht wird. d ist das in Abschnitt 3.6 genutzte arithmetische Mittel der Werte vom Scan. b wird mithilfe der Fourier-Transformation geschätzt [25]. Und c konnte nicht ohne weiteres geschätzt weren und wurde daher mit 0 initialisiert.
Die Modellfunktion, die Daten und die geschätzten Startwerte werden in die `scipy.curve_fit`-Methode gegeben, welche die Startwerte iterativ anpasst, sodass die Summe der kleinsten Quadrate minimal wird. Abbildung 3.6 zeigt auf der linken Seite einen Narrow Scan mit Coulomb-Oszillationen und auf der rechten die angepasste Sinus-Modellfunktion.

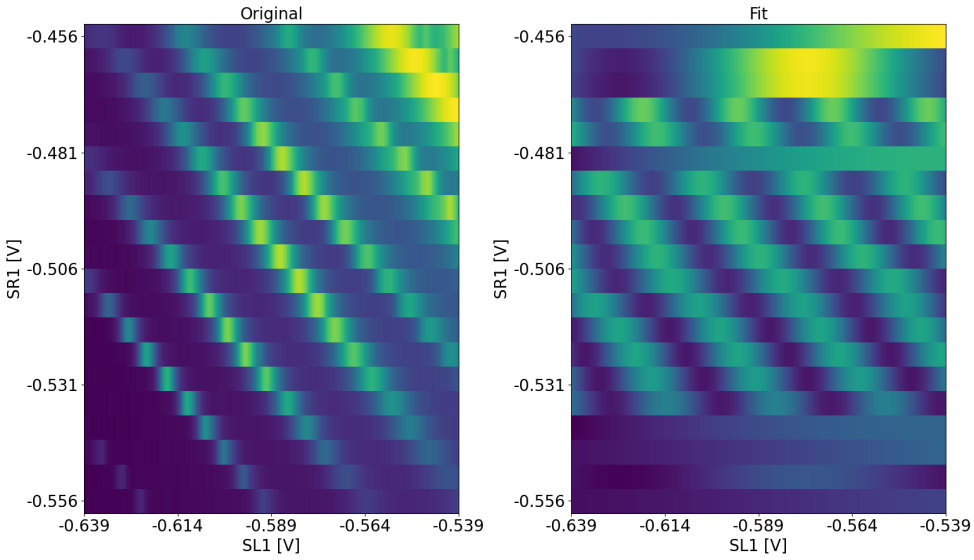


Abbildung 3.6: Narrow Scan und angepasste Sinus-Modellfunktion

3.7.2 Template basierend auf Gaußfunktionen

Die Gauß-Modellfunktion für einen einzelnen Coulomb-Peak sieht wie folgt aus:

$$f(x) = a \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{3.14}$$

Da mithilfe der Gauß-Glockenfunktion nur ein Peak darstellbar ist, muss eine Summe von Gaußfunktionen verwendet werden, um eine Modellfunktion für eine Zeile zu erhalten.
Die Position der Peaks ist bereits durch den Aufruf von `scipy.find_peaks` bekannt. Mithilfe des Ergebnisses dieser Funktion wird ein Intervall basierend auf der Breite und der Position des jeweiligen Peaks erstellt. Nun werden die Parameter der Gauß-Glockenfunktion für jedes Intervall geschätzt. μ ist die Verschiebung bzw. die Position des Maximums der Funktion und wird mithilfe des Mittelwertes des Peak-Intervalls berechnet. a ist die Höhe des Maximums der Funktion und wird mit der Höhe des Peaks geschätzt. σ beeinflusst die Breite der Funktion bzw. des Peaks und wird mithilfe der empirischen Standardabweichung des Intervalls um einen Peak geschätzt. Diese Startwerte werden ebenfalls in die `scipy.curve_fit`-Funktion gegeben. Am Ende werden die so erhaltenen Gauß-Funktionen aufsummiert und bilden so ein Template für eine Zeile. Abbildung 3.7 zeigt auf der linken Seite einen Narrow Scan mit Coulomb-Oszillationen und auf der rechten die angepasste Summe aus Gauß-Modellfunktionen.

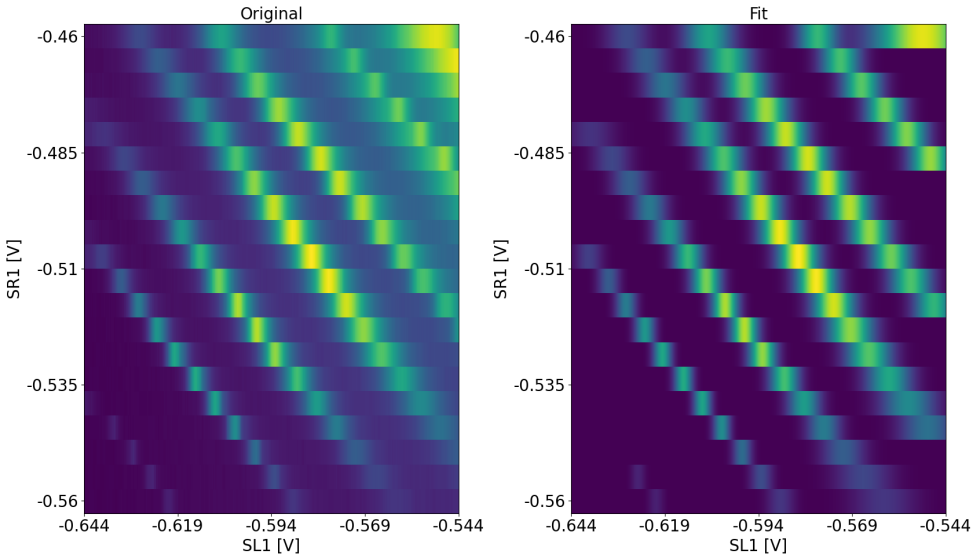


Abbildung 3.7: Narrow Scan und angepasste Gauß-Modellfunktion

3.8 Schrittweite

Die Schrittweite für die Verschiebung der Narrow Scans hat eine hohe Bedeutung für die Effizienz des Algorithmus. Sie darf nicht zu groß sein, da der Algorithmus sonst Kanten und den Oszillationbereich übersehen kann. Diese darf aber auch nicht zu klein sein, weil der Algorithmus sonst viele Narrow Scans benötigt, was zu einer höheren Laufzeit führt. Zunächst werden zwei verschieden Fälle für die Schrittweite unterschieden. Im ersten Fall muss der Narrow Scan verschoben werden, da keine Oszillationen vorliegen. Im zweiten Fall liegen Oszillationen vor, aber die linke untere Ecke des Oszillationsbereiches ist noch nicht gefunden.
Wenn der Algorithmus noch keine Oszillation gefunden hat, wird die Größe des Narrow Scans, welcher eine Standardgröße von 200mV pro Gate hat, halbiert und als Schrittweite verwendet. Diese wird je nach Fall, zur Spannung der Gates SL1 und SR1 addiert oder von dieser subtrahiert. Durch die Überlappung aufeinanderfolgender Scans, ist die Wahrscheinlichkeit gering, dass der Algorithmus eine Kante bzw. den Oszillationsbereich übersieht, weil sie sich immer am Rand der Narrow Scans befindet. Die Richtung der Verschiebung hängt in dem Fall davon ab, ob sich die horizontale oder vertikale Kante im Scan befindet oder ob es sich um den leitenden oder nichtleitenden Bereich handelt, vergleiche dazu Abbildung 3.2.
Im zweiten Fall befindet sich der Algorithmus schon im Oszillationsbereich und der Quotient der Korrelationswerte aus Formel (3.12) wurde für die Teilstücke des Narrow Scans berechnet. Überschreitet der Quotient den in Formel (3.12) angegebenen Schwellenwert, wird er mit der Schrittweite multipliziert und der Scan in die Richtung des höchsten Quotienten, wie in Abschnitt 3.7 beschrieben, verschoben. Quotienten mit einem Wert größer als 100%, werden auf 100% gesetzt bevor sie als Faktor verwendet werden. Durch die Multiplikation mit dem Quotienten wird die Schrittweite verkleinert und die Chance erhöht, dass der Algorithmus die linke untere Ecke findet.

4 Ergebnisse

In diesem Kapitel wird zunächst erklärt, wie die Daten, die in Abschnitt 3.1 beschrieben sind, zur Bewertung gelabelt wurden. Anschließend werden auf Basis dieser Labels Bewertungsmetriken definiert. Darauf folgend werden mithilfe der definierten Metriken die einzelnen Teilergebnisse des Algorithmus bewertet und als letztes die Bewertung des Endergebnis des kompletten Algorithmus dargestellt, welches daraus besteht, wie gut die linke untere Ecke des Coulomb-Oszillationsbereiches detektiert wurde.

4.1 Labels

Um zu bewerten, wie gut die Ergebnisse des Algorithmus sind, werden gelabelte Daten benötigt, die das erwartete Ergebnis enthalten.
Um das Gesamtergebnis des Algorithmus bewerten zu können, bildet der Punkt, welcher die linke untere Ecke des Coulomb-Oszillationsbereiches markiert, dieses Label. Die simulierten Daten enthalten bereits ein solches Label in der Form eines Punktes aus SL1- und SR1-Spannung. Dieses Label fehlt bei den experimentellen Daten und muss manuell hinzugefügt werden. Hierfür werden zunächst die Endpunkte der diagonal im Scan verlaufenden Coulomb-Peaks gesucht. Anschließend wird eine Regressionsgerade durch die jeweiligen Endpunkte gelegt. Der Schnittpunkt der beiden Regressionsgeraden wird als Label verwendet, wie in Abbildung 4.1 dargestellt.

Das Label wird zusammen mit dem Namen der Matlab-Datei in eine JSON-Datei gespeichert, wie in Listing 1 beispielhaft gezeigt wird.

```
{
    "Datei_001.mat":
    {
        "x": -0.234,
        "y": -0.623
    },
    "Datei_002.mat":
    {
        "x": ...,
        "y": ...
    },
    ...
}
```

Listing 1: JSON Datei zur Speicherung des Labels für die linke untere Ecke des Oszillationsbereichs

Es gibt aber auch experimentelle Daten, bei denen es nicht möglich ist, den Oszillationsbereich genau zu bestimmen und damit auch ein Label zu setzen. Zwei Beispiele für

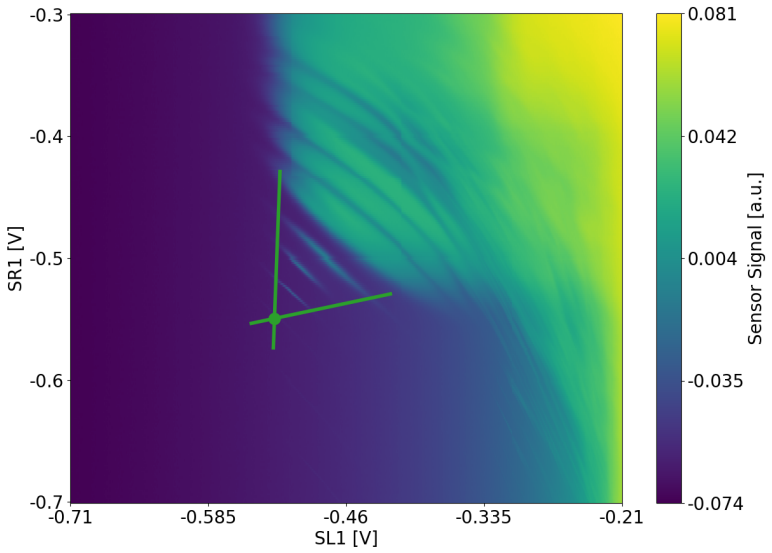
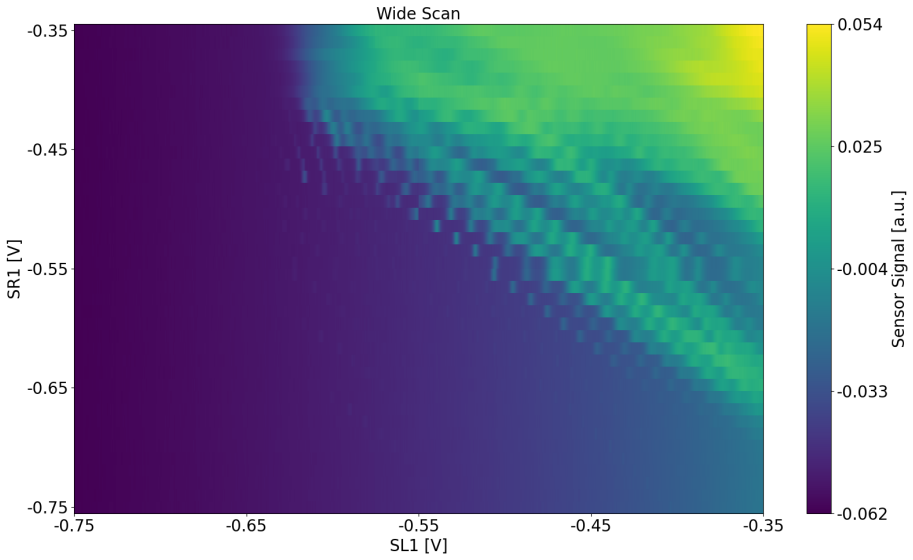


Abbildung 4.1: Beispielbild zum Vorgehen bei der Bestimmung des Labels für die linke untere Ecke des Oszillationsbereichs

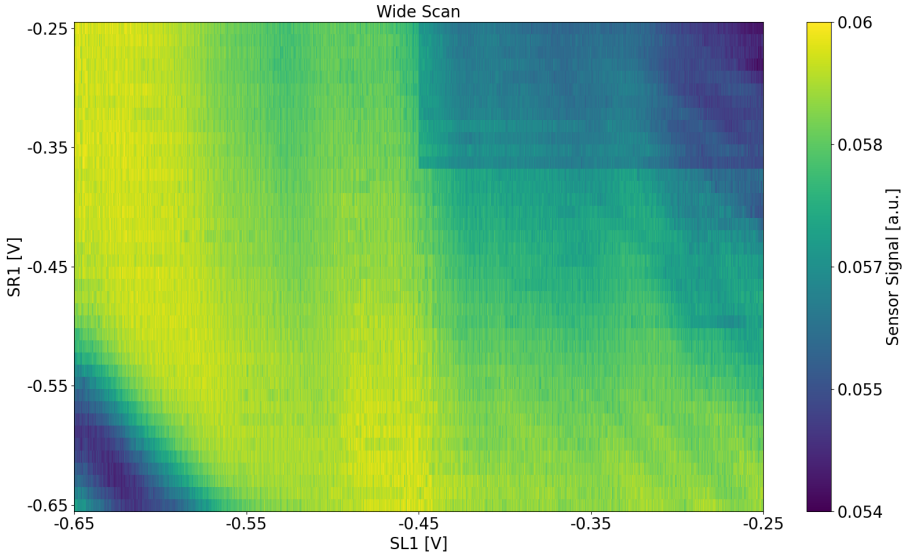
solche Scans befinden sich in Abbildung 4.2. Scans wie diese wurden aus dem dazugehörigen Datensatz entfernt, da ohne Label das Ergebnis des Algorithmus nicht sinnvoll bewertet werden kann.
Um die Verfahren für die jeweiligen Teilschritte des Algorithmus (siehe Abschnitt 3.3) bewerten zu können, wurden zusätzlich Narrow Scans gelabelt. Für die einzelnen Schritte des Algorithmus wird überprüft, ob der Algorithmus erkennt, welche Art von Messung vorliegt. Hierfür wurden die Scans in die Kategorien

- leitender Bereich
- nichtleitender Bereich
- horizontale Kante
- vertikale Kante
- Oszillationsbereich

aufgeteilt. Beispiele zu jeder Kategorie sind in Abbildung 4.3 gezeigt. Die zur Bewertung verwendeten Scans wurden zu gleichen Anteilen aus dem experimentellen und dem simulierten Datensatz ausgewählt.

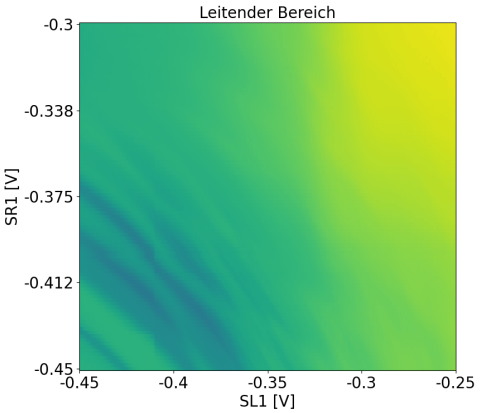


(a)

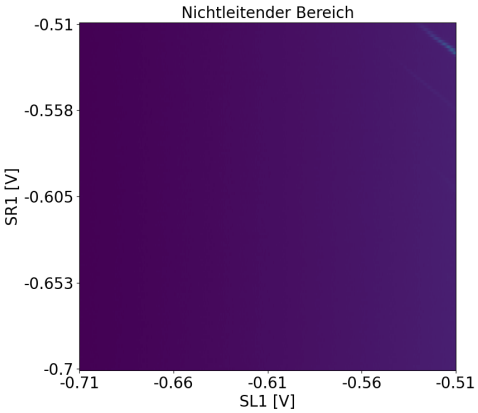


(b)

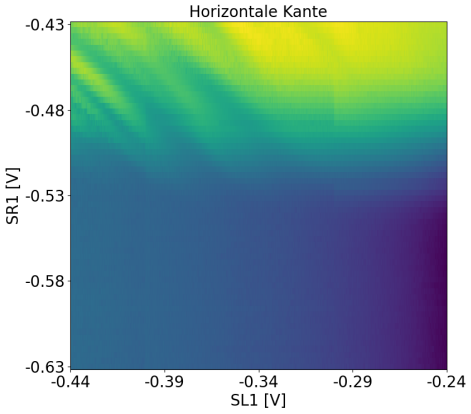
Abbildung 4.2: Wide Scans ohne klare Ecke



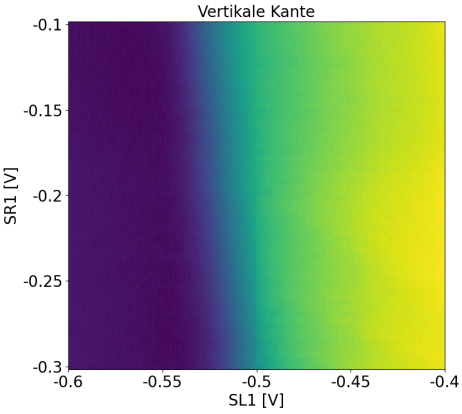
(a) Leitender Bereich



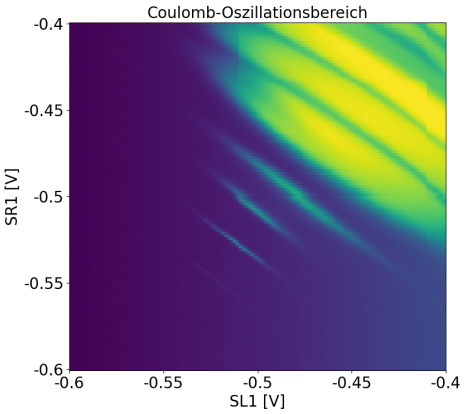
(b) Nichtleitender Bereich



(c) Horizontale Kante



(d) Vertikale Kante



(e) Coulomb-Oszillationsbereich

Abbildung 4.3: Beispiele für Narrow Scans pro gelabelter Kategorie

4.2 Bewertungsmetrik

Zur Bewertung des Erfolgs des Algorithmus wird zum einen die Qualität der Ergebnisse bewertet und zum anderen die Laufzeit im Vergleich zur Nutzung von Wide Scans betrachtet.
Der Algorithmus gibt die Koordinaten der linken unteren Ecke des Coulomb-Oszillationsbereichs zurück. Zur Bewertung der Genauigkeit des Ergebnisses wird der Abstand zwischen den Koordinaten des Labels und der Ausgabe des Algorithmus verwendet. Der Abstand wird hierbei mit der euklidischen Norm bestimmt, die in Formel (4.1) abgebildet ist.

$$f_d = \|x - y\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{4.1}$$

Hierbei ist x der vom Algorithmus bestimmte Punkt und y das Label. Je kleiner die Distanz zwischen den Werten ist, desto besser wurde der Beginn des Oszillationsbereiches vom Algorithmus detektiert. Wenn der Abstand aber eine gewisse Länge überschreitet, sodass die Koordinaten des Labels nicht mehr auf dem Narrow Scan zu finden sind, dann war der Algorithmus nicht erfolgreich. Der Algorithmus ist ebenfalls nicht erfolgreich, wenn keine Oszillationen im finalen Narrow Scan zu sehen sind.

Ein weiterer wichtiger Aspekt zur Bewertung des Suchalgorithmus ist die Anzahl der Narrow Scans, die benötigt werden, um die Ecke des Oszillationsbereichs erfolgreich zu finden. Wenn die Anzahl von 45 Scans überschritten wird, gibt es keine Laufzeitverbesserung gegenüber einem Wide Scan. Die Aufnahme eines Wide Scans dauert 10-20 Minuten, während die Aufnahme eines Narrow Scans 10 - 30 Sekunden dauert. Je weniger Narrow Scans der Algorithmus also benötigt, desto besser schneidet er ab, solange der Algorithmus nach obiger Definition noch erfolgreich ist.
Da der Algorithmus aus vielen Teilschritten besteht, ist nicht nur das Endergebnis interessant, sondern auch welches Vorgehen sich für welchen Teilschritt am besten eignet. Um dies zu bewerten, wird geprüft, ob der Algorithmus entsprechende Scans korrekt den in Abschnitt 4.1 erwähnten Kategorien zuordnet. Für die Untersuchungen wurden jeweils 30 Scans aus den experimentellen und simulierten Daten verwendet.

4.3 Bewertung der Normalisierungsverfahren

Der erste Schritt des Algorithmus ist die Normalisierung der Daten, um in den folgenden Schritten vergleichbare Ergebnisse zu erhalten. Hierfür wurden vier verschiedene Techniken, welche in Abschnitt 3.4 beschrieben sind, implementiert und getestet. Im Folgenden werden Beispiele und Probleme dieser Techniken aufgezählt und zusammengefasst.
Der `MinMaxScaler` skaliert die Daten auf eine einheitliche Skala und ist dadurch für weitere Verfahren und Normalisierungsschritte besonders wichtig. Im Vergleich zu anderen angewandten Normalisierungsverfahren wird der Kontrast im Bild nicht verändert,

sodass es bei dieser Art der Normalisierung schwierig sein kann, Kanten im Bild zu erkennen.
Der `LogisticScaler` und `SignScaler` sind vom Prinzip sehr ähnlich. Während der `LogisticScaler` einen Verlauf der Daten produziert, auch soft thresholding genannt, erzeugt der `SignScaler` eine harte Kante, auch hard thresholding genannt. Wenn Coloumb-Peaks zu niedrig sind, werden diese von `SignScaler` komplett entfernt. Deshalb ist die richtige Einstellung des Parameters k wichtig. Der optimale Wert des Parameters ist schwierig zu bestimmen, da er adaptiv an die jeweils sichtbaren Kanten angepasst werden müsste. In Abbildung 4.4 sind die Ergebnisse für verschiedene Werte von k beispielhaft gezeigt. Stattdessen wurde hier als Alternative der `LogisticScaler` benutzt.

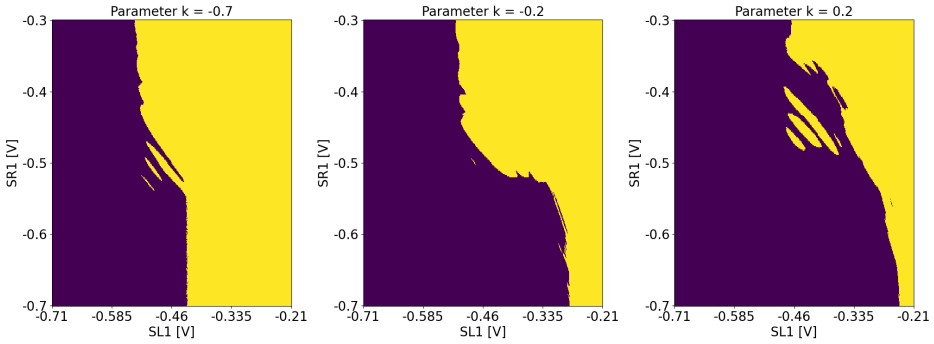


Abbildung 4.4: `SignScaler`-Beispiele für verschiedene Parameterwerte

Dieser kann über den Faktor p Kontraste verstärken. Je höher der Faktor p gewählt wird, desto ähnlicher wird das Ergebnis dem Ergebnis des `SignScaler`s. Auch hier kann es passieren, dass schwache Coulomb-Peaks weiter geschwächt werden, bis hin zum Verschwinden der Peaks, aber nicht so extrem wie beim `SignScaler`. Der Parameter p sollte hier im Bereich zwischen 1 und 10 gehalten werden, um nicht zu viele Coulomb-Peaks zu entfernen. Durch die Anwendung auf 40 Beispielbilder wurde für $p = 5$ als geeigneter Standardwert festgelegt. In Abbildung 4.5 sind Beispielbilder für unterschiedliche Werte von p dargestellt. Hier ist inbesondere zu erkennen, dass mit höheren Werten mehr Peaks verschwinden.
Sowohl der `SignScaler` als auch der `LogisticScaler` haben Probleme damit schwache Peaks zu erhalten, wenn die Höhe der Peaks vom Beginn des Oszillationsbereich an stetig steigt. Bei einer Stichprobe von 30 Scans verschwinden bei der Anwendung des `SignScaler`s bei 86,6% der Scans Coulomb-Peaks. Kanten verschwinden nicht, können sich aber etwas verschieben. Beim `LogisticScaler` verschwinden nur bei 23,3% der Testdaten Peaks komplett. Diese Peaks sind aber so schwach, dass diese auch ohne die Verwendung des `LogisticScaler` in nachfolgenden Schritten kaum vom Algorithmus erkannt werden. Kanten verschwinden nicht, können sich aber ebenfalls wie beim `SignScaler` verschieben. Die Verschiebung der Kante ist beim `SignScaler` aber wesentlich ausgeprägter als beim `LogisticScaler`, wie man schon in Abbildung 4.4 und 4.5 erkennen kann.

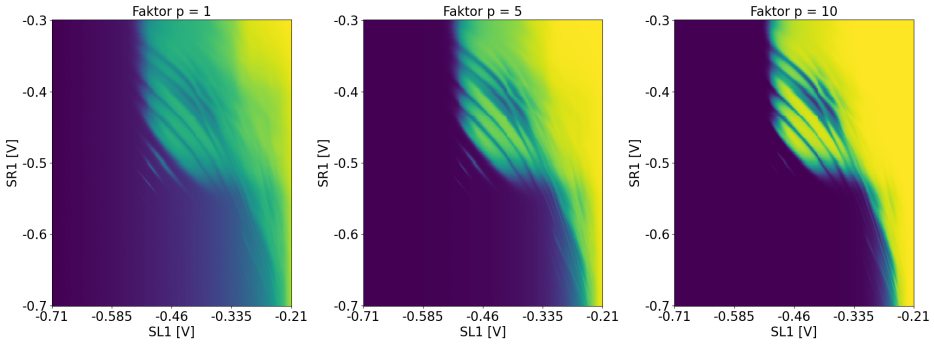


Abbildung 4.5: `LogisticScaler`-Beispiele für verschiedene Parameterwerte

Ein weiteres in Abschnitt 3.4 vorgestelltes Verfahren, ist der `ZScoreScaler`. Für die Anwendung des `ZScoreScaler` wird das arithmetische Mittel und die Standardabweichung des Wide Scans benötigt. Da dem Algorithmus nur Narrow Scans zur Verfügung stehen, aber der Mittelwert und die Standardabweichung des Wide Scans benötigt würde, ist es schwierig diese Skalierung anzuwenden. Der Mittelwert könnte mit dem minimalen und maximalen Leitwert geschätzt werden. Dies ist für die Standardabweichung nicht möglich. Da sich die Narrow Scans am Anfang entweder im leitenden oder nichtleitenden Bereich, an einer Kante oder im Oszillationsbereich befinden, wären die erzeugten Werte nicht vergleichbar, wenn statt der Standardabweichung des Wide Scans die Standardabweichung des jeweiligen Narrow Scans verwendet werden würde. Aus diesem Grund wurde diese Transformation nicht angewandt.

Für den Algorithmus wird der `MinMaxScaler` zusammen mit dem `LogisticScaler` mit dem Faktor $p = 5$ verwendet, da er durch die Kontrasterhöhung die Detektion von Kanten erleichtert und mit dem Parameter $p = 5$ nur einen kleinen Teil der schwachen Coulomb-Oszillationen entfernt.

4.4 Bewertung der Dispersionsmaße zur Kantenerkennung

Ein entscheidender Schritt im Algorithmus ist die Erkennung von Kanten. Dies wird durch den Vergleich eines Schwellenwerts mit dem Ergebnis eines Dispersionsmaßes erreicht. Für den Vergleich mit dem Schwellenwert wird das 75%-Quantil der Dispersionswerte in den Zeilen bzw. Spalten verwendet. Die Wahl des 75%-Quantils hat sich als passend erwiesen, da es Ausreißer in den Daten ignoriert und sicherstellt, dass mindestens 25% der Zeilen bzw. Spalten über dem Schwellenwert liegen. Letzteres ist insbesondere wichtig, wenn eine Kante leicht vom 0° bzw. 90° Winkel abweicht, da in diesem Fall im Bereich der Kante sowohl entlang der Zeilen als auch entlang der Spalten eine hohe Dispersion vorhanden ist. Würde der maximale Wert statt des 75%-Quantils verwendet

werden, so würden solche Kanten als diagonale Kanten interpretiert werden.
Es wurden für Testzwecke 80 Scans ausgewählt bzw. erzeugt, um zu testen, mit welchem Dispersionsmaß Kanten und Oszillationen am besten vom nichtleitenden und leitenden Bereich unterschieden werden können. Die Auswahl der Scan-Beispiele erfolgte wie folgt: Für die Dispersion an Kanten oder Oszillationen wurden vertikale Kanten in einem Wide Scan ausgewählt. Es wurden 5 Narrow Scans um die Kante herum erstellt, sodass die Kante sowohl am Rand als auch in der Mitte erfasst wurde. Gleiches gilt für den Oszillationsbereich, wobei die Narrow Scans diagonal nach rechts oben bewegend aufgenommen wurden. Für die Dispersion im nichtleitenden Bereich wurde die linke untere Ecke des Wide Scan verwendet. Der Narrow Scan wurde diagonal nach rechts oben verschoben, ohne dabei auf eine Kante oder einen Oszillationsbereich zu stoßen. Ein ähnlicher Ansatz wurde für den leitenden Bereich angewendet, wobei der Narrow Scan beginnend ab der rechten oberen Ecke des Wide Scans diagonal nach links unten verschoben wurde. Dadurch wurden vergleichbare Mengen an Daten für beide Szenarien erfasst. Die Ergebnisse der Dispersionsmaße der Scans werden in Abbildung 4.6 als Boxplots dargestellt.

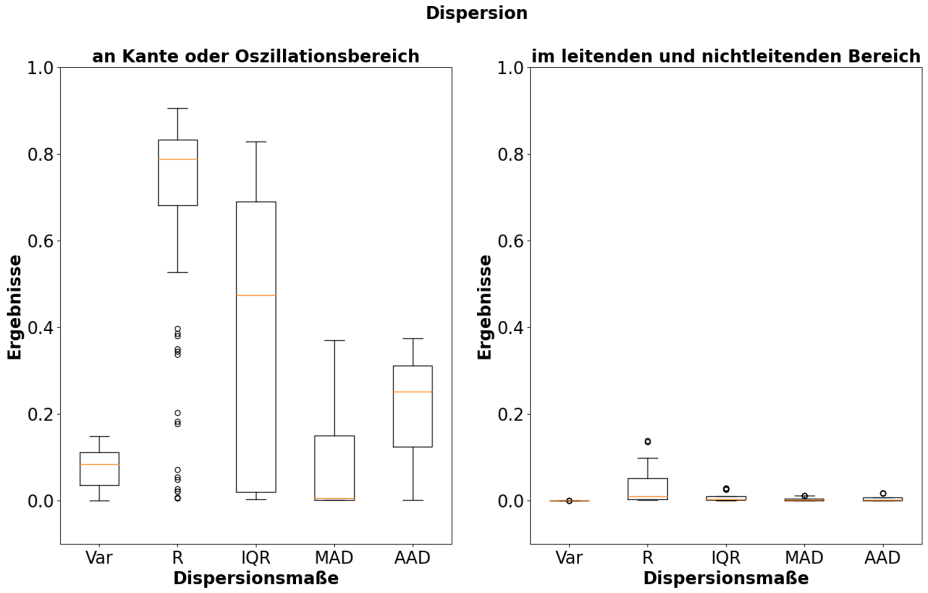


Abbildung 4.6: Ergebnisse der Dispersion an Oszillationsbereich und vertikaler Kanten im Vergleich zum leitenden und nichtleitenden Bereich aus insgesamt 80 Beispiel Scans

In Abbildung 4.6 fällt auf, dass keines der Dispersionsmaße für den nichtleitenden/leitenden Bereich signifikante Ausschläge zeigt, außer der Spannweite R. Dies deutet darauf hin, dass dieses Maß auch auf Rauschen reagiert. Bei vertikalen Kanten reagiert die Spannweite R ebenfalls mit hohen Werten. Aus diesem Grund ist die Wahl eines angemessenen Schwellenwerts besonders wichtig, um Kanten von Rauschen zu unterscheiden. Der Median der absoluten Abweichung vom Median kurz MAD zeigt einen Median nahe

Dispersionmaß	Schwellenwert
Varianz	0,05
Spannweite	0,5
Interquartilsabstand	0,1
Mediale Abs. Abweichung	-
Mittlere Abs. Abweichung	0,1

Tabelle 4.1: Dispersionmaße mit dazugehörigem Schwellenwert

null, da die Kante sich mittig im Scan befinden muss, um hohe Werte für MAD zu erhalten. Daher ist es mit diesem Dispersionsmaß nicht möglich einen passenden Schwellenwert zu finden, der es ermöglicht den leitenden/nichtleitenden Bereich von Bereichen mit Kanten oder Oszillationen zu unterscheiden. Varianz, Interquartilsabstand und mittlere absolute Abweichung (AAD) reagieren kaum auf Kanten am Rand des Scans, weshalb ihre Minima nahe null liegen. Der Interquartilsabstand zeigt jedoch große Schwankungen zwischen den Quartilen. Varianz und mittlere absolute Abweichung sind stabiler.
Auf Basis der Ergebnisse in Abbildung 4.6 wurden die Schwellenwerte in Tabelle 4.1 gewählt.
Wenn man die Dispersionsmaße mit diesen Schwellenwerten auf den zur Beginn des Abschnitts beschriebenen Datensatz und ähnliche Scans anwendet, schneiden Varianz und IQR am besten ab. Beide erkennen die Kanten und können diese auch richtig zuordnen. Dies bedeutet, dass sie zwischen horizontalen und vertikalen Kanten sowie diagonalen Kanten, wie sie beim Oszillationsbereich zu finden sind, unterscheiden können. Der IQR erkennt den Oszillationsbereich zu 73,3%, die Varianz zu 93,3%. Beim IQR muss der Oszillationsbereich einen größeren Bereich abdecken, damit er erkannt wird. Daher kommt der niedrigere Prozentwert bei diesem Dispersionsmaß. Die Spannweite R erkennt ebenfalls, ob Kanten vorhanden sind, aber kann diese oft nicht richtig zuordnen. Kanten, die nah am Oszillationsbereich liegen, zeigen oft auch schon leichte Oszillationen. Diese beeinflussen die Spannweite, wodurch eine Kante zu 37,5% zum Oszillationsbereich gezählt wird. Handelt es sich wirklich um den Oszillationsbereich, so wird dieser durch die Spannweite auch als Oszillationsbereich erkannt.
Der MAD ist nicht brauchbar, da dieser nur Kanten die sehr mittig durch den Scan gehen detektiert. Das liegt an dem Median in $|x_i - \text{median}(x_i)|$. Dieser gibt bei einer Kante am Rand des Scans entweder einen recht niedrigen oder hohen Wert zurück. Daher muss die Kante sehr mittig durch den Scan gehen, damit dieser Wert ein gutes Mittelmaß zurückgibt und dieses Dispersionsmaß ausschlagen lässt. Aus diesem Grund wird der MAD auch hier nicht mehr weiter betrachtet.
Der AAD schneidet ähnlich wie der IQR ab. Dieser entdeckt 87,5% der Kanten und ordnet diese ebenfalls richtig zu. Der Oszillationsbereich wurde zu 71,2% richtig detektiert. Im Durchlauf des Algorithmus erkennt dieses Dispersionsmaß jedoch teilweise keine der Kanten in den Zeilen/Spalten, in denen sie eigentlich vorhanden sind, was für die weiteren Schritte sehr problematisch ist.
Um Ergebnisse in einer guten Qualität zu erreichen, kann der Algorithmus Varianz und

IQR verwenden. AAD liefert ebenfalls nutzbare Ergebnisse, jedoch mit einer niedrigeren Qualität als Varianz und IQR. Für die folgenden Schritte wurde die Varianz zur Detektion der Kanten und des Oszillationsbereichs verwendet.

4.5 Bewertung des Mittelwerts zur Unterscheidung von leitendem und nichtleitendem Bereich

Wenn weder Oszillationen noch Kanten im Scan gefunden werden, überprüft der Algorithmus, ob sich der Scan im leitenden oder nichtleitenden Bereich befindet. Hierfür wird der Abstand des arithmetischen Mittels des Narrow Scans zum minimalen bzw. maximalen Leitwert verwendet. Mögliche Probleme sind hier sehr starkes Rauschen und ein schlecht geschätztes Minimum bzw. Maximum für die Leitwerte. Praktisch unterscheidet dieser Ansatz korrekt zwischen leitendem und nichtleitendem Bereich in allen Beispielen aus beiden Datensätzen.

4.6 Bewertung des Pattern-Matching-Verfahrens

Wenn eine diagonale Kante gefunden wurde, muss überprüft werden, ob es sich nur um eine Kante handelt oder um Oszillationen. Hierfür werden die Peaks gezählt. Wenn durchschnittlich mehr als 2 Peaks in den Zeilen vorkommen, werden Oszillationen vermutet. In den experimentellen Daten zeigen diagonale Kanten oft Oszillationen, aber der gesuchte Beginn der Oszillationen liegt irgendwo anders. In Abbildung 4.7 ist die rechte Kante des Bildes ein Beispiel dafür. Hier würde der Algorithmus eine Oszillation finden, aber die linke untere Ecke des eigentlichen Coulomb-Oszillationbereichs liegt fast mittig im Bild. Da es jedoch keine Entscheidungsmerkmale für den Algorithmus gibt, um zu überprüfen, ob es sich um eine Kante mit Oszillationen oder um den gesuchten Coulomb-Oszillationsbereich handelt, ist es möglich, dass der Algorithmus, statt dem Startpunkt des Oszillationsbereiches, einen Punkt auf einem anderen Coulomb-Peak findet.
Ein weiteres Problem ist, dass die generierten Templates stark von den Daten abweichen. Dies liegt meistens daran, dass die Merkmale in den Daten von den erwarteten Merkmalen abweichen. Ein Beispiel dazu ist in Abbildung 4.8 gezeigt. Für dieses Beispiel wurde eine Sinuskurve zum Erstellen des Templates verwendet. Hier wird sichtbar, dass das Template deutlich von den Daten abweicht und die Sinuskurve komplett durch den Scan geht, obwohl an der linken Seite gar keine Peaks mehr zu sehen sind. Ein weiteres Problem, welches in der Abbildung 4.8 schwer zu erkennen ist, sind die ansteigenden Höhen der Peaks. In Abbildung 4.9 ist das deutlicher erkennbar. Da die Maxima der Sinuskurve immer die gleiche Höhe haben, wird diese Funktion für dieses Beispiel nie brauchbare Ergebnisse liefern. In Abbildung 4.9 sieht man auch, dass die Daten kaum Ähnlichkeiten zur Sinusfunktion haben, sondern eher einer Gauß-Glockenfunktion äh-

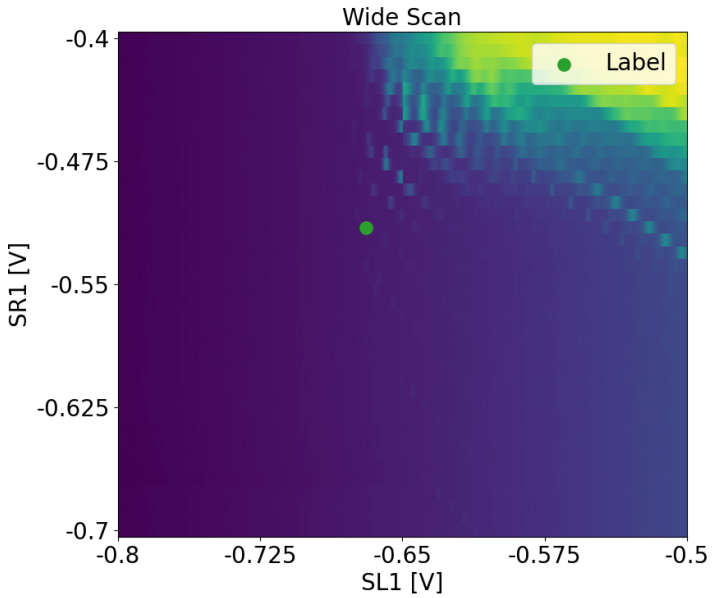


Abbildung 4.7: 45°-Kante mit Oszillationen

neln. Die Gauß-Glockenfunktion wurde in Abbildung 4.10 angewandt. Wenn die Peaks zu dicht aneinander liegen oder zu verrauscht sind, was im Beispiel der Fall ist, dann weicht auch die Gauß-Glockenfunktion stark von den Daten ab. Da die Gauß-Glockenfunktion häufiger als die Sinuskurve brauchbare Templates erzeugt, wird die Gauß-Funktion im Algorithmus angewandt.
Trotz der großen Abweichung zwischen Template und Daten, haben empirische Untersuchungen an 20 Scans gezeigt, dass Pattern-Matching angewendet werden kann, wobei das Template bei 50% der Test-Scans stark abweicht und bei den anderen 50% kaum abweicht.
Ein weiteres mögliches Problem ist, dass das Ergebnis des Pattern-Matching zu einer falschen Richtung für den nächsten Scan führt. Wenn zum Beispiel nur in einem kleinen Bereich des Narrow Scans Coulomb-Oszillationen sichtbar sind, könnte es vorkommen, dass das Pattern-Matching in den anderen Teilstücken eine hohe Korrelation zeigt. Dies liegt aber nicht an übereinstimmenden Oszillationen in den Teilstücken, sondern an einem hohen Anteil des nichtleitenden Bereichs im Template. Dies konnte aber in den untersuchten Beispieldaten nicht beobachtet werden.
Ein weiteres Problem lässt sich in Scans mit einem sehr großen Oszillationsbereich beobachten. Wenn der Oszillationbereich mehr als fünf Oszillationen besitzt und dadurch größer ist als ein Narrow Scan, bricht der Algorithmus oft im Oszillationsbereich ab, nicht an der linken unteren Ecke dieses Bereiches. Das passiert insbesondere, wenn der Algorithmus von einer Kante aus den Oszillationsbereich betritt und das Pattern-Matching mit dem Vergleich der Teilstücke den prozentualen Schwellenwert nicht überschreitet

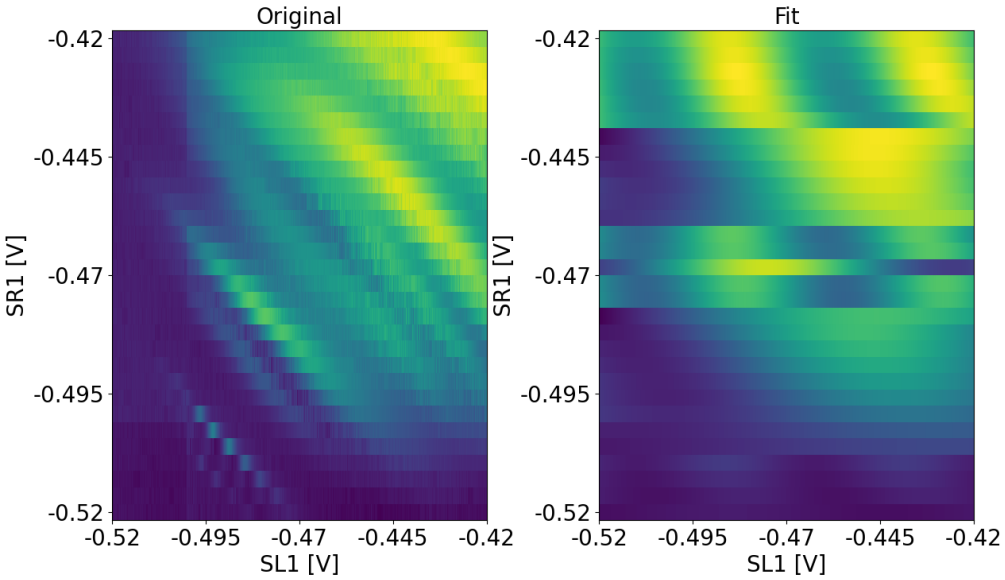


Abbildung 4.8: Original-Daten und generiertes Template, welches stark von den Original-Daten abweicht (Sinus)

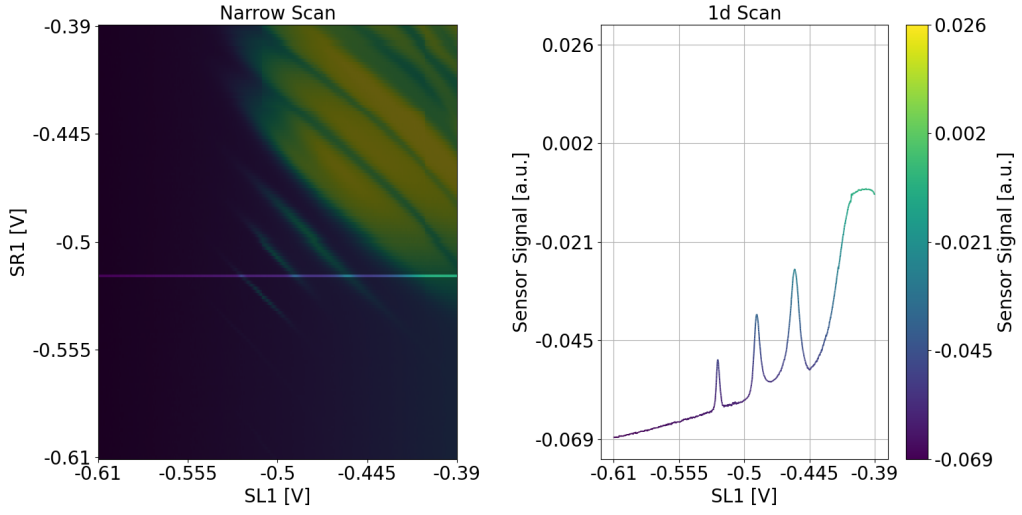


Abbildung 4.9: Höhenunterschiede der Peaks im 1d Scan in der markierten Zeile des Narrow Scans

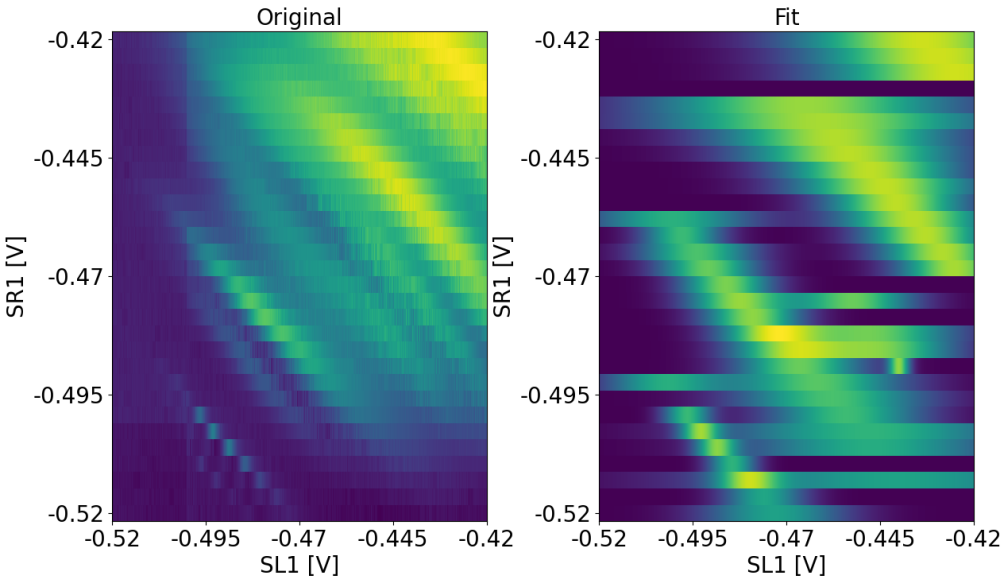


Abbildung 4.10: Original-Daten und generiertes Template, welches stark von den Original-Daten abweicht (Gauß)

und dadurch der Algorithmus terminiert. Ein Beispiel für dieses Problem ist in Abbildung 4.11 dargestellt.

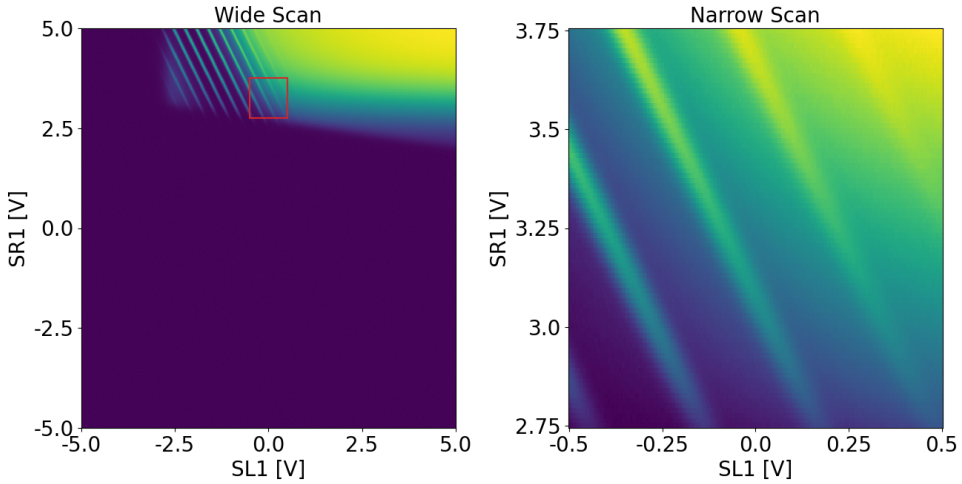


Abbildung 4.11: Gefundene Oszillationen, welche weit entfernt von der linken unteren Ecke des Oszillationsbereichs liegen

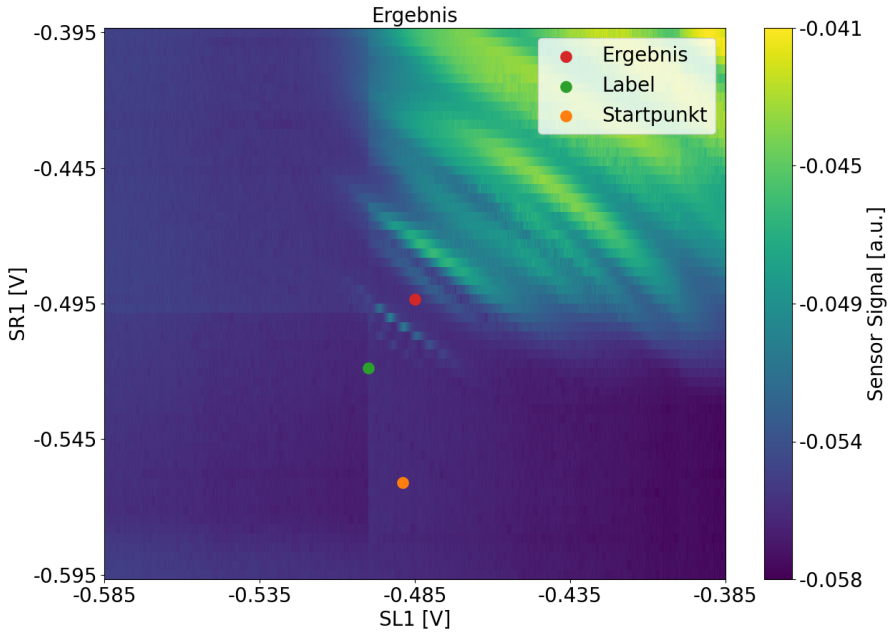


Abbildung 4.12: Ergebnis mit guter Pinch-Off Messung und geringer Abweichung der Kantenwinkel

4.7 Bewertung des Gesamtergebnisses vom Algorithmus

Für den Algorithmus wurde als Standardverfahren für das Normalisieren der `MinMaxScaler` und der `LogisticScaler` mit dem Faktor $p = 5$ gewählt. Als Dispersionsmaß wurde die Varianz mit dem Schwellenwert 0,1 verwendet. Für das Erzeugen des Templates wurde als Standard die Gauß-Funktion ausgewählt. Der Algorithmus bricht in den Datensatz zuverlässig im Oszillationsbereich ab. Wenn die beiden Kanten nahezu horizontal und vertikal liegen, dann liegt auch der Startpunkt, der über die Pinch-Off-Messung bestimmt wird, oft schon im Oszillationsbereich oder er ist innerhalb von ein bis zwei Scans erreichbar. In solchen Fällen ist die Anzahl der Scans auch relativ niedrig, durchschnittlich 1,3 Scans, und durch das Verfahren wird viel Zeit gespart, im Vergleich zur Aufnahme eines Wide Scans. Ein Beispiel für eine solche Messung ist in Abbildung 4.12 dargestellt. Der über die Pinch-Off-Messung gefundene Startpunkt liegt direkt unter dem Oszillationsbereich, mit einer diagonalen Verschiebung nach oben rechts und mit zwei Verschiebungen durch das Pattern-Matching-Verfahren nimmt der Algorithmus vier Narrow Scans auf.
Wenn die Pinch-Off-Messung keinen guten Startpunkt liefert oder der Winkel der Kanten von 0° bzw. 90° abweicht, werden wesentlich mehr Scans benötigt, um den Oszillationsbereich zu entdecken. Die maximale Anzahl von Scans in den Daten waren 32, was den

Algorithmus wesentlich ineffizienter macht, als die Detektion des Oszillationsbereichs in einem Wide Scan. Die Effizienz des Algorithmus ist daher stark abhängig davon, wie gut der Startpunkt gewählt wurde.
Das Label liegt in den Beispielen zu 81,6% in dem letzten Narrow Scan des Algorithmus, aber oft ist der Mittelpunkt des Narrow Scans nicht nah am Label. Grund hierfür ist vor allem die Qualität der Templates, da das Ergebnis des Pattern-Matching von dieser abhängig ist. Ist die Qualität des Templates zu gering, bricht der Algorithmus oft zu früh ab. Ein weiterer Grund für einen zu frühen Abbruch ist auch die Größe des Oszillationsbereichs. Wenn der Algorithmus, wie am Ende von Abschnitt 4.6 erklärt, von einer Kante zum Oszillationsbereich kommt, kann auch hier das Pattern-Matching oft falsche Ergebnisse liefern. Da vor allem die simulierten Daten einen großen Oszillationsbereich besitzen, dafür aber die Templates besser zu den Peaks passen, passiert dies aber selten. Die Auswertung der Bewertungsmetrik wurde mit den 50 simulierten Scans und den gelabelten experimentellen Daten durchgeführt. Durchschnittlich ist das Label von der berechneten linken unteren Ecke 0,056V entfernt. Da eine Narrow Scan in der Arbeit die Größe von 0,2V hat, lässt der Abstand der Bewertungsmetrik schließen, dass der Algorithmus die linke untere Ecke nicht gut detektiert. Da der vom Algorithmus bestimmte Startpunkt aber immer rechts oberhalb vom tatsächlichen Startpunkt liegt, sind immer genug Oszillationen auf dem finalen Scan enthalten, sodass man diese für weitere Kalibrierungsschritte nutzen kann. Die Laufzeit des Algorithmus beträgt im Durchschnitt ca. 0,8s. Ein Großteil der Laufzeit entfällt dabei auf den rekursiven Aufruf des Algorithmus und das Erzeugen des Templates. Durchschnittlich 0,4s entfallen dabei auf das Pattern-Matching-Verfahren.
Zwei allgemeine Probleme sind, wie im vorherigen Abschnitt bereits erklärt, 45°-Kanten, da dem Algorithmus bei diesen nicht bekannt ist, in welcher Richtung der Oszillationsbereich liegt. In dem Fall bricht der Algorithmus einfach ab. Für weitere Kalibrierungsschritt ist außerdem die Qualität des Coulomb-Oszillationsbereichs wichtig. Damit direkt mit dem finalen Narrow Scans des Suchalgorithmus weitergearbeitet werden kann, wäre eine zusätzliche Bewertung der Coulomb-Peaks im gefundenen Oszillationsbereich hilfreich.

5 Fazit und Ausblick

Die Bachelorarbeit hat sich mit dem Detektieren der linken unteren Ecke des Coulomb-Oszillationsbereichs ausschließlich über Narrow Scans befasst. Die Detektion dieses Bereichs ist notwendig für die Kalibrierung von Sensorpunkten, die zum Auslesen von Qubits verwendet werden.
Für die Arbeit wurden experimentelle und simulierte Wide Scans von der Quantum Technology Group des JARA-Instituts für QI [3] zur Verfügung gestellt. Diese wurden zunächst selektiert und anschließend gelabelt. Bei den experimentellen Daten wurde die linke untere Ecke des Coulomb-Oszillationsbereichs gelabelt und aus beiden Datensätzen wurden Narrow Scans als Teilscan eines Wide Scans ausgeschnitten und ebenfalls gelabelt. Es wurden Merkmale, anhand derer sich der Algorithmus orientieren kann, gesucht und die Aufgabe wurde in kleinere Schritte unterteilt. Mithilfe dieser Merkmale wurde ein Algorithmus entworfen, der die einzelnen Schritte der Aufgabe durchführt. Die Schritte des Algorithmus wurden implementiert, einzeln getestet und Vor- und Nachteile dieser Schritte aufgezeigt. Der Algorithmus kann noch nicht für reale Experimente verwendet werden, da die entsprechende Schnittstelle noch fehlt. Jedoch wird der Oszillationsbereich zuverlässig genug gefunden, dass der finale Narrow Scan für weitere Kalibrierungsschritte des Sensorpunktes genutzt werden könnte. Dazu ist insbesondere wichtig, dass Coulomb-Peaks im finalen Scan sichtbar sind, was für alle Testdaten erreicht werden konnte.
Insgesamt lässt sich sagen, dass der Algorithmus den Oszillationsbereich zuverlässig findet. In einem Großteil der untersuchten Fälle liegt auch die gesuchte linke untere Ecke dieses Bereiches auf dem finalen Narrow Scan. Das Detektieren von Kanten und Unterscheiden von nichtleitendem und leitendem Bereich funktioniert gut genug, damit der Algorithmus sich zuverlässig in die Richtung des Startpunktes des Oszillationsbereichs bewegt. Um die Laufzeit des Algorithmus zu reduzieren, könnte die Schrittweite für die Verschiebungen der Narrow Scans vergrößert werden. Das Generieren von Templates zur Detektion der linken unteren Ecke des Oszillationsbereichs ist noch relativ fehleranfällig, da die Templates oft stark von den Daten abweichen, was teilweise zu einem vorzeitigen Abbruch des Algorithmus führt. Dadurch wird die linke untere Ecke des Oszillationsbereich weniger zuverlässig detektiert als der Oszillationsbereich selbst.

Für den aktuellen Stand des Algorithmus gibt es Verbesserungs- und Erweiterungsmöglichkeiten, auf die im Folgenden genauer eingegangen wird.
Eine Verbesserungsmöglichkeit ist die Optimierung der Parameter des Algorithmus. Dafür würde sich zum Beispiel die Schrittgröße für Verschiebungen der Narrow Scans anbieten, welche in Abschnitt 3.8 beschrieben wurde. Dadurch könnte auch die Gesamtlaufzeit des Algorithmus inklusive der Datenaufnahme verringert werden.
Eine weitere Verbesserungsmöglichkeit, wäre den genauen Winkel der horizontalen und vertikalen Kante zu bestimmen, damit eine genauere Richtung beim Verschieben der Narrow Scans angegeben werden kann. Hierfür könnte man auch Kantensuche-Algorithmen verwenden [26].

Für Kanten, die diagonal durch den Scan gehen und daher teilweise als Anfangspunkt des Coulomb-Oszillationsbereich erkannt werden, könnte man ein Verfahren entwickeln, dass entscheiden kann, in welcher Richtung sich der tatsächliche Anfangsbereich der Coulomb-Oszillationen befindet, indem es sich beispielsweise die vorherigen Scans nochmal anschaut. Eine andere Möglichkeit wäre, wenn der Algorithmus über einer Kante zum Oszillationsbereich gelangt, solange weiter zu verschieben, bis der Algorithmus in einer gewissen Anzahl von Zeilen und Spalten keine Dispersion mehr findet. Somit kann der Algorithmus verhindern, dass Pattern-Matching mitten im Oszillationsbereich durchgeführt wird. Stattdessen sollte es nur, wenn die linke untere Ecke des Oszillationsbereichs mit einer hohen Wahrscheinlichkeit schon im Scan zu finden oder relativ nahe ist, durchgeführt werden. Dieses Verfahren würde auch zu einer geringeren Laufzeit des Algorithmus führen, da das Pattern-Matching-Verfahren einer der aufwendigsten Schritte des Algorithmus ist.
Da die generierten Templates für das Pattern-Matching teilweise stark von den tatsächlichen Daten abweichen, wäre eine weitere Verbesserungsmöglichkeit bessere Verfahren für das Generieren der Templates zu entwickeln, zum Beispiel indem man andere Modellfunktionen verwendet. Eine Alternative wäre es, andere Verfahren für die Bestätigung des Oszillationsbereich auszuprobieren, da Pattern-Matching nicht optimal funktioniert.
Ein weiterer möglicher Ansatz um die linke untere Ecke besser und etwas schneller zu detektieren, wäre eine weitere Pinch-Off-Messung im Bereich des Narrow Scans zu machen, sobald dieser Oszillationen enthält. Dadurch kann die gesuchte linke untere Ecke schneller gefunden werden, als durch kleine Verschiebungen, des Narrow Scans, die aktuell über das Pattern-Matching-Verfahren bestimmt werden.
Eine Erweiterungsmöglichkeit für den Algorithmus wäre, nicht einfach den Anfangspunkt des Oszillationsbereichs zu suchen, sondern auch zu beurteilen, wie gut die vorhandenen Oszillationen für weitere Kalibrierungsschritte der Quantenpunkte geeignet sind.
Da die Laufzeit des Algorithmus für die Kalibrierung der Quantenpunkte ein wichtiger Faktor ist, sollte auch die Laufzeitoptimierung des Algorithmus für Verbesserungen in Betracht gezogen werden. Dazu könnten zusätzlich zu den schon vorgeschlagenen Möglichkeiten zum Beispiel die Normalisierungsschritte oder das Generieren der Templates parallelisiert werden.
Um einen besseren Eindruck zu bekommen, welche Probleme bei der Echtzeit-Kalibrierung eines Sensorpunktes auftreten können, ist es außerdem wichtig, die Schnittstelle zum Experiment zu implementieren.

Literaturverzeichnis

[1] Forschungszentrum Jülich GmbH. *Zentralinstitut für Engineering, Elektronik und Analytik (ZEA)*. URL: https://www.fz-juelich.de/de/zea (besucht am 10. Oktober 2023).

[2] Forschungszentrum Jülich GmbH. *Systeme der Elektronik (ZEA-2)*. URL: https://www.fz-juelich.de/de/zea/zea-2 (besucht am 10. Oktober 2023).

[3] *Quantum Technology Group*. URL: https://www.quantuminfo.physik.rwth-aachen.de/cms/Quantuminfo/Forschung/~xwpl/Quantum-Technology-Group/ (besucht am 25. August 2023).

[4] RWTH Aachen University Forschungszentrum Jülich GmbH. *JARA-Institut für Quanteninformation*. URL: https://www.jara.org/de/jara (besucht am 10. Oktober 2023).

[5] Ian Parberry. „An efficient algorithm for the Knight's tour problem". In: *Discrete Applied Mathematics* 73.3 (1997), S. 251–260. URL: https://mimuw.edu.pl/~rytter/TEACHING/ALCOMB/efficient%20algorithm%20recursion%20knight.pdf (besucht am 9. Oktober 2023).

[6] Albert Messiah. *Quantum mechanics*. Courier Corporation, 2014.

[7] *Erste Schritte auf dem Weg zur Spin-Elektronik*. 2002. URL: https://www.weltderphysik.de/gebiet/materie/spintronik/erste-schritte/ (besucht am 10. Oktober 2023).

[8] Ronald Hanson u. a. „Spins in few-electron quantum dots". In: *Reviews of modern physics* 79.4 (2007), S. 1217. DOI: https://doi.org/10.1103/RevModPhys.79.1217. URL: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.79.1217 (besucht am 10. Oktober 2023).

[9] Robert S Smith, Michael J Curtis und William J Zeng. „A practical quantum instruction set architecture". In: *arXiv preprint arXiv:1608.03355* (2016). URL: https://arxiv.org/abs/1608.03355 (besucht am 18. Oktober 2023).

[10] S. Fleitmann. „Characterization of Distortions in Charge Stability Diagrams and Their Simulation in Modeled Data". Masterarbeit. Fachhochschule Aachen, 2022.

[11] Paul Adrien Maurice Dirac. „On the theory of quantum mechanics". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 112.762 (1926), S. 661–677. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1926.0133 (besucht am 6. November 2023).

[12] scikti learn. *MinMaxScaler*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html#sklearn.preprocessing.MinMaxScaler (besucht am 8. November 2023).

[13] DataScientist. *Sigmoid function*. URL: https://datascientest.com/de/sigmoid-funktion-was-ist-das-wozu-dient-sie (besucht am 8. November 2023).

[14] Wolfram. *Sign function*. URL: https://mathworld.wolfram.com/Sign.html (besucht am 8. November 2023).

[15] Microsoft. *Normalize Data (Normalisieren von Daten) Komponente*. URL: https://learn.microsoft.com/de-de/azure/machine-learning/component-reference/normalize-data?view=azureml-api-2 (besucht am 10. Oktober 2023).

[16] Wikipedia. *Statistical dispersion*. URL: https://en.wikipedia.org/wiki/Statistical_dispersion (besucht am 10. Oktober 2023).

[17] Statistics How To. *Median Absolute Deviation*. URL: https://www.statisticshowto.com/median-absolute-deviation/ (besucht am 8. November 2023).

[18] Statistics How To. *Average Deviation*. URL: https://www.statisticshowto.com/average-deviation/ (besucht am 8. November 2023).

[19] Tony Hak und Jan Dul. „Pattern matching“. In: (2009). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1433934 (besucht am 18. Oktober 2023).

[20] OpenCV. *OpenCV Homepage*. URL: https://opencv.org/ (besucht am 18. Oktober 2023).

[21] OpenCV. *cv2.matchTemplate*. URL: https://docs.opencv.org/4.x/df/dfb/group__imgproc__object.html#ga586ebfb0a7fb604b35a23d85391329be (besucht am 10. Oktober 2023).

[22] SciPy. *scipy.signal.find_peaks*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html (besucht am 10. Oktober 2023).

[23] SciPy. *scipy.optimize.curve_fit*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html (besucht am 10. Oktober 2023).

[24] Kenneth Levenberg. „A method for the solution of certain non-linear problems in least squares“. In: *Quarterly of applied mathematics* 2.2 (1944), S. 164–168. URL: https://www.ams.org/journals/qam/1944-02-02/S0033-569X-1944-10666-0/S0033-569X-1944-10666-0.pdf (besucht am 6. November 2023).

[25] Duraisamy Sundararajan. *The discrete Fourier transform: theory, algorithms and applications*. World Scientific, 2001.

[26] Djemel Ziou, Salvatore Tabbone u. a. „Edge detection techniques-an overview“. In: *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii* 8 (1998), S. 537–559. URL: https://www.researchgate.net/profile/Djemel-Ziou/publication/312890367_'Edge_detection_techniques_An_overview'/links/59d3c61c0f7e9b4fd7ffbfbb/Edge-detection-techniques-An-overview.pdf (besucht am 8. November 2023).

Jül-4445 • Januar 2024
ISSN 0944-2952

Mitglied der Helmholtz-Gemeinschaft