

Optimierung verschränkender Quantengatter für Experimente mit Ionenfallen

Volckmar Nebendahl

Diplomarbeit

Fachbereich Physik
Universität Hamburg

durchgeführt am
Institut für Experimentalphysik
Universität Innsbruck

in Zusammenarbeit mit dem
Institut für Quantenoptik und Quanteninformation
Österreichische Akademie der Wissenschaften

Juni 2008

Diese Diplomarbeit entstand in der Arbeitsgruppe von Prof. Dr. Rainer Blatt an der Universität Innsbruck und wurde von Prof. Dr. Heinrich Heyszenau an der Universität Hamburg betreut.

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen und Hilfsmittel angefertigt zu haben.

Ich bin mit einer späteren Ausleihe meiner Diplomarbeit einverstanden.

Inhaltsverzeichnis

Einleitung	vii
1 Grundlagen	1
1.1 Einführung in den Quantencomputer	1
1.1.1 Die Logik des Quantencomputers	2
1.1.2 Darstellung der Operatoren	4
1.1.3 Verwirklichung des Quantencomputers	5
1.2 Ionenfallen	6
1.3 Ion-Laser-Wechselwirkung	7
1.3.1 Mehrere Ionen im bichromatischen Laserfeld	9
1.3.2 Interpretation der Ion-Laser-Wechselwirkung	11
1.4 Mølmer-Sørensen-Gatter	12
1.5 $\sigma_z \otimes \sigma_z$ -Gatter	14
1.5.1 Interpretation des $\sigma_z \otimes \sigma_z$ -Gatters	16
1.6 Ideale und reale Gatter	19
2 Optimierungsalgorithmen	21
2.1 Optimal control theory (OCT)	21
2.2 Numerische Suche nach Extrema	23
2.3 Einzel-Variablen-Optimierung (EVO)	25
2.4 Nebenbedingungen	28
2.5 Lokale und globale Extrema	32
2.5.1 Generierung zufälliger Wahrscheinlichkeiten	34
3 Zeitentwicklung	37
3.1 Die Struktur des Zeitentwicklungsoperators	37
3.2 Recycling	38
3.3 Wechselwirkungsbilder	40
3.4 Numerische Lösung der Dysonentwicklung	43
3.4.1 Numerische Integration	47
3.5 Verkürzung der Summation	49
3.5.1 Weitere Ansätze zur Verbesserung	51

4	Gatteroptimierung	55
4.1	Rahmenbedingungen	55
4.2	Optimierungsansatz	56
4.3	Fehlertoleranz	59
4.3.1	Phasenabhängigkeit der Ergebnisse	60
4.3.2	Fluktuationen in der Rabifrequenz	64
4.4	Spinecho	66
4.4.1	Spinecho + Offset	69
4.5	Ergebnisse	69
4.5.1	$\sigma_z \otimes \sigma_z$ -Gatter	71
4.5.2	Mølmer-Sørensen-Gatter	78
4.6	Ausblick	80
5	Impulssequenzen	83
5.1	Grundbausteine	83
5.2	Optimierungsverfahren	85
5.3	Resultate	88
5.3.1	Globale Strukturen	89
5.3.2	CNOT in einer anderen Basis	91
5.3.3	Variationen zum NOT	94
5.4	Analyse der Resultate	97
6	Quantenfehlerkorrektur	103
6.1	Grundlagen und ein erstes Modell	103
6.2	Korrektur ohne Messung	105
6.3	Mehr durch weniger?	107
6.4	Phasenfehler	110
6.5	Algorithmische Umsetzung	112
6.6	Optimieren in der Praxis	115
7	Zusammenfassung und Ausblick	119

Einleitung

1982 veröffentlichte R. P. Feynman seinen Artikel *Simulating Physics with Computers* [1], was häufig als Geburtsstunde des Quantencomputers angesehen wird. In diesem Artikel stellte Feynman fest, dass es extrem schwierig ist, quantenmechanische Systeme auf klassischen Computern zu berechnen. Als Ausweg überlegte er, ob nicht eventuell jedes beliebige quantenmechanische System von einer Maschine simuliert werden könnte, die selbst auf quantenmechanischen Elementen basiert. Diese Idee erwies sich als fruchtbar und wurde 1985 von David Deutsch [2] zu einer mathematischen Theorie über den universellen Quantencomputer ausgebaut. Es folgten weitere theoretische Beiträge, die 1994 von Peter Shors Faktorisierungsalgorithmus gekrönt wurden [3]. Dieser Algorithmus erlangte Berühmtheit, weil er es erlauben würde, das RSA-Kryptosystem zu knacken, auf der fast die gesamte Verschlüsselung im Internet beruht.

Trotz großer theoretischer Erfolge war es zu diesem Zeitpunkt (1994) noch weitgehend unklar, wie all dies in die Praxis umgesetzt werden kann. Einen großen Schritt in diese Richtung leisteten J. Ignacio Cirac und Peter Zoller im Jahre 1995, als sie ein Schema zur Realisierung eines CNOT-Gatters¹ mit gefangenen Ionen vorschlugen [4]. Ionenfallen erlauben es, Ionen im Hochvakuum gefangen zu halten und deren Quantenzustand mit Hilfe von Lasern zu manipulieren. Diese Technik wurde 1995 bereits relativ gut beherrscht, so dass eine abgewandelte Umsetzung des CNOTs auf einem Ion noch im gleichen Jahr erfolgen konnte [5]. Das erste CNOT mit zwei Ionen wurde aber erst im Jahre 2003 experimentell verwirklicht [6].

Inzwischen sind eine Vielzahl an Verfahren zur Realisierung von Quantengattern auf unterschiedlichen quantenmechanischen Systemen bekannt. Bis zum heutigen Tag kann aber kein Verfahren mit hinreichenden Zuverlässigkeiten aufwarten, die für eine Erweiterung auf beliebig komplexe Quantenschaltkreise nötig wären [7].

Zu den aussichtsreichsten Kandidaten für zukünftige Fortschritte auf dem Gebiet des Quantencomputers zählen weiterhin die Ionenfallen. In den Jahren nach 1995 wurden noch weitere Schemata zur Realisierung von Quantengattern mit gefangenen Ionen gefunden. Als bedeutsam erwiesen sich

¹Das CNOT ist ein ist ein elementarer Schaltkreis für Quantencomputer.

hierbei das so genannte Mølmer-Sørensen- und $\sigma_z \otimes \sigma_z$ -Gatter [8, 9]. Mit einem Mølmer-Sørensen-Gatter gelang in unserer Gruppe z.B. vor Kurzem der Nachweis der höchsten je erreichten Güte für ein verschränkendes Quantengatter mit 99,3% [10]. Auch diese Diplomarbeit wird sich mit solchen Gattertypen beschäftigen. Einer der interessantesten Aspekte dieser Gatter ist ihre hohe Resistenz gegen äußere Störungen. Diese beruht auf der weitgehenden Unabhängigkeit vom (thermischen) Bewegungszustand² der Ionen in der Falle. Dies ist zu vergleichen mit dem Cirac-Zoller-CNOT, das nur dann korrekt funktioniert, wenn die Ionen sich in einem ganz bestimmten Bewegungszustand (meist dem Grundzustand) befinden.

Wie in den Abschnitten 1.3.2 und 1.4 gezeigt wird, fußen die Grundideen des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters allerdings auf einigen Näherungen. Als Folge dessen wird es ihnen daher selbst bei idealen praktischen Bedingungen nie möglich sein, Güten von 100% zu erzielen. Andererseits lässt die Durchführung dieser Gatter die Variation einiger Parameter der beteiligten Laserstrahlen zu, was die Frage aufwirft, ob sich diese Freiheit nutzen lässt, um die Güten weiter zu verbessern. Derartige Probleme gehören der *optimal control theory* an. In den Kapiteln 3 und 4 wird dem weiter nachgegangen.

Eine weitere interessante Eigenschaft des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters besteht in ihrer Anwendung auf n Ionen. Beim Cirac-Zoller-CNOT wird der Laser immer nur auf einzelne Ionen gerichtet – auch wenn mehrere Ionen verschränkt werden sollen. Beim Mølmer-Sørensen- und $\sigma_z \otimes \sigma_z$ -Gatter wird hingegen ein globaler, bichromatischer Laserstrahl eingesetzt, der alle n Ionen auf einmal anspricht und verschränkt. Dies kann genutzt werden, um mit Hilfe des Mølmer-Sørensen-Gatters beliebige Quantengatter auf n Ionen zu konstruieren. Für $n = 2$ ist es noch relativ leicht zu erkennen, wie dies zu erfolgen hat. Es wurde aber bisher noch kein allgemeines Verfahren zur Erzeugung beliebiger Quantengatter für den Fall vorgestellt, dass das Mølmer-Sørensen-Gatter durch einen bichromatischen Laserstrahl verwirklicht wird, der mehr als zwei Ionen auf einmal anspricht. Diese Lücke wird jetzt in Kapitel 5 dieser Diplomarbeit für eine beliebige Anzahl an Ionen geschlossen. Dabei beschränkt sich die vorliegende Arbeit nicht auf das Aufzeigen theoretischer Möglichkeiten, sondern gibt auch verschiedene Sequenzen aus Laserimpulsen (kurz: Impulssequenzen) für den praktischen Einsatz an.

Der Ansatz aus Kapitel 5 wird in Kapitel 6 fortgeführt und es werden weitere Impulssequenzen vorgestellt, die für eine einfache Quantenfehlerkorrektur verwendet werden können.

Sowohl zum Optimieren der Güten als auch zum Auffinden der Impuls-

²Die gefangenen Ionen sind durch die Coulomb-Wechselwirkung aneinander gebunden. Ihre gemeinsame Bewegung kann in guter Näherung als harmonische Schwingungen beschrieben werden.

quenzen wurde der Computer (ein einfacher PC) eingesetzt. Die Weiterentwicklung bzw. Anpassung entsprechender Algorithmen der *optimal control theory* auf die jeweiligen Bedürfnisse ist daher ein immer wiederkehrendes Thema.

Die Arbeit ist wie folgt gegliedert:

Kapitel 1 stellt den Versuch dar, diese Diplomarbeit weitgehend selbstkonsistent zu gestalten und liefert das notwendige physikalische Grundwissen.

Kapitel 2 liefert die mathematischen Grundlagen zur Erstellung von Optimierungsalgorithmen. Neben Standardmaterial werden auch neue Verfahren vorgestellt, die Rahmen dieser Diplomarbeit entwickelt wurden.

Kapitel 3 beschäftigt sich mit der effizienten Lösung der nicht trivialen Zeitentwicklung der Mølmer-Sørensen- und der $\sigma_z \otimes \sigma_z$ -Gatter.

Kapitel 4 baut auf dem Wissen der beiden vorherigen Kapitel auf und geht das Problem der Optimierung der Güten des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters an. Die daraus resultierenden Ergebnisse werden ebenfalls in diesem Kapitel diskutiert.

Kapitel 5 nimmt sich des Problems der Auffindung von Impulssequenzen mit Mølmer-Sørensen-Gattern auf n Ionen an. Ziel ist es, beliebige Quantengatter erzeugen zu können. Neben der Erweiterungen des Optimierungsalgorithmus beinhaltet dieses Kapitel auch die Präsentation der Ergebnisse sowie eine Analyse der Impulssequenzen.

Kapitel 6 ist eine Weiterführung der Ideen des vorherigen Kapitels. Hier werden spezielle Impulssequenzen zur Quantenfehlerkorrektur besprochen, und eine leichte Abwandlung der Schaltkreise zur Effizienzsteigerung wird vorgeschlagen.

Kapitel 7 schließt die Diplomarbeit mit einer Zusammenfassung und einem Ausblick ab.

Kapitel 1

Grundlagen

In diesem Kapitel werden die physikalischen Grundlagen dieser Diplomarbeit bereitgestellt. Abschnitt 1.1 liefert eine kurz gehaltene Einführung in die Welt des Quantencomputers. Wer sich mehr Details zu diesem Thema wünscht, sei an [11] verwiesen. Abschnitt 1.2 beschäftigt sich mit Ionenfallen. Trotz der großen Bedeutung der Ionenfallen im Experiment ist ein tieferes Verständnis für diese Diplomarbeit nicht erforderlich, so dass dieser Abschnitt kurz gehalten ist. Die Ion-Laser-Wechselwirkung – mit der sich Abschnitt 1.3 befasst – ist hingegen recht ausführlich beschrieben, um ein gründliches Verständnis des zugrunde liegenden Modells und den damit implizierten Näherungen zu gewährleisten. Abschnitt 1.4 und 1.5 erklären zwei Spezialfälle der Ion-Laser-Wechselwirkung: Das Mølmer-Sørensen- und $\sigma_z \otimes \sigma_z$ -Gatter. Beide spielen eine zentrale Rolle in dieser Diplomarbeit. Abgeschlossen wird dieses Kapitel durch Abschnitt 1.6, der die Überleitung zu den folgenden Kapiteln darstellt.

1.1 Einführung in den Quantencomputer

Jeder reale Computer ist zugleich ein physikalisches Objekt und damit den Gesetzen der Physik unterworfen. Insbesondere nimmt die Bedeutung der Quantenmechanik bei fortschreitender Miniaturisierung beständig zu. Für die herkömmliche Computerarchitektur bedeutet dies in erster Linie ein Problem. Es bieten sich aber auch neue Chancen. Durch explizites Nutzen der Gesetze der Quantenmechanik lassen sich zumindest gewisse Berechnungen sehr viel effizienter durchführen, als es bisherige Computer gestatten. Hierfür ist es allerdings nicht ausreichend, nur die Architektur des Computers zu verändern. Vielmehr muss die Boolesche Logik, die dem herkömmlichen Computer zugrunde liegt, durch eine der Quantenmechanik angepassten Logik ersetzt werden. Diese Quantenlogik¹ ist wie die Boolesche Logik ein immate-

¹Der Begriff *Quantenlogik* wird hier recht unbedarft genutzt und meint die Rechenregeln eines Quantencomputers. Dieser Begriff bezeichnet aber auch eine sehr viel weitrei-

rielles, mathematisches Konstrukt. Als solches können wir bereits heutzutage Berechnungen auf Basis dieser Quantenlogik anstellen und auf ihr basierende Algorithmen und Computerprogramme entwerfen. So präsentierte z.B. Peter Shor 1994 seinen Quantenalgorithmus zur Primfaktorzerlegung [3], der auf einem Quantencomputer exponentiell schneller arbeiten würde als der beste herkömmliche Algorithmus. Natürlich kann man die Faktorisierung auch auf einem herkömmlichen Computer mittels Shors Algorithmus betreiben – nur der Effizienzvorteil bliebe aus.

Während also bereits erste Software für den Quantencomputer zur Verfügung steht, mangelt es z.Z. noch immer an der Hardware. Letztere ist Objekt intensiver Forschung. Verschiedene Quantensysteme bieten sich dabei als potentielle Plattform zur Implementierung von Quantenalgorithmen an. Bevor wir dazu kommen, müssen wir uns aber etwas eingehender damit beschäftigen, wie die Logik des Quantencomputers aussieht.

1.1.1 Die Logik des Quantencomputers

Die klassische Boolesche Computerlogik kann in zwei Komponenten zerlegt werden:

- BITS mit ihren zwei Zuständen 0 und 1
- OPERATOREN, die auf die Bits einwirken (z.B. AND, OR, NOT)

Diese beiden Komponenten können in abgewandelter Form auch beim Quantencomputer identifiziert werden. Wir beginnen unsere Betrachtungen mit der Erweiterung des klassischen Bits.

Qubits

Der Begriff Qubit steht für Quanten-Bit. Ein Qubit kann prinzipiell durch jedes quantenmechanische (Teil-)System realisiert werden, das durch zwei Basiszustände beschreibbar ist. Das Paradebeispiel für ein solches Zwei-Niveau-System sind Teilchen mit Spin $\frac{1}{2}$. Die beiden Basiszustände bezeichnen wir mit $|0\rangle$ und $|1\rangle$. Damit lässt sich der Zustand des Qubits beschreiben durch

$$\alpha |0\rangle + \beta |1\rangle \quad \text{mit} \quad \alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1. \quad (1.1)$$

Abgesehen von der Komplexwertigkeit der α und β mag dies wie eine Fuzzy-Logik erscheinen, die auch mit der klassischen Physik zu verwirklichen ist. Der Unterschied wird aber deutlich, wenn wir zu einem Quantenregister übergehen, das aus mehreren Qubits besteht. Der Hilbertraum \mathcal{H} eines n -Qubit-Registers ist gegeben durch

$$\mathcal{H} = \bigotimes_{i=1}^n \mathcal{H}_i^{\text{Qubit}} \quad \text{mit} \quad \dim(\mathcal{H}) = 2^n. \quad (1.2)$$

chendere Theorie, die in diesem Zusammenhang nicht gemeint ist.

Im Fall $n = 8$ gibt es 256 Basisvektoren, die als $|0\rangle, |1\rangle \dots |255\rangle$ bezeichnet seien. Der entscheidende Unterschied zu klassischen Computern besteht nun darin, dass jede Linearkombination dieser Basisvektoren

$$\sum_{i=0}^{255} \alpha_i |i\rangle \quad \text{mit} \quad \alpha_i \in \mathbb{C} \quad \sum_{i=0}^{255} |\alpha_i|^2 = 1 \quad (1.3)$$

einen erlaubten Zustand für das Quantenregister darstellt. Selbst wenn wir nur zwei Werte für die α_i zuließen, gäbe es 2^{256} verschiedene Zustände. Bei der gleichen Einschränkung der α_i erfordert die Simulation eines Quantenregisters aus 300 Qubits die Darstellung von $2^{(2^{300})}$ Zuständen. Dafür bedarf es $2^{300} \approx 2 \cdot 10^{90}$ klassischer Bits. Dies übersteigt bereits die geschätzte Anzahl der Teilchen in dem uns bekannten Universum!

Die Geschichte hat aber einen Haken: Die gewaltige Datenmenge in den Qubits kann nicht ausgelesen werden, da die Wellenfunktion kollabiert, sobald die Qubits gemessen werden. Dabei geht ein Großteil der Information verloren. Die Menge der Information, die letztlich zugänglich ist, entspricht der klassischer Bits. Der Vorteil des Quantencomputers muss sich demnach vor dem Ausleseprozess, also während der Berechnung entfalten. Damit leiten wir über zu den Operatoren, die auf die Qubits wirken.

Operatoren

Während der Ausführung eines Computerprogrammes verändert sich der Zustand der Bits eines Computers. Das Gleiche gilt für den Quantencomputer. Die Zustandsveränderung der Qubits gehorcht der Quantenmechanik und wird durch zwei Arten von Operatoren beschrieben:

- unitäre Zeitentwicklungsoperatoren
- projektive Messoperatoren

Letztere werden in der Regel erst am Ende der Berechnungen benötigt, und wir wollen sie daher vernachlässigen². Sämtliche Operatoren des Quantencomputers gehören somit der unitären Gruppe $U(2^n)$ an, wobei n die Anzahl der Qubits darstellt. Aber gilt auch die Umkehrung dieser Aussage? Kann jeder Operator der Gruppe $U(2^n)$ auf einem hypothetischen Quantencomputer verwirklicht werden? Die Antwort ist ja. Es reicht ein kleiner Satz an elementaren Operatoren, um jedes Element der $U(2^n)$ zu erzeugen. Diese elementaren Operatoren werden in Anlehnung an die Digitaltechnik auch Gatter genannt. Ein Satz an Gattern, mit dem jeder Operator der $U(2^n)$ erzeugt werden kann, wird als universell bezeichnet. Ein solcher universeller

²Beim Messungs-basierten Quantencomputer (auch Einweg-Quantencomputer genannt) [12] spielen die Messoperatoren die entscheidende Rolle. Der gewollte Effekt der Messungen ist in diesem Fall aber auch nur eine unitäre Entwicklung der verbleibenden Qubits.

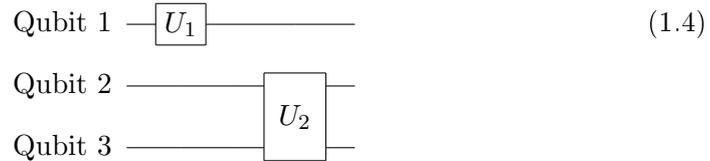
Satz an Gattern ist z.B. gegeben durch das in (1.5) beschriebene CNOT in Verbindung mit allen Ein-Qubit-Gattern [11].

Klassische Computerprogramme beinhalten neben den Elementen der reinen Booleschen Logik auch noch weitere Elemente. Das vielleicht wichtigste ist die bedingte Verzweigung (if-Befehl). Dies ist allein durch unitäre Transformationen nicht auf den Quantencomputer übertragbar und setzt zumindest eine teilweise Messung der Qubits voraus.

1.1.2 Darstellung der Operatoren

Quantenschaltkreis

Eine der gebräuchlichsten Darstellungen eines Quantenalgorithmus bzw. Quantenoperators ist der Quantenschaltkreis, für den es unterschiedliche Konventionen gibt. In dieser Diplomarbeit werden Qubits durch horizontale Linien (Drähte) dargestellt. Operatoren, die auf die Qubits einwirken, liegen über diesen Linien. Im folgenden Beispiel wirkt der Operator U_1 nur auf das erste Qubit, während der Operator U_2 auf Qubit 2 und 3 wirkt



Die zeitliche Reihenfolge geht von links nach rechts; U_1 ist also vor U_2 aktiv. Eine besondere Art an elementaren Operatoren stellen die kontrollierten Gatter dar, deren bekanntester Vertreter das CNOT (Controlled NOT) sein dürfte



Hier symbolisiert \oplus die NOT-Operation (d.h. $|0\rangle$ und $|1\rangle$ werden beim Zielqubit vertauscht). Dieses NOT wird aber nur wirksam, wenn das Kontrollqubit im Zustand $|1\rangle$ ist. Das CNOT wirkt wie ein (reversibles) XOR, z.B.

$$(\alpha|0\rangle + \beta|1\rangle)_{\text{Kontrollqubit}} \otimes |0\rangle_{\text{Zielqubit}} \xrightarrow{\text{CNOT}} \alpha|0,0\rangle + \beta|1,1\rangle. \quad (1.6)$$

Es sind auch mehrere Kontrollqubits möglich, wie beim Toffoli-Gatter



Hier wird die Operation auf dem Zielqubit (das NOT) nur ausgeführt, wenn beide Kontrollqubits im Zustand $|1\rangle$ sind.

Weitere graphische Symbole jenseits der unitären Operatoren sind

$$\begin{aligned} \text{---} \boxed{\text{---}} \text{---} &= \text{Messung} \\ \text{===} &= \text{klassische Leitung.} \end{aligned}$$

Matrixdarstellung

Natürlich können die Operatoren auch als $2^n \times 2^n$ -Matrix dargestellt werden, was aber für $n > 3$ kaum noch zu überblicken ist. Die Quantenzustände werden nun durch einen Spaltenvektor repräsentiert. Hierbei entspricht

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle. \quad (1.8)$$

Indem wir den Wert i in $|i\rangle$ als Binärzahl darstellen, erhalten wir eine Zerlegung von $|i\rangle$ in die Qubit-Komponenten. Bei drei Qubits erhalten wir z.B.

$$|6\rangle = |(110)_{\text{binär}}\rangle = |1, 1, 0\rangle = |1\rangle_1 \otimes |1\rangle_2 \otimes |0\rangle_3. \quad (1.9)$$

Als Beispiel für eine Matrixdarstellung eines Drei-Qubit-Operators geben wir das Toffoli-Gatter (1.7) an

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (1.10)$$

1.1.3 Verwirklichung des Quantencomputers

Die Theorie rund um den Quantencomputer ist bereits relativ weit fortgeschritten, während die praktische Umsetzung noch enorme Probleme bereitet. Trotz großer Fortschritte und erster Erfolge befindet sich der Quantencomputer immer noch in den Kinderschuhen. Wie ein zukünftiger Quantencomputer aussehen wird und welche Leistungen von ihm zu erwarten sind, ist noch nicht abschätzbar. Zur Zeit werden verschiedene Ansätze zu seiner Realisierung verfolgt. Wir werden uns darauf beschränken, den für diese Diplomarbeit relevanten Ansatz der Verwirklichung mittels Ionenfallen zu beschreiben, auf die wir im nächsten Abschnitt näher eingehen. Bei diesem

Ansatz werden Ionen als Qubits genutzt, die mit Laserstrahlen manipuliert werden. Dazu müssen wir solche Ionen auswählen, deren Elektronenhülle in guter Näherung als Zwei-Niveau-System beschreibbar ist. In unserer Gruppe erfolgt die Darstellung der Qubitzustände durch das metastabile D- und das S-Niveau von Kalzium-Ionen. Ein anderer erfolgreicher Ansatz besteht in der Verwendung zweier Hyperfeinniveaus.

1.2 Ionenfallen

In diesem Abschnitt beschränken wir uns auf einige grundlegende Elemente zur Physik der Ionenfallen. Eine ausführliche Darstellung findet sich z.B. in [13, 14]. Die Idee der Ionenfallen ist es, einzelne oder mehrere Ionen nur aufgrund ihrer elektrischen Ladung im Vakuum gefangen zu halten. Dies ist allein mit einem statischen elektrischen Feld nicht zu leisten. In der Penning-Falle wird deshalb zusätzlich noch ein statisches Magnetfeld angelegt. In unserer Gruppe benutzen wir hingegen das Prinzip der Paul-Falle. Der Einschluss der Ionen erfolgt hier durch ein hochfrequentes Wechselfeld, welches das statische Feld ergänzt. Wie eine exakte Lösung der Bewegungsgleichung für die Ionen zeigt, kann die Wirkung der Falle auf die Ionen in guter Näherung durch ein effektives statisches und harmonisches Potential beschrieben werden. Durch Laserkühlung „kristallisieren“ die Ionen im Fallenpotential. Dabei werden idealerweise sämtliche Schwingungsmoden auf den Grundzustand gekühlt. Beim Kristallisieren in einer linearen Paul-Falle ordnen sich die Ionen als Ionenkette entlang der Fallenachse an, da hier der Einschluss (das Potential) am schwächsten ist (siehe Abbildung 1.1). Durch die Kombination aus Fallenpotential und gegenseitiger Abstoßung sind die Ionen und deren Bewegungen aneinander gekoppelt. So ist dann auch die energieärmste Schwingungsmode die des Schwerpunktes der Ionenkette entlang der Fallenachse. Diese oder eine andere gemeinsame Schwingungsmode kann als Quantenbus genutzt werden – d.h. zur Informationsübertragung von einem Ion auf ein anderes. Durch eine zustandsabhängige Anregung *eines* Ions mittels Laser kann die ganze Ionenkette zum Schwingen gebracht werden. Dieser Schwingungszustand kann von einem anderen Ion abgefragt werden. Auf diesem Prinzip fußt der erste Vorschlag zur Umsetzung eines CNOTs, der 1995 von I. Cirac und P. Zoller präsentiert wurde [4]. Auch die in dieser Diplomarbeit behandelten laserinduzierten $\sigma_z \otimes \sigma_z$ - und Mølmer-Sørensen-Gatter benötigen eine gemeinsame Schwingungsmode

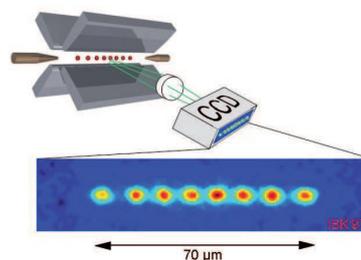


Abbildung 1.1: Lineare Paul-Falle mit acht gefangenen Ionen

zur Verschränkung der Ionen.

1.3 Ion-Laser-Wechselwirkung

In diesem Abschnitt soll der Hamilton-Operator für *ein* Ion hergeleitet werden, das mit Laserlicht beschienen wird. Das Laserlicht selbst können wir dabei als klassische monochromatische elektromagnetische Welle $\sim \cos(k_L x - \omega_L t + \phi)$ beschreiben. Die Elektronenhülle des Ions soll nur durch zwei Niveaus beschrieben werden: den Grundzustand $|g\rangle$ und den angeregten Zustand $|e\rangle$. Dies ist gerechtfertigt, wenn die Kreisfrequenz ω_L des Lasers dicht bei der Übergangsfrequenz ω_0 der beiden Niveaus liegt und weit verstimmt gegenüber anderen Übergängen ist. Wir betrachten auch nur eine Schwingungsmode. Schwingungsmoden führen zu Seitenbändern, für die auch das Resonanzargument herangezogen werden kann, um die Beschränkung auf eine Mode zu begründen.

Dieses System wird vielerorts beschrieben z.B. in den Referenzen [13, 15, 16, 17]. Der Hamilton-Operator H des Gesamtsystems kann in drei Anteile zerlegt werden:

ELEKTRONENHÜLLE: Diese beschreiben wir durch

$$\begin{aligned} H^{(\text{Elek.})} &= \hbar \cdot (\omega_e |e\rangle \langle e| + \omega_g |g\rangle \langle g|) \\ &= \hbar \frac{\omega_e + \omega_g}{2} \underbrace{(|e\rangle \langle e| + |g\rangle \langle g|)}_{\mathbb{1}} + \hbar \frac{\omega_e - \omega_g}{2} \underbrace{(|e\rangle \langle e| - |g\rangle \langle g|)}_{\sigma_z}. \end{aligned} \quad (1.11)$$

Den Anteil $\sim \mathbb{1}$ können wir weglassen; dies entspricht nur einer Eichung auf einen andern Energienullpunkt. Damit erhalten wir

$$H^{(\text{Elek.})} = \hbar \frac{\omega_0}{2} \sigma_z \quad \text{mit} \quad \omega_0 = \omega_e - \omega_g. \quad (1.12)$$

SCHWINGUNG DES IONS: Diese können wir als harmonisch betrachten und durch die Fallenfrequenz ν charakterisieren.

$$H^{(\text{harm. Osz.})} = \hbar \nu (\hat{a}^\dagger \hat{a} + \frac{1}{2}). \quad (1.13)$$

Dabei sind \hat{a}^\dagger und \hat{a} die Erzeugungs- und Vernichtungsoperatoren, die zu dieser Mode gehören. Da das Spektrum des harmonischen Oszillators keine Begrenzung nach oben aufweist, müssen wir bei numerischen Berechnungen einen maximalen Wert für $n = \hat{a}^\dagger \hat{a}$ festlegen.

WECHSELWIRKUNG: Die Stärke der die Wechselwirkung (W.W.) wird durch die Rabi-Frequenz Ω beschrieben. In Ω sind alle zeitunabhängigen Anteile des Übergangsmatrixelements $\langle e| H |g\rangle$ enthalten. Da die Zustände $|g\rangle$ und $|e\rangle$ nur bis auf eine Phase festgelegt sind, können wir Ω

immer reell wählen. Damit ergibt sich der Wechselwirkungs-Hamilton-Operator zu

$$H^{(\text{W.W.})} = \hbar\Omega \left(\underbrace{|e\rangle\langle g|}_{\sigma_+} + \underbrace{|g\rangle\langle e|}_{\sigma_-} \right) \cdot \left(e^{i(k_L \hat{x} - \omega_L t + \phi)} + e^{-i(k_L \hat{x} - \omega_L t + \phi)} \right). \quad (1.14)$$

Dabei ist \hat{x} der Ortsoperator des schwingenden Ions und kann ausgedrückt werden als

$$\hat{x} = \sqrt{\frac{\hbar}{2m\nu}} (\hat{a} + \hat{a}^\dagger). \quad (1.15)$$

Da wir die \hat{x} auf eine Dimension beschränkt haben, können wir den Wellenvektor k_L durch eine Zahl ersetzen und den dimensionslosen Lamb-Dicke-Faktor η definieren

$$\eta := k_L \cdot \sqrt{\frac{\hbar}{2m\nu}}. \quad (1.16)$$

Dieser beschreibt das Verhältnis der Schwingungsamplitude zur Wellenlänge des Lichtes (bei den Berechnungen für Kapitel 4 wurde z.B. durchgehend $\eta = 0,05$ verwendet). Damit erhalten wir

$$H^{(\text{W.W.})} = \hbar\Omega (\sigma_+ + \sigma_-) \cdot \left(e^{i(\eta(\hat{a} + \hat{a}^\dagger) - \omega_L t + \phi)} + e^{-i(\eta(\hat{a} + \hat{a}^\dagger) - \omega_L t + \phi)} \right). \quad (1.17)$$

Der Hamilton-Operator H des Gesamtsystems ergibt sich als Summe der drei Komponenten

$$H = H^{(\text{Elek.})} + H^{(\text{harm. Osz.})} + H^{(\text{W.W.})}. \quad (1.18)$$

Der nächste Schritt besteht in der Wahl eines anderen Bezugssystems; d.h. wir verlassen das Schrödinger-Bild und verwenden im Folgenden das Dirac- bzw. Wechselwirkungsbild zur Beschreibung unseres Systems. Wir werden zwei Transformationen vornehmen, die zu den Hamilton-Operatoren H^I und H^{II} führen. Beide Formen werden wir später verwenden. Im ersten Schritt transformieren wir gemäß

$$\begin{aligned} H &\rightarrow H^I \\ H^I &= e^{+\frac{i}{\hbar} H^{(\text{Elek.})} \cdot t} (H - H^{(\text{Elek.})}) e^{-\frac{i}{\hbar} H^{(\text{Elek.})} \cdot t}. \end{aligned} \quad (1.19)$$

Mit Hilfe von

$$e^{i\frac{\omega_0}{2}\sigma_z} \sigma_\pm e^{-i\frac{\omega_0}{2}\sigma_z} = e^{\pm i\omega_0} \sigma_\pm \quad (1.20)$$

ergibt sich

$$\begin{aligned} H^I &= \hbar\nu \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) \\ &+ \hbar\Omega \left[\sigma_+ e^{-i((\omega_L - \omega_0)t + \phi)} e^{+i\eta(\hat{a} + \hat{a}^\dagger)} + \sigma_- e^{+i((\omega_L - \omega_0)t + \phi)} e^{-i\eta(\hat{a} + \hat{a}^\dagger)} \right] \\ &+ \hbar\Omega \left[\sigma_+ e^{+i((\omega_L + \omega_0)t + \phi)} e^{-i\eta(\hat{a} + \hat{a}^\dagger)} + \sigma_- e^{-i((\omega_L + \omega_0)t + \phi)} e^{+i\eta(\hat{a} + \hat{a}^\dagger)} \right]. \end{aligned} \quad (1.21)$$

Da $\omega_L \approx \omega_0$ ist $\delta := \omega_L - \omega_0 \ll \omega_L + \omega_0$. Wir führen deshalb die als Drehwellennäherung (engl. rotating-wave approximation) bekannte Näherung durch, welche die schnell oszillierenden Terme $\sim e^{\pm i(\omega_L + \omega_0)t}$ vernachlässigt, da sich diese im Zeitentwicklungsoperator $U(t, t_0) = \mathcal{T}[\exp(-\frac{i}{\hbar} \int_{t_0}^t H^I dt)]$ weitestgehend wegheben ($\mathcal{T}[\dots]$ ist dabei der Zeitordnungsoperator). Nach der Drehwellennäherung verbleiben wir mit

$$H^I = \hbar\nu(\hat{a}^\dagger \hat{a} + \frac{1}{2}) + \hbar\Omega[\sigma_+ e^{-i(\delta \cdot t + \phi)} e^{+i\eta(\hat{a} + \hat{a}^\dagger)} + \text{h.c.}]. \quad (1.22)$$

Zwei typische Werte aus unseren Experimenten sind $\frac{\omega_L}{2\pi} \approx \frac{\omega_0}{2\pi} \approx 4 \cdot 10^{14} \text{Hz}$ und $\frac{\delta}{2\pi} \approx 10^6 \text{Hz}$. Da $\frac{\delta}{\omega_L} < 10^{-8}$, wird δ oft als *Verstimmung* bezeichnet. Wir hingegen wollen uns den Begriff der Verstimmung für die noch einzuführende Größe ε (1.34) & (1.41) aufsparen. Deshalb werden wir fortan auch δ als „Laserfrequenz“ bezeichnen und bei Bedarf „im mitrotierenden Bezugssystem“ hinzufügen.

Wie oben angekündigt, wollen wir noch eine zweite Transformation durchführen. Diese hat die Gestalt

$$\begin{aligned} H^I &\rightarrow H^{\text{II}} \\ H^{\text{II}} &= e^{+\frac{i}{\hbar} H^{(\text{harm. Osz.})} \cdot t} (H^I - H^{(\text{harm. Osz.})}) e^{-\frac{i}{\hbar} H^{(\text{harm. Osz.})} \cdot t}. \end{aligned} \quad (1.23)$$

Das liefert uns

$$H^{\text{II}} = \hbar\Omega\sigma_+ e^{-i(\delta \cdot t + \phi)} \exp[+i\eta(\hat{a} e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})] + \text{h.c.} \quad (1.24)$$

1.3.1 Mehrere Ionen im bichromatischen Laserfeld

Bisher haben wir uns bei unseren Betrachtungen auf *ein* Ion in *einem* Laserfeld beschränkt. Wie die Überschrift schon erkennen lässt, wollen wir jetzt die Erweiterung auf mehrere Ionen in einem Feld aus zwei Laserstrahlen vornehmen. Als ersten Schritt lassen wir jetzt n Ionen zu, die alle gleichzeitig und mit derselben Intensität vom Laser beleuchtet werden sollen.

- Jedes Ion hat nun die Möglichkeit, mit dem Laser zu interagieren. Deshalb müssen wir die Stufenoperatoren σ_\pm ersetzen

$$\sigma_+ \rightarrow S_+ = \sum_{i=1}^n \sigma_+^{(i)} \quad (1.25)$$

$$\sigma_- \rightarrow S_- = \sum_{i=1}^n \sigma_-^{(i)}, \quad (1.26)$$

wobei $\sigma_\pm^{(i)}$ auf das i te Ion wie σ_\pm wirkt und auf alle anderen wie $\mathbb{1}$.

- Wir lassen auch weiterhin nur eine Schwingungsmode zu – in unserem Fall die Schwerpunktbewegung der Ionenkette. Formal ändert sich damit nichts an den Formeln, aber \hat{a} und \hat{a}^\dagger sind jetzt die Erzeuger und Vernichter der gemeinsamen Schwingung und im Lamb-Dicke-Faktor η steht die Masse der Ionenkette.

Die zweite Erweiterung besteht in der Einführung eines zweiten Laserfeldes. Die beiden Laserstrahlen sollen co-propagieren, d.h. sie haben die gleiche Laufrichtung und werden durch die gleiche Ortsfunktion beschrieben. Des Weiteren möge sich die Kreisfrequenz δ der beiden Laser im mitrotierenden Bezugssystem nur durch das Vorzeichen unterscheiden. Das Gleiche möge für die Phase ϕ gelten (Dies kann durch Umdefinieren der Phase der Energieniveaus erreicht werden).

$$\delta_{\text{Laser 1}} = -\delta_{\text{Laser 2}} \quad \phi_{\text{Laser 1}} = -\phi_{\text{Laser 2}}. \quad (1.27)$$

Die beiden Laserstrahlen können auch als *ein* amplitudenmodulierter Laserstrahl betrachtet werden [18], wovon wir hier aber keinen weiteren Gebrauch machen werden.

Die beiden Hamilton-Operatoren H^{I} und H^{II} zur Beschreibung mehrerer Ionen im bichromatischen Laserfeld nehmen damit folgende Gestalt an

$$H^{\text{I}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})e^{+i\eta(\hat{a} + \hat{a}^\dagger)} + \text{h.c.} \quad (1.28)$$

$$+ \hbar\nu(\hat{a}^\dagger \hat{a} + \frac{1}{2})$$

$$H^{\text{II}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})e^{+i\eta(\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})} + \text{h.c.} \quad (1.29)$$

Wir wollen jetzt noch eine letzte Erweiterung vornehmen, indem wir einen Störterm $\hbar\frac{\gamma}{2}S_z$ mit aufnehmen, wobei $S_z = \sum_{i=1}^n \sigma_z^{(i)}$. Dieser Term beschreibt eine relative Verschiebung der Niveaus $|g\rangle$ und $|e\rangle$ zueinander. Diese kann durch eine ungewollte AC-Starkverschiebung (Kopplung an ein weiteres Niveau) oder durch magnetische Streufelder verursacht werden. Wenden wir auf $\hbar\frac{\gamma}{2}S_z$ die zu (1.19) analoge Transformation an, so sehen wir, dass dieser Term im neuen Bezugssystem durch eine Multiplikation mit $e^{+i\gamma t}$ in den Hamilton-Operator eingeht und somit einer gemeinsamen Verschiebung der beiden Laserfrequenzen um γ entspricht.

In Abschnitt 3.3 werden wir die folgenden beiden Darstellungen des Hamilton-Operators verwenden

$$H^{\text{I}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})e^{i\eta(a + a^\dagger)} + \text{h.c.} \\ + \hbar\nu(a^\dagger a + \frac{1}{2}) + \hbar\frac{\gamma}{2}S_z \quad (1.30)$$

$$H^{\text{II}} = \hbar\Omega e^{+i\gamma t} S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})e^{i\eta(ae^{-i\nu t} + a^\dagger e^{i\nu t})} \\ + \text{h.c.} \quad (1.31)$$

1.3.2 Interpretation der Ion-Laser-Wechselwirkung

In diesem Abschnitt geht es uns ausschließlich um ein Verständnis der Physik hinter der Ion-Laser-Wechselwirkung. Für die computergestützte Optimierung sind die hier betriebenen Analysen nicht erforderlich – dort greifen wir auf die Hamilton-Operatoren (1.30) und (1.31) zurück. Die hier gewonnenen Kenntnisse benötigen wir in den beiden kommenden Abschnitten 1.4 und 1.5, in denen es um das Mølmer-Sørensen- und das $\sigma_z \otimes \sigma_z$ -Gatter geht. Beide Gatter sind verschränkend und fußen auf Zwei-Photon-Prozessen. Vorher müssen wir aber erst einmal verstehen, welche Ein-Photon-Prozesse es gibt. Unser Ausgangspunkt ist der Hamilton-Operator H^{II} ohne den Störterm (1.29)

$$H^{\text{II}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})e^{+i\eta(\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})} + \text{h.c.}$$

Schwierigkeiten bereitet hier besonders der Term $e^{+i\eta(\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})}$ mit seinem Exponenten im Exponenten. Da $\eta \ll 1$, können wir für die Exponentialfunktion die Lamb-Dicke-Näherung verwenden

$$H^{\text{II}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)})[1 + i\eta(\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})] + \text{h.c.} \quad (1.32)$$

Die beiden Klammern können wir ausmultiplizieren und drei verschiedenen Prozessen zuordnen

TRÄGER-ÜBERGANG $|g\rangle |n\rangle \longleftrightarrow |e\rangle |n\rangle$
 $H^{\text{Träger}} = \hbar\Omega S_+(e^{-i(\delta \cdot t + \phi)} + e^{+i(\delta \cdot t + \phi)}) + \text{h.c.}$

ROTES SEITENBAND $|g\rangle |n+1\rangle \longleftrightarrow |e\rangle |n\rangle$
 $H^{\text{rot}} = i\hbar\eta\Omega S_+\hat{a}(e^{-i((\delta+\nu)t+\phi)} + e^{+i((\delta-\nu)t+\phi)}) + \text{h.c.}$

BLAUES SEITENBAND $|g\rangle |n\rangle \longleftrightarrow |e\rangle |n+1\rangle$
 $H^{\text{blau}} = i\hbar\eta\Omega S_+\hat{a}^\dagger e^{-i((\delta-\nu)t+\phi)} + e^{+i((\delta+\nu)t+\phi)}) + \text{h.c.}$

Zur Verdeutlichung der Prozesse haben wir die Veränderungen des elektronischen Zustands *eines* Ions ($|g\rangle, |e\rangle$) und der gemeinsamen Schwingungsmode ($|n\rangle, |n+1\rangle$) angegeben. Unsere nächste Aufgabe wird darin bestehen, ein Verständnis für die relative Stärke der einzelnen Prozesse zu gewinnen. Vier Faktoren sind für uns von Bedeutung

LAMB-DICKE-FAKTOR: Die Seitenbänder sind gegenüber dem Träger-Übergang um den Lamb-Dicke-Faktor η unterdrückt.

RESONANZ: Nichtresonante Übergänge erscheinen im Hamilton-Operator mit einer oszillierenden Exponentialfunktion, die zur Unterdrückung dieser Übergänge führt (siehe *Drehwellennäherung*, Seite 9).

SCHWINGUNGSZUSTAND: Die Kopplungsstärke der Erzeuger und Vernichter \hat{a}^\dagger, \hat{a} ist zustandsabhängig

$$\hat{a}^\dagger |n\rangle = \sqrt{n+1} |n+1\rangle \quad \hat{a} |n\rangle = \sqrt{n} |n-1\rangle. \quad (1.33)$$

INTERFERENZ: Vor allem bei den für uns relevanten Zwei-Photon-Prozessen wird die Interferenz eine Schlüsselrolle einnehmen.

Als nächsten Schritt müssen wir die Kreisfrequenz $\pm\delta$ der Laser im mitrotierenden Bezugssystem festlegen. Da dieser Wert für das Mølmer-Sørensen- und das $\sigma_z \otimes \sigma_z$ -Gatter unterschiedlich ausfällt, werden wir trotz weiterer Gemeinsamkeiten fortan eine getrennte Diskussion der beiden Gatter führen.

1.4 Mølmer-Sørensen-Gatter

Beim Mølmer-Sørensen-Gatter [9, 18, 19] wird die Kreisfrequenz $\pm\delta$ der beiden Laser so gewählt, dass sie annähernd mit dem roten bzw. blauen Seitenband zusammenfällt.

$$\delta = \nu - \varepsilon \quad \varepsilon \ll \nu. \quad (1.34)$$

Dabei ist ν wieder die Fallenfrequenz. Den Wert ε werden wir als *Verstimmung* bezeichnen (zur Wortwahl siehe auch Seite 9). Dieser Wert ist klein aber nicht infinitesimal, wie der übliche Gebrauch von ε suggerieren könnte. Für $\varepsilon > 0$ ist der eine Laser leicht rotverstimmt gegenüber dem blauen Seitenband und der andere ist leicht blauverstimmt gegenüber dem roten Seitenband (Abbildung 1.2).

Durch die Verstimmung ε gibt es keinen resonanten Ein-Photon-Prozess, aber es gibt resonante Zwei-Photon-Prozesse. Man könnte erwarten, dass der dominante Zwei-Photon-Prozess der doppelte Träger-Übergang sei, der mit der Absorption eines roten $\sim \exp(+i(\nu - \varepsilon)t)$ und eines blauen $\sim \exp(-i(\nu - \varepsilon)t)$ Photons einher geht. Dieser Prozess fällt aber der destruktiven Interferenz zum Opfer. Dies erkennen wir anhand der effektiven Rabifrequenz $\tilde{\Omega}$, die sich aus der Störungsrechnung zweiter Ordnung ergibt³ [9]

$$\tilde{\Omega}_{\text{Zustand1} \rightarrow \text{Zustand2}} = \frac{1}{\hbar} \sum_m \frac{\langle \text{Zustand2} | H | m \rangle \cdot \langle m | H | \text{Zustand1} \rangle}{E_m - (E_{\text{Zustand1}} + \hbar\omega_1)}. \quad (1.35)$$

Die Summation läuft formal über alle möglichen Prozesse. Der Ausdruck $\hbar\omega_1$ im Nenner von (1.35) repräsentiert die Energie des ersten absorbierten Photons – d.h. des Photons, welches den Übergang $|\text{Zustand1}\rangle \rightarrow |m\rangle$ einleitet. Für $|g, g, n\rangle \rightarrow |e, e, n\rangle$ erhalten wir

$$\tilde{\Omega} = \frac{1}{\hbar} \sum_m \frac{\langle e, e, n | H | m \rangle \cdot \langle m | H | g, g, n \rangle}{E_m - (E_{gg} + \hbar\omega_1)}. \quad (1.36)$$

³Abgesehen von der Summierung handelt es sich bei (1.35) um die gleiche Formel, die zur Berechnung der effektiven Rabifrequenz eines stimulierten Raman-Übergangs verwendet wird. Bei nichtresonanten Prozessen enthält (1.35) noch eine zeitabhängige Exponentialfunktion. Letztere ist notwendig, um die Kopplung korrekt zu beschreiben, wird aber nicht zur effektiven Rabifrequenz gezählt.

Wenn wir uns auf die Träger-Übergänge beschränken, gilt es zwei Fälle zu unterscheiden:

- Zuerst wird ein rotes Photon absorbiert

$$E_m - (E_{ggn} + \hbar\omega_1) = \hbar(\nu - \varepsilon).$$

- Zuerst wird ein blaues Photon absorbiert

$$E_m - (E_{ggn} + \hbar\omega_1) = -\hbar(\nu - \varepsilon).$$

Da der Zähler in beiden Fällen der gleiche ist, summieren sich diese Beiträge zu null.

Dieses Argument gilt nicht für *nichtresonante* doppelte Träger-Übergänge, also die Absorption zweier blauer oder zweier roter Photonen. Diese unterscheiden sich aber nicht von zwei unabhängigen Ein-Photon-Prozessen und sind somit nicht geeignet, eine Verschränkung zu erzeugen.

Betrachten wir wieder den Fall der Absorption eines roten und eines blauen Photons, wobei jetzt die Seitenbänder die virtuellen Zwischen-niveaus $|m\rangle$ (1.36) für den Übergang $|g, g, n\rangle \rightarrow |e, e, n\rangle$ stellen sollen. Dann gilt

$$E_m - (E_{ggn} + \hbar\omega_1) = \pm\hbar\varepsilon. \quad (1.37)$$

Das Vorzeichen hängt wieder davon ab, welches Photon zuerst absorbiert wird. Wir haben es also erneut mit destruktiver Interferenz zu tun. Aber es gibt einen entscheidenden Unterschied: Da die Kopplung von \hat{a} und \hat{a}^\dagger zustandsabhängig ist, hängt der Wert des Zählers von (1.36) davon ab, welchen Wert wir für $|m\rangle$ annehmen. Somit wird die effektive Rabi-Frequenz (1.36) proportional zu

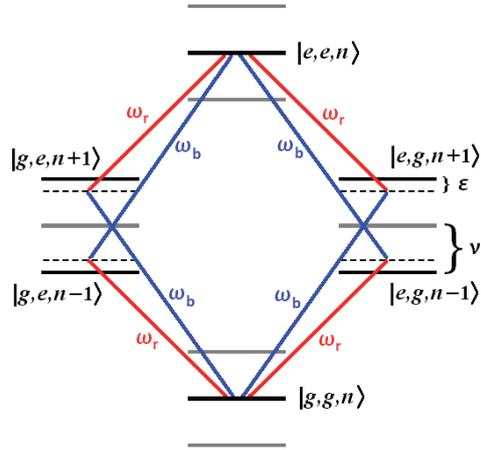


Abbildung 1.2: Beim Mølmer-Sørensen-Gatter erfolgt der Übergang $|g, g, n\rangle \leftrightarrow |e, e, n\rangle$ durch Interferenz von vier Zwei-Photon-Prozessen mit jeweils $2\omega_0 = \omega_b + \omega_r$.

$$\begin{aligned} \tilde{\Omega} &\sim \underbrace{\langle n | \hat{a} | n+1 \rangle}_{\text{rot}} \cdot \underbrace{\langle n+1 | \hat{a}^\dagger | n \rangle}_{\text{blau}} - \underbrace{\langle n | \hat{a}^\dagger | n-1 \rangle}_{\text{blau}} \cdot \underbrace{\langle n-1 | \hat{a} | n \rangle}_{\text{rot}} \\ &= \sqrt{n+1} \cdot \sqrt{n+1} - \sqrt{n} \cdot \sqrt{n} \\ &= 1. \end{aligned} \quad (1.38)$$

Nicht genug, dass der Beitrag der Seitenbänder nicht verschwindet – er ist sogar noch unabhängig von der Besetzungszahl der Schwingungsmode! Somit ist im Gegensatz zum Cirac-Zoller-CNOT die Kühlung der Schwingungsmode auf den Grundzustand nicht unbedingt erforderlich. Einen ähnlichen Effekt beobachten wir bei der Absorption und Emission zweier Photonen

gleicher Farbe. Dies ermöglicht den Übergang

$$|g, e, n\rangle \longleftrightarrow |e, g, n\rangle. \quad (1.39)$$

Es sei aber daran erinnert, dass wir z.Z. mit einem genäherten Hamilton-Operator rechnen.

Wie weiter oben erwähnt, gibt es keine resonanten Ein-Photon-Prozesse. Das heißt aber nicht, dass diese Prozesse bedeutungslos sind. Durch geschickte Wahl der Gatterzeit lässt sich aber dafür sorgen, dass die unerwünschten Effekte dieser Prozesse verschwinden. So führt der nichtresonante Träger-Übergang z.B. zu Rabioszillationen, die regelmäßig wieder auf null abfallen. Der Effekt der nichtresonanten Seitenbänder hingegen ist etwas komplizierter und lässt sich besser verstehen, nachdem das $\sigma_z \otimes \sigma_z$ -Gatter behandelt wurde. Wir werden deshalb am Ende des Abschnitts 1.5 noch einmal darauf zurückkommen.

Beschreibung des Gatters

Das idealisierte Mølmer-Sørensen-Gatter kann beschrieben werden durch [9]

$$\begin{aligned} U_{\text{MS}}(\vartheta) &= \exp[+i\vartheta(S_y)^2] \otimes \mathbb{1}_{\text{Schwing.}} \quad (1.40) \\ &= \exp\left[+i\vartheta \sum_{j \neq k} \sigma_y^{(j)} \otimes \sigma_y^{(k)}\right] \cdot \underbrace{\exp\left[+i2\vartheta \sum_{j=k} \sigma_y^{(j)} \otimes \sigma_y^{(k)}\right]}_{= e^{+i2\vartheta(n\mathbb{1})} = \text{globale Phase}} \otimes \mathbb{1}_{\text{Schwing.}}, \end{aligned}$$

wobei $S_y = \sum_{i=1}^n \sigma_y^{(i)}$ und $\vartheta = \frac{2\pi\eta^2\Omega^2}{\varepsilon^2}$. Die $\sigma_y^{(j)} \otimes \sigma_y^{(k)}$ beschreiben kollektive Niveau-Flips, ausgelöst durch die Zwei-Photon-Übergänge. Dass in (1.40) σ_y erscheint und nicht etwa σ_x , hängt mit der Phasenbeziehung zwischen den Lasern und den Atomniveaus zusammen⁴, die sich auch in der Wahl der Rabifrequenz als reelle Zahl widerspiegelt. Durch eine Veränderung der Phasenbeziehung können wir σ_y durch jede Linearkombination aus σ_y und σ_x ersetzen.

1.5 $\sigma_z \otimes \sigma_z$ -Gatter

Beim $\sigma_z \otimes \sigma_z$ -Gatter [8, 18, 19] wird die Kreisfrequenz der beiden Laser so gewählt, dass sie annähernd in der Mitte zwischen einem der Seitenbänder und dem Träger liegt.

$$\delta = \frac{\nu}{2} - \varepsilon \quad \varepsilon \ll \nu. \quad (1.41)$$

⁴Dies sollte nicht mit der explizit mitgeführten Phase ϕ verwechselt werden, welche nur die Phasenbeziehung zwischen den beiden Lasern beeinflusst.

Wie beim Mølmer-Sørensen-Gatter gibt es keine resonanten Ein-Photon-Übergänge. Die resonanten doppelten Seitenband-Übergänge, die beim Mølmer-Sørensen-Gatter die entscheidende Rolle spielen, sind beim $\sigma_z \otimes \sigma_z$ -Gatter stark unterdrückt, da in (1.35) jetzt

$$E_m - (E_{\text{Zustand1}} + \hbar\omega_1) = \pm\hbar(\nu - (\frac{\nu}{2} - \varepsilon)) = \pm\hbar(\frac{\nu}{2} + \varepsilon) \quad (1.42)$$

im Nenner erscheint anstatt $\pm\hbar\varepsilon$. Die wichtigsten Zwei-Photon-Prozesse sind jetzt:

- Absorption eines blauen Photons und Emission eines roten
 $|n\rangle \rightarrow |n+1\rangle$
- Absorption eines roten Photons und Emission eines blauen
 $|n\rangle \rightarrow |n-1\rangle$

Beide Prozesse sind als Kombination eines Seitenband-Übergangs plus Träger-Übergang darstellbar, was mit einer stärkeren Kopplung als beim doppelten Seitenband-Übergang einhergeht. Andererseits ist diese Kombinationen mit 2ε leicht verstimmt (siehe Abbildung 1.3). Des Weiteren können beide Prozesse nur stattfinden, wenn Absorption und Emission *vom selben* Ion erfolgen. Übergänge der Art

$$|g, e, n\rangle \longleftrightarrow |e, g, n \pm 1\rangle \quad (1.43)$$

würden sich wieder wegheben, da es zwei destruktiv interferierende Wege gibt. Jedes der beiden Ionen kann den „ersten Schritt“ machen, während es bei einem Ion für

$$|g, n\rangle \rightarrow |g, n \pm 1\rangle \quad |e, n\rangle \rightarrow |e, n \pm 1\rangle \quad (1.44)$$

jeweils nur einen Weg gibt, da der Zustand $|g\rangle$ immer erst ein Photon absorbieren muss, bevor er emittieren kann (und umgekehrt für $|e\rangle$). Es gibt allerdings immer noch die Wahl zwischen zwei virtuellen Zwischenzuständen für $|m\rangle$ in (1.35), je nachdem ob man für $|m\rangle$ den Träger- oder den Seitenband-Übergang wählt. Da aber die Kopplung der Seitenbänder mit $i\hbar\eta\Omega$ rein imaginär ist, haben die Seitenband-Übergänge bei der Absorption und Emission (\leftarrow das hermitisch Konjugierte) unterschiedliche Vorzeichen, was den annähernden⁵ Vorzeichenunterschied im Nenner von (1.35) kompensiert.

⁵ $E_{\text{blau S.B.}} - (E_{Z.1} + \hbar\omega_{\text{blau}}) = \hbar(\frac{\nu}{2} + \varepsilon)$ und $E_{\text{Träger}} - (E_{Z.1} + \hbar\omega_{\text{blau}}) = -\hbar(\frac{\nu}{2} - \varepsilon)$

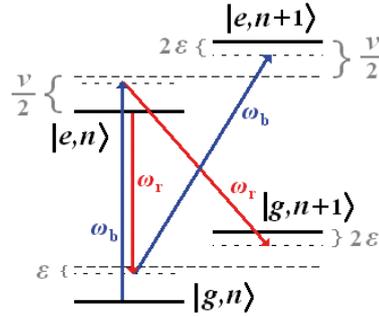


Abbildung 1.3: Beim $\sigma_z \otimes \sigma_z$ -Gatter ermöglicht die gleichzeitige Absorption eines blauen Photons ω_b und die Emission eines roten ω_r den Übergang $|n\rangle \rightarrow |n+1\rangle$.

Da bei $|g, n\rangle \rightarrow |g, n+1\rangle$ und $|e, n\rangle \rightarrow |e, n+1\rangle$ die Reihenfolge von Absorption und Emission vertauscht ist, liefert (1.35) unterschiedliche Vorzeichen für deren effektive Rabifrequenzen $\tilde{\Omega}$

$$\tilde{\Omega}_{|e,n\rangle \rightarrow |e,n+1\rangle} = -\tilde{\Omega}_{|g,n\rangle \rightarrow |g,n+1\rangle}. \quad (1.45)$$

Wenn wir bereit sind, für den Rest dieser Seite die effektive Rabifrequenz als Operator darzustellen, können wir obige Aussage wie folgt zusammenfassen

$$\tilde{\Omega}_{|\alpha g + \beta e, n\rangle \rightarrow |\alpha g + \beta e, n+1\rangle} = \tilde{\Omega}_{|e,n\rangle \rightarrow |e,n+1\rangle} \sigma_z. \quad (1.46)$$

1.5.1 Interpretation des $\sigma_z \otimes \sigma_z$ -Gatters

Im letzten Abschnitt haben wir die dominanten Zwei-Photon-Übergänge ausgemacht. Hierbei handelt es sich um Ein-Ion-Prozess. Dies mag etwas erstaunen, da das $\sigma_z \otimes \sigma_z$ -Gatter ein verschränkendes Gatter ist. Verschränkung ist aber ein Effekt, der die Kopplung mehrere Teilchen bzw. Freiheitsgrade erfordert. Dem soll jetzt nachgegangen werden.

Mit dem Wissen aus dem letzten Kapitel können wir die Zwei-Photon-Prozesse in einem phänomenologischen Hamilton H zusammenfassen

$$H = \gamma \sigma_z \hat{a}^\dagger e^{i2\epsilon t} + \text{h.c.} \quad (1.47)$$

Die Anwendung von \hat{a}^\dagger auf $|n\rangle$ erzeugt einen zusätzlichen Faktor $\sqrt{n+1}$, der bei unserer Herleitung bereits in $\tilde{\Omega}_{|e,n\rangle \rightarrow |e,n+1\rangle}$ enthalten ist. Deshalb wurde in (1.47) die Kopplung $\gamma = \frac{1}{\sqrt{n+1}} \tilde{\Omega}_{|e,n\rangle \rightarrow |e,n+1\rangle}$ eingeführt, die unabhängig von $|n\rangle$ ist.

Wir nehmen jetzt die Erweiterung auf n Ionen vor, die mit $S_z = \sum_{i=1}^n \sigma_z^{(i)}$ trivial ist

$$H = \gamma S_z \hat{a}^\dagger e^{i2\epsilon t} + \text{h.c.} \quad (1.48)$$

Wir wollen uns hier auf zwei Ionen beschränken. Dann gilt

$$\langle e, e | H | e, e \rangle = 2\gamma \hat{a}^\dagger e^{i2\epsilon t} + \text{h.c.} = -\langle g, g | H | g, g \rangle \quad (1.49)$$

$$\langle g, e | H | g, e \rangle = \langle e, g | H | e, g \rangle = 0. \quad (1.50)$$

$H^{ee} = \langle e, e | H | e, e \rangle$ beschreibt einen getriebenen harmonischen Oszillator im Wechselwirkungsbild. Im Folgenden wollen wir uns die Konsequenzen dieser Interpretation anschauen. Auf eine exakte mathematische Ausarbeitung werden wir dabei allerdings verzichten und verweisen statt dessen auf [19].

Die den harmonischen Oszillator treibende Kraft ist in unserem Bild um 2ϵ verstimmt. Dies führt dazu, dass der Oszillator während der Zeitspanne

$$\tau = \frac{2\pi}{2\epsilon} \quad (1.51)$$

genauso stark beschleunigt wie gebremst wird und nach der Zeit τ wieder im Ausgangszustand ist

$$U(\tau) |\alpha\rangle = \mathcal{T} \left[\exp \left(-\frac{i}{\hbar} \int_{t_0}^{t_0+\tau} 2\gamma \hat{a}^\dagger e^{i2\epsilon t} - 2\gamma \hat{a} e^{-i2\epsilon t} dt \right) \right] |\alpha\rangle = e^{i\Phi} |\alpha\rangle, \quad (1.52)$$

wobei $|\alpha\rangle$ einen beliebigen kohärenten Schwingungszustand bezeichnet. $|\alpha\rangle$ lässt sich mit $\langle x \rangle \sim \frac{1}{\sqrt{2}}(\alpha^* + \alpha)$ und $\langle p \rangle \sim \frac{i}{\sqrt{2}}(\alpha^* - \alpha)$ als Punkt im Phasenraum darstellen. Dieser Punkt durchläuft in der Zeit τ einen Kreis im Phasenraum. Dabei wird eine geometrische Phase Φ aufgesammelt, die der eingeschlossenen Kreisfläche im Phasenraum entspricht. Der spezielle Schwingungszustand selbst hat keinen Einfluss auf die Phase, womit auch hier die zwingende Notwendigkeit der Kühlung der Ionen in den Grundzustand der Schwingungsmode entfällt. Ebenfalls bemerkenswert ist das Vorzeichen dieser Phase, welches durch den Umlaufsinn gegeben ist, mit dem der Phasenraumkreis durchlaufen wurde. Der Umlaufsinn ist wiederum durch das Vorzeichen der Verstimmung gegeben und damit für $H^{ee} = \langle e, e | H | e, e \rangle$ und $H^{gg} = \langle g, g | H | g, g \rangle$ der gleiche! Das heißt, die Zustände $|g, g\rangle$ und $|e, e\rangle$ sammeln die gleiche geometrische Phase Φ auf, während $|g, e\rangle$ und $|e, g\rangle$ keine extra Phase erhalten, da ihr Hamilton-Operator null ist bzw. die Kräfte sich gegenseitig wegheben.

Somit können wir das $\sigma_z \otimes \sigma_z$ -Gatter für zwei Ionen wie folgt beschreiben

$$U_{\sigma_z \otimes \sigma_z}(\tau) = \begin{pmatrix} e^{i\Phi} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\Phi} \end{pmatrix} \otimes \mathbb{1}_{\text{Schwingung}} \quad \Phi \approx \frac{8\pi\eta^2\Omega^4}{\delta^2\epsilon^2}. \quad (1.53)$$

Dieses Gatter ist sensitiv für den *gemeinsamen* Zustand beider Ionen und somit verschränkend. Es sei nochmals auf die besondere Eigenschaft hingewiesen, dass $|g, g\rangle$ und $|e, e\rangle$ die gleiche Phase mit dem gleichen Vorzeichen erhalten. Dies ist zu vergleichen mit

$$U_{\text{trivial}} = \begin{pmatrix} e^{-i\Phi} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{+i\Phi} \end{pmatrix} \otimes \mathbb{1}_{\text{Schwingung}}, \quad (1.54)$$

welches durch eine einfache Transformation zu erhalten ist

$$|e\rangle \rightarrow e^{-\frac{i}{2}\Phi} |e\rangle \quad |g\rangle \rightarrow e^{+\frac{i}{2}\Phi} |g\rangle. \quad (1.55)$$

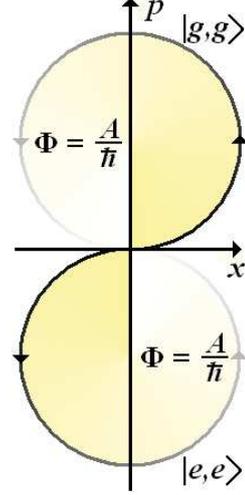


Abbildung 1.4: Beim $\sigma_z \otimes \sigma_z$ -Gatter sammeln $|g, g\rangle$ und $|e, e\rangle$ die gleiche geometrische Phase Φ auf.

Damit wird auch erkennbar, dass $\Phi = \pi$ wegen $e^{+i\pi} = e^{-i\pi}$ keine Verschränkung erzeugt. Die maximale Verschränkung ergibt sich für $\Phi = \frac{\pi}{2}$, also $e^{i\Phi} = i$. Für diesen Wert lässt sich aus $U_{\sigma_z \otimes \sigma_z}$ und U_{trivial} ein kontrolliertes σ_z -Gatter (CZ) erzeugen

$$\begin{aligned}
 \begin{array}{c} \bullet \\ | \\ \square Z \\ | \\ \bullet \end{array} &= \begin{array}{c} \square Z \\ | \\ \bullet \end{array} = U_{\sigma_z \otimes \sigma_z}(\Phi = \frac{\pi}{2}) \cdot U_{\text{trivial}}(\Phi = \frac{\pi}{2}) \quad (1.56) \\
 &= \begin{pmatrix} i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix} \begin{pmatrix} -i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}
 \end{aligned}$$

Damit lässt sich wiederum ein CNOT erzeugen

$$\begin{array}{c} \bullet \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \\ | \\ \square H \square Z \square H \end{array} \quad (1.57)$$

mit der Hadamard-Matrix H

$$\square H \square = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.58)$$

Die Verwandtschaft zum Mølmer-Sørensen-Gatter enthüllt sich anhand einer alternativen Darstellung für $U_{\sigma_z \otimes \sigma_z}$, welche der des Mølmer-Sørensen-Gatters (1.40) ähnelt

$$U_{\sigma_z \otimes \sigma_z}(\tau) = e^{i\frac{\Phi}{2}\sigma_z \otimes \sigma_z} \otimes \mathbb{1}_{\text{Schwingung}}. \quad (1.59)$$

Hierbei wurde die globale Phase $e^{i\frac{\Phi}{2}}$ weggelassen.

Gatterzeiten beim Mølmer-Sørensen-Gatter

Auch das Mølmer-Sørensen-Gatter ist in der Lage, durch Absorption und Emission zweier Photonen unterschiedlicher Farbe eine geometrische Phase aufzusammeln. Im Gegensatz zum $\sigma_z \otimes \sigma_z$ -Gatter ist dies beim Mølmer-Sørensen-Gatter durch zwei Ein-Photon-Prozesse (nichtresonante Seitenband-Übergänge) möglich. Da es auch hier gilt, eine geschlossene Phasenraumbahn zu durchlaufen, bestimmt dieser Effekt letztlich die Gatterzeit des Mølmer-Sørensen-Gatters.

Der entsprechende Zwei-Photonen-Prozess des $\sigma_z \otimes \sigma_z$ -Gatters ermöglicht den Übergang $|n\rangle \leftrightarrow |n+1\rangle$, während der doppelte Ein-Photon-Prozess des

Mølmer-Sørensen-Gatters zum Übergang $|n\rangle \leftrightarrow |n+2\rangle$ führt. Beide Übergänge sind nicht resonant. Da beim $\sigma_z \otimes \sigma_z$ -Gatter aber zwei „Schritte“ für den Übergang $|n\rangle \leftrightarrow |n+1\rangle \leftrightarrow |n+2\rangle$ benötigt werden, ist dieser Übergang doppelt so stark verstimmt wie der vergleichbare Übergang beim Mølmer-Sørensen-Gatter (gleiches ε vorausgesetzt). Dies hat zur Konsequenz, dass die Gatterzeit des Mølmer-Sørensen-Gatters τ_{MS} der doppelten Gatterzeit des $\sigma_z \otimes \sigma_z$ -Gatters $\tau_{\sigma_z \otimes \sigma_z}$ (1.51) entspricht

$$\tau_{\sigma_z \otimes \sigma_z} = \frac{2\pi}{2\varepsilon}, \quad \tau_{\text{MS}} = \frac{2\pi}{\varepsilon}. \quad (1.60)$$

1.6 Ideale und reale Gatter

In Abschnitt 1.4 und 1.5 wurden das Mølmer-Sørensen- und $\sigma_z \otimes \sigma_z$ -Gatter erklärt. Dabei haben wir einige Näherungen vorgenommen, so dass es sich um idealisierte Beschreibungen handelt. Diese idealisierten Beschreibungen stellen zugleich die gewünschte Form der Gatter dar, wie wir sie zu erzeugen beabsichtigen. Daher geht unser Bestreben dahin, dass die realen Gatter den idealisierten möglichst nahe kommen sollen. Dieser Aufgabe nehmen wir uns in den Kapiteln 3 und 4 an. Dabei verzichten wir zum Einen auf einige Näherungen und verwenden die Hamilton-Operatoren (1.30) & (1.31). Zum Anderen erlauben wir eine zeitliche Variation der Rabi-Frequenz Ω und der Phase ϕ . Wie diese neuen Freiheitsgrade möglichst gewinnbringend genutzt werden können, ist nicht trivial zu erkennen. Deshalb greifen wir auf Algorithmen der *optimal control theory* zurück. Letztere werden im nun folgenden Kapitel erklärt.

Kapitel 2

Optimierungsalgorithmen

In diesem Kapitel werden grundlegende Ansätze zur Lösung eines Optimierungsproblems präsentiert. Die Darstellung ist möglichst allgemein gehalten, und die Einbeziehung der später zu lösenden physikalischen Probleme geschieht vorwiegend, um konkrete Beispiele zu haben. Eine Ausnahme bildet diesbezüglich Abschnitt 2.3, in dem ein Algorithmus präsentiert wird, der die Faktorisierbarkeit des quantenmechanischen Zeitentwicklungsoperators nutzt. Denjenigen, die sich mit der in diesem Kapitel präsentierten Materie bereits auskennen, sei trotzdem ein Blick in Abschnitt 2.5 ans Herz gelegt, da hier eine vorteilhafte Variante des Simulated Annealing präsentiert wird, die im Rahmen dieser Diplomarbeit entwickelt wurde.

2.1 Optimal control theory (OCT)

Optimal control theory (OCT) findet Einsatz in Bereichen wie Wirtschaft, Chemie, Physik und Technik. In allen Fällen haben wir es mit einem System zu tun, auf dessen zeitliche Entwicklung wir Einfluss nehmen können, indem wir die Werte gewisser Systemvariablen verändern. Diesen Einfluss versuchen wir derart zu nutzen, dass das System am Ende gewisse Kriterien bzw. Zielvorgaben erfüllt. In den meisten Fällen wird es aber nicht trivial ersichtlich sein, welche Wahl der Systemvariablen die geeignetste ist, um die Zielvorgaben möglichst optimal zu erfüllen. Genau diese Aufgabe – die Auffindung der geeignetsten Systemvariablen – bemüht sich die OCT zu lösen.

In dieser Diplomarbeit besteht das System aus Ionen in einer Ionenfalle, die mit Laserstrahlen wechselwirken. Unser Ziel ist es, auf diesen Ionen ein Quantengatter zu implementieren. Was die Systemvariablen anbelangt, verfolgen wir zwei verschiedene Ansätze. Im ersten Teil (Kapitel 3 & 4) betrachten wir die Phasen und Rabi-Frequenzen der Laserstrahlen als Systemvariablen, die es optimal zu bestimmen gilt. Im zweiten Teil (Kapitel 5 & 6) werden wir etwas abstrakter und stellen ein Repertoire an Standardoperationen zur Verfügung. Die Aufgabe besteht dann darin, diese Standardoperationen in

der richtigen Reihenfolge und Länge aneinanderzureihen.

Die Komplexität der Aufgabenstellung erlaubt es meistens nicht, die optimalen Systemvariablen analytisch zu bestimmen. Wir werden uns somit um numerische Verfahren zur Lösung des OCT-Problem bemühen müssen. Für diesen Zweck versuchen wir, das Problem so zu formulieren, dass es durch einen Algorithmus zum Auffinden von Maxima und Minima einer Funktion (bzw. eines Funktionals) gelöst werden kann. Anders formuliert: Wir suchen eine von den Systemvariablen abhängige Funktion, die ihr globales Extremum genau bei den Werten der Systemvariablen annimmt, die auch das OCT-Problem optimal lösen. Je nachdem, ob wir ein Maximum oder Minimum suchen, bezeichnen wir diese Funktion als *Kostenfunktion* (Minimum) oder *Gewinnfunktion* (Maximum). Unsere physikalischen Probleme werden wir alle durch eine Gewinnfunktion beschreiben. Als erstes abstraktes Beispiel soll uns aber eine Kostenfunktion dienen. Nehmen wir dafür an, unser Ziel besteht darin, einen bestimmten Endzustand $\mathcal{S}_{\text{Ziel}}$ des Systems zu erreichen. Nehmen wir ferner an, wir können eine Art Metrik $d(\mathcal{S}_1, \mathcal{S}_2)$ für die verschiedenen Zustände des Systems definieren — d.h. eine Abbildung, die zwei Zustände des Systems $\mathcal{S}_1, \mathcal{S}_2$ als Eingabe erhält und als Ausgabe eine positive Zahl ausgibt, die den Abstand (bzw. die Verschiedenheit) der Zustände beschreibt. Dann können wir eine Kostenfunktion \mathfrak{K} wie folgt konstruieren

$$\mathfrak{K}(\mathbf{x}) := d(\mathcal{S}(\mathbf{x}), \mathcal{S}_{\text{Ziel}}), \quad (2.1)$$

wobei der erste der beiden Systemzustände $\mathcal{S}(\mathbf{x})$ in der Metrik als Funktion der Systemvariablen \mathbf{x} gegeben ist. Diese Abhängigkeit von den Systemvariablen überträgt sich damit auch auf die Kostenfunktion \mathfrak{K} . Da der kleinste Ausgabewert der Metrik null ist und dieser nur vergeben wird, wenn die beiden Eingabezustände übereinstimmen, löst die Minimierung von \mathfrak{K} nach den Systemvariablen \mathbf{x} das OCT-Problem. (Die Existenz einer Lösung sei dabei vorausgesetzt.)

Betrachten wir nun das Problem der in Fallen gefangenen Ionen, auf denen wir Quantengatter implementieren möchten. Wir sind jetzt nicht mehr an einem speziellen Endzustand des Systems interessiert, sondern möchten, dass der Zeitentwicklungsoperator $U(\Delta t)$ einem bestimmten Gatter¹ U_{Ziel} entspricht. Die Gatteroperation, die sich aus der Anwendung eines bestimmten Variablensatzes \mathbf{x} ergibt, bezeichnen wir als $U_{\text{Real}}(\mathbf{x})$. Es bietet sich nun an, eine Gewinnfunktion \mathfrak{G} wie folgt zu definieren [20]

$$\mathfrak{G}(\mathbf{x}) := \langle U_{\text{Real}}(\mathbf{x}) | U_{\text{Ziel}} \rangle = \text{tr}(U_{\text{Real}}(\mathbf{x}) \cdot U_{\text{Ziel}}^\dagger). \quad (2.2)$$

Hier verwenden wir das Skalarprodukt für Matrizen nach Hilbert-Schmidt bzw. Frobenius. Es nimmt seinen maximalen Wert nur für $U_{\text{Real}}(\mathbf{x}) = U_{\text{Ziel}}$

¹Wir verwenden den Begriff *Gatter* hier auch für eine Kombination aus mehreren (elementaren) Gattern.

an.

Zwei quantenmechanische Systeme, die sich nur durch eine *globale* Phase unterscheiden, sind physikalisch als identisch zu behandeln. Es ist daher unangebracht, eine phasengenaue Übereinstimmung von $U_{\text{Real}}(\mathbf{x})$ mit U_{Ziel} zu fordern. Dies berücksichtigen wir durch folgende Definition einer neuen Gewinnfunktion

$$\mathfrak{G}(\mathbf{x}) := |\langle U_{\text{Real}}(\mathbf{x}) | U_{\text{Ziel}} \rangle|^2. \quad (2.3)$$

Sie wird maximal für alle $U_{\text{Real}}(\mathbf{x})$, die sich nur durch eine globale Phase von U_{Ziel} unterscheiden. Wir werden später noch weitere Varianten für \mathfrak{G} sehen ((4.11) und (6.40)).

Es sei noch angemerkt, dass

$$\langle A | B \rangle = \sum_{i=1}^n \sum_{j=1}^n A_{ij} \cdot \bar{B}_{ij}. \quad (2.4)$$

Da die Rechenzeit für Matrixmultiplikationen mit der dritten Potenz von n wächst, ist in der numerischen Umsetzung diese, quadratisch mit n in der Rechenzeit wachsende Form der Variante (2.2) vorzuziehen.

2.2 Numerische Suche nach Extrema

Im letzten Abschnitt haben wir gesehen, wie man ein OCT-Problem angeht, indem man das Extremum einer entsprechend gewählten *Kosten-* bzw. *Gewinnfunktion* sucht. In diesem Abschnitt lernen wir Strategien kennen, um Extrema von Funktionen zu finden, die von n Variablen abhängen. Dies ist bereits eine weitreichende Einschränkung für OCT-Probleme. Denn auch wenn wir gewisse *Variablen* des Systems kontrollieren, darf das nicht darüber hinwegtäuschen, dass diese Variablen meistens Funktionen der Zeit sind und der Wert zu jedem Zeitpunkt eine Unbekannte darstellt, die es zu bestimmen gilt. Demnach hätten wir überabzählbar viele Freiheitsgrade. Anders ausgedrückt, wir haben es mit *Kosten-/Gewinn-Funktionalen* und Variationsrechnung zu tun. Wer sich für diesen Ansatz interessiert, sei an [21, 22] verwiesen. Wir werden uns darauf beschränken, die Variablen des Systems nur zu ganz bestimmten Zeitpunkten zu verändern. Die Funktion der Variablen über die Zeit wird also die Form einer Treppenfunktion annehmen mit einer begrenzten Anzahl an verschiedenen Werten. Dieser Ansatz wurde in NMR Experimenten erfolgreich verwendet [20]. Es sei darauf hingewiesen, dass dies keine rein numerische Konzession ist, sondern durchaus der physikalischen Wirklichkeit und den technischen Möglichkeiten entspricht. Wir können in unseren Experimenten nicht beliebig oft und schnell die Eigenschaften des Laserstrahls verändern.

Eine notwendige Bedingung für ein Extremum einer n -dimensionalen Funktion besteht darin, dass alle ersten Ableitungen den Wert Null haben.

Dies ergibt ein Gleichungssystem aus n Gleichungen. Für nichtlineare Gleichungen ist der Versuch, dieses Gleichungssystem direkt zu lösen, meist wenig vorteilhaft, und andere Verfahren sind vorzuziehen.

Eine erfolgreiche Strategie der numerischen Lösungssuche fußt auf der Tatsache, dass es leicht ist, für einen Punkt \mathbf{x} im n -dimensionalen Variablenraum festzustellen, in welche Richtung die Funktion $F(\mathbf{x})$ ansteigt und in welche Richtung sie abfällt. Wenn man von einem gegebenen Punkt \mathbf{x}_0 im Variablenraum ein kleines Stückchen in die richtige Richtung weiter geht, sollte man zu einem neuen Punkt \mathbf{x}_1 gelangen, der dem gesuchten Extremum etwas näher liegt als sein Vorgänger. Diese Prozedur wird nun wiederholt, bis man ein Extremum erreicht hat. Zwei Fragen gilt es dabei zu präzisieren:

1. In welche Richtung bewegen wir uns im Variablenraum?
2. Wie weit bewegen wir uns in besagte Richtung?

Die Antwort auf die erste Frage scheint schnell gefunden. Der Gradient $\nabla F(\mathbf{x}_0)$ zeigt immer in die Richtung des stärksten Anstiegs. Je nachdem, ob wir ein Maximum oder Minimum suchen, bewegen wir uns in die positive oder negative Richtung des Gradienten. Die Antwort auf die zweite Frage, wie weit wir uns zu bewegen haben, ergibt sich aus der Optimierung von

$$F(\mathbf{x}_0 + \alpha \cdot \nabla F(\mathbf{x}_0)). \quad (2.5)$$

Die Optimierung erfolgt nur noch nach einer Variablen: α . Hierfür gibt es verschiedene Strategien: Man kann versuchen, das Extremum durch Intervallschachtelung einzugrenzen, oder die Funktion durch eine Parabel (oder ein höheres Polynom) anzunähern, deren Extremum leicht zu bestimmen ist. Die Parabelnäherung kann aus einer Taylorentwicklung gewonnen werden, oder man bestimmt neben $F(\mathbf{x}_0)$ noch zwei weitere, benachbarte Funktionswerte. Diese drei Punkte legen eine Parabel eindeutig fest.

Wir kommen noch einmal zurück auf die erste Frage, in welche Richtung man sich im Variablenraum bewegen sollte. Der Gradient $\nabla F(\mathbf{x}_0)$ schien die optimale Antwort zu sein. Dass dies aber nicht der Fall ist, erkennen wir, wenn wir mehrere Einzelschritte aneinander reihen. Der erste Optimierungsschritt liefert:

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha \cdot \nabla F(\mathbf{x}_0). \quad (2.6)$$

Nehmen wir an, wir haben α optimal bestimmt. Dann würde jede Veränderung von \mathbf{x}_1 in Richtung von $\nabla F(\mathbf{x}_0)$ das Ergebnis nur verschlechtern. Daraus folgt, dass der für den nächsten Optimierungsschritt benötigte Gradient $\nabla F(\mathbf{x}_1)$ keine Komponente in Richtung des alten Gradienten $\nabla F(\mathbf{x}_0)$ haben kann. Anders gesagt: Aufeinander folgende Gradienten stehen senkrecht aufeinander (optimales α vorausgesetzt). Wenn aber die einzelnen Optimierungsschritte senkrecht aufeinander stehen, heißt dies nichts anderes,

als dass wir uns im Zickzack durch den Variablenraum bewegen – eine Bewegungsart, die kaum als optimal aufgefasst werden kann. Die Methode des konjugierten Gradienten versucht, hier Abhilfe zu schaffen, indem Informationen aus früheren Optimierungsschritten mit berücksichtigt werden. Für weitere Informationen siehe [23, 24]

2.3 Einzel-Variablen-Optimierung (EVO)

Im letzten Abschnitt haben wir die Grundideen zur Extremumsuche einer Funktion $F(\mathbf{x})$ im n -dimensionalen Variablenraum kennen gelernt. In diesem Abschnitt wird der EVO-Algorithmus vorgestellt, der auf einer Einzel-Variablen-Optimierung beruht. Es sei darauf hingewiesen, dass der EVO-Algorithmus im Rahmen dieser Diplomarbeit entwickelt wurde und somit weder „Einzel-Variablen-Optimierung“ noch „EVO-Algorithmus“ Begriffe aus der Fachliteratur sind.

Der EVO-Algorithmus verwendet als Optimierungsrichtung nicht den (konjugierten) Gradienten, sondern immer jeweils eine der Koordinatenachsen. Nach den Ausführungen des Abschnitts 2.2 erscheint dies als keine überaus geschickte Wahl. Bei senkrecht aufeinander stehenden Koordinatenachsen gilt hier in einem noch viel extremeren Maße als beim Gradientenverfahren der Vorwurf, dass dies zu einem Zickzackkurs im Koordinatenraum führt. Im Gegensatz zum Gradientenverfahren wird man aber fast nie in den Genuss kommen, den maximalen Anstieg gewählt zu haben. Wie wir im Folgenden sehen werden, kann der EVO-Algorithmus aber auch mit einigen Vorteilen aufwarten.

Der EVO-Algorithmus ermöglicht die effiziente Verwendung einer im Rahmen dieser Diplomarbeit entwickelten Variante des *Simulated Annealing* zur Auffindung globaler Extrema (Abschnitte 2.5). Darüber hinaus gestattet er eine elegante Einbettung von Nebenbedingungen (Abschnitte 2.4). Es gibt aber noch ein weiteres Argument, das direkt den Kern einer jeden Optimierung betrifft: Die Geschwindigkeit. Mag der Einzelschritt auch viel schlechter sein als etwa beim Verfahren des konjugierten Gradienten, so ist er doch sehr viel schneller berechnet. In der gleichen Rechenzeit viele kleine Schritte zu machen, kann einen durchaus weiterbringen als nur ein großer Schritt.

Die Quelle des Geschwindigkeitszuwachses liegt in der Struktur unseres quantenmechanischen Zeitentwicklungsoperators begründet. Wir möchten einen Zeitentwicklungsoperator $U(t) = U_{\text{Real}}(\mathbf{x})$ optimieren, der sich als Produkt vieler „kleiner“ Zeitentwicklungsoperatoren schreiben lässt. Unser Ansatz sieht wie folgt aus:

$$U(t_n, t_0) = U(t_n, t_{n-1}) \cdot U(t_{n-1}, t_{n-2}) \cdot U(t_{n-2}, t_{n-3}) \cdots U(t_1, t_0). \quad (2.7)$$

Im Folgenden benutzen wir hierfür die verkürzte Schreibweise:

$$U(t) = U_n \cdot U_{n-1} \cdot U_{n-2} \cdots U_1. \quad (2.8)$$

Der Zeitentwicklungsoperator ist in n Intervalle unterteilt. Während eines Intervalls werden die von uns kontrollierten Systemvariablen konstant gehalten. Im Folgenden gehen wir immer davon aus, dass es sich bei diesen Systemvariablen um die Rabi-Frequenz Ω und die Phase ϕ der Laser handelt. Veränderungen dieser Werte finden nur am Übergang von einem zum nächsten Intervall statt

$$t_{i-1} \leq t < t_i \Rightarrow \begin{aligned} \Omega(t) &= \Omega_i \\ \phi(t) &= \phi_i \end{aligned} \quad (2.9)$$

Die Rabi-Frequenzen Ω_i und die Phasen ϕ_i stellen die $2n$ Variablen dar, nach denen optimiert werden soll. Dank der Faktorisierung (2.8) lässt eine Veränderung der Variablen im k ten Intervall alle $U_{i \neq k}$ unverändert, was eine enorme Einsparung beim Rechenaufwand ermöglicht.

$$U(t) = \underbrace{U_n \cdots U_{k+1}}_{U_{\text{Danach}}} \cdot U_k \cdot \underbrace{U_{k-1} \cdots U_1}_{U_{\text{Davor}}} \quad (2.10)$$

Ableitungen nach den Variablen im k ten Intervall lassen sich ebenso zerlegen. Der Vorteil dieser Zerlegung ist, dass wir U_{Danach} und U_{Davor} nicht bei jedem Optimierungsschritt komplett neu durch Multiplizieren aller U_i berechnen müssen, sondern sie ohne viel Aufwand aus ihren Werten vom letzten Optimierungsschritt erhalten. Im Folgenden gehen wir davon aus, dass wir im k ten Optimierungsschritt die Variablen des k ten Intervalls optimieren und U_k neu berechnen. Dieses U_k^{neu} nutzen wir, um $U_{\text{Davor}}^{[k+1]}$ und $U_{\text{Danach}}^{[k+1]}$ zu berechnen, welche wir im $k+1$ ten Optimierungsschritt verwenden². Der hochgestellte, geklammerte Index $[k+1]$ zeigt die Zugehörigkeit zum entsprechenden Optimierungsschritt an. Es gilt

$$U_{\text{Davor}}^{[k+1]} = U_k^{\text{neu}} \cdot U_{\text{Davor}}^{[k]} \quad (2.11)$$

Zur Berechnung von $U_{\text{Danach}}^{[k+1]}$ müssen wir das alte U_{k+1} – das im $k+1$ ten Optimierungsschritt neu berechnet wird – aus $U_{\text{Danach}}^{[k]}$ entfernen.

$$U_{\text{Danach}}^{[k+1]} = U_{\text{Danach}}^{[k]} \cdot U_{k+1}^{\text{alt}^{-1}} = U_{\text{Danach}}^{[k]} \cdot U_{k+1}^{\text{alt}^\dagger} \quad (2.12)$$

Wir haben gesehen, wie man $U(t)$ effizient berechnet. Die eigentliche Aufgabe lautet aber, die Gewinnfunktion und deren Ableitungen zu berechnen. Bei allen in dieser Diplomarbeit verwendeten Gewinnfunktionen lässt sich aber der gleiche Trick in entsprechender Abwandlung anwenden. Für Gewinnfunktionen der Art $\text{tr}(U(t) \cdot U_{\text{Ziel}}^\dagger)$ (2.2) bietet sich noch folgende

²Man erspart sich eine Matrixmultiplikation, wenn man $U_{\text{Davor}}^{[k+1]}$ bereits im k ten Schritt verwendet, um $U(t)^{[k]}$ zu berechnen. Dann wird (2.10) zu $U(t)^{[k]} = U_{\text{Danach}}^{[k]} \cdot U_{\text{Davor}}^{[k+1]}$

Umformung an

$$\begin{aligned}
\langle U(t)|U_{\text{Ziel}} \rangle &= \text{tr}(U(t) \cdot U_{\text{Ziel}}^\dagger) \\
&= \text{tr}(U_{\text{Danach}} \cdot U_k \cdot U_{\text{Davor}} \cdot U_{\text{Ziel}}^\dagger) \\
&= \text{tr}(U_k \cdot U_{\text{Davor}} \cdot U_{\text{Ziel}}^\dagger \cdot U_{\text{Danach}}) \\
&= \text{tr}(U_k \cdot U_{\text{Davor}} \cdot (U_{\text{Danach}}^\dagger \cdot U_{\text{Ziel}})^\dagger) \\
&= \text{tr}(U_k \cdot U_{\text{Davor}} \cdot (U_{\text{Danach}}^{-1} \cdot U_{\text{Ziel}})^\dagger) \\
&= \langle U(t_k, t_0)|U^{-1}(t_n, t_k) \cdot U_{\text{Ziel}} \rangle. \tag{2.13}
\end{aligned}$$

$U_{\text{Danach}}^{-1} \cdot U_{\text{Ziel}} = U^{-1}(t_n, t_k) \cdot U_{\text{Ziel}}$ kann als die in der Zeit rückwärts propagierte Information über den gewünschten Zieloperator aufgefasst werden. Anstatt bei jedem Schritt $U_{\text{Danach}}^{[k+1]}$ zu berechnen, kann man gleich

$$\underbrace{U_{\text{Danach}}^{[k+1]^{-1}} \cdot U_{\text{Ziel}}}_{=: U_{\text{Ziel}}^{[k+1]}} = U_{k+1}^{\text{alt}} \cdot \underbrace{(U_{\text{Danach}}^{[k]^{-1}} \cdot U_{\text{Ziel}})}_{=: U_{\text{Ziel}}^{[k]}} \tag{2.14}$$

bestimmen und so eine Matrixmultiplikation pro Optimierungsschritt einsparen.

In Abschnitt 2.2 haben wir festgestellt, dass wir für einen Optimierungsschritt zwei Informationen benötigen: In welche Richtung wir uns im Variablenraum bewegen und wie weit die Bewegung in besagte Richtung zu erfolgen hat. Als Richtung wählt der EVO-Algorithmus die Koordinatenachsen. Bleibt zu klären, wie weit die Bewegung zu erfolgen hat. Dazu nähern wir die zu optimierende Gewinnfunktion für jede Variable via Taylor-Entwicklung durch eine Parabel an. Der optimale Wert der Variable ergibt sich dann aus der Lage des Scheitelpunktes. Damit hat die Näherungsparabel aber noch nicht ausgedient. Ihre Kenntnis wird uns auch in den kommenden beiden Abschnitten wertvolle Hilfe leisten, wenn es darum geht Nebenbedingungen (Abschnitt 2.4) einzubauen und durch Simulated Annealing globale Extrema aufzuspüren (Abschnitt 2.5). Da der EVO-Algorithmus in gleicher Zeit viel mehr Optimierungsschritte absolviert als z.B. das Gradientenverfahren, erzeugt er auch viel mehr Näherungsparabeln, was die Qualität des Simulated Annealing deutlich verbessern sollte.

Für den seltenen Fall, dass die Optimierung mittels Näherungsparabel nicht klappt, sollte es immer noch einen „Notfallplan“ geben. Dieser besteht einfach darin, eine Standarddistanz in die richtige Richtung zu gehen (Die Richtung ergibt sich aus dem Gradienten). Sollte dies zu einer Verschlechterung führen, die größer als die Toleranz ist, wird die Distanz so lange halbiert, bis eine Verbesserung eintritt bzw. die Verschlechterung kleiner als die Toleranz ist. Dieser simple Notfallplan reicht vollkommen aus, denn er wird fast nie benötigt (die Parabelnäherung ist erstaunlich gut).

Vergleichen wir abschließend die Rechenleistung des EVO-Algorithmus mit der des Gradientenverfahrens. Letzteres muss jeweils n einzelne U_i, U'_i und je nach Algorithmus U''_i für einen Optimierungsschritt bestimmen. Mit dem gleichen Aufwand (plus zusätzliche Multiplikationen) schafft der EVO-Algorithmus n Optimierungsschritte. Andererseits optimiert der EVO-Algorithmus auch nur eine Variable pro Optimierungsschritt, während das Gradientenverfahren n Variablen auf einmal anpasst. Für diesen einen großen Optimierungsschritt benutzt das Gradientenverfahren nur eine einzige Näherungsparabel, während der EVO-Algorithmus für jede Variable eine individuelle, maßgeschneiderte Parabel berechnet. Dafür ist der EVO-Algorithmus wiederum blind für positive Effekte, die sich aus der gemeinsamen Anpassung mehrerer Variablen ergeben. Ein entscheidender Rechenvorteil des EVO-Algorithmus besteht in der viel schnelleren Einbeziehung von Informationen aus vorangegangenen Optimierungsschritten. Warum berechnet das Gradientenverfahren denn bei jedem Optimierungsschritt einen neuen Gradienten? Einfach deshalb, weil der Informationsgehalt des letzten Gradienten sehr schnell aufgebraucht ist, wenn die Variablen verändert werden. Diesem Informationsverlust wirkt der EVO-Algorithmus entgegen, indem er n Updates vornimmt, wo das Gradientenverfahren nur eins schafft.

Eine auf den Erkenntnissen der Praxis beruhende Bewertung der Qualität des EVO-Algorithmus muss hier jedoch ausbleiben, obwohl auch Erfahrungen mit einem anderen Algorithmus vorliegen. Im Rahmen dieser Diplomarbeit wurde bis zum vierten Monat der auf dem Gradientenverfahren beruhende GRAPE-Algorithmus [20] verwendet. Da bei der anschließenden Umstellung auf den EVO-Algorithmus aber auch viele andere Routinen geändert wurden, ist ein objektiver Vergleich beider Algorithmen nicht möglich.

2.4 Nebenbedingungen

In Abschnitt 2.1 haben wir gesehen, wie man ein OCT-Problem löst, indem man eine entsprechende Gewinnfunktion definiert. Mit (2.3) wurde auch schon eine für unsere Belange geeignete Gewinnfunktion vorgestellt. Die Optimierung dieser Gewinnfunktion stellt aber nur sicher, dass der Zeitentwicklungsoperator des Systems einem vorgegebenen Gatter entspricht – mehr nicht. Eine gute Lösung des OCT-Problem sollte aber noch ein paar zusätzliche Anforderungen erfüllen, wie z.B. die Möglichkeit einer problemlosen technischen Umsetzung. Diese zusätzlichen Anforderungen müssen wir in Form von Nebenbedingungen einfügen [23].

Ein Standardverfahren zur analytischen Optimierung mit Nebenbedingungen besteht in der Einführung *Lagrangischer Multiplikatoren*. Dieses Verfahren ist auch auf die Numerik ausdehnbar. Dies setzt aber Nebenbedingungen in Form von Gleichungen voraus (mit einem kleinen Trick sind sogar Un-

gleichungen implementierbar). Oft hat man es aber mit Anforderungen der Art „*Erreiche folgendes Ziel mit möglichst wenig Aufwand*“ zu tun. „...*mit möglichst wenig Aufwand*“ ist eine Nebenbedingung, die meist nicht als Gleichung darstellbar sein wird. Nebenbedingungen dieser Art bieten sich aber an, selbst einem Optimierungsprozess unterworfen zu werden. Ein Standardverfahren besteht darin, die Nebenbedingung in Form einer *Strafffunktion* \mathfrak{S} zu formulieren. Diese Strafffunktion sollte die Eigenschaft haben, umso größer auszufallen, je schlechter die Nebenbedingungen erfüllt sind. Optimiert wird dann die Differenz aus Gewinnfunktion minus Strafffunktion. Der Nachteil dieses Verfahrens ist aber, dass am Ende weder das eigentliche OCT-Problem noch die Nebenbedingung für sich allein genommen optimal sein werden, sondern nur der Kompromiss aus beiden. Ein Kompromiss mag in vielen Fällen ein sinnvolles Ergebnis sein – für unseren Fall ist das jedoch nicht so. Unsere Nebenbedingungen sind eher grobe Richtlinien als zwingende Erfordernisse. Dies gilt sowohl für die Nebenbedingung, die in Abschnitt 4.1 zum Einsatz kommt, als auch für die aus Abschnitt 5.2. Da letztere aber wenig anschaulich ist, werden wir hier die Nebenbedingung aus Abschnitt 4.1 als Beispiel anführen, welche auf die Werte der Rabifrequenz Ω der Laserstrahlen abzielt. Die Rabifrequenzen Ω nach oben zu begrenzen, ist physikalisch sicher sinnvoll, so dass die Summe der Quadrate der Rabifrequenzen Ω eine brauchbare Strafffunktion \mathfrak{S} darstellt

$$\text{Strafe} = \mathfrak{S} = \sum_{i=1}^n \Omega_i^2, \quad (2.15)$$

wobei Ω_i der Wert von Ω im i ten Zeitintervall ist (2.9). Als noch sinnvollere Nebenbedingung erwies sich die Forderung nach einem glatten Kurvenverlauf für die Rabifrequenzen („glatt“ ist hier umgangssprachlich aufzufassen – die Rabifrequenz ist vom Ansatz her eine Treppenfunktion (2.9)). Konkret lautet die Forderung, dass die Rabifrequenz Ω_i sich idealerweise als Mittelwert ihrer beiden zeitlichen Nachbarn ergeben sollte

$$\mathfrak{S} = \sum_{i=1}^n (2\Omega_i - (\Omega_{i-1} + \Omega_{i+1}))^2 \quad \text{mit} \quad \Omega_0 = \Omega_{n+1} = 0. \quad (2.16)$$

Nur wenn der zeitliche Verlauf der Rabifrequenz eine konstante Steigung aufweist, wird durch (2.16) keine Strafe vergeben. Da die Randpunkte Ω_0 & Ω_{n+1} aber auf null festgelegt sind, müssten unsinnigerweise alle $\Omega_i = 0$ werden, damit die Strafffunktion \mathfrak{S} vollständig verschwindet. Es kann also gar nicht unser Ziel sein, der Strafffunktion optimal Genüge zu tun. Deshalb ist ein Kompromiss aus optimaler Maximierung der Gewinnfunktion und optimaler Minimierung der Strafffunktion für uns auch nicht erstrebenswert. Darauf zu verzichten, das primäre Optimierungsziel bestmöglich zu lösen, um als Gegenleistung etwas bessere Nebenbedingungen zu erhalten, ist kein

guter Tausch. Unsere Aufgabenstellung könnte man daher wie folgt verbalisieren: „Löse das primäre Problem so gut es geht. Unter allen gleichwertigen Lösungen wähle diejenige aus, die den Nebenbedingung am besten entspricht.“ Der Trick besteht nun in der Art der Einbettung der Nebenbedingungen. Die Straffunktion \mathfrak{S} wird nicht einfach von der Gewinnfunktion \mathfrak{G} des Primärzieles subtrahiert, sondern zuvor mit einem Kopplungsparameter α multipliziert, der die relative Wichtigkeit der Nebenbedingung repräsentiert. Wir suchen also das Maximum

$$\max(\mathfrak{G} - \alpha \cdot \mathfrak{S}). \quad (2.17)$$

Da wir α im Laufe der Optimierung anpassen, verändert sich auch die Lage des Maximums. Anfangs, wenn die Veränderungen noch groß sind, wird der Nebenbedingung via α eine große Bedeutung beigemessen. Je mehr die Optimierung sich dem Idealwert nähert, desto kleiner wird der Kopplungsparameter α gewählt, so dass wir asymptotisch auf ein Maximum von \mathfrak{G} allein zusteuern und das Primärziel optimal erfüllen. Da zum Ende hin die Veränderungen nur noch klein ausfallen, sollte sich im Kurvenverlauf der Ω_i immer noch die zuvor betriebene Optimierung unter starker Einbindung der Nebenbedingungen bemerkbar machen.

Dieses Rezept mag einfach klingen, aber es ist gar nicht so leicht in die Praxis umzusetzen: Woher soll der Computer wissen, wann sich die Optimierung dem Idealwert wie weit genähert hat und was der angebrachteste Wert für den Kopplungsparameter α ist? Sicherlich gibt es dafür einige Indikatoren. Die durchschnittliche Verbesserung pro Rechenschritt könnte z.B. herangezogen werden. Diese wird sicherlich immer kleiner, je näher das Optimum rückt. Leider sind aber auch andere Ursachen möglich.

Im Folgenden wird die beste im Rahmen dieser Diplomarbeit ersonnene Methode zur Bestimmung des Kopplungsparameters α erklärt. Hierbei wird das $\alpha \cdot \mathfrak{S}$ in (2.17) als störender Term aufgefasst, der die optimale Suche nach dem Maximum von \mathfrak{G} behindert. Ziel ist es, durch Anpassung von α diese Behinderung während des gesamten Optimierungsprozesses auf einem konstanten Wert zu halten. D.h. wir erlauben dem Computer z.B. 70% der möglichen Verbesserung von \mathfrak{G} pro Optimierungsschritt zu opfern, um dafür den Nebenbedingungen besser nachzukommen. Das Opfern der 70% führt nur zu einer Verlangsamung des Optimierungsprozesses, aber nicht zu einer Verschlechterung des Endergebnisses. Am Ende erhalten wir ein pures Maximum von \mathfrak{G} , das den Nebenbedingungen weitestgehend entgegenkommt. Die Frage ist nun, wie wir die Stärke des störenden Einflusses von $\alpha \cdot \mathfrak{S}$ bestimmen. Dies geschieht am effizientesten, wenn wir auf die Näherungsparabel zurückgreifen, die bei der Optimierung der Gewinnfunktion \mathfrak{G} berechnet wird (siehe Abschnitt 2.3). Wir bestimmen diese Näherungsparabel auch weiterhin nur für \mathfrak{G} allein ohne Einfluss der Straffunktion \mathfrak{S} . Diese Näherungsparabel beschreibt die Gewinnfunktion, wenn nur eine Variabel als

veränderlich angenommen wird. Diese Variable sei Ω_i und $P_{\mathfrak{G}}(\Omega_i)$ bezeichne die Näherungsparabel. Wenn wir nur Ω_i als Veränderliche zulassen, so reduziert sich auch die Straffunktion \mathfrak{S} in (2.16) auf eine Parabel $P_{\mathfrak{S}}(\Omega_i)$. Mittels dieser beiden Parabeln lassen sich folgende Werte bestimmen:

1. Der maximale Wert der Gewinnfunktion ohne Nebenbedingung $\max(P_{\mathfrak{G}}(\Omega_i))$
2. Die Lage $x_{\mathfrak{G}-\alpha \cdot \mathfrak{S}}$ – d.h. der Wert der Abszisse, nicht der Ordinate – des maximalen Werts für die Kombination $P_{\mathfrak{G}} - \alpha \cdot P_{\mathfrak{S}}$. Dieses $x_{\mathfrak{G}-\alpha \cdot \mathfrak{S}}$ stellt den neuen Wert für die Variable Ω_i dar.
3. Der Wert $P_{\mathfrak{G}}(\Omega_i^{\text{neu}})$ der reinen Gewinnfunktion ohne Nebenbedingung für $\Omega_i^{\text{neu}} = x_{\mathfrak{G}-\alpha \cdot \mathfrak{S}}$

Mit $\max(P_{\mathfrak{G}}(\Omega_i))$ haben wir die maximal mögliche Verbesserung der Gewinnfunktion in diesem Optimierungsschritt bestimmt und dank $P_{\mathfrak{G}}(\Omega_i^{\text{neu}})$ kennen wir die effektive Verbesserung (oder auch Verschlechterung) der Gewinnfunktion bei aktiver Nebenbedingung. Daraus können wir leicht den störenden Einfluss von $\alpha \cdot \mathfrak{S}$ auf die Optimierung von \mathfrak{G} bestimmen und daraufhin die Größe von α anpassen. Diese Anpassung sollte hinreichend gedämpft erfolgen, damit α nicht zu oszillieren anfängt.

Neben der Formulierung der Nebenbedingung als Straffunktion wurden auch komplett andere Wege besprochen. Da das Ziel der Nebenbedingung (2.16) ein glatter Kurvenverlauf ist, lag es nahe, gänzlich auf das Konzept der Straffunktion zu verzichten und dafür am Ende eines jeden Optimierungsschrittes einen Algorithmus zur Kurvenglättung aufzurufen. Dieser Glättungsalgorithmus muss *idempotent* konstruiert werden, wie in der Fußnote³ erklärt wird.

Letztlich hat sich die Optimierung via Straffunktion aber allen anderen Verfahren gegenüber als überlegen erwiesen. Ein weiteres Plus ist die große Flexibilität. In Abschnitt 5.2 werden z.B. gänzlich andere Nebenbedingungen vonnöten sein. Die einfache Ersetzung der Straffunktion (2.16) durch (5.9) reicht aus, um die Anpassung vorzunehmen.

³Wenn die Veränderungen durch die Optimierung der Gewinnfunktion gegen null gehen, müssen die Veränderungen des Glättungsalgorithmus dies ebenfalls tun (sonst konvergiert der Algorithmus nicht). Ein nichtidempotenter Operator bewirkt aber auch dann Veränderungen, wenn er ein zweites Mal hintereinander Einsatz findet. Dieser Fall läge vor, wenn die Optimierung der Gewinnfunktion keine Veränderungen bewirkt hat – also in einem der gesuchten Maxima. Der Glättungsalgorithmus würde also dafür sorgen, dass die Gewinnfunktion nicht im gesuchten Maximum bleiben kann bzw. es nie erreicht.

2.5 Lokale und globale Extrema

Die bisher beschriebenen Algorithmen vermögen nicht, lokale und globale Extrema zu unterscheiden. Man könnte sogar sagen, dass diese Algorithmen die Eigenschaft haben, von lokalen Extrema gefangen zu werden. Einmal in einem Extremum gefangen, gibt es für einen Algorithmus, der nie weiter als einen Schritt vorausschaut und bei jedem Schritt eine Verbesserung erwartet, kein Entrinnen. Nur ein Algorithmus, der auch mal eine Verschlechterung akzeptiert, kann einem lokalen Extremum enttrinnen. Ein Algorithmus, der gleichermaßen gute wie schlechte Schritte akzeptiert, tappt aber nur blind umher. Es gilt hier, einen guten Kompromiss zu finden. Ein Beispiel, wie so ein Algorithmus aussehen kann, finden wir in der Natur beim Kristallisationsprozess verwirklicht. Verläuft die Abkühlung zu schnell, wird man einen eher ungeordneten Kristall erhalten. Verläuft die Abkühlung hingegen sehr langsam, hat man eine gute Chance, einen über weite Bereiche wohl geordneten Kristall zu bekommen. In beiden Fällen sucht der Kristall ein Minimum der (freien) Energie auf. Im ersten Fall ist dies ein „minderwertiges“ lokales Minimum, während im zweiten Fall eine starke Annäherung an das globale Minimum stattgefunden hat. Wir könnten versuchen, diesen Kristallisationsprozess auf dem Computer zu simulieren, indem wir bei jedem Optimierungsschritt einem Atom gestatten, sein lokales Energieminimum anzunehmen. Diese Simulation wäre aber nur für eine Temperatur von null Kelvin adäquat. Bei höheren Temperaturen erlaubt die Natur den Atomen auch Plätze einzunehmen, die vom Minimum abweichen. Diese Abweichung wird jedoch mit einem Boltzmannfaktor unterdrückt.

Im Metropolis-Algorithmus für Monte Carlo Simulationen wurde diese Idee zum ersten Mal aufgegriffen. Heute laufen derartige Algorithmen unter der Bezeichnung *Simulated Annealing* [23]. Einsatz finden sie hauptsächlich bei diskreten Problemen, aber sie sind auch bei kontinuierlicher Optimierung verwendbar. Die Grundstrategie sieht wie folgt aus:

1. Wähle vom Startpunkt \mathbf{x}_0 aus zufällig einen benachbarten Punkt \mathbf{x}_1 und berechne den dortigen Funktionswert $F(\mathbf{x}_1)$.
2. Ist der neue Funktionswert besser, dann akzeptiere \mathbf{x}_1 als neuen Startpunkt. Ist der neue Wert schlechter, dann akzeptiere \mathbf{x}_1 trotzdem, wenn

$$\text{Zufallszahl}(0 \dots 1) < \exp(-|F(\mathbf{x}_1) - F(\mathbf{x}_0)|/T). \quad (2.18)$$

Anderenfalls bleibe beim alten Startpunkt \mathbf{x}_0

3. Senke eventuell die Temperatur T leicht ab und gehe wieder zu Punkt 1.

Die Nachteile dieses Verfahrens sind offensichtlich: Die Wahl des Punktes \mathbf{x}_1 wird dem Zufall überlassen. Auch wenn der Weg zur Verbesserung ganz eindeutig ist, werden immer wieder falsche Routen eingeschlagen. Trotz dieser

offensichtlichen Mängel ist dies einer der besten Algorithmen zum Auffinden globaler Extrema, so dass es nicht überrascht, dass es inzwischen viele verbesserte Varianten gibt.

Im Rahmen dieser Diplomarbeit wurde eine neue Variante entwickelt, die den Textbuchvarianten aus [23, 24] dahin gehend überlegen ist, dass sie gegen den zuvor beschriebenen, gezielt optimierenden EVO-Algorithmus (Abschnitt 2.3) konvergiert, wenn die Temperatur absinkt (s.u.). Darüber hinaus wird nur in Ausnahmefällen ein einmal berechneter Wert zu verwerfen sein. Zum besseren Verständnis der zugrunde liegenden Idee dieser neuen Variante, betrachten wir das Problem, das Verhalten eines Atoms beim Kristallisationsprozess auf Basis der klassischen Physik simulieren:

Als erstes nähern wir das Potential, in welchem das Atom sich befindet, durch eine Parabel P an. Damit können wir jedem Ort x via Boltzmannverteilung eine Besetzungswahrscheinlichkeit zuweisen

$$\rho(x) = \frac{1}{Z} \exp\left(-\frac{P(x) - P_{\min}}{k \cdot T}\right) \quad Z = \text{Zustandssumme.} \quad (2.19)$$

Als nächsten Schritt invertieren wir die Boltzmannverteilung. Da x in Form der Parabel $P(x)$ in $\rho(x)$ eingeht, können wir zu jedem gegebenen ρ zwei Ortskoordinaten $x_1(\rho)$ und $x_2(\rho)$ benennen, von denen wir eine per Zufall auswählen. Welches ρ wir zur Berechnung von $x(\rho)$ verwenden, soll ebenfalls durch eine Zufallsentscheidung bestimmt werden. Da ρ eine Wahrscheinlichkeit ist, darf es den Wert 0,9 aber nicht mit der gleichen Häufigkeit annehmen wie den Wert 0,01. Die Zufallsentscheidung muss sicherstellen, dass ρ den Wert ϱ_{Zufall} genau mit der Wahrscheinlichkeit ϱ_{Zufall} annimmt. Dieses Problems nehmen wir uns in (2.5.1) an und gehen hier davon aus, dass wir über derartige ϱ_{Zufall} verfügen. Damit können wir

$$\rho := \varrho_{\text{Zufall}} \quad (2.20)$$

setzen und daraus den neuen Ort des Atoms $x(\varrho_{\text{Zufall}})$ bestimmen. Die Wahrscheinlichkeit für ein so berechnetes x entspricht jetzt genau der Boltzmannverteilung (2.19).

Dieses, für ein Atom beschriebenes Verfahren lässt sich leicht auf unser Optimierungsproblem übertragen. Zu jeder Optimierung einer Variablen berechnen wir eine Näherungsparabel (Hierbei handelt es sich um die Näherungsparabel für die Kombination [Gewinnfunktion $- \alpha \cdot$ Straffunktion] – siehe Abschnitt 2.4). Statt nun wie bisher immer den Scheitelpunkt der Parabel als neuen Wert der Variablen zu nehmen, wählen wir den neuen Wert jetzt mittels einer Zufallszahl gemäß dem oben beschriebenen Verfahren.

Heben wir noch einmal den Hauptunterschied der hier vorgestellten, neuen Variante des Simulated Annealing zum Standardverfahren hervor: Das Standardverfahren wählt erst per Zufall einen Koordinatenpunkt aus, um dann den dazugehörigen Funktionswert zu errechnen und über die Boltzmannverteilung zu bewerten und eventuell zu verwerfen. Die neue Variante hingegen

wählt erst per Zufall eine Wahrscheinlichkeit, gewinnt daraus einen Funktionswert und errechnet damit den neuen Koordinatenpunkt. Dieser Wert wird nie verworfen, es sei denn, der Funktionswert erweist sich als viel schlechter, als die Parabelnäherung es erwarten ließ. Dies passiert aber fast nie⁴. Durch Absenken der Temperatur können wir erreichen, dass nur noch Werte ausgewählt werden, die dicht am Optimum liegen. Damit geht die neue Variante des Simulated Annealing für $T \rightarrow 0$ in den EVO-Algorithmus aus Abschnitt 2.3 über, womit die oben erwähnte Konvergenz zum gezielt optimierenden Algorithmus gezeigt ist.

Es bleibt noch zu klären, welche Temperatur zu wählen ist und wie die Abkühlung erfolgen soll. Die Antworten auf diese Fragen hängen vom Problem ab. Die Optimierung bzw. Auffindung von Impulssequenzen (Kapitel 5 & 6) erforderte z.B. oft das direkte Verändern der Temperatur per Hand. Die generelle Idee besteht jedoch in einem langsamen Abklingen der Temperatur. Dieses Abklingen geschah anfangs noch exponentiell. Dies erwies sich jedoch nicht als ideal, und wurde deshalb durch eine Feedbackschleife ersetzt. Die Temperatur wird dann so geregelt, dass die durchschnittliche Verbesserung in einem festen Verhältnis zur durchschnittlichen Schwankung der Funktionswerte steht. Je näher das (hoffentlich globale) Optimum rückt, desto kleiner werden die Verbesserungen, und die Temperatur wird abgesenkt, was sich wiederum in kleineren Schwankungen widerspiegelt. Sollte man den Verdacht hegen, in einem lokalen Extremum gefangen zu sein, muss man einem guten Schmied gleich die Temperatur von Hand wieder erhöhen.

Das hier vorgestellte Verfahren zum Simulated Annealing ist nicht zwangsläufig an den in Abschnitt 2.3 präsentierten EVO-Algorithmus gebunden. Es ist aber sehr fraglich, ob z.B. bei einer Optimierung entlang des Gradienten ähnliche Leistungen erzielt werden können. Der EVO-Algorithmus ist sehr schnell und produziert viele kleine Schritte. Hierbei unterliegt jede Variable – also jeder Freiheitsgrad – einer *individuellen* thermischen Schwankung. Beim Gradientenverfahren würden die Schwankungen immer nur im Gleichtakt, in Richtung des Gradienten erfolgen.

2.5.1 Generierung zufälliger Wahrscheinlichkeiten

Im letzten Abschnitt sollte der Zufallsgenerator des Computers eine Wahrscheinlichkeit $\rho(x)$ auswählen. Die Wahrscheinlichkeit, mit der $\rho(x)$ den Wert ϱ_{Zufall} annimmt, sollte dabei genau ϱ_{Zufall} sein. In diesem Abschnitt wird gezeigt, wie ein derartiges ϱ_{Zufall} erzeugt wird.

Bevor wir dieses Problem angehen, ändern wir noch die Normierung der

⁴Anfangs diente noch ein Polynom dritten Grades zur Approximation der Funktion. Es erwies sich aber, dass eine Parabel genügt

Wahrscheinlichkeiten

$$\rho(x) \rightarrow \frac{\rho(x)}{\max(\rho(x))} \quad \Rightarrow \quad \rho \in]0 \cdots 1], \quad (2.21)$$

d.h. $\rho(x)$ ist jetzt eine *relative* Wahrscheinlichkeit (Wir werden den Zusatz „relativ“ aber nicht weiter mitführen). Dies entspricht dem Wegfall der Zustandssumme Z in der Boltzmannverteilung (2.19).

Wir gehen im Folgenden davon aus, dass der Computer Zufallszahlen $z \in]0 \cdots 1]$ erzeugt⁵, die einer Gleichverteilung folgen. Damit gilt für die Wahrscheinlichkeit P , die Zufallszahlen z_1 oder z_2 zu erhalten

$$P(z_1) = P(z_2) \quad \forall z_1, z_2. \quad (2.22)$$

Wir hingegen möchten Zufallszahlen $\varrho \in]0 \cdots 1]$ generieren, die folgende Eigenschaften haben

$$\frac{P(\varrho_1)}{P(\varrho_2)} = \frac{\varrho_1}{\varrho_2} \quad \forall \varrho_1, \varrho_2. \quad (2.23)$$

Der Wert $\varrho_1 = 0,9$ sollte z.B. neunmal so häufig vorkommen wie $\varrho_2 = 0,1$. Ein derartiges ϱ kann erstaunlich einfach aus den gleichverteilten Zufallszahlen z (2.22) gewonnen werden

$$\varrho = \sqrt{z}. \quad (2.24)$$

Der Rest dieses Abschnitts dient dem Beweis dieser Behauptung.

Es ist trivial zu sehen, dass $]0 \cdots 1] \rightarrow]0 \cdots 1]$ unter (2.24). Eine Aussage über die Wahrscheinlichkeit von ϱ lässt sich gewinnen, indem wir $P(\varrho)$ als Wahrscheinlichkeitsdichte erachten und die Abbildungen des infinitesimalen Intervalls $[z \cdots z + \varepsilon]$ unter (2.24) für zwei verschiedenen z_1, z_2 betrachten

$$\begin{aligned} [z_1 \cdots z_1 + \varepsilon] &\rightarrow [\varrho_1 \cdots \varrho_1 + \epsilon_1] \\ [z_2 \cdots z_2 + \varepsilon] &\rightarrow [\varrho_2 \cdots \varrho_2 + \epsilon_2]. \end{aligned} \quad (2.25)$$

Für die Wahrscheinlichkeitsdichten gilt $P(\varrho_1) \sim \frac{\varepsilon}{\epsilon_1}$ und $P(\varrho_2) \sim \frac{\varepsilon}{\epsilon_2}$. Da die z gleichverteilt sind und in beiden Fällen ein Intervall der gleichen Länge ε abgebildet wird, ist

$$\frac{P(\varrho_1)}{P(\varrho_2)} = \frac{\epsilon_2}{\epsilon_1}. \quad (2.26)$$

⁵Normalerweise erzeugen Computer Zufallszahlen $z \in [0 \cdots 1[$. Dann ist aber $\mathfrak{z} \in]0 \cdots 1]$ mit $\mathfrak{z} := 1 - z$

Nun lassen sich ϵ_1 und ϵ_2 mittels (2.24) leicht aus ϵ bestimmen

$$\begin{aligned}
 \epsilon_{1,2} &= (\varrho_{1,2} + \epsilon_{1,2}) - \varrho_{1,2} \\
 &= \sqrt{z_{1,2} + \epsilon} - \sqrt{z_{1,2}} \\
 &= \sqrt{z_{1,2}} \cdot \sqrt{1 + \frac{\epsilon}{z_{1,2}}} - \sqrt{z_{1,2}} \\
 &\approx \sqrt{z_{1,2}} \cdot \left(1 + \frac{\epsilon}{2 \cdot z_{1,2}}\right) - \sqrt{z_{1,2}} \\
 &= \sqrt{z_{1,2}} \frac{\epsilon}{2 \cdot z_{1,2}} \\
 &= \frac{\epsilon}{2 \cdot \sqrt{z_{1,2}}} \\
 &= \frac{\epsilon}{2 \cdot \varrho_{1,2}}.
 \end{aligned} \tag{2.27}$$

Setzen wir dies in (2.26) ein, erhalten wir wie gefordert

$$\frac{P(\varrho_1)}{P(\varrho_2)} = \frac{\varrho_1}{\varrho_2}, \tag{2.28}$$

was den Beweis abschließt.

Kapitel 3

Zeitentwicklung

In Kapitel 2 wurde in allgemeiner Form beschrieben, wie ein Optimierungsproblem angegangen werden kann. Nun wenden wir uns den Spezifikationen unseres ersten Problems zu: Der Optimierung eines $\sigma_z \otimes \sigma_z$ - bzw. Mølmer-Sørensen-Gatters auf zwei Ionen. Die Bestimmung des quantenmechanischen Zeitentwicklungsoperators erweist sich hierbei als die zeitintensivste Berechnung. Das ganze Kapitel 3 über werden wir uns damit zu beschäftigen haben, wie wir diese Aufgabe lösen wollen. Im Laufe der Jahre wurde hierfür eine Vielzahl an Werkzeugen entworfen. In unserem Fall kommt allerdings als neuer Aspekt hinzu, dass sehr oft einander ähnliche Rechnungen durchgeführt werden müssen. Das zentrale Anliegen besteht deshalb darin, Rechnungen wiederzuverwenden, um dadurch Ressourcen einzusparen.

3.1 Die Struktur des Zeitentwicklungsoperators

Im Abschnitt 2.3 wurde die Zeitentwicklung unseres Optimierungsproblems durch den allgemeinen Ansatz

$$U(t_n, t_0) = U(t_n, t_{n-1}) \cdot U(t_{n-1}, t_{n-2}) \cdot U(t_{n-2}, t_{n-3}) \cdots U(t_1, t_0) \quad (3.1)$$

beschrieben. Jetzt gilt es, den $U(t_i, t_{i-1})$ eine konkrete Gestalt zu geben. Bevor wir uns den Ionen in einer Ionenfalle zuwenden, werfen wir zuerst einen kurzen Blick auf bereits bestehende Lösungsmethoden. In der Chemie, wo es darum geht, Moleküle maßzuschneidern, wird das OCT Problem meist in Form einer Variationsrechnung gelöst. Eine Übernahme dieser Methoden ist für unsere Zwecke nicht ideal. Sehr viel größere Ähnlichkeit entdecken wir bei NMR Problemen. Hier lässt sich der (effektive) Hamilton-Operator in folgender Form schreiben [20]

$$H(t) = H_0 + \sum_{i=1}^m \alpha_i(t) \cdot H_i. \quad (3.2)$$

Die H_i werden als zeitunabhängig angenommen – die ganze Zeitabhängigkeit steckt in den Parametern $\alpha_i(t)$. Die $\alpha_i(t)$ sind zugleich die kontrollierten Variablen, die es optimal anzupassen gilt. Gemäß unserem Ansatz (siehe Abschnitt 2.3) verändern wir diese Variablen nur an den Intervallgrenzen und halten sie sonst konstant

$$\alpha_i(t) \rightarrow \alpha_{i,j} = \text{const} \quad \text{für } t_{j-1} \leq t < t_j. \quad (3.3)$$

Damit gewinnen wir für $U(t_j, t_{j-1})$ den einfachen Ausdruck

$$U(t_j, t_{j-1}) = e^{-\frac{i}{\hbar}(H_0 + \sum_{i=1}^m \alpha_{i,j} \cdot H_i) \cdot (t_j - t_{j-1})}. \quad (3.4)$$

Die Ableitungen von $U(t_j, t_{j-1})$ hingegen gestalten sich etwas komplizierter, da die H_i in der Regel nicht untereinander kommutieren. Für ein hinreichend kleines $\Delta t = t_j - t_{j-1}$ wird man diesen Fehler aber tolerieren können, da die Ableitungen nur zum Abschätzen der Optimierungsschritte benötigt werden.

Bei dem zweiten in dieser Diplomarbeit behandelten physikalischen Problem – dem Auffinden von Impulssequenzen – werden wir es mit Hamilton-Operatoren zu tun haben, die etwas einfacher sind als (3.2). Bei dem ersten physikalischen Problem – dem Optimieren einer einzelnen $\sigma_z \otimes \sigma_z$ - oder Mølmer-Sørensen-Gatteroperation – haben wir es allerdings mit komplizierteren Hamilton-Operatoren zu tun. Diese Hamilton-Operatoren haben die Form

$$H_j(t) = H_0 + \sum_{i=1}^m \alpha_{i,j} \cdot H_i(t). \quad (3.5)$$

(Zur Erinnerung: H_i ist eine der Komponenten des vollständigen Hamilton-Operators. H_j ist der vollständige Hamilton-Operator im Zeitintervall j .) Für den Rest dieses Kapitels werden wir uns der Aufgabe widmen, den aus (3.5) resultierenden Zeitentwicklungsoperator zu berechnen. Der entscheidende (und lästige) Unterschied zu (3.2) ist die Zeitabhängigkeit der H_i , die zudem noch so stark ausfällt, dass eine Entwicklung viele Ordnungen umfassen muss. Die gute Nachricht ist, dass wir die kontrollierten Variablen wieder in Form von $\alpha_{i,j}$ vor die Hamilton-Operatoren ziehen können, was keine Selbstverständlichkeit ist. Des Weiteren können wir in ein System transformieren, in dem H_0 verschwindet.

3.2 Recycling

Im Laufe der Optimierung müssen die $U_j = U(t_j, t_{j-1})$ aus (3.1) viele tausend Male berechnet werden. Das einzige, worin sich die Berechnungen unterscheiden, sind die Werte der $\alpha_{i,j}$ (3.5). Können diese Berechnungen ökonomisiert werden, indem man eine Art Prototyp dieser Gleichung löst, auf den dann immer wieder zurückgegriffen werden kann? Wünschenswert wäre es

z.B., einen effektiven, zeitunabhängigen Hamilton-Operator zu formulieren. Dies ist möglich, jedoch wird das Ergebnis von den $\alpha_{i,j}$ abhängen

$$H_{\text{eff}}(\alpha_{i,j}) \neq \alpha_{i,j} \cdot H_{\text{eff}}. \quad (3.6)$$

$H_{\text{eff}}(\alpha_{i,j})$ könnte man nun einer Taylorentwicklung nach den $\alpha_{i,j}$ unterwerfen. Dieses Ergebnis erhält man auch nach einer Magnusentwicklung [25, 26], da die $\alpha_{i,j}$ vor die dort auftretenden Integrale gezogen werden können.

$$\begin{aligned} H_{\text{eff}}^{\text{Magnus}} = & \int_{t_0}^t dt_1 H(t_1) - \frac{i}{2\hbar} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 [H(t_1), H(t_2)] \\ & - \frac{1}{6\hbar^2} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 \int_{t_0}^{t_2} dt_3 \left([H(t_1), [H(t_2), H(t_3)]] \right. \\ & \quad \left. + [[H(t_1), H(t_2)], H(t_3)] \right) \\ & + \dots \end{aligned} \quad (3.7)$$

Wenn man aber schon bereit ist, eine Entwicklung nach den $\alpha_{i,j}$ durchzuführen, stellt sich die Frage, ob es nicht geschickter wäre, gleich den Zeitentwicklungsoperator $U_j(\alpha_{i,j})$ zu entwickeln, anstatt den Umweg über die $H_{\text{eff}}(\alpha_{i,j})$ zu gehen. Dieses Vorgehen entspricht einer Dysonentwicklung.

$$U(t, t_0) = 1 + \sum_{n=1}^{\infty} \left(-\frac{i}{\hbar}\right)^n \int_{t_0}^t dt_1 H(t_1) \int_{t_0}^{t_1} dt_2 H(t_2) \cdots \int_{t_0}^{t_{n-1}} dt_n H(t_n). \quad (3.8)$$

Die Dysonentwicklung konvergiert nicht so gut wie die Magnusentwicklung, erspart aber dafür die Operation $e^{-\frac{i}{\hbar} H_{\text{eff}} T}$. Für unsere Zwecke erwies sich die Dysonentwicklung als günstiger.

An dieser Stelle sollten wir uns noch einmal klarmachen, was wir gewonnen haben. Unser Hamilton-Operator (3.5) ist eine Summe und somit zerfallen die Integrale der Dysonentwicklung (3.8) auch in eine Summe aus Mehrfachintegralen. Aus diesen Mehrfachintegralen können wir die $\alpha_{i,j}$ herausziehen. Die verbleibenden, nun von den $\alpha_{i,j}$ unabhängigen Mehrfachintegrale müssen wir genau einmal lösen und können die jeweiligen Ergebnisse immer wieder verwenden. Da wir den unendlichdimensionalen Hamilton-Operator durch eine $n \times n$ -Matrix annähern, werden auch die verbleibenden Mehrfachintegrale zu (konstanten) $n \times n$ -Matrizen.

Der Zeitentwicklungsoperator U_j errechnet sich also aus einer Summe von $n \times n$ -Matrizen, die mit einer entsprechenden Anzahl an $\alpha_{i,j}$ multipliziert werden

$$U_j(\alpha_{1,j}, \dots, \alpha_{m,j}) = \sum_{k_1 \cdots k_m} \underbrace{(\alpha_{1,j})^{k_1} \cdot (\alpha_{2,j})^{k_2} \cdots (\alpha_{m,j})^{k_m}}_{k_i = \text{Exponent}} \cdot \underbrace{\mathcal{M}_{k_1 \cdots k_m}}_{n \times n\text{-Matrix}}. \quad (3.9)$$

Korrekterweise müsste die obige Matrix $\mathcal{M}_{k_1 \dots k_m}$ auch noch den Index j tragen, denn es ist nicht offensichtlich, warum die Dysonentwicklung (3.8) für unterschiedliche Zeiten zu den gleichen Matrizen führen sollte. Durch entsprechende Tricks, die im folgenden Abschnitt (3.3) präsentiert werden, können wir aber genau dies erreichen.

Werfen wir nun einen Blick auf den Rechenaufwand, den wir zur Berechnung der Zeitentwicklung nach (3.9) betreiben müssen. Die einzigen Matrixoperationen, die wir benötigen, sind die Multiplikation mit Skalaren und die Addition. Beide sind vom Rechenaufwand $\sim n^2$. Operationen höherer Ordnung (wie z.B. Matrixmultiplikationen) kommen nicht vor. Dies ist der große Vorteil gegenüber anderen Methoden zur Bestimmung und Entwicklungen von $U(t)$.

Sei m weiterhin die Anzahl der Summanden von $H(t)$ in (3.5) und der H_0 -Term sei nicht vorhanden. Dann besteht die Entwicklung bis zur k ten Ordnung von U_j nach (3.9) aus $\sum_{j=0}^k \binom{j+m-1}{j} = \binom{k+m}{k}$ Summanden (Die Verteilung der Exponenten k_i auf die m α_i in (3.9) entspricht der Verteilung von k Bosonen auf m Niveaus). Bei großen m sollte deshalb eine Zerlegung der Art

$$U(t_j, t_{j-1}) = U(t_j, \frac{t_j + t_{j-1}}{2}) \cdot U(\frac{t_j + t_{j-1}}{2}, t_{j-1}) \quad (3.10)$$

erwogen werden, um mit einer niedrigeren Ordnung auszukommen.

3.3 Wechselwirkungsbilder

Im letzten Abschnitt wurde erörtert, warum es vorteilhaft ist, die Zeitentwicklungsoperatoren U_j via Dysonentwicklung (3.8) zu berechnen. Die Berechnung der darin auftauchenden Mehrfachintegrale wird uns in diesem und im folgenden Abschnitt beschäftigen. Hier geht es vor allem um die Frage, was für ein Bezugssystem wir zur Berechnung heranziehen.

Der gleiche physikalische Sachverhalt kann in unterschiedlichen Bildern repräsentiert werden. Diese Bilder laufen unter den Namen Schrödinger-, Heisenberg- und Dirac- bzw. Wechselwirkungsbild. Das Wechselwirkungsbild unterteilt den Hamilton-Operator in zwei Anteile:

$$H = H_0 + H_{\text{WW}}. \quad (3.11)$$

Die Art der Aufteilung wird in der Regel vom physikalischen System nahe gelegt, ist aber letztlich beliebig. Zur Berechnung der Dysonentwicklung (3.8) werden wir zwei verschiedene Wechselwirkungsbilder bemühen; beide mit spezifischen Vor- und Nachteilen. Durch geschickte Transformationen zwischen den Bildern können wir die wichtigsten Vorteile für uns nutzen. Für den Hamilton-Operator verwenden wir die beiden Darstellungen (1.30)

und (1.31), die hier nochmals angegeben seien

$$\begin{aligned}
H^I &= \hbar\Omega S_+(e^{-i(\delta\cdot t+\phi)} + e^{+i(\delta\cdot t+\phi)})e^{i\eta(a+a^\dagger)} + \text{h.c.} \\
&+ \underbrace{\hbar\nu(a^\dagger a + \frac{1}{2}) + \hbar\frac{\gamma}{2}S_z}_{H_0}
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
H^{II} &= \hbar\Omega e^{+i\gamma t} S_+(e^{-i(\delta\cdot t+\phi)} + e^{+i(\delta\cdot t+\phi)})e^{i\eta(ae^{-i\nu t} + a^\dagger e^{+i\nu t})} \\
&+ \text{h.c.}
\end{aligned} \tag{3.13}$$

wobei $S_{+/z} = \mathbb{1} \otimes \sigma_{+/z} + \sigma_{+/z} \otimes \mathbb{1}$. Die Umrechnung erfolgt durch:

$$H^{II}(t) = e^{+\frac{i}{\hbar}H_0\cdot(t-t_0)}(H^I(t) - H_0)e^{-\frac{i}{\hbar}H_0\cdot(t-t_0)} \tag{3.14}$$

$$U^{II}(t, t_0) = e^{+\frac{i}{\hbar}H_0\cdot(t-t_0)}U^I(t, t_0). \tag{3.15}$$

Zuerst müssen wir klären, wie wir H^I & H^{II} in die gewünschte Form (3.5) bringen, wobei wir hier und im Rest dieses Abschnitts den Zeitintervallsindex j unterdrücken

$$H(t) = H_0 + \sum_{i=1}^n \alpha_i \cdot H_i(t). \tag{3.16}$$

Die α_i stellen die Variablen des Systems dar. In unserem Fall sind dies die Variablen Ω und ϕ . Diese müssen wir nun herausziehen. Dazu benutzen wir

$$e^{\pm i(\delta t + \phi)} = e^{\pm i(\delta t)} \cdot (\cos(\phi) \pm i \sin(\phi)). \tag{3.17}$$

Mit dieser Zerlegung können wir die Hamilton-Operatoren (3.12) und (3.13) darstellen als

$$H^I = \Omega \cos(\phi) \cdot H_{\cos}^I + \Omega \sin(\phi) \cdot H_{\sin}^I + H_0 \tag{3.18}$$

$$H^{II} = \Omega \cos(\phi) \cdot H_{\cos}^{II} + \Omega \sin(\phi) \cdot H_{\sin}^{II}. \tag{3.19}$$

Was mit

$$\alpha_1 = \Omega \cos(\phi) \quad \alpha_2 = \Omega \sin(\phi) \tag{3.20}$$

der gewünschten Form (3.5) bzw. (3.16) entspricht¹. Für die kommende Argumentation werden wir aber die Darstellungen (3.12) & (3.13) verwenden. H^I (3.12) hat den großen Vorteil, eine sehr einfache Zeitabhängigkeit aufzuweisen. Dies würde uns sogar erlauben, die bei der Dysonentwicklung anfallenden Integrale analytisch zu lösen. Diesen Vorteil werden wir aber nicht nutzen und stattdessen H^{II} (3.13) für die Dysonentwicklung verwenden, was auf eine numerische Lösung der Integrale hinausläuft. Eine Entwicklung, die

¹Die α_i sind die Variablen für die Dysonentwicklung. Die anschließende Optimierung des Systems kann ebenfalls nach diesen α_i erfolgen, aber es ist sinnvoller, die physikalischen Variablen Ω und ϕ dafür zu wählen – u.a. weil die Nebenbedingungen natürlicherweise in Ω und ϕ gegeben sein werden.

auf H^I fußt, würde nämlich Entwicklungskoeffizienten liefern, die nicht ideal sind. Ein Beispiel soll uns helfen, dies besser zu verstehen: Nehmen wir an, wir hätten eine beliebige Funktion $F(x)$ bis zur zweiten Ordnung via Taylorentwicklung approximiert

$$F(x) \approx a_0 + a_1 \cdot x + \frac{1}{2}a_2 \cdot x^2. \quad (3.21)$$

Wenn wir jetzt noch die dritte Ordnung berechnen, erhalten wir einen weiteren Term der Form $\frac{1}{3!}a_3 \cdot x^3$. Für uns ist aber viel wichtiger, was *nicht* passiert: Die Entwicklungskoeffizienten a_0 , a_1 und a_2 werden nicht verändert – sie sind bereits ideal und können nicht verbessert werden. Das Gleiche gilt für eine Dysonentwicklung, die auf H^{II} basiert; eine Verwendung von H^I hingegen würde keine idealen Entwicklungskoeffizienten liefern.

Was unterscheidet also die Entwicklung nach H^I von einer nach H^{II} ? Betrachten wir dazu zuerst eine Dysonentwicklung, die auf H^{II} aufbaut. Alle Summanden der n ten Ordnung werden die Form

$$\left(\prod_{k=1}^n \alpha_{i_k} \right) \cdot \underbrace{\mathcal{M}_{i_1 \dots i_n}}_{\text{Entwicklungskoeffizient}} \quad (3.22)$$

haben, wobei $\prod_{k=1}^n \alpha_{i_k}$ ein Produkt aus genau n α_i 's ist. Verwenden wir hingegen H^I , so sorgt das α -lose H_0 in H^I für Summanden der n ten Ordnung mit weniger als n α 's. Diese Terme mit „zuwenig“ α 's führen dazu, dass höhere Ordnungen die Entwicklungskoeffizienten der niedrigeren Ordnungen ständig korrigieren.

Es sei noch angemerkt, dass die Dysonentwicklung selbstverständlich ideal wäre, wenn wir uns für eine Entwicklung nach den Potenzen von $\frac{1}{\hbar}$ und nicht nach den Potenzen der α_i interessieren würden.

Wir werden also H^{II} verwenden, um daraus die Entwicklung für U^{II} zu berechnen. Haben wir alle Entwicklungskoeffizienten bestimmt, werden wir (3.15) nutzen, um daraus die Entwicklung für U^I zu gewinnen (den Grund dafür sehen wir gleich). Dies geht mit einer *einmaligen* Transformation aller Entwicklungskoeffizienten einher.

$$\begin{aligned} U^I &= e^{-\frac{i}{\hbar}H_0 \cdot (t-t_0)} \cdot U_j^{II} \\ &= e^{-\frac{i}{\hbar}H_0 \cdot (t-t_0)} \sum_{k_1 \dots k_m} (\alpha_1)^{k_1} \cdot (\alpha_2)^{k_2} \dots (\alpha_m)^{k_m} \cdot \mathcal{M}_{k_1 \dots k_m}^{(\text{für } U^{II})} \\ &= \sum_{k_1 \dots k_m} \alpha_1^{k_1} \cdot \alpha_2^{k_2} \dots \alpha_m^{k_m} \cdot \underbrace{e^{-\frac{i}{\hbar}H_0 \cdot (t-t_0)} \cdot \mathcal{M}_{k_1 \dots k_m}^{(\text{für } U^{II})}}_{\mathcal{M}_{k_1 \dots k_m}^{(\text{für } U^I)}}. \end{aligned} \quad (3.23)$$

Der Sinn dieser Transformation wird verständlich, wenn wir uns den Ansatz (2.7) in Erinnerung rufen:

$$U(t_n, t_0) = U(t_n, t_{n-1}) \cdot U(t_{n-1}, t_{n-2}) \cdot U(t_{n-2}, t_{n-3}) \dots U(t_1, t_0).$$

Wir müssen jedes $U(t_{\text{Stopp}}, t_{\text{Start}})$ in diesem Produkt separat berechnen. Dazu möchten wir immer auf die gleiche Dysonentwicklung nach (3.9) zurückgreifen können. Aber wie soll das funktionieren? Die Start- und Stoppzeiten der $U(t_{\text{Stopp}}, t_{\text{Start}})$ gehen als Integralgrenzen in die Entwicklungskoeffizienten ein. Mit $U^{\text{II}}(t_{\text{Stopp}}, t_{\text{Start}})$ wäre dies in der Tat ein hoffnungsloses Unterfangen – nicht jedoch mit $U^{\text{I}}(t_{\text{Stopp}}, t_{\text{Start}})$. Wir können das $U^{\text{I}}(t_{\text{Stopp}}, t_{\text{Start}})$, das wir über die Transformation (3.23) gewonnen haben, so behandeln, als hätten wir es direkt aus der Integration der H^{I} erhalten. Somit können wir die einfache Zeitabhängigkeit der H^{I} nutzen, um das Problem der Abhängigkeit von den Start- und Stoppzeiten zu lösen. Dazu legen wir uns zuerst darauf fest, alle Zeitintervalle gleich lang zu wählen (etwas, das ohnehin vorgesehen war). Als nächstes halten wir fest, dass die Zeit in H^{I} (3.12) immer in Form von $(\delta \cdot t + \phi)$ eingeht. Dieses ϕ können wir nutzen, um die Integrale auf passende Zeiten zu transformieren. Dazu nehmen wir folgende Zerlegung vor

$$\phi^{\text{Gesamt}} = \phi^{\text{Physik}} + \phi^{\text{Zeitshift}} \quad \phi^{\text{Zeitshift}} = \delta \cdot (t_{j-1} - t_0). \quad (3.24)$$

ϕ^{Gesamt} wird zur Berechnung der α_i (3.20) verwendet, bzw. es ist der Wert, den der Computer am Ende des Optimierungsprozesses ausgibt. ϕ^{Physik} ist der Wert der Phase, den wir im Experiment einstellen müssen, während $\phi^{\text{Zeitshift}}$ ein rein mathematisches Artefakt ist. Damit erhalten wir

$$\begin{aligned} \int_{t_0}^{t_0+\Delta t} dt H(\delta \cdot t + \phi^{\text{Ges.}}) &= \int_{t_0}^{t_0+\Delta t} dt H(\delta \cdot t + [\phi^{\text{Phys.}} + \phi^{\text{Zeitshift}}]) \\ &= \int_{t_0}^{t_0+\Delta t} dt H(\delta \cdot t + [\phi^{\text{Phys.}} + \delta \cdot (t_{j-1} - t_0)]) \\ &= \int_{t_0}^{t_0+\Delta t} dt H(\delta \cdot (t + [t_{j-1} - t_0]) + \phi^{\text{Phys.}}) \\ &= \int_{t_{j-1}}^{t_j=t_{j-1}+\Delta t} dt H(\delta \cdot t + \phi^{\text{Phys.}}). \end{aligned} \quad (3.25)$$

Man beachte die verschobenen Integralgrenzen des letzten Integrals. Hierin äußert sich der Sinn der Einführung von $\phi^{\text{Zeitshift}}$. Die gleiche Substitution lässt sich auch bei den Mehrfachintegralen der Dysonentwicklung anwenden. Wir können uns somit auf die Entwicklung des Zeitentwicklungsoperators $U(t_1, t_0)$ beschränken und daraus mittels (3.24) alle anderen $U(t_j, t_{j-1})$ gewinnen.

3.4 Numerische Lösung der Dysonentwicklung

Im letzten Abschnitt haben wir erfahren, welche Bezugssysteme wir heranziehen müssen, um die Dysonentwicklung unseres Zeitentwicklungsoperators

durchzuführen. Wir konnten uns davon überzeugen, dass *eine* Dysonentwicklung für alle $U(t_j, t_{j-1})$ in (3.1) ausreichend ist. Jetzt müssen wir uns der Frage zuwenden, wie wir die geschachtelten Mehrfachintegrale in der Entwicklung (3.8)

$$U(t, t_0) = 1 + \sum_{n=1}^{\infty} \left(-\frac{i}{\hbar}\right)^n \int_{t_0}^t dt_1 H(t_1) \int_{t_0}^{t_1} dt_2 H(t_2) \cdots \int_{t_0}^{t_{n-1}} dt_n H(t_n)$$

numerisch berechnen wollen. Leider werden die mathematischen Standardroutinen, die von den Computerprogrammen bereitgestellt werden, sich hierfür als unzureichend erweisen.

Zwecks besseren Verständnisses werden wir die Integrale fürs Erste durch obere Riemannsummen lösen und uns erst im Abschnitt 3.4.1 um eine geschicktere Integrationsmethode bemühen. Im Folgenden ersetzen wir also

$$\int_0^t dt_1 H(t_1) \int_0^{t_1} dt_2 H(t_2) \int_0^{t_2} dt_3 H(t_3) \cdots \int_0^{t_{k-1}} dt_k H(t_k) \quad (3.26)$$

durch

$$\mathcal{S}_k^n := \sum_{i_1=1}^n \frac{t}{n} H^{i_1} \sum_{i_2=1}^{i_1} \frac{t}{n} H^{i_2} \sum_{i_3=1}^{i_2} \frac{t}{n} H^{i_3} \cdots \sum_{i_k=1}^{i_{k-1}} \frac{t}{n} H^{i_k} \quad (3.27)$$

mit

$$H^j = H\left(\frac{j}{n}t\right). \quad (3.28)$$

\mathcal{S}_k^n beschreibt den Beitrag der k ten Ordnung der Dysonentwicklung, wobei n Stützstellen zur Berechnung der Integrale herangezogen wurden. Die Summe \mathcal{S}_k^n umfasst $\binom{n+k-1}{k}$ Summanden, wie in der Fußnote² gezeigt wird. Die gesamte Dysonentwicklung bis zur k ten Ordnung umfasst somit $\sum_{j=0}^k \binom{j+n-1}{j} = \binom{k+n}{k}$ Summanden. Da n die Anzahl der Stützstellen zur Näherung des Integrals darstellt, ist dies keine ganz so kleine Zahl. Bei $n = 100$ und $k = 20$ hätten wir $2,9 \cdot 10^{22}$ Summanden. Dabei wurde bisher noch nicht berücksichtigt, dass wir Produkte von Hamilton-Operatoren integrieren, die selbst noch über eine Unterstruktur aus Kombinationen der $\alpha_i H_i$ verfügen (3.5). Leider ist all dies keine bloße Zahlenspielerei. Wenn wir auf die vorgefertigten Integrationsroutinen zurückgreifen wollten, würde der Computer versuchen, genau diese horrende Anzahl an Kalkulationen vorzunehmen. Daher müssen wir den Komfortbereich der vorgefertigten Integrationsroutinen verlassen und uns selbst um eine trickreiche Summierung bemühen.

²Die Anzahl der Summanden in (3.27) entspricht der möglichen Anzahl an Verteilungen von $n-1$ Bosonen auf $k+1$ Niveaus. Wir können nämlich jeder erlaubten Kombination $H^{i_1} H^{i_2} H^{i_3} \cdots H^{i_k}$ mit $n \geq i_1$ und $i_j \geq i_{j+1} \geq 1$ eindeutig eine Bosonenverteilung zuordnen. Dabei befinden sich im ersten Niveau $n - i_1$ Bosonen und im j ten Niveau $i_{j-1} - i_j$. Wir brauchen noch ein $k+1$ tes Niveau, damit die Anzahl der Bosonen immer $n-1$ ist; d.h. in diesem Niveau befinden sich $i_k - 1$ Bosonen.

Im Folgenden werden wir eine Rekursionsformel (3.30) erstellen, die eine problemlose Berechnung der Dysonentwicklung gestattet. Dazu definieren wir zuerst $\mathcal{S}_k^{s,n}$ als Untersumme von \mathcal{S}_k^n aus (3.27) gemäß

$$\mathcal{S}_k^{s,n} := \sum_{i_1=1}^s \frac{t}{n} H^{i_1} \sum_{i_2=1}^{i_1} \frac{t}{n} H^{i_2} \sum_{i_3=1}^{i_2} \frac{t}{n} H^{i_3} \dots \sum_{i_k=1}^{i_{k-1}} \frac{t}{n} H^{i_k}, \quad (3.29)$$

wobei gegenüber (3.27) nur die erste Summationsgrenze von n auf s gesetzt wurde; d.h. $\mathcal{S}_k^{n,n} = \mathcal{S}_k^n$. Jetzt können wir die Untersummen der k ten Ordnung durch die der $(k-1)$ ten Ordnung ausdrücken

$$\begin{aligned} \mathcal{S}_k^{s+1,n} &= \sum_{i=1}^{s+1} \frac{t}{n} H^i \cdot \mathcal{S}_{k-1}^{i,n} \quad \text{mit} \quad \mathcal{S}_0^{i,n} := 1 \\ &= \sum_{i=1}^s \frac{t}{n} H^i \cdot \mathcal{S}_{k-1}^{i,n} + \frac{t}{n} H^{s+1} \cdot \mathcal{S}_{k-1}^{s+1,n} \\ &= \mathcal{S}_k^{s,n} + \frac{t}{n} H^{s+1} \cdot \mathcal{S}_{k-1}^{s+1,n}. \end{aligned} \quad (3.30)$$

Dank dieser Rekursionsformel können wir die gesamte Dysonentwicklung mit $(n-1) \cdot k$ Summationen und $n \cdot (k-1)$ Multiplikationen berechnen. D.h. der Aufwand für die Berechnung sämtlicher Summanden wächst nur linear mit der Ordnung! Die Anzahl der Berechnungen des Hamilton ist mit n sogar konstant.

Die Summenstruktur des Hamilton-Operators

Als nächstes haben wir die Summenstruktur des Ansatzes (3.5) für den Hamilton-Operator zu berücksichtigen, den wir hier angepasst wiedergeben

$$H = \sum_{i=1}^m \alpha_i \cdot H_i. \quad (3.31)$$

Wir müssen dabei im Hinterkopf behalten, was eigentlich unser Ziel bei der Dysonentwicklung ist. Deshalb geben wir hier auch die Formel (3.9) noch einmal angepasst wieder

$$U(\alpha_1, \dots, \alpha_m) = \sum_{p_1 \dots p_m} \underbrace{(\alpha_1)^{p_1} \cdot (\alpha_2)^{p_2} \dots (\alpha_m)^{p_m}}_{p_i = \text{Exponent}} \cdot \underbrace{\mathcal{M}_{p_1 \dots p_m}}_{n \times n \text{-Matrix}}. \quad (3.32)$$

Die Dysonentwicklung soll uns die Entwicklungskoeffizienten $\mathcal{M}_{p_1 \dots p_m}$ für die Produkte $\alpha_1^{p_1} \cdot \alpha_2^{p_2} \dots \alpha_m^{p_m}$ liefern. In diesem Sinne kommt es uns zupass, dass zwar für die Produkte der Hamilton-Operatoren $H_j \cdot H_k \neq H_k \cdot H_j$ gilt, aber beide zu $\alpha_j \alpha_k$ gehören. Diese beiden Beiträge können also zusammengefasst (addiert) werden. Für den Fall $H = \alpha_1 H_1 + \alpha_2 H_2$ liefert die 20te Ordnung 2^{20} Anordnungen der H_i , aber es gibt nur 21 Kombinationen

für $(\alpha_1)^n(\alpha_2)^{20-n}$. Die Entwicklungskoeffizienten können in jeder Ordnung in zusammengefasster Form berechnet werden; d.h. die $(k+1)$ te Ordnung kommt mit Rückgriffen auf die bereits zusammengefassten Terme der k ten Ordnung aus. Dies wird ermöglicht durch eine Erweiterung der Rekursionsformel (3.30) auf die weiter unten folgende Form (3.34). Dafür benötigen wir aber zuerst die Verallgemeinerung von $\mathcal{S}_k^{s,n}$ (3.29)

$$\mathcal{S}_k^{s,n} \rightarrow \mathcal{S}_{p_1 p_2 \dots p_m}^{s,n} \quad \text{wobei} \quad \sum_{i=1}^m p_i = k. \quad (3.33)$$

Die Indices von $\mathcal{S}_{p_1 p_2 \dots p_m}^{s,n}$ zeigen an, dass dies der Entwicklungskoeffizient zu $(\alpha_1)^{p_1}(\alpha_2)^{p_2} \dots (\alpha_m)^{p_m}$ ist, wobei die Summierung bisher über s von insgesamt n Stützstellen erfolgte. Jetzt können wir wie angekündigt die Rekursionsformel (3.30) erweitern

$$\begin{aligned} \mathcal{S}_{p_1 p_2 \dots p_m}^{s+1,n} = \mathcal{S}_{p_1 p_2 \dots p_m}^{s,n} &+ \left(H_1^{s+1} \cdot \mathcal{S}_{(p_1-1)p_2 \dots p_m}^{s+1,n} + H_2^{s+1} \cdot \mathcal{S}_{p_1(p_2-1) \dots p_m}^{s+1,n} + \right. \\ &\dots + \left. H_m^{s+1} \cdot \mathcal{S}_{p_1 p_2 \dots (p_m-1)}^{s+1,n} \right) \cdot \frac{t}{n}. \end{aligned} \quad (3.34)$$

Wiederholen wir diese Berechnung, bis alle n Stützstellen aufsummiert sind, ergibt sich für die $\mathcal{M}_{p_1 p_2 \dots p_m}$ aus (3.32)

$$\mathcal{M}_{p_1 p_2 \dots p_m} \equiv \mathcal{S}_{p_1 p_2 \dots p_m}^{n,n}. \quad (3.35)$$

Anhand dieser Identität erkennen wir auch, dass nicht mehr Summen berechnet werden, als die Dysonentwicklung Summanden hat: Zur Berechnung von $\mathcal{S}_{p_1 p_2 \dots p_m}^{n,n}$ benötigen wir z.B. für jedes $p_j > 0$ die Summe $\mathcal{S}_{p_1 \dots (p_j-1) \dots p_m}^{n,n}$. Da aber

$$\mathcal{M}_{p_1 \dots (p_j-1) \dots p_m} \equiv \mathcal{S}_{p_1 \dots (p_j-1) \dots p_m}^{n,n}, \quad (3.36)$$

sind dies keine „unnützen“ Zwischenergebnisse, sondern vollwertige Summanden der Dysonentwicklung. Damit wächst die Rechenzeit für Dysonentwicklung nur linear mit der Anzahl der Summanden $\mathcal{M}_{p_1 \dots p_m}$ der Entwicklung.

Die Umsetzung auf dem Computer gestaltet sich etwas unübersichtlich. Die Rechenergebnisse können zwar platzsparend in einem linearen Array abgelegt werden, aber die Zugehörigkeiten sind dann nicht mehr trivial erkennbar. Welche Terme des linearen Arrays muss man addieren, wenn man zur nächst höheren Ordnung gelangen will? Hier empfiehlt es sich, am Anfang ein Pointerfeld³ mit Zusatzinformationen über die „Verwandtschaften“ innerhalb des linearen Arrays zu erstellen. In der hier vorliegenden Arbeit

³Im Rahmen dieser Diplomarbeit wurde MATLAB verwendet, welches nicht über den Variablentyp „Pointer“ verfügt. Hier kann sich mit Sets beholfen werden, die andere Sets als Elemente enthalten.

wurde dafür ein rekursiv arbeitenden Algorithmus herangezogen.

Magnusentwicklung

Zum Ende dieses Abschnitts kommen wir noch einmal auf die Magnusentwicklung (3.7) zurückkommen. Dort scheitert der Versuch, die Summation gemäß (3.30) vorzunehmen, da die H^j durch die Kommutatoren nicht mehr in der zeitgeordneten Reihenfolge auftreten. Interessanterweise ist es aber mit moderatem Aufwand möglich, aus einer fertig berechneten Dysonentwicklung eine Magnusentwicklung zu gewinnen. Das vermutlich bekannteste Verfahren dafür stammt von Burum [26, 27].

3.4.1 Numerische Integration

Im letzten Abschnitt haben wir gesehen, wie wir die Integrale der Dysonentwicklung durch Summation numerisch berechnen können. Entscheidend war, die einzelnen Summanden geschickt zusammenzufassen, was in der Rekursionsformel (3.34) gipfelte. Zum einfacheren Verständnis hatten wir dabei in (3.27) die Integrale durch Riemannsummen genähert. Zur wirklichen Berechnung sind Riemannsummen aber herzlich wenig geeignet, denn sie konvergieren extrem schlecht und haben einen Fehler⁴ proportional zur Intervalllänge Δt zwischen zwei Stützstellen. Die Sehnentrapezformel, die den Mittelwert aus oberer und unterer Riemannsumme bildet, kommt bereits auf einen Fehler $\sim (\Delta t)^2$.

Es gibt viele Integralformeln, die wesentlich geschickter agieren. Sie alle greifen auf zusätzliche Stützstellen innerhalb des Intervalls Δt zurück. Dies ist nicht das Gleiche wie die Verkleinerung der Intervalllänge Δt , was im Endeffekt auch mit mehr Stützstellen einhergeht. Da sich dies als wichtiger Punkt für uns erweisen wird, wollen wir hier etwas ausführlicher werden. Die Sehnentrapezformel macht folgende Ersetzung

$$\int_{t_1}^{t_2} f(t) dt \rightarrow (f(t_1) + f(t_2)) \cdot \frac{t_2 - t_1}{2} \quad t_2 = t_1 + \Delta t. \quad (3.37)$$

Halbieren wir die Intervalllänge für die Riemannsummen, so ergibt sich für das gleiche Integral

$$\begin{aligned} \int_{t_1}^{t_2} f(t) dt &= \int_{t_1}^{\frac{t_1+t_2}{2}} f(t) dt + \int_{\frac{t_1+t_2}{2}}^{t_2} f(t) dt \\ &\rightarrow (f(t_1) + 2 \cdot f(\frac{t_1+t_2}{2}) + f(t_2)) \cdot \frac{t_2 - t_1}{4}. \end{aligned} \quad (3.38)$$

⁴Zur Vorbeugung von Missverständnissen sei darauf hingewiesen, dass in der Literatur oft der Fehler pro Intervalllänge angegeben wird. Dieser wird trivialerweise kleiner, wenn die Intervalllänge kleiner wird. Die hier angegebenen Fehler beziehen sich hingegen auf konstante Integralgrenzen.

Belassen wir die Intervalllänge hingegen bei ihrem alten Wert und benutzen stattdessen eine zusätzliche Stützstelle pro Intervall, so berechnet sich das Integral mit der Simpsonregel zu

$$\int_{t_1}^{t_2} f(t)dt \rightarrow (f(t_1) + 4 \cdot f(\frac{t_1+t_2}{2}) + f(t_2)) \cdot \frac{t_2-t_1}{6}. \quad (3.39)$$

Der Fehler bei der Simpsonregel ist nur noch $\sim (\Delta t)^4$. Die Simpsonregel – ebenso wie alle anderen elaborierten Integralformeln – hat für unsere Zwecke aber einen entscheidenden Nachteil: Wir müssen die Information über den Funktionswert an der Stützstelle $t_i + \frac{\Delta t}{2} = \frac{t_i+t_{i+1}}{2}$ hineinstecken, bekommen aber keine Information zurück, wie der Wert des Integrals an dieser Stelle lautet. Wir stecken also mehr Information hinein, als wir herausbekommen. Dies ist ein schwerwiegendes Problem, denn die Rekursionsformeln (3.30) und (3.34) funktionieren nur dann vernünftig, wenn wir für jede Stützstelle auch eine Ausgabe in Form einer Integralnäherung bekommen. Diese benötigen wir bei der Berechnung der nächsten Ordnung.

Mit den als Newton-Cotes-Formeln bekannten Integralnäherungen lässt sich ein Ausweg finden. Bei diesen Verfahren wird die Funktion durch ein Polynom n ten Grades angenähert. Dieses Polynom lässt sich dann problemlos integrieren. Normalerweise „verbraucht“ dieses Verfahren auch mehr Informationen, als es liefert. Aber dadurch, dass wir im Besitz des Näherungspolynoms sind, können wir jetzt für jede Stützstelle einen zusätzlichen Integralwert berechnen. Nun mag sich dieses Verfahren recht aufwendig in der Umsetzung anhören. Das Schöne ist aber, dass wir fast alle Berechnungen auch ohne Kenntnis der zu integrierenden Funktion bereits im Voraus durchführen können. Was am Ende bleibt, ist ein Satz an Zahlen (3.40), mit denen wir die Werte der Funktion an den unterschiedlichen Stützstellen gewichten müssen.

In der vorliegenden Arbeit wurde ein Polynom 8ten Grades verwendet, was den Speicherbedarf während der Berechnung gegenüber der Riemannsumme ungefähr verneunfacht, aber dafür auch nur mit einem Fehler $\sim (\Delta t)^{10}$ statt $\sim \Delta t$ behaftet ist. Die Essenz des Verfahrens ist die Matrix \mathcal{I} , die im Fall des Polynom 8ten Grades gegeben ist durch

$$\mathcal{I} = \begin{pmatrix} 1070017 & 4467094 & -4604594 & 5595358 & -5033120 & 3146338 & -1291214 & 312874 & -33953 \\ 1036064 & 5842688 & -1359808 & 3842816 & -3715840 & 2391296 & -996928 & 243968 & -26656 \\ 1043361 & 5743062 & 278478 & 6474654 & -4548960 & 2789154 & -1139022 & 275562 & -29889 \\ 1040128 & 5779456 & 62464 & 8384512 & -2324480 & 2363392 & -1012736 & 249856 & -27392 \\ 1042625 & 5753750 & 188750 & 7958750 & -100000 & 4273250 & -1228750 & 286250 & -30625 \\ 1039392 & 5785344 & 46656 & 8356608 & -933120 & 6905088 & 409536 & 186624 & -23328 \\ 1046689 & 5716438 & 340942 & 7601566 & 384160 & 5152546 & 3654322 & 1562218 & -57281 \\ 1012736 & 6029312 & -950272 & 10747904 & -4648960 & 10747904 & -950272 & 6029312 & 1012736 \end{pmatrix} \quad (3.40)$$

Damit gilt

$$\begin{pmatrix} \int_{t_0}^{t_0+\frac{1}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{2}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{3}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{4}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{5}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{6}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{7}{8}\Delta t} f(t)dt \\ \int_{t_0}^{t_0+\frac{8}{8}\Delta t} f(t)dt \end{pmatrix} = \frac{1}{29030400} \cdot \mathcal{I} \cdot \begin{pmatrix} f(t_0 + \frac{0}{8}\Delta t) \\ f(t_0 + \frac{1}{8}\Delta t) \\ f(t_0 + \frac{2}{8}\Delta t) \\ f(t_0 + \frac{3}{8}\Delta t) \\ f(t_0 + \frac{4}{8}\Delta t) \\ f(t_0 + \frac{5}{8}\Delta t) \\ f(t_0 + \frac{6}{8}\Delta t) \\ f(t_0 + \frac{7}{8}\Delta t) \\ f(t_0 + \frac{8}{8}\Delta t) \end{pmatrix} \cdot \Delta t. \quad (3.41)$$

Der Faktor $\frac{1}{29030400}$ in (3.41) entstammt der Matrix \mathcal{I} und wurde der Übersicht wegen herausgezogen. So ist z.B. die Summe der Einträge der untersten Zeile von \mathcal{I} exakt 29030400. Allgemeiner formuliert gilt

$$\frac{1}{29030400} \sum_k \mathcal{I}_{j,k} = \frac{j}{8}, \quad (3.42)$$

wie sich sofort ergibt, wenn man in (3.41) die zu integrierende Funktion $f(t) \equiv 1$ setzt.

3.5 Verkürzung der Summation

Im letzten Abschnitt haben wir gesehen, wie die Entwicklungskoeffizienten der Dysonentwicklung des Zeitentwicklungsoperators numerisch berechnet werden können. Letzterer liegt uns jetzt in der in (3.9) angekündigten Form vor

$$U_j(\alpha_{1,j}, \dots, \alpha_{m,j}) = \sum_{k_1 \dots k_m} \underbrace{\alpha_{1,j}^{k_1} \cdot \alpha_{2,j}^{k_2} \cdots \alpha_{m,j}^{k_m}}_{k_i = \text{Exponent}} \cdot \underbrace{\mathcal{M}_{k_1 \dots k_m}}_{n \times n \text{-Matrix}}. \quad (3.43)$$

Diese Entwicklung hat den Vorteil, nur aus Rechenoperationen zu bestehen, die mit n^2 skalieren. Trotz dieses Vorteils ist die Berechnung von U_j via (3.43) sehr zeitaufwendig. In diesem Kapitel untersuchen wir, inwieweit wir diese Berechnungen noch vereinfachen können.

Als ersten Schritt geben wir den $\alpha_{i,j}$ gemäß (3.20) eine konkrete Gestalt

$$\alpha_{1,j} = \Omega_j \cos(\phi_j) \quad \alpha_{2,j} = \Omega_j \sin(\phi_j). \quad (3.44)$$

Damit wird (3.43) zu

$$U_j = \sum_{k_{\cos}, k_{\sin}} (\Omega_j \cos(\phi_j))^{k_{\cos}} (\Omega_j \sin(\phi_j))^{k_{\sin}} \cdot \mathcal{M}_{k_{\cos} k_{\sin}}. \quad (3.45)$$

Als nächstes betrachten wir nur die Terme einer Ordnung. Als Beispiel haben wir die vierte Ordnung gewählt

$$\Omega_j^4 \cdot [\cos^4(\phi_j)\mathcal{M}_{40} + \cos^3(\phi_j)\sin(\phi_j)\mathcal{M}_{31} + \cos^2(\phi_j)\sin^2(\phi_j)\mathcal{M}_{22} + \cos(\phi_j)\sin^3(\phi_j)\mathcal{M}_{13} + \sin^4(\phi_j)\mathcal{M}_{04}]. \quad (3.46)$$

Wenn ϕ_j keine Variable, sondern eine Konstante wäre, könnten wir die ganzen Terme in der eckigen Klammer in einer einzigen $n \times n$ -Matrix zusammenfassen. Dergleichen gilt natürlich nicht nur für die vierte Ordnung, sondern für alle

$$\phi_j = \text{const} \rightarrow U_j = \sum_n (\Omega_j)^n \cdot \mathcal{M}_n^{(\phi_j)}. \quad (3.47)$$

Bei variablem ϕ_j umfasst die Entwicklung von (3.45) bis zur 20ten Ordnung⁵ 231 Summanden; bei konstantem ϕ_j reichen (3.47) hingegen 21. Wir können enorm an Rechenzeit einsparen, wenn wir ϕ_j zeitweise als konstant behandeln. Es sei daran erinnert, dass wir bei jedem Optimierungsschritt immer nur eines der Ω_j, ϕ_j als variabel betrachten und alle anderen als konstant. Bisher schien die natürliche Reihenfolge für die Wahl der zu optimierenden Variable $\Omega_1, \phi_1, \Omega_2, \phi_2, \Omega_3, \phi_3$ usw. zu sein. Unsere neuen Erkenntnisse legen nun aber nahe, für mehrere Runden nur $\Omega_1, \Omega_2, \Omega_3$ usw. zu optimieren und danach mehrere Runden nur die ϕ_j . Bei der ersten Runde der Ω_j Optimierung fassen wir die Entwicklungsterme von (3.45) zusammen und können dann in den folgenden Runde auf die verkürzte Form (3.47) zurückgreifen. Hierbei gewinnen wir Rechenzeit auf Kosten des Speicherbedarfs: Wir müssen nämlich für jedes $\phi_j = \text{const}_j$ eine eigene Entwicklung speichern. Da eingesparte Rechenzeit aber meist ein höheres Gut darstellt als Speicher, ist dies ein ratsamer Handel.

Wenn wir uns nun die Runden ansehen, in denen wir die Ω_j als konstant ansehen können, dann werden wir etwas ernüchtert: Wir sparen in (3.45) keinen einzigen Summanden ein, da die trigonometrischen Ausdrücke sich nicht zusammenfassen lassen – zumindest nicht auf trivial ersichtliche Weise. Eine nur *einmalig* durchzuführende Transformation ermöglicht es, anschließend auch bei konstantem Ω_j die Terme einfach zusammenzufassen. Rein lineare Matrixoperationen reichen aus, um zu folgender Entwicklung zu gelangen

$$U_j = \sum_{n=0}^{\dots} \sum_{m=0}^n [(\Omega_j)^n \cos(m \cdot \phi_j)\mathcal{M}_{n,m}^{\cos} + (\Omega_j)^n \sin(m \cdot \phi_j)\mathcal{M}_{n,m}^{\sin}]. \quad (3.48)$$

Diese hat genauso viele Summanden wie (3.45), lässt sich bei konstantem Ω_j aber verkürzen zu

$$U_j = \sum_m [\cos(m \cdot \phi_j)\mathcal{M}_m^{(\Omega_j)\cos} + \sin(m \cdot \phi_j)\mathcal{M}_m^{(\Omega_j)\sin}]. \quad (3.49)$$

⁵Bei einigen in dieser Diplomarbeit nicht erwähnten Modellen war eine Entwicklung bis zur zu 40ten Ordnung notwendig – die Entwicklung erfolgt nämlich über einen recht großen Zeitraum.

Aus den 231 Summanden der Entwicklung bis zur 20ten Ordnung von (3.45) bzw. (3.48) werden bei konstantem Ω_j 41 Summanden⁶ in (3.49).

3.5.1 Weitere Ansätze zur Verbesserung

Wie im letzten Abschnitt geht es auch in diesem Abschnitt um die Frage, wie wir den Zeitentwicklungsoperator U_j möglichst effektiv berechnen. Drei Ansätze werden beschrieben, von denen sich die letzten beiden als unbrauchbar erwiesen haben.

Verkleinerung der Entwicklung

Bei jeder nichttrivialen Näherung gilt es, die gewünschte Präzision mit dem dafür erforderlichen Aufwand in Einklang zu bringen. Für uns heißt das, wir müssen darüber nachdenken, wie genau wir die Entwicklung der U_j betreiben wollen. Es erweist sich als klug, die Optimierung mit einer geringeren Genauigkeit für U_j zu starten und den damit verbundenen Vorteil kürzerer Rechenzeiten zu nutzen, um zum Ende hin eine hohe Präzision an den Tag zu legen. Zwei Näherungen gilt es vorzunehmen:

1. Die Ordnung der Dysonentwicklung für U_j
2. Die Festlegung einer maximalen Anzahl für die Phononen, um \hat{a} und \hat{a}^\dagger als Matrix ausdrücken zu können

Hat man einmal eine präzise Entwicklung für die U_j berechnet, so stellt es kein Problem dar, diese auf eine weniger genaue, aber schneller zu berechnende Form zu verkürzen. Einsparungen bei der Anzahl der Phononen führt in (3.48) und in allen anderen Darstellungen zu kleineren Matrizen \mathcal{M} . Die Reduktion der Ordnung hingegen resultiert einfach nur im Streichen von Termen höherer Ordnung.

Tschebyschow-Polynome

Der Numeriker mag sich fragen, ob das oben erwähnte simple Streichen von Termen höherer Ordnung wirklich den geschicktesten Weg zur Verkürzung der Dysonentwicklung darstellt. Wenn man eine Taylorentwicklung verkürzen möchte, ermöglicht die Verwendung von Tschebyschow-Polynomen, Informationen der höheren Ordnungen auf die niedrigeren zu übertragen und so zu einer besseren Entwicklung zu gelangen [23]. Dies klingt unmöglich, da das Ergebnis selbst wieder ein Polynom ist und in Abschnitt 3.3 behauptet wurde, die Taylorentwicklung sei ideal. Damit der „Trick“ funktioniert, muss die Variable, nach der entwickelt wird, beschränkt sein. Dann besteht die Leistung der Tschebyschow-Polynome darin, den Approximationsfehler über

⁶Der besseren Übersicht wegen wurden in (3.49) auch für den Fall $m = 0$ zwei Summanden aufgeführt, obwohl einer reicht.

das ganze Intervall gleichmäßig zu „verschmieren“. Die Taylorentwicklung hingegen hat einen stark erhöhten Fehler beim Maximalwert der Entwicklungsvariablen x , was aus dem Restglied $\sim x^{n+1}$ ersichtlich ist.

Eine genauere Rechnung zeigt aber, dass die Verwendung von Tschebyschow-Polynomen zum Wegfall der aus \sum_m^n folgenden Eigenschaft $m \leq n$ in (3.48) führt, was eine annähernde Verdopplung der Summanden nach sich ziehen würde.

„Doppelte“ Präzision bei gleichem Aufwand

Die Entwicklung (3.48)

$$U_j = \sum_{n=0}^{\infty} \sum_{m=0}^n [(\Omega_j)^n \cos(m \cdot \phi_j) \mathcal{M}_{n,m}^{\cos} + (\Omega_j)^n \sin(m \cdot \phi_j) \mathcal{M}_{n,m}^{\sin}]$$

eröffnet uns die Möglichkeit, die Symmetrien der trigonometrischen Funktionen besser auszunutzen

$$\phi \rightarrow \phi \pm \pi \quad \longrightarrow \quad \begin{array}{l} \sin(m \cdot \phi) \rightarrow (-1)^m \sin(m \cdot \phi) \\ \cos(m \cdot \phi) \rightarrow (-1)^m \cos(m \cdot \phi). \end{array} \quad (3.50)$$

Dies ist für uns interessant, da gemäß (3.24) die Transformation $\phi \rightarrow \phi + \pi$ auf die Berechnung von U_j die gleiche Auswirkung hat, wie die Transformation $t \rightarrow t + \frac{\pi}{\delta}$, wobei $\pm \delta$ die Laserfrequenzen im mitrotierenden Bezugssystem darstellen. Wählen wir die Zeitintervalle Δt für die Zeitentwicklung U_j gemäß

$$U_j(\Delta t, 0) := U_j\left(\frac{2\pi}{\delta}, 0\right) \quad \text{also} \quad \Delta t := \frac{2\pi}{\delta}, \quad (3.51)$$

so können wir folgende Zerlegung vornehmen.

$$U_j\left(\frac{2\pi}{\delta}, 0\right) = U_j\left(\frac{2\pi}{\delta}, \frac{\pi}{\delta}\right) \cdot U_j\left(\frac{\pi}{\delta}, 0\right). \quad (3.52)$$

Da $U_j\left(\frac{2\pi}{\delta}, \frac{\pi}{\delta}\right)$ aus $U_j\left(\frac{\pi}{\delta}, 0\right)$ durch die Transformation $\phi \rightarrow \phi + \pi$ zu gewinnen ist, erfordern beide Entwicklungen zusammen nur unwesentlich mehr Rechenaufwand als ein U_j allein.

$$U_j\left(\frac{\pi}{\delta}, 0\right) = \underbrace{\sum_{V_{\text{gerade}}} \text{Terme mit geradem } m}_{V_{\text{gerade}}} + \underbrace{\sum_{V_{\text{ungerade}}} \text{Terme mit ungeradem } m}_{V_{\text{ungerade}}} \quad (3.53)$$

$$U_j\left(\frac{2\pi}{\delta}, \frac{\pi}{\delta}\right) = V_{\text{gerade}} - V_{\text{ungerade}}. \quad (3.54)$$

Während die Zerlegung (3.52) die Rechenzeit nur leicht erhöht, hat sie den großen Vorteil, nur noch eine Entwicklung über ein halb so großes Zeitintervall zu benötigen. Dies steigert die Präzision bzw. erlaubt die Reduzierung der Ordnung der Dysonentwicklung.

Es sei hier angemerkt, dass die Wahl des Zeitintervalls $\Delta t = \frac{2\pi}{\delta}$ auch aus anderen Gründen vorteilhaft erscheint; so wird z.B. die Zeitshift-Transformation (3.24) trivial und kann weggelassen werden. Wie aber die Praxis gezeigt hat, ist ausgerechnet $\Delta t = \frac{2\pi}{\delta}$ mit Problemen behaftet, so dass von der Zerlegung (3.52) abzuraten ist. Wir kommen darauf in Abschnitt 4.3.1 zurück.

Kapitel 4

Gatteroptimierung

Im letzten Kapitel haben wir uns mit der Lösung des Zeitentwicklungsoperators für das $\sigma_z \otimes \sigma_z$ - und das Mølmer-Sørensen-Gatter auf zwei Ionen beschäftigt. In diesem Kapitel wollen wir uns der Optimierung dieser Gatter zuwenden. Aber was soll eigentlich optimiert werden? Die Wirkung der beiden Gatter wird meist durch zwei idealisierte Operatoren beschrieben, denen die realen Gatter auch unter optimalen Bedingungen nur genähert zu entsprechen vermögen. Diesen intrinsischen Fehler zu minimieren, ist unser erstes Optimierungsziel. Im Weiteren werden wir auch versuchen, experimentelle Unzulänglichkeiten zu kompensieren.

4.1 Rahmenbedingungen

Das $\sigma_z \otimes \sigma_z$ - und das Mølmer-Sørensen-Gatter sind bereits in hoher Güte realisiert worden [10, 28]. Da eine experimentelle Umsetzung der hier optimierten Gatteroperationen auf bestehenden Experimenten aufbauen wird, wollen wir einige der für uns interessanten Aspekte näher beleuchten.

Die bisherige experimentelle Umsetzung via Impulsformung besteht im wesentlichen aus einem bichromatischen Rechteck-Impuls mit abgeweichten Kanten [10, 18]. D.h. die Standardwahl der Rabifrequenz Ω und der Phase ϕ sieht so aus, dass Ω die meiste Zeit einen konstanten Werte hat, und nur am Anfang und Ende des Impulses etwas geglättet wird. Die (zusätzliche) Phase ϕ wird die ganze Zeit konstant gehalten. Betrachtet man die beiden Laserstrahlen der bichromatischen Gatter als *ein* amplitudenmoduliertes Lichtfeld, dann ergibt sich das ideale ϕ aus der Forderung, dass wir in einem Knotenpunkt der Einhüllenden starten wollen.

Die beiden Frequenzanteile des bichromatischen Impulses werden mit Hilfe eines akustooptischen Modulators aus *einem* Laserstrahl gewonnen. Veränderungen der Rabifrequenz Ω und der Phase ϕ geschehen vor dem Passieren des akustooptischen Modulators, so dass beide Frequenzanteile immer mit dem gleichen Ω und ϕ behaftet sind.

Die angestrebte Verbesserung der Gatteroperationen erfolgt durch eine neue Wahl für die Ω_i und ϕ_i (wobei der Index wieder die Zugehörigkeit zum Zeitintervall t_i anzeigt). Neben dem Primärziel – der Optimierung des Zeitentwicklungsoperators – gilt es hierbei auch, auf eine möglichst unkomplizierte technische Umsetzung zu achten. Dies zu antizipieren, ist natürlich nur begrenzt möglich. Das Augenmerk lag daher darauf, unnötige Sprünge in den Variablen Ω und ϕ zu vermeiden, sowie – wie beim bisherigen Verfahren mit Impulsformung – ein zu abruptes Hochfahren der Rabifrequenz Ω von null auf den Maximalwert zu unterbinden. Die schon in (2.16) aufgeführte Nebenbedingung

$$\text{Strafe} = \mathfrak{S} = \sum_{i=1}^n (2\Omega_i - (\Omega_{i-1} + \Omega_{i+1}))^2 \quad \text{mit } \Omega_0 = \Omega_{n+1} = 0 \quad (4.1)$$

scheinen dies am ehesten zu gewährleisten, wobei es sich als günstig erwies, auch noch

$$(2\Omega_0 - (\Omega_{-1} + \Omega_1))^2 + (2\Omega_{n+1} - (\Omega_n + \Omega_{n+2}))^2 := \Omega_1^2 + \Omega_n^2$$

zur Strafe zu addieren. Die Nebenbedingungen (4.1) favorisieren auch kleine Rabifrequenzen – mehr dazu in der Fußnote¹. Für die Phase ϕ ist eine entsprechende Nebenbedingung zu setzen, nur mit offenen Endpunkten, da es keinen Grund gibt, warum ϕ am Anfang und am Ende gleich null sein sollte. Bei den meisten Berechnungen wurde es allerdings vorgezogen, die Umsetzung der Gatteroperation zu vereinfachen, indem von einer Variation der Phase abgesehen und ϕ als konstant gesetzt wurde. Für die Optimierung reicht es aus, nur die Rabifrequenz Ω zu variieren. Mathematisch ist ϕ aber weiterhin als Variable zu behandeln, da sonst der am Ende des Abschnitts 3.3 in (3.24) beschriebene Trick des Zeitshifts nicht durchführbar wäre.

4.2 Optimierungsansatz

In diesem Abschnitt geht es darum, unser Optimierungsziel in Gestalt einer Gewinnfunktion \mathfrak{G} mathematisch zu formulieren.

In den Abschnitten 1.4 und 1.5.1 haben wir die idealisierte Form des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters kennen gelernt. Dort erfolgte die Darstel-

¹Die Strafe \mathfrak{S} (4.1) entspricht der Summe der zweiten Ableitungen bzw. der Krümmungen. Da die Endpunkte auf null fixiert sind, haben hohe Werte der Rabifrequenz auch immer eine größere Krümmung zur Folge und sollten deshalb vom Computer gemieden werden. Sollte diese indirekte Bestrafung nicht reichen, kann \mathfrak{S} noch um einen Term $\sim \sum_{i=1}^n \Omega_i^2$ erweitert werden. Da dieser Term viel größer ausfällt als das bisherige \mathfrak{S} , sollten beide Terme durch eine während der Optimierung anpassbare Kopplungskonstante verbunden werden, die das Verhältnis der beiden Terme regelt.

lung mit den offenen Parametern ϑ, Φ (1.40) & (1.53), die wir nun fixieren

$$U_{\sigma_z \otimes \sigma_z} := e^{i\frac{\pi}{4}\sigma_z \otimes \sigma_z} = \begin{pmatrix} i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix} \otimes \mathbb{1}_{\text{Schwingung}} \quad (4.2)$$

$$U_{\text{MS}} := e^{i\frac{\pi}{4}\sigma_y \otimes \sigma_y} = \frac{1}{\sqrt{(2)}} \begin{pmatrix} 1 & 0 & 0 & -i \\ 0 & 1 & i & 0 \\ 0 & i & 1 & 0 \\ -i & 0 & 0 & 1 \end{pmatrix} \otimes \mathbb{1}_{\text{Schwingung}}. \quad (4.3)$$

Gemäß der Nomenklatur in Abschnitt 2.1 bezeichnen wir die Gatter (4.2) & (4.3) von nun ab als U_{Ziel} . Mit $U_{\text{Real}} := \prod_i U_i(\Omega_i, \phi_i)$ hingegen bezeichnen wir die Gatteroperation, welche sich unter Verwendung der Einstellungen Ω_i, ϕ_i ergeben würde.

Wir fordern nun *nicht* die exakte Übereinstimmung von U_{Real} mit U_{Ziel} . Erinnern wir uns an die Gewinnfunktion \mathfrak{G} (2.3) aus Abschnitt 2.1

$$\mathfrak{G} = |\langle U_{\text{Real}} | U_{\text{Ziel}} \rangle|^2. \quad (4.4)$$

Dort führten wir die Quadratur ein, um eine größere Klasse an U_{Real} als Lösungen aufzunehmen; nämlich all diejenigen, welche sich nur durch eine globale Phase von U_{Ziel} unterscheiden. Je größer die Klasse der erlaubten Gatteroperationen ist, desto höher ist die Wahrscheinlichkeit, eine gute Umsetzung zu finden.

Im Folgenden wollen wir die Klasse der erlaubten Gatteroperationen noch etwas größer fassen. Wir gehen dabei zuerst der Frage nach, wie der allgemeinste Operator U_{Erlaubt} aussieht, den wir als *gleichwertigen* Ersatz für U_{Ziel} akzeptieren können. Anschließend konstruieren wir eine Gewinnfunktion, die durch U_{Erlaubt} maximiert wird.

Wir beginnen mit der Feststellung, dass in den Schwingungszuständen keine für uns relevante Information gespeichert wird. Die gesamte uns interessierende Quanteninformation ist in den Spins² abgelegt. Das heißt aber nicht, dass wir beliebige Operationen auf der Schwingungsmode akzeptieren können. Wir müssen nämlich aufpassen, dass wir keine ungewollten Verschränkungen zwischen den Schwingungszuständen und den Spins erzeugen. Als erste Bedingung ist daher zu fordern, dass die Schwingungsmode und der Spin separierbar sind

$$U_{\text{Erlaubt}} = U_{\text{Spin}} \otimes U_{\text{Schwingung}}. \quad (4.5)$$

Als zweite Bedingung fordern wir, dass die Schwingungsmode nicht aufgeheizt werden darf, d.h. jede Zuweisung der Art $|n\rangle \rightarrow |n+k\rangle$ mit $k > 0$

² *Spin* ist in einem verallgemeinerten Sinn zu verstehen und umfasst jedes Zwei-Niveausystem.

ist für U_{Erlaubt} verboten. Da U_{Erlaubt} unitär und damit reversibel sein muss, folgt aus diesen beiden Bedingungen, dass auch eine Kühlung der Schwingungsmoden für U_{Erlaubt} nicht zulässig ist, was in der Fußnote³ gezeigt wird. Das allgemeinste U_{Erlaubt} unter diesen Bedingungen sieht somit wie folgt aus

$$U_{\text{Erlaubt}} = U_{\text{Spin}} \otimes \left(\begin{array}{ccccc} e^{i\xi_0} & 0 & 0 & 0 & \dots \\ 0 & e^{i\xi_1} & 0 & 0 & \dots \\ 0 & 0 & e^{i\xi_2} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \right)_{\text{Schwingung}}. \quad (4.6)$$

Jedem Schwingungszustand wird eine eigene, vom Spin unabhängige Phase ξ_i zugestanden. Dies bereitet keine Probleme und wir können dieses U_{Erlaubt} als gleichwertigen Ersatz für U_{Ziel} akzeptieren, sofern die beiden U_{Spin} übereinstimmen. Ähnlich wie in (4.4) ist es nun unser Bestreben, durch Quadrieren eine Gewinnfunktion zu generieren, die unabhängig von den Phasen ξ_i ist. Da das Quadrieren aber nur gegen *eine* Phase hilft, muss jede Phase ξ_i individuell bekämpft werden. Deshalb berechnen wir für jeden Schwingungszustand $|n = i\rangle$ eine eigene Gewinnfunktion \mathfrak{G}_i

$$\mathfrak{G}_i = |\langle U_{\text{Real}} \cdot P_i | U_{\text{Ziel}} \cdot P_i \rangle|^2. \quad (4.7)$$

Dabei ist P_i ein Projektor, der den i ten Schwingungszustand heraus projiziert

$$P_i = \mathbb{1}_{\text{Spins}} \otimes \mathcal{P}_{\text{Schwingung}}^{(i)}, \quad (4.8)$$

wobei

$$\mathcal{P}_{\text{Schwingung}}^{(i)} = \begin{pmatrix} p_{11} & p_{12} & \dots \\ p_{21} & p_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \text{mit } p_{jk} = \begin{cases} 1 & \text{wenn } j = k = i \\ 0 & \text{sonst,} \end{cases}$$

d.h. $\sum_{i=1}^{\infty} P_i = \mathbb{1}$. Die Gewinnfunktion \mathfrak{G} erhalten wir, indem wir alle Bewertungen \mathfrak{G}_i (4.7) addieren

$$\mathfrak{G} = \sum_{i=1}^m \mathfrak{G}_i = \sum_{i=1}^m |\langle U_{\text{Real}} \cdot P_i | U_{\text{Ziel}} \cdot P_i \rangle|^2. \quad (4.9)$$

³Es soll gezeigt werden, dass die beiden Bedingungen im Haupttext verbieten, dass U_{Erlaubt} die Schwingungsmoden kühlt. Wegen der Separabilität von Schwingungsmoden und Spin reicht es aus, die Schwingungszustände zu betrachten.

Angenommen, U_{Erlaubt} würde die Abbildung $|n\rangle \rightarrow |n-k\rangle$ vornehmen. Dies schliesse die Abbildung $|n-k\rangle \rightarrow |n-k\rangle$ aus, da U_{Erlaubt} anderenfalls nicht reversibel wäre ($|?\rangle \leftarrow |n-k\rangle$). Da nun $|n-k\rangle \rightarrow |n-k\rangle$ ebenso ausgeschlossen wäre wie die Abbildung in einen höheren Schwingungszustand $|n-k\rangle \rightarrow |n-k+l\rangle$ (zweite Bedingung), wäre nur noch eine Abbildung in einen tieferen Schwingungszustand möglich $|n-k\rangle \rightarrow |n-k-l\rangle$. Dies verschiebt aber nur das ursprüngliche Problem. Dieses Verschieben des Problems ist jedoch nicht beliebig oft wiederholbar, da die Schwingungsmoden mit $|0\rangle$ nach unten beschränkt ist.

Da U_{Ziel} auf der Schwingungsmode der $\mathbb{1}$ entspricht, könnten die P_i wahlweise auch links der U stehen. Wie man leicht überprüft, nimmt \mathfrak{G} für alle $U_{\text{real}} = U_{\text{Erlaubt}}$ ⁴ den Maximalwert an.

Die Summation in (4.9) läuft bis m . Welchen Wert haben wir für m zu wählen? Zum Einen müssen wir die unendliche Anzahl der Schwingungszustände auf n beschränken, um ihrer numerisch Herr werden zu können. Es wäre aber nun vollkommen unangebracht, $m = n$ zu setzen. Da wir meist bemüht sein werden, die Schwingungsmode auf den Grundzustand zu kühlen, sind höher angeregte Zustände nur „Ausreißer“, die selten auftreten. Die Güte eines Gatters nach seinem Verhalten bei Fällen zu beurteilen, die fast nie eintreten, ist wenig sinnvoll (jedenfalls nicht, wenn diese seltenen Fälle genauso gewichtet werden, wie die viel wahrscheinlicheren). Der Wert von m ist daher eher klein anzusetzen. Alternativ kann man auch einen Operator W zur Gewichtung der Schwingungszustände definieren

$$W = \mathbb{1}_{\text{Spins}} \otimes \mathcal{W}_{\text{Schwingung}}, \quad (4.10)$$

wobei

$$\mathcal{W}_{\text{Schwingung}} = \begin{pmatrix} p^1 & 0 & 0 & 0 & \dots \\ 0 & p^2 & 0 & 0 & \dots \\ 0 & 0 & p^3 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad \text{mit } p^{i+1} < p^i \leq 1$$

und damit die Gewinnfunktion \mathfrak{G} wie folgt festlegen:

$$\mathfrak{G} = \sum_{i=1}^m |\langle U_{\text{Real}} \cdot P_i \cdot W | U_{\text{Ziel}} \cdot P_i \cdot W \rangle|^2. \quad (4.11)$$

4.3 Fehlertoleranz

Es gibt viele Störungen, denen die Gatter ausgesetzt sind. Ein Problem ist z.B. die Fluktuation der Laserfrequenz ω_L . Auch wenn Letztere nur einen sehr kleinen Fehler aufweist, kann der relative Fehler im mitrotierenden Bezugssystem – also nach der Drehwellennäherung (siehe Seite 9) – schon recht groß ausfallen, da hier nur die kleine Differenz der Laserfrequenz ω_L zur Frequenz ω_0 des Übergangs $|g\rangle \leftrightarrow |e\rangle$ betrachtet wird. Durch einen geeigneten Wechsel des Bezugssystems lässt sich dieser Fehler mathematisch gleichwertig als ein störendes Magnetfeld beschreiben (siehe Seite 10). Diese beiden Fehler können beim $\sigma_z \otimes \sigma_z$ -Gatter z.B. durch ein *Spinecho* unterdrückt werden. Dieses Verfahren wird in Abschnitt 4.4 beschrieben. Eine Optimierung

⁴ U_{Erlaubt} repräsentiert eine ganze Klasse an Operatoren. $U_{\text{Real}} = U_{\text{Erlaubt}}$ soll heißen, dass U_{Real} ein beliebiges Element aus U_{Erlaubt} ist.

auf Fehlerunanfälligkeit durch den Computer scheint auch möglich, wäre aber mit dem hier gewählten Ansatz relativ aufwendig, da wir für jede Variation der Laserfrequenz eine eigene Dysonentwicklung bereitstellen müssten (siehe Abschnitt 3.3). Wir beschränken uns hier deshalb auf Fehler, die durch die leicht zugänglichen Variablen Ω_i und ϕ_i verursacht werden. Erinnern wir uns: Wir bauen die Gatter gemäß

$$U_{\text{Gatter}} = \prod_i U_i(\Omega_i, \phi_i) \quad (4.12)$$

auf. Ω_i und ϕ_i sind zugleich die Variablen, die wir zum Optimieren der Gatteroperationen heranziehen. Vom mathematischen Standpunkt sollte dies ein Vorteil sein: Durch den Optimierungsprozess ist die Güte der Gatteroperation für jedes Ω_i und ϕ_i ein Maximum. Somit verschwindet die erste Ableitung nach diesen Werten, und ein Fehler sollte sich erst in der zweiten Ordnung bemerkbar machen.

Zur Erzeugung von Fehlertoleranz müssen wir uns vorab Gedanken machen, welches Fehlermodell wir zugrunde legen wollen. Hier behandeln wir einfache Modelle, für die wir sowohl die physikalischen Auswirkungen als auch die algorithmischen Umsetzungen beleuchten wollen.

4.3.1 Phasenabhängigkeit der Ergebnisse

In unseren Experimenten können wir die Phase ϕ_i relativ gut einstellen. Trotzdem wäre es sehr viel komfortabler, wenn wir uns über den Wert der Phase beim Start des Experimentes keine Gedanken machen müssten. Wir setzen deshalb folgendes Fehlermodell an

$$\phi_i = \phi_{\text{Start}} + \Delta\phi_i. \quad (4.13)$$

Alle ϕ_i haben den gleichen Fehler, nämlich das unbestimmte ϕ_{Start} . Diesem Fehler legen wir nicht die Bedingung auf, klein zu sein. Wir gehen davon aus, dass wir während des Experimentes die Phase stabil halten können, so dass wir individuelle Fehler der ϕ_i nicht berücksichtigen. Wir erklären es fortan zu unserem Optimierungsziel, Unabhängigkeit von der Phase ϕ_{Start} zu erreichen; d.h. die Gatteroperationen sollen für alle Werte von ϕ_{Start} annähernd gleich gute Ergebnisse liefern.

Bei den meisten Berechnungen wurde sogar gänzlich auf eine Variation der Phase verzichtet und $\phi_i = \phi_{\text{Start}}$ gesetzt, was die anschließende experimentelle Umsetzung wesentlich einfacher gestaltet. Von einer zufälligen, aber konstanten Phase unabhängig zu werden, dürfte nicht allzu schwierig ausfallen: Während die Realisierung der Gatter durch Rechteck-Impulse sich noch als recht sensitiv für die Phase erweist, reicht bereits einfache Impulsformung, um weitestgehend unabhängig von der Phase zu werden. Umso größer war die Überraschung, als der Computer bei den ersten Tests offensichtlich nicht

in der Lage zu sein schien, dieses Problem zu meistern. Es war dem Computer nicht nur unmöglich, von der Phase unabhängig zu werden, sondern er scheiterte sogar schon bei gewissen, fest vorgegebenen Phasen. Das Problem wurzelte in der damaligen Wahl des Zeitintervalls Δt , nach dem die Rabifrequenz Ω verändert wird

$$\Omega(t) \rightarrow \Omega_i \quad \text{für} \quad (i-1) \cdot \Delta t \leq t < i \cdot \Delta t. \quad (4.14)$$

Dieses war derart gewählt, dass

$$\Delta t = \frac{2\pi}{\delta}, \quad (4.15)$$

wobei $\pm\delta$ der Winkelfrequenz der Laser im mitrotierenden Bezugssystem entspricht. Dies schien eine sehr sinnvolle Wahl, was im letzten Teil von Abschnitt 3.5.1 dargelegt wurde.

Im Folgenden versuchen wir nachzuvollziehen, wieso die Festlegung von Δt gemäß (4.15) es vereitelt, die Gatteroperationen von ϕ unabhängig werden zu lassen. Dazu müssen wir erst den positiven Effekt der Impulsformung verstehen. Die Erklärung hierfür stammt von Christian F. Roos. In seiner Veröffentlichung [18] entwickelt er einen effektiven Hamilton-Operator

$$H_{\text{eff}} \sim S_z \cos \psi - S_y \sin \psi \quad \text{mit} \quad \psi = \frac{4\Omega}{\delta} \sin \phi \quad (4.16)$$

$$S_{y/z} = \mathbb{1} \otimes \sigma_{y/z} + \sigma_{y/z} \otimes \mathbb{1}.$$

Für $\phi = 0$ sind die Eigenvektoren des H_{eff} identisch mit denen von S_z . Dies gilt nicht mehr, wenn $\phi \neq 0$; dann kommt S_y als „störendes“ Element hinzu. Bei der Impulsformung wird der Wert von Ω am Anfang langsam hochgefahren und am Ende langsam wieder abgesenkt. Bei sehr kleinen Werten von Ω geht der Wert von $\sin \psi$ ebenfalls gegen null, so dass die Eigenvektoren von H_{eff} und S_z wieder zusammenfallen. Die entscheidende Idee ist nun, dass die ursprünglichen Eigenvektoren von S_z dem Hamilton-Operator *adiabatisch* folgen, wenn Ω hoch- und wieder runtergefahren wird. Die ursprünglichen Eigenvektoren von S_z bleiben also immer Eigenvektoren des Hamilton-Operators und werden am Ende wieder als Eigenvektoren von S_z „abgeliefert“.

Die Feststellung, dass Eigenvektoren wieder auf Eigenvektoren abgebildet werden, reicht natürlich nicht für einen rigorosen Beweis der Phasenunabhängigkeit aus. An dieser Stelle wollen wir es aber dabei belassen und uns der Frage zuwenden, warum sich das System nach der Diskretisierung gemäß (4.14) und (4.15) offensichtlich der obigen Erklärung zu entziehen vermag. Am Beispiel des klassischen harmonischen Oszillators werden wir sehen, dass das soeben benutzte Adiabatentheorem nicht gilt, wenn im Geiste von (4.14) und (4.15) diskretisiert wird. Betrachten wir eine schwingende Masse an einer Feder. Bewegen wir nun rasch den Aufhängepunkt der Feder entlang der

Schwingungsachse, so wird dies nicht ohne Einfluss auf die Schwingung bleiben. Bewegen wir den Aufhängepunkt hingegen langsam und gleichmäßig, bleibt die Masse in ihrem Schwingungsverhalten weitestgehend ungestört. Nun kommt der für uns relevante Fall: Wir verschieben den Aufhängepunkt um die Strecke s in n diskreten, sprunghaften Schritten zu je $\frac{s}{n}$ mit der prinzipiellen Näherung $n \rightarrow \infty$. Der i te Schritt verändert die potentielle Energie E um

$$\Delta_i E = E(x_i + \frac{s}{n}) - E(x_i) = \frac{1}{2}D \cdot (x_i + \frac{s}{n})^2 - \frac{1}{2}D \cdot x_i^2 \simeq D \frac{s}{n} x_i, \quad (4.17)$$

wobei x_i die Position der Masse relativ zum Aufhängepunkt beim i ten Schritt ist. Die Energieänderung nach n Schritten ist

$$\Delta E = D \cdot s \cdot \frac{1}{n} \sum_{i=1}^n x_i. \quad (4.18)$$

Ist die Abfolge der Schritte nicht in Resonanz mit der Schwingung des Oszillators, so wird $\frac{1}{n} \sum_{i=1}^n x_i = 0$ und somit $\Delta E = 0$. Bei Resonanz hingegen sind alle $x_i = x_\phi = \text{const}$ und somit $\Delta E = D \cdot s \cdot x_\phi$ trotz $n \rightarrow \infty$.

Jetzt müssen wir den Übergang vom harmonischen Oszillator zu unseren Gattern bewerkstelligen. Dazu machen wir uns zuerst klar, dass die Feder-schwingung entlang einer Achse trivial zu erweitern ist auf den Fall, dass die Masse eine Kreisbahn um den Aufhängepunkt der Feder beschreibt. Dieses Bild ist dann leicht auf das *Bloch-Vektor-Bild* eines Zwei-Niveau-Atoms in einem Laserfeld zu übertragen. Im letzteren präzediert der Blochvektor des Atoms um den Vektor des Laserfeldes, der beschrieben werden kann durch

$$\vec{\Omega}_{\text{Bloch}} = \begin{pmatrix} \Omega \\ 0 \\ \Delta \end{pmatrix} \quad \text{mit} \quad \Delta = \omega_{\text{Licht}} - \omega_{\text{Atom}}. \quad (4.19)$$

Für $\Omega = 0$ zeigt $\vec{\Omega}_{\text{Bloch}}$ in z -Richtung. Die Blochvektoren des Atoms, die entlang der z -Achse liegen, können $\vec{\Omega}_{\text{Bloch}}$ adiabatisch folgen, wenn Ω langsam und gleichmäßig erhöht wird. Erfolgt die Veränderung von Ω hingegen sprunghaft und in Resonanz, dann hat dies Auswirkungen auf die Präzession des Atoms, die von dessen Phasenlage zu den Zeitpunkten der Sprünge abhängt. Die Präzession erfolgt mit der Winkelfrequenz $|\vec{\Omega}_{\text{Bloch}}|$. Bei unseren Gattern gilt

$$|\Delta| = |\pm \delta| \gg |\Omega| \quad \text{und somit} \quad |\vec{\Omega}_{\text{Bloch}}| \approx |\delta|. \quad (4.20)$$

Bei dem letzten Argument muss man natürlich vorsichtig sein. Wenn wir $\vec{\Omega}_{\text{Bloch}}$ als völlig unabhängig von der Rabi-Frequenz deklarieren, hätte eine Erhöhung letzterer auch keinen Einfluss auf das Schwingungsverhalten des Atoms. Letztlich ist das einzelne Atom im monochromatischen Laserfeld aber

auch nur ein vereinfachtes Aushilfsmodell für unser wirkliches System. Leider kann hier kein anschauliches Bild für unseren Fall mit zwei Ionen in einem bichromatischen Lichtfeld angegeben werden. Die Erklärung muss sich deshalb auf folgende drei Punkte stützen

1. Christian Roos konnte unter Verwendung des Adiabatentheorem zeigen, dass durch Impulsformung Phasenunabhängigkeit erreicht werden kann.
2. Bereits bei einfachen Modellen kann das Adiabatentheorem seine Gültigkeit verlieren, wenn es sprunghafte Veränderungen gibt, die mit dem System in Resonanz sind.
3. Der Computer ist nicht in der Lage, den Einfluss der Phase auszuschalten, wenn die Zeitintervalle zur Veränderung der Rabi-Frequenz in Resonanz mit der Laserfrequenz im mitrotierenden Bezugssystem sind. Dies entspricht genau der Vorhersage des einfachen Modells eines Zwei-Niveau-Atoms in einem Laserfeld.

Folgt man dem Modell des Zwei-Niveau-Atoms im Laserfeld, so scheint die naheliegendste Wahl für die Zeitintervalle eine halbe Schwingungsperiode $\Delta t = \frac{1}{2} \frac{2\pi}{\delta}$ zu sein. Aufeinander folgende Veränderungen hätten dann entgegengesetzte Vorzeichen und sollten sich gut wegheben. Beim $\sigma_z \otimes \sigma_z$ -Gatter gilt $\frac{1}{2\delta} \approx \frac{1}{\nu}$ und daher $\Delta t = \frac{1}{2} \frac{2\pi}{\delta} = \frac{2\pi}{\nu}$. Diese Wahl für die Zeitintervalle ist somit in Resonanz mit der Fallenfrequenz. Das Ausbleiben von Problemen bei dieser Einstellung zeigt, dass die Schwingung des Lichtfeldes und nicht die vergleichsweise kleine Schwingung der Ionen der entscheidende Faktor ist.

Algorithmische Umsetzung

Bisher wurde nur über die notwendige Bedingung zum Erreichen der Phasenunabhängigkeit berichtet aber nicht, wie die Umsetzung auf dem Computer zu erfolgen hat. Letztere gestaltet sich derart, dass wir die Gewinnfunktion \mathfrak{G} aus (4.11) durch eine Summe solcher Gewinnfunktionen zu unterschiedlichen, aber nach wie vor konstanten Phasen ϕ_k ersetzen

$$\mathfrak{G}_\Sigma = \sum_{k=1}^m \mathfrak{G}(\phi_k). \quad (4.21)$$

Der „einfachste“ Weg, \mathfrak{G}_Σ zu optimieren, besteht dann in der Erzeugung einer phasenunabhängigen Gatteroperation. In der Praxis hat es sich meist als ausreichend erwiesen, drei verschiedene Phasen zu wählen. Zum Schluss des Optimierungsprozesses kann man diese Anzahl dann noch einmal erhöhen. Dies sollte aber nicht zu früh passieren, da die Rechenzeit linear mit der Anzahl der gleichzeitig optimierten Phasen wächst.

4.3.2 Fluktuationen in der Rabifrequenz

Für die Rabifrequenz Ω_i betrachten wir zwei verschiedene Fehlermodelle:

1. Eine für alle Ω_i gleiche prozentuale Abweichung vom Sollwert

$$\Omega_i \rightarrow (1 \pm \varepsilon) \cdot \Omega_i \quad (4.22)$$

2. Individuelle Abweichungen für jedes Ω_i , die unkorreliert, normalverteilt und prozentual erfolgen

$$\Omega_i \rightarrow (1 \pm \varepsilon_i) \cdot \Omega_i \quad (4.23)$$

Die beiden Modelle unterscheiden sich in der Zeitskala der Fluktuation der Ω_i . Im ersten Fall ist die Qualität des Fehlers gleichbleibend für die gesamte Gatteroperation, während sie im zweiten Fall mit jedem Ω_i anders ausfällt. Beide Modelle sind algorithmisch leicht umzusetzen. Unangenehm und nur schwer simulierbar wären Fehler, die für ein paar aufeinander folgende Ω_i von der gleichen Art sind, aber nicht für alle. Die zweite Annahme in beiden Modellen ist ein prozentualer, also linear mit Ω_i wachsender Fehler. Dies ist nicht zwingend und es stellt auch algorithmisch kein Problem dar, die Modelle diesbezüglich zu umzugestalten.

Algorithmische Umsetzung

Das erste Fehlermodell mit seinem zeitlich konstanten Fehler dürfte für die Praxis das wesentlich bedeutendere sein. Der Optimierungsansatz für dieses Fehlermodell ist von der Idee her identisch mit dem zum Erreichen der Phasenunabhängigkeit (4.21): Die Gewinnfunktion \mathfrak{G} wird durch eine Summe von Gewinnfunktionen mit unterschiedlichen Fehlern f_k ersetzt

$$\mathfrak{G}_\Sigma = \sum_{k=1}^m \mathfrak{G}(f_k). \quad (4.24)$$

Dem zweiten Fehlermodell liegt die Idee zugrunde, dass der Fehler beim „Umschalten“ von Ω_i auf Ω_{i+1} zustande kommt. Diese Fehler werden als unkorreliert angenommen, was sicherlich nicht 100%ig den physikalischen Realitäten entspricht. So soll denn auch nicht verschwiegen werden, dass ein starkes Motiv für dieses Modell die Möglichkeit einer eleganten mathematischen Umsetzung war. Beim ersten Fehlermodell erfolgt die Optimierung gemäß Gleichung (4.24) exemplarisch an m Fehlern, wodurch die Rechenzeit sich auch gleich um den Faktor m erhöht. Beim zweiten Fehlermodell können *alle* Fehler gleichzeitig mit nur *einer* Gewinnfunktion (bzw. mit einer Gewinnfunktion mit nur einem Summanden) abgehandelt werden. Die Gewichtung der Wahrscheinlichkeit eines Fehlers erfolgt dabei mit einer Gauß-

Verteilung⁵. Um die Attraktivität des Modells zu sehen, müssen wir uns zuerst an (2.8) erinnern

$$U_{\text{Gatter}} = U_n \cdot U_{n-1} \cdot U_{n-2} \cdots U_1. \quad (4.25)$$

Wenn nun jedes dieser U_i als Summe aller möglichen fehlerbehafteten $U_i(\mathbf{f}_k)$ mit der Wahrscheinlichkeit $\rho(\mathbf{f}_k)$ gewichtet dargestellt werden kann,

$$U_i \rightarrow U_i^{(\Sigma \mathbf{f})} = \sum_{k=1}^{\infty} \rho(\mathbf{f}_k) \cdot U_i(\mathbf{f}_k), \quad (4.26)$$

dann ist in

$$U_{\text{Gatter}}^{(\Pi \Sigma \mathbf{f})} = U_n^{(\Sigma \mathbf{f})} \cdot U_{n-1}^{(\Sigma \mathbf{f})} \cdot U_{n-2}^{(\Sigma \mathbf{f})} \cdots U_1^{(\Sigma \mathbf{f})} \quad (4.27)$$

jede mögliche Fehlerkombination korrekt gewichtet enthalten. (Hierfür ist es notwendig, dass die Fehler *unkorreliert* sind und somit jeder Fehler mit gleicher Wahrscheinlichkeit an jeden anderen Fehler koppelt)

Die Frage ist nun, ob sich $U_i^{(\Sigma \mathbf{f})}$ gemäß (4.26) verwirklichen lässt. Glücklicherweise läuft die unendliche Summe nur auf die Bildung einiger Mittelwerte der Art $\overline{(\Omega_i)^n}$ hinaus. Jedes U_i wird über eine Dysonentwicklung (z.B. (3.47)) berechnet, die sich darstellen lässt als

$$U_i = \sum_{j=1}^n (\Omega_i)^j \cdot \mathcal{M}_j. \quad (4.28)$$

Damit folgt

$$\begin{aligned} U_i^{(\Sigma \mathbf{f})} &= \sum_{k=1}^{\infty} \rho(\mathbf{f}_k) \cdot U_i(\mathbf{f}_k) \\ &= \sum_{k=1}^{\infty} \rho(\mathbf{f}_k) \cdot \sum_{j=1}^n (\Omega_i(\mathbf{f}_k))^j \cdot \mathcal{M}_j \\ &= \sum_{j=1}^n \left(\sum_{k=1}^{\infty} \rho(\mathbf{f}_k) \cdot (\Omega_i(\mathbf{f}_k))^j \right) \cdot \mathcal{M}_j \\ &= \sum_{j=1}^n \overline{(\Omega_i)^j} \cdot \mathcal{M}_j. \end{aligned} \quad (4.29)$$

Da wir normalverteilte Fehler annehmen, kann $\overline{(\Omega_i)^j}$ analytisch berechnet werden

$$\overline{(\Omega_i)^j} = \int_{-\infty}^{\infty} \frac{1}{\sigma \sqrt{2\pi}} (\Omega_i^f)^j \exp\left[-\frac{1}{2} \left(\frac{\Omega_i^f - \bar{\Omega}_i}{\sigma}\right)^2\right] d\Omega_i^f. \quad (4.30)$$

⁵Es wird eine Gauß-Verteilung um den Sollwert der Rabifrequenz angenommen. Diese Gauß-Verteilung liefert mit sehr geringer Wahrscheinlichkeit auch negative Werte. Dies ist sicherlich nicht richtig aber andererseits auch wieder so selten, dass dieser Einwand vernachlässigt werden kann

Dieses Integral lässt sich für jedes j in ein Polynom j ten Grades in $\bar{\Omega}_i = \Omega_i$ umformen, wodurch sich die Berechnungen so weit vereinfachen lassen, dass Optimierungen mit diesem Fehlermodell ähnlich schnell vonstatten gehen wie Optimierungen ohne Fehlermodell.

Auswirkungen der Fehler

Während im ersten Fehlermodell die Fehler für alle Ω_i immer von der gleichen Art sind, sind diese im zweiten Fehlermodell unkorreliert. Man sollte daher vermuten, dass die Fehler sich im zweiten Modell z.T. gegenseitig wegheben und damit weit weniger ins Gewicht fallen als beim ersten Modell. So mag es denn auf den ersten Blick überraschen, dass die Gatter sich als weniger anfällig für Fehler der ersten Art erweisen als für die der zweiten Art.

Eine anschauliche Erklärung erhalten wir beim $\sigma_z \otimes \sigma_z$ -Gatter mit Hilfe des Phasenraumdiagramms (siehe Abbildung 1.4). Die Basiszustände – die „richtige“ Wahl vorausgesetzt, – durchlaufen geschlossene Bahnen. Es sind diese geschlossenen Bahnen, die dafür sorgen, dass die Fehler im ersten Modell sich am Ende fast vollständig wegheben. Als primärer Effekt bleibt nur eine entsprechende Vergrößerung oder Verkleinerung der von den Graphen eingeschlossenen Flächen. Dies schlägt sich in einer Abweichung der aufgesammelten geometrischen Phase nieder.

Bedauerlicherweise muss am Ende dieses Abschnitts verkündet werden, dass sich durch die Einbindung der hier beschriebenen Fehlermodelle keine signifikante Verbesserung der Fehlerresistenz der Gatter gegenüber Schwankungen der Rabifrequenz feststellen ließ. Es konnte lediglich bestätigt werden, dass sich kleine Abweichungen der Rabifrequenz nur quadratisch in der Güte der Gatter bemerkbar machen.

4.4 Spinecho

Das *Spinecho* ist eine Technik aus der NMR-Spektroskopie zur Refokussierung auseinander präzedierter Spins. Die dem Spinecho zugrunde liegende Idee kann auch auf andere Bereiche der Physik ausgedehnt werden. Wir wollen das Spinecho nutzen, um die unweigerlich in realen Gatteroperationen auftretenden Fehler zu reduzieren. Wir beginnen unsere Ausführungen sehr allgemein. In diesem Sinne führen wir den Spinecho-Operator \mathfrak{S} ein und legen dessen Eigenschaften erst später sukzessive fest, da uns verschiedene Arten der Realisierung zur Verfügung stehen. Unser Ziel ist die Reduktion von Fehlern. Letztere drücken wir als Differenz der realen Gatteroperation U_{Real} von ihrem theoretischen, perfekten Pendant U_{Perfekt} aus.

Wir machen folgende Zerlegung:

$$U_{\text{Real}} = U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_- \quad (4.31)$$

$$\mathfrak{S} \cdot U_{\text{Real}} \cdot \mathfrak{S}^{-1} = U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ - \varepsilon_- \tilde{\Delta}_-. \quad (4.32)$$

Eine Zerlegung dieser Art ist bei gegebenen U_{Real} und \mathfrak{S} immer möglich

$$\begin{aligned}\varepsilon_+ \tilde{\Delta}_+ &= \frac{1}{2}(U_{\text{Real}} + \mathfrak{S} \cdot U_{\text{Real}} \cdot \mathfrak{S}^{-1}) - U_{\text{Perfekt}} \\ \varepsilon_- \tilde{\Delta}_- &= \frac{1}{2}(U_{\text{Real}} - \mathfrak{S} \cdot U_{\text{Real}} \cdot \mathfrak{S}^{-1}).\end{aligned}\quad (4.33)$$

Die durch ε_{\pm} suggerierte Kleinheit der Terme $\varepsilon_+ \tilde{\Delta}_+$ und $\varepsilon_- \tilde{\Delta}_-$ gilt hingegen nicht in voller Allgemeingültigkeit. An einen *sinnvollen* Operator \mathfrak{S} stellen wir nun die Anforderung, dass

$$|\varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_-| \underbrace{\approx}_{\text{gleiche Größenordnung}} |\varepsilon_+ \tilde{\Delta}_+ - \varepsilon_- \tilde{\Delta}_-| \quad (4.34)$$

gilt. Da $\varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_- = U_{\text{Real}} - U_{\text{Perfekt}}$ sollte dieser Wert von Haus aus klein sein, und Selbiges fordern wir jetzt auch von $\varepsilon_+ \tilde{\Delta}_+ - \varepsilon_- \tilde{\Delta}_-$. Hat man einen Operator gefunden, der dies erfüllt, dann verstehen wir unter einem *Spinecho* die folgende Sequenz

$$\begin{aligned}\mathfrak{S} \cdot U_{\text{Real}} \cdot \mathfrak{S}^{-1} \cdot U_{\text{Real}} &= (U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ - \varepsilon_- \tilde{\Delta}_-) \cdot \\ &\quad (U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_-) \\ &= U_{\text{Perfekt}} \cdot U_{\text{Perfekt}} + \\ &\quad \varepsilon_+ \{U_{\text{Perfekt}}, \tilde{\Delta}_+\} + \varepsilon_- [U_{\text{Perfekt}}, \tilde{\Delta}_-] + O(\varepsilon_{\pm}^2),\end{aligned}\quad (4.35)$$

wobei $[\dots, \dots]$ den Kommutator und $\{\dots, \dots\}$ den Antikommutator repräsentiert. Da unsere Gatter problemlos aus zwei „halben“ Gattern aufgebaut werden können

$$U(\varphi) = U\left(\frac{\varphi}{2}\right) \cdot U\left(\frac{\varphi}{2}\right) \quad \text{z.B.} \quad e^{i\varphi\sigma_z \otimes \sigma_z} = e^{i\frac{\varphi}{2}\sigma_z \otimes \sigma_z} \cdot e^{i\frac{\varphi}{2}\sigma_z \otimes \sigma_z}, \quad (4.36)$$

vermögen wir mit der Spinecho-Sequenz (4.35) jedes Gatter – mit kleinem Fehler – zu erzeugen.

Vergleichen wir nun (4.35) mit

$$\begin{aligned}U_{\text{Real}} \cdot U_{\text{Real}} &= (U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_-) \cdot \\ &\quad (U_{\text{Perfekt}} + \varepsilon_+ \tilde{\Delta}_+ + \varepsilon_- \tilde{\Delta}_-) \\ &= U_{\text{Perfekt}} \cdot U_{\text{Perfekt}} + \\ &\quad \varepsilon_+ \{U_{\text{Perfekt}}, \tilde{\Delta}_+\} + \varepsilon_- \{U_{\text{Perfekt}}, \tilde{\Delta}_-\} + O(\varepsilon_{\pm}^2).\end{aligned}\quad (4.37)$$

Wenn wir die Terme $O(\varepsilon_{\pm}^2)$ vernachlässigen, so besteht der einzige Unterschied der Spinecho-Sequenz gegenüber (4.37) in einer Ersetzung von $\varepsilon_- \{U_{\text{Perfekt}}, \tilde{\Delta}_-\}$ durch $\varepsilon_- [U_{\text{Perfekt}}, \tilde{\Delta}_-]$.

Besonders wirkungsvoll ist ein Spinecho, wenn

$$|\varepsilon_- \tilde{\Delta}_-| \gg |\varepsilon_+ \tilde{\Delta}_+| \quad \text{und} \quad [U_{\text{Perfekt}}, \tilde{\Delta}_-] \approx 0. \quad (4.38)$$

Einen solchen Fall wollen wir uns nun beim $\sigma_z \otimes \sigma_z$ -Gatter ansehen. Ein Studium der real erzeugbaren Gatter zeigt, dass der größte intrinsische Fehler durch den Operator

$$\tilde{\Delta}_{\text{gro\ss}} = |e, g\rangle \langle g, e| + |g, e\rangle \langle e, g| \quad (4.39)$$

beschrieben werden kann. Wir nähern das reale Gatter daher durch⁶

$$\begin{aligned} U_{\text{Real}}\left(\frac{\varphi}{2}\right) &\approx U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right) + \varepsilon \tilde{\Delta}_{\text{gro\ss}} \\ &= e^{i\frac{\varphi}{2}\sigma_z \otimes \sigma_z} + \varepsilon \tilde{\Delta}_{\text{gro\ss}}. \end{aligned} \quad (4.40)$$

Für den Spinecho-Operator \mathfrak{S} wählen wir

$$\mathfrak{S} = \mathfrak{S}_{1z} := e^{i\frac{\pi}{2}\sigma_z^{(2)}} = i\sigma_z^{(2)} \quad \text{mit} \quad \sigma_z^{(2)} = \mathbb{1} \otimes \sigma_z. \quad (4.41)$$

Damit gilt

$$\mathfrak{S}_{1z} \cdot \left(U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right) + \varepsilon \tilde{\Delta}_{\text{gro\ss}} \right) \cdot \mathfrak{S}_{1z} = U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right) - \varepsilon \tilde{\Delta}_{\text{gro\ss}}. \quad (4.42)$$

Dies ergibt für die Spinecho-Sequenz

$$\begin{aligned} \mathfrak{S}_{1z} \cdot U_{\text{Real}}\left(\frac{\varphi}{2}\right) \cdot \mathfrak{S}_{1z}^{-1} \cdot U_{\text{Real}}\left(\frac{\varphi}{2}\right) &\approx U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right) \cdot U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right) + \\ &\quad \underbrace{\varepsilon [U_{\text{Perfekt}}\left(\frac{\varphi}{2}\right), \tilde{\Delta}_{\text{gro\ss}}]}_{=0} \\ &= U_{\text{Perfekt}}(\varphi). \end{aligned}$$

Zum gleichen Ergebnis – wenn auch nicht so leicht zu erkennen – wären wir gelangt, hätten wir statt \mathfrak{S}_{1z} den Operator

$$\mathfrak{S}_{xy} := e^{-i\frac{\pi}{2}(\sigma_x^{(1)} + \sigma_y^{(2)})} = -\sigma_x \otimes \sigma_y \quad (4.43)$$

gewählt. Der Operator \mathfrak{S}_{xy} hat aber noch den Vorteil, nicht nur den größten intrinsischen Fehler zu bekämpfen, sondern auch den Großteil der externen Störungen zu beheben, die durch den Term $\sim S_z$ in (1.30) verursacht werden.

Zusammenfassend lässt sich sagen, dass ein Spinecho gewisse Fehler zu beheben vermag, während andere Fehler verstärkt oder gar neu erzeugt ($\sim O(\varepsilon_{\pm}^2)$) werden können. Nun scheint es eine nahe liegende Frage zu sein, ob man nicht noch bessere Ergebnisse erzielen kann, wenn man mehrere Spinechos miteinander kombiniert. Das sieht dann z.B. so aus:

$$\mathfrak{U}\left(\frac{\varphi}{2}\right) = \mathfrak{S}^{(1)} \cdot U_{\text{Real}}\left(\frac{\varphi}{4}\right) \cdot \mathfrak{S}^{(1)-1} \cdot U_{\text{Real}}\left(\frac{\varphi}{4}\right) \quad (4.44)$$

$$\mathfrak{U}(\varphi) = \mathfrak{S}^{(2)} \cdot \mathfrak{U}\left(\frac{\varphi}{2}\right) \cdot \mathfrak{S}^{(2)-1} \cdot \mathfrak{U}\left(\frac{\varphi}{2}\right). \quad (4.45)$$

⁶Da dies eine *Näherung* ist, kümmern wir uns nicht um die Frage, inwieweit die Summe der beiden Operatoren U_{Perfekt} und $\tilde{\Delta}_{\text{gro\ss}}$ wieder unitär ist.

Als besonders geschickte Wahl erweist es sich, wenn $\mathfrak{S}^{(1)}$ einen Großteil des Fehlers vernichtet, der normalerweise von $\mathfrak{S}^{(2)}$ verstärkt wird; ein nicht mehr vorhandener Fehler kann nicht verstärkt werden. Die Wahl $\mathfrak{S}^{(1)} = \mathfrak{S}_{xx} := -\sigma_x \otimes \sigma_x$ in (4.44) und $\mathfrak{S}^{(2)} = \mathfrak{S}_{xy}$ in (4.45) hat sich in diesem Sinne als erfolgreich erwiesen. Zusammengefasst ergibt das

$$\begin{aligned} \mathcal{U}(\varphi) &= \mathfrak{S}_{xy} \mathfrak{S}_{xx} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{xx}^{-1} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{xy}^{-1} \mathfrak{S}_{xx} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{xx}^{-1} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \\ &= \mathfrak{S}_{1z} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{xx} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{1z} U_{\text{Real}}\left(\frac{\varphi}{4}\right) \mathfrak{S}_{xx} U_{\text{Real}}\left(\frac{\varphi}{4}\right). \end{aligned} \quad (4.46)$$

4.4.1 Spinecho + Offset

In Abschnitt 4.4 haben wir gesehen, dass ein Spinecho mit dem Operator \mathfrak{S}_{xy} einen Großteil sowohl des intrinsischen Fehlers korrigiert als auch des Fehlers durch eine externe Störquelle $\sim S_z$. Von beiden Fehlerquellen verbleiben aber ungewollte Terme. Nun haben wir das seltene Glück, dass diese Fehlerterme sich z.T. gegenseitig korrigieren. Konkret bedeutet dies, dass eine Spinecho-Gatteroperation mit Störfeld groteskerweise erheblich besser ausfallen kann als ohne. Es muss aber kein zusätzliches Magnetfeld in das Experiment integriert werden, um in den Genuss dieses Effektes zu gelangen. Wie anhand von (1.31) erkennbar, entspricht der Term $\hbar \frac{\gamma}{2} S_z$ in einem anderen Wechselwirkungsbild einer Verschiebung der Winkelfrequenz der Laser um γ . Diese Verschiebung wird *Offset* genannt und hat für beide Laserstrahlen das gleiche Vorzeichen.

Eine korrekte Transformation würde auch eine Anpassung des Spinecho-Operators \mathfrak{S}_{xy} erfordern. Zum Erreichen unseres Primärzieles – einer Reduktion des Fehlers – ist dies allerdings nicht erforderlich.

4.5 Ergebnisse

Im Folgenden sollen einige Optimierungsergebnisse dargestellt werden. Diese hängen natürlich von den jeweiligen Systemparametern ab. Wir beginnen deshalb diesen Abschnitt mit einigen Festlegungen und Definitionen. Standardmäßig rechnen wir mit einem ein Lamb-Dicke-Faktor von

$$\eta = 0,05. \quad (4.47)$$

Die Fallenfrequenz ν dient als Referenzwert zum Festlegen der Zeiteinheit

$$\nu := 2\pi \quad \Rightarrow \quad \text{Zeiteinheit} = \frac{2\pi}{\nu} = 1. \quad (4.48)$$

Eine Zeiteinheit entspricht somit einem Fallenzyklus. Dementsprechend wird auch die Rabi-Frequenz Ω ohne Einheiten angegeben, gemäß $\Omega = 1 \Leftrightarrow \Omega = \frac{\nu}{2\pi}$.

Im Folgenden sollen das Mølmer-Sørensen- und das $\sigma_z \otimes \sigma_z$ -Gatter einheitlich beschrieben werden. Dazu führen wir die Konstante ζ ein

$$\zeta = \begin{cases} 1 & \text{für Mølmer-Sørensen} \\ 2 & \text{für } \sigma_z \otimes \sigma_z \end{cases} \quad (4.49)$$

Damit schreiben sich die Kreisfrequenzen $\pm\delta$ der Laser im mitrotierenden Bezugssystem als

$$\delta = \frac{\nu}{\zeta} - \varepsilon \quad (4.50)$$

(zu vergleichen mit (1.34) & (1.41)). Die Verstimmung ε wird nun so gewählt, dass es ein ganzzahliges n gibt mit

$$\frac{\delta}{\nu} = \frac{n-1}{\zeta \cdot n} \quad \Rightarrow \quad \varepsilon = \frac{\nu}{\zeta \cdot n}. \quad (4.51)$$

Bei der Realisierung der Gatter durch Rechteck-Impulse ist $\frac{2\pi}{\zeta \cdot \varepsilon} = n \cdot \frac{2\pi}{\nu}$ die Zeit, die zur Durchführung der Gatter benötigt wird (1.60). Wir führen deshalb die Kenngröße T_{RI} ein

$$T_{\text{RI}} := n \cdot \frac{2\pi}{\nu}. \quad (4.52)$$

Die Zeitdauer für einen Rechteck-Impuls T_{RI} und die Verstimmung ε bedingen einander gegenseitig. Da wir aber keine Rechteck-Impulse verwenden, haben wir die Freiheit, eine andere Zeitdauer T_{G} für die Gatteroperationen zu wählen. Es erweist sich, dass die Erscheinungsbilder von Gatter-Realisierungen mit gleichem Quotienten $\Lambda = \frac{T_{\text{G}}}{T_{\text{RI}}}$ einander ähnlich sind (siehe Abbildung 4.2). Wir beschreiben die Gatter deshalb durch die Angabe

$$T_{\text{G}} = \Lambda \times T_{\text{RI}}. \quad (4.53)$$

Bei den folgenden Gatter-Realisierungen wurde die Gatterzeit T_{G} in $\Lambda \cdot 80$ Intervalle $\Delta T_{\text{Intervall}}$ unterteilt, in welchen die jeweilige Rabifrequenz Ω_i optimiert wurde ($1 \leq i \leq \Lambda \cdot 80$). Ein Intervall hat damit die Länge

$$\Delta T_{\text{Intervall}} = \frac{T_{\text{G}}}{\Lambda \cdot 80} = \frac{T_{\text{RI}}}{80}. \quad (4.54)$$

Zur Beurteilung der Güte einer Gatter-Realisierung ist zu klären, auf welche Zustände das Gatter wirken soll. Wir werden uns daher im Folgenden auf den Grundzustand der Schwingungsmode $|n=0\rangle$ beschränken, um den Vergleich einfach zu gestalten. Zur Charakterisierung der Güte verwenden wir die beiden Kenngrößen F_1 & F_2 (engl. fidelity), die im Folgenden beschrieben werden. F_1 basiert auf der Gewinnfunktion \mathfrak{G} (4.9), die schon während der Optimierung das entscheidende Maß für die Güte war

$$F_1 = \sqrt{\frac{\mathfrak{G}}{\max(\mathfrak{G})}} \leq 1. \quad (4.55)$$

Die Division durch $\max(\mathfrak{G})$ dient nur der Normierung, während die Wurzel der Quadratur in (4.9) entgegenwirkt.

Als zweite Kenngröße F_2 verwenden wir ein in der Praxis oft genutztes Maß, das angibt, wie gut ein bestimmter Bell-Zustand mit einem Gatter erzeugt werden kann. Hierbei geht es vor allem um die Verschränkungseigenschaften der Gatter. Daher erlauben wir die Ergänzung der Gatter durch rein lokale Drehungen, deren Zuverlässigkeit mit 100% angenommen wird: $U_{\text{Real}} \rightarrow U_{\text{Real+Lokal}}$. Damit definieren wir F_2 wie folgt

$$F_2 = \left| \frac{1}{\sqrt{2}} (\langle g, g, 0 | + \langle e, e, 0 |) U_{\text{Real+Lokal}} |g, g, 0_{\text{Schwingung}} \rangle \right|^2 \leq 1. \quad (4.56)$$

Die lokalen Drehungen werden dabei so gewählt, dass F_2 für $U_{\text{Real}} = U_{\text{Ziel}}$ maximal wird.

Der besseren Übersicht halber geben wir statt der Güten F_1 & F_2 deren Abweichungen \mathcal{E}_1 & \mathcal{E}_2 vom perfekten Gatter an

$$\mathcal{E}_1 = 1 - F_1 \quad \mathcal{E}_2 = 1 - F_2. \quad (4.57)$$

Zur weiteren Charakterisierung der Gatter geben wir noch die durchschnittliche und die maximale Rabifrequenz ($\bar{\Omega}$, Ω_{max}) an.

4.5.1 $\sigma_z \otimes \sigma_z$ -Gatter

Tabelle 4.1 zeigt einige Optimierungsergebnisse für verschiedene $T_{\text{G}} = \Lambda \times T_{\text{RI}}$ (4.53), wobei nur die Rabifrequenz Ω optimiert wurde. Deren Verlauf über die Zeit wird für sechs dieser Gatter-Realisierungen in Abbildung 4.1 wiedergegeben. Die Phase ϕ kann dabei jeweils einen beliebigen, aber konstanten Wert annehmen.

Wie deutlich zu erkennen ist, fallen $\bar{\Omega}$ und Ω_{max} umso größer aus, je kleiner Λ und T_{RI} sind. Bei zu großer Rabifrequenz ist das Gatter nicht mehr brauchbar zu optimieren. Dieser Fall liegt in Tabelle 4.1 bei $T_{\text{G}} = 0,75 \times 40 \cdot \frac{2\pi}{\nu}$ vor. Erstaunlicherweise ist ansonsten aber die Tendenz festzustellen, dass die Güte F_1 für höhere Rabifrequenzen besser ausfällt als für niedrige⁷. Dies ist erst einmal eine sehr gute Nachricht, denn wir sind sowohl an einer hohen Güte als auch an kurzen Gatterzeiten interessiert. Wie es scheint, bekommen wir beides zusammen! Hier gilt es allerdings auch vorsichtig zu sein: Unser Hamilton-Operator ist nur eine Näherung für die komplexe Wechselwirkung

⁷Dass die höheren Rabifrequenzen und nicht die kürzeren Gatterzeiten die entscheidende Größe sind, ist allein aus Tabelle 4.1 nicht erkennbar. Zwei andere Indizien sprechen aber dafür:

- Wenn ein bereits gut optimiertes Gatter noch weiter optimiert wird, steigt Ω_{max} meist stark an.
- Optimiert man auch die Phase ϕ , fällt Ω_{max} kleiner aus, was zu schlechteren Ergebnissen führt.

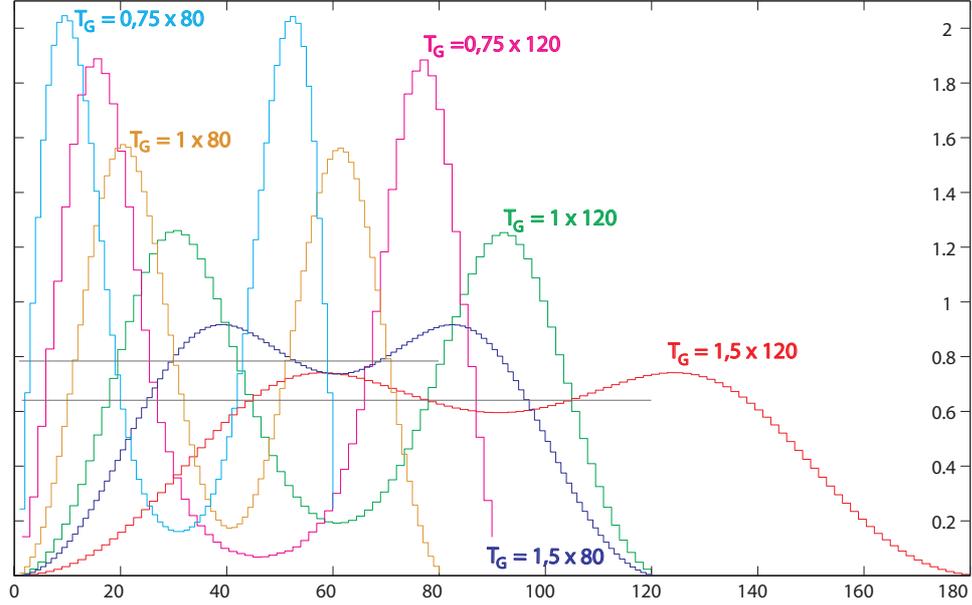


Abbildung 4.1: Aufgetragen ist die Rabifrequenz Ω über die Zeit für Realisierungen des $\sigma_z \otimes \sigma_z$ -Gatters mit unterschiedlichen $T_G = \Lambda \times T_{RI}$ (4.53). Die Einheit der Zeitachse (horizontal) ist ein Fallenzyklus $\frac{2\pi}{\nu}$. Die Einheit der Rabifrequenz ist $\frac{\nu}{2\pi}$. Die nicht beschrifteten, grauen Linien entsprechen der Umsetzung des Gatters durch Rechteck-Impulse.

$\Lambda \backslash T_{RI}$	40	80	120
0,75	$\mathcal{E}_1 = 1,9 \cdot 10^{-2}$ $\mathcal{E}_2 = 3,3 \cdot 10^{-2}$ $\bar{\Omega} = 1,60$ $\Omega_{\max} = 2,75$	$\mathcal{E}_1 = 4,0 \cdot 10^{-4}$ $\mathcal{E}_2 = 8,9 \cdot 10^{-4}$ $\bar{\Omega} = 1,02$ $\Omega_{\max} = 2,05$	$\mathcal{E}_1 = 8,0 \cdot 10^{-4}$ $\mathcal{E}_2 = 3,8 \cdot 10^{-4}$ $\bar{\Omega} = 0,775$ $\Omega_{\max} = 1,89$
1	$\mathcal{E}_1 = 9,4 \cdot 10^{-5}$ $\mathcal{E}_2 = 2,4 \cdot 10^{-4}$ $\bar{\Omega} = 1,28$ $\Omega_{\max} = 2,24$	$\mathcal{E}_1 = 1,2 \cdot 10^{-3}$ $\mathcal{E}_2 = 4,1 \cdot 10^{-4}$ $\bar{\Omega} = 0,781$ $\Omega_{\max} = 1,58$	$\mathcal{E}_1 = 2,1 \cdot 10^{-3}$ $\mathcal{E}_2 = 9,6 \cdot 10^{-4}$ $\bar{\Omega} = 0,617$ $\Omega_{\max} = 1,26$
1,5	$\mathcal{E}_1 = 1,1 \cdot 10^{-3}$ $\mathcal{E}_2 = 3,0 \cdot 10^{-4}$ $\bar{\Omega} = 0,880$ $\Omega_{\max} = 1,38$	$\mathcal{E}_1 = 2,2 \cdot 10^{-3}$ $\mathcal{E}_2 = 6,8 \cdot 10^{-4}$ $\bar{\Omega} = 0,589$ $\Omega_{\max} = 0,919$	$\mathcal{E}_1 = 2,7 \cdot 10^{-3}$ $\mathcal{E}_2 = 8,9 \cdot 10^{-4}$ $\bar{\Omega} = 0,472$ $\Omega_{\max} = 0,741$

Tabelle 4.1: Ergebnisse für das $\sigma_z \otimes \sigma_z$ -Gatter für verschiedene Gatterzeiten $T_G = \Lambda \times T_{RI}$ (4.53), wobei T_{RI} in Einheiten des Fallenzyklus $\frac{2\pi}{\nu}$ angegeben ist. Zu jeder Realisierung sind die Gatterfehler \mathcal{E}_1 und \mathcal{E}_2 (4.57) sowie die durchschnittliche und maximale Rabifrequenz $\bar{\Omega}$, Ω_{\max} angegeben.

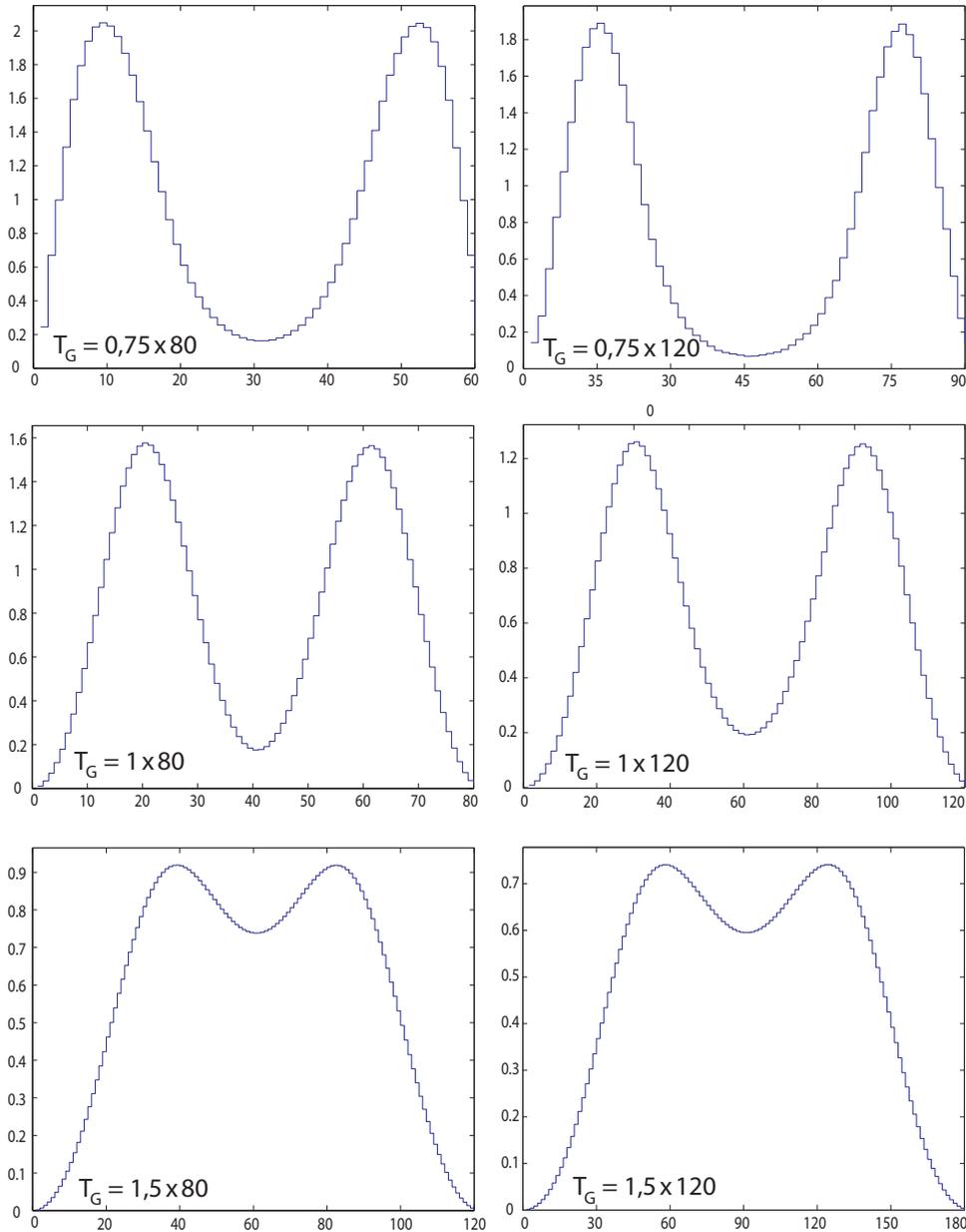


Abbildung 4.2: $\sigma_z \otimes \sigma_z$ -Gatter mit unterschiedlichen $T_G = \Lambda \times T_{RI}$ (4.53). Aufgetragen ist jeweils die Rabi-Frequenz Ω (vertikal) in Einheiten $\frac{\nu}{2\pi}$ über der Gatterzeit T_G in Einheiten des Fallenzklus $\frac{2\pi}{\nu}$. Die Skalierung der Graphen ist relativ zu T_G und Ω_{\max} gewählt, wodurch die Ähnlichkeit der Gatter mit gleichem Λ zutage tritt.

der Laser mit der Elektronenhülle der Ionen. Es ist zu erwarten, dass diese Näherung sich mit zunehmender Rabifrequenz verschlechtert.

Der vielleicht interessanteste Gatterfehler besteht im unerwünschten Übergang $|g, e\rangle \leftrightarrow |e, g\rangle$. Für großes T_G bzw. kleines Ω ist dies der dominante Fehler, der alle anderen überschattet. Je kleiner T_G wird, desto mehr geht dieser Fehler zurück. Bei $T_G = 1 \times 40 \cdot \frac{2\pi}{\nu}$ (dem Gatter mit der höchsten Güte in Tabelle 4.1) ist dieser Fehler bedeutungslos verglichen mit den anderen verbliebenen Fehlern. Eine einfache Erklärung für diesen Effekt kann in dieser Arbeit leider nicht geliefert werden.

Da der Fehler $|g, e\rangle \leftrightarrow |e, g\rangle$ keinen Einfluss auf die Güte F_2 hat, wird begreiflich, warum F_2 einer sehr viel geringeren Schwankung ausgesetzt ist als F_1 .

Werfen wir jetzt einen Blick auf Abbildung 4.2, wo der Verlauf der Ω_i über die Gatterzeit T_G in angepasster Skalierung aufgetragen ist. Hier erkennen wir die schon angedeutete Ähnlichkeit von Gattern mit gleichem Λ . Die Form des Kurvenverlaufs ergibt sich aus der Kombination der physikalischen Notwendigkeiten und den Nebenbedingungen (4.1). Im Folgenden werden wir uns bemühen, diesen Zusammenhang besser zu verstehen.

Die Nebenbedingungen favorisieren eine starke Annäherungen der Rabifrequenz Ω an den Wert null für die Enden des Graphen. Weiterhin erkennen wir für $\Lambda = 1$ und $\Lambda = 1,5$ ein parabolisches Ansteigen bzw. Abklingen von Ω zu den Enden hin, was ebenfalls durch die spezielle Wahl der Nebenbedingungen bedingt ist (bei anderen Nebenbedingungen kann dieser Effekt ausbleiben). Vielleicht etwas überraschend mag das Abfallen von Ω zur Mitte hin sein, was einen Graphen bedingt, der an den Buchstaben „M“ erinnert. Dies ist eine direkte Folge der niedrigen Werte für Ω an den Enden des Graphen. Dies wird verständlich, wenn wir uns Abbildung 4.3 ansehen. Hohe Rabifrequenzen erlauben in gleicher Zeit sehr viel weitere Strecken im Phasenraum zurückzulegen als niedrige Rabifrequenzen. Die Veränderung der Bewegungsrichtung pro Zeit ist hingegen unabhängig von der Rabifrequenz und hängt nur von der Verstimmung $\varepsilon = \text{const}$ ab. Somit ist die Krümmung – also die Veränderung der Richtung pro Bogenlänge – der Phasenraumbahn umso größer, je kleiner die Rabifrequenz ist. Die niedrigen Rabifrequenzen am Anfang und am Ende des Gatters korrespondieren also mit einer stark gekrümmten Phasenraumbahn. Die einfachste Möglichkeit, um unter dieser Vorgabe immer noch eine geschlossene Phasenraumbahn zu durchlaufen, ist eine ebenfalls starke Krümmung in der Bahnmitte – also niedrige Rabifrequenzen im Bereich der halben Gatterzeit T_G .

Im Fall $T_G = \Lambda \times T_{RI} = 1 \times T_{RI}$ entspricht das Integral der Krümmung entlang der gesamten Phasenraumbahn einem Winkel von 360° . Für $\Lambda = 0,75$ weist dieser Winkel nur noch eine Größe von $0,75 \cdot 360^\circ = 270^\circ$ auf (gemäß der idealisierten Gatterbeschreibung aus Abschnitt 1.5). Mit etwas Wohlwollen erkennen wir in Abbildung 4.3, wie für $T_G = 0,75 \times 80 \cdot \frac{2\pi}{\nu}$ die beiden Enden der Phasenraumbahn im rechten Winkel aufeinander treffen.

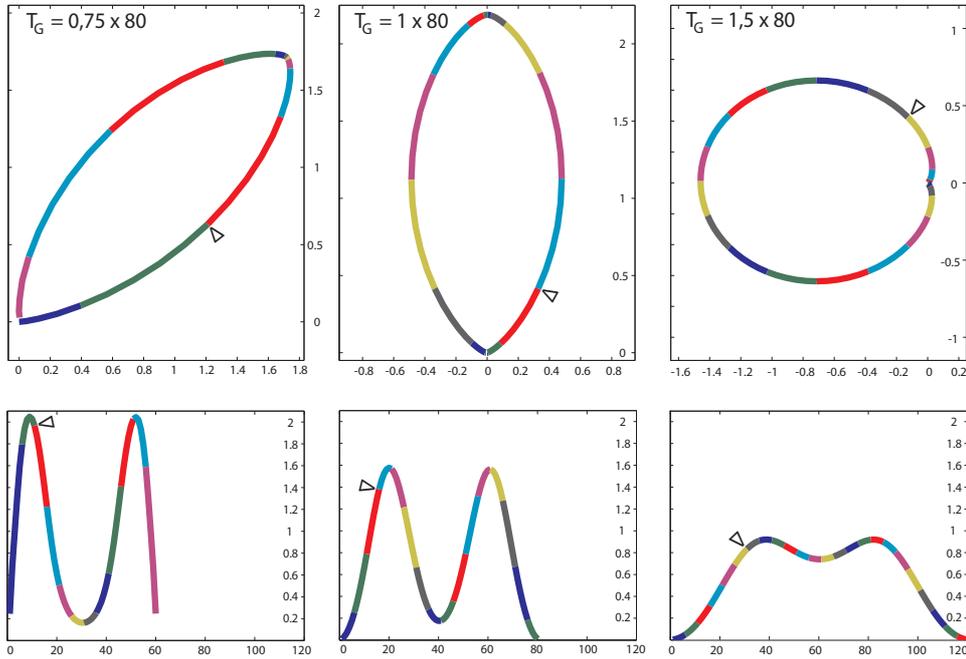


Abbildung 4.3: Die oberen drei Graphen zeigen Phasenraumbahnen für den Startzustand $|g, g, n = 0\rangle$ unter Einwirkung eines $\sigma_z \otimes \sigma_z$ -Gatters mit jeweils unterschiedlichen $T_G = \Lambda \times T_{\text{RI}}$ (4.53). Darunter sind die drei Graphen der dazugehörigen Rabifrequenzen. In den oberen Graphen sind die Erwartungswerte $\langle \frac{i}{\sqrt{2}}(\hat{a}^\dagger - \hat{a}) \rangle \sim p$ über $\langle \frac{1}{\sqrt{2}}(\hat{a}^\dagger + \hat{a}) \rangle \sim x$ aufgetragen. In den unteren Graphen ist die Rabifrequenz in Einheiten $\frac{\nu}{2\pi}$ über die Zeit in Einheiten des Fallenzyklus $\frac{2\pi}{\nu}$ aufgetragen. Die Graphen wechseln alle fünf Fallenzyklen die Farbe, wobei korrespondierende Zeitabschnitte in den Graphen der Phasenräume und in den Graphen der Rabifrequenzen die gleiche Farbe aufweisen. Wie gut zu erkennen ist, werden zu Zeiten hoher Rabifrequenzen viel größere Strecken im Phasenraum zurückgelegt als zu Zeiten kleiner Rabifrequenzen. Dies führt dazu, dass Abschnitte mit sehr niedriger Rabifrequenz (wie z.B. die ersten und letzten zehn Fallenzyklen im Falle $T_G = 1,5 \times 80$) im Phasenraumgraphen z.T. nicht mehr identifizierbar sind. Zur besseren Orientierung wurde deshalb noch eine Markierung (schwarzes Dreieck) gesetzt. Alle Phasenraumbahnen starten im Punkt $(0, 0)$. Die Startrichtung zeigt horizontal entlang der positiven x-Richtung, was aufgrund der starken Krümmung bei kleinen Rabifrequenzen (siehe Haupttext) nicht mehr zu erkennen ist. Die Phasenraumbahnen werden gegen den Uhrzeigersinn durchlaufen und schließen jeweils eine Fläche der Größe $\frac{\pi}{2}$ ein. Während eines Farbabschnitts summiert sich die Krümmung entlang der Phasenraumbahn zu $\frac{\pi}{8}$ bzw. $22,5^\circ$ (gemäß der idealisierten Gatterbeschreibung aus Abschnitt 1.5).

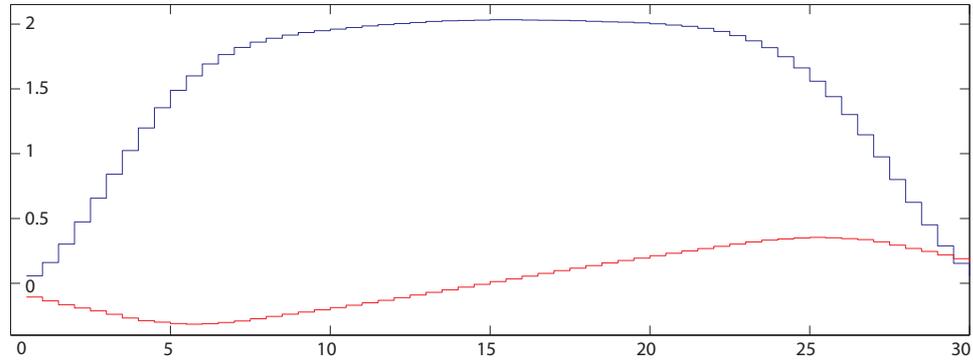


Abbildung 4.4: $\sigma_z \otimes \sigma_z$ -Gatter mit $T_G = 0,75 \times 40$ (4.53) mit variabler Phase ϕ . Aufgetragen sind die Rabifrequenz Ω und die Phase ϕ über der Gatterzeit T_G in Einheiten des Fallenzklus $\frac{2\pi}{\nu}$. Die Einheiten der Y-Achse sind: $1 \leftrightarrow \frac{\nu}{2\pi}$ für Ω und $1 \leftrightarrow \frac{\pi}{2}$ für ϕ .

Für $T_G = 1 \times 80 \cdot \frac{2\pi}{\nu}$ scheint dies auch der Fall zu sein – das ist aber eine Täuschung, die auf der extremen Verkürzung der Phasenraumbahn bei kleinen Rabifrequenzen beruht.

Der insgesamt durchlaufene Winkel hängt nur von Λ ab. Das Krümmungsverhalten der Phasenraumbahnen muss sich diesem Winkel entsprechend gestalten. Da die Krümmung wiederum durch die Höhe der Rabifrequenz gegeben ist, erklärt sich die Ähnlichkeit der Graphen der Rabifrequenz bei gleichem Λ (Abbildung 4.2).

Die Rabifrequenz Ω ist nicht die einzige Größe, welche die Krümmung der Phasenraumbahn bedingt (1.53). Diese wird auch von der Laserfrequenz im mitrotierenden Bezugssystem δ bzw. deren Verstimmung ε beeinflusst. Je resonanter das Laserfeld, desto größer werden die Erwartungswerte für x und p . Dies ist gleichbedeutend mit einer großen Phasenraumbahn und kleiner Krümmung. Nun halten wir δ zwar konstant, aber wir können ein stufenweise variables δ mit Hilfe der Phase ϕ simulieren, da in unserem Hamilton-Operator (3.12) immer nur die Kombination $\delta \cdot t + \phi$ eingeht. Lassen wir den Computer nicht nur Ω , sondern auch die Werte der Phase ϕ optimieren, erhalten wir annähernd kreisförmige Phasenraumbahnen. Im Unterschied zum Rechteck-Impuls werden diese Phasenraumkreise aber nicht mit konstanter Phasenraum-Geschwindigkeit durchfahren.

Wird die Phase ϕ angepasst, ist ein starkes Absinken der Rabifrequenz Ω im Bereich der halben Gatterzeit T_G nicht mehr notwendig, wie in Abbildung 4.4 zu erkennen ist. Dies hat zur Folge, dass die Differenz $\Omega_{\max} - \bar{\Omega}$ kleiner ausfällt, als bei Gattern mit konstantem ϕ . Da aber bei beiden Varianten $\bar{\Omega}$ ungefähr den gleichen Wert hat, macht sich der Unterschied vor allem bei Ω_{\max} bemerkbar.

Weiter oben wurde behauptet, dass höhere Rabifrequenzen Ω zu besseren

Güten F_1 führen (sofern sie gewisse Grenzen nicht überschreiten). Dies wird durch die Beobachtung gestützt, dass Gatter eine geringere Güte aufweisen, wenn die Phase ϕ variabel ist und somit Ω_{\max} kleiner ausfällt. Zumindest ist dies ein erstaunlicher Umstand, denn eigentlich sollte die Güte durch die Optimierung eines zusätzlichen Freiheitsgrades ansteigen und nicht abfallen. Dass Letzteres trotzdem passiert, ist nur möglich, weil der zugrunde liegende Algorithmus lediglich *lokal* zu optimieren vermag.

Die Regel, dass höhere Rabifrequenzen Ω bessere Güten F_1 liefern, bricht bei zu hohen Werten von Ω zusammen. Dies geschieht z.B. bei $T_G = 0,75 \times 40 \cdot \frac{2\pi}{\nu}$ (siehe Tabelle 4.1). In diesem Fall kann der Effekt der reduzierten Werte für Ω_{\max} durch Optimierung der Phase ϕ sehr hilfreich sein. Abbildung 4.4 zeigt den Graphen für $T_G = 0,75 \times 40 \cdot \frac{2\pi}{\nu}$ mit variabler Phase. Die Güte dieses Gatters übertrifft alle Gatter in Tabelle 4.1. Sogar die Güte F_2 hat sich deutlich erhöht. Die entsprechenden Werte lauten:

$$\begin{aligned} \mathcal{E}_1 &= 2,0 \cdot 10^{-5} & \bar{\Omega} &= 1,66 \cdot \frac{\nu}{2\pi} \\ \mathcal{E}_2 &= 3,8 \cdot 10^{-5} & \Omega_{\max} &= 2,03 \cdot \frac{\nu}{2\pi}. \end{aligned} \quad (4.58)$$

Spinecho

In Abschnitt 4.4 wurde die Technik des Spinechos erklärt. Wir werden im Folgenden immer annehmen, dass der Spinecho-Operator \mathfrak{S} fehlerfrei durchgeführt werden kann. Weiterhin wollen wir die Gatter durch die Angabe $T_G = \Lambda \times T_{\text{RI}}$ (4.53) charakterisieren. Da für ein Spinecho mindestens zwei Impulse⁸ notwendig sind, legen wir fest, dass T_{RI} die Gesamtzeitdauer des Gatters bei einer Realisierung durch Rechteck-Impulse darstellen soll. Zur Optimierung der Ω_i wurde die Gatterzeit T_G in $p \cdot \Lambda \cdot 80$ Intervalle unterteilt, wobei p die Anzahl der Impulse darstellt (also $p = 2$ beim einmaligen Spinecho). Die Zeitdauer des Operators \mathfrak{S} wird weder in der Angabe von T_{RI} noch in T_G berücksichtigt. Bei der Optimierung hingegen wurden vier Fallenzyklen ($4 \cdot \frac{2\pi}{\nu}$) für \mathfrak{S} veranschlagt.

Tabelle 4.2 zeigt einige der Ergebnisse für ein einmaliges Spinecho ($\mathfrak{S} = \sigma_x \otimes \sigma_y$) mit und ohne Offset (Abschnitt 4.4.1). Das Spinecho macht das Gatter unempfindlicher gegen äußere Störungen und erhöht dessen Güte, indem der unerwünschte Übergang $|g, e\rangle \leftrightarrow |e, g\rangle$ ausgelöscht wird. Bei größeren Gatterzeiten T_G und somit kleineren Rabifrequenzen Ω besteht der verbleibende dominante Fehler in den unerwünschten Übergängen $|g, g, 0_{(\text{Schwingung})}\rangle \rightarrow |e, e, 1\rangle$ (doppelte Absorption eines blauen Photons) sowie $|e, e, 0\rangle \rightarrow |g, g, 1\rangle$ (doppelte Emission eines roten Photons). Beide Prozesse sind mit 2ε leicht verstimmmt. Fügt man einen Offset hinzu, so addiert sich dessen doppelter Wert zur Verstimmung, was zu einer Unterdrückung dieser Übergänge führt. Dies erklärt die stark verbesserte Güte der Gatter mit Offset für $T_{\text{RI}} = 160$ gegenüber denen ohne Offset.

⁸Zu einem *Impuls* zählen wir alles, was nicht durch einen Spinecho-Operator \mathfrak{S} unterbrochen ist.

Offset \ T_G	$0,75 \times 80$	1×80	$0,75 \times 160$	1×160
ja	$\mathcal{E}_1 = 8,1 \cdot 10^{-5}$ $\mathcal{E}_2 = 1,6 \cdot 10^{-4}$ $\bar{\Omega} = 1,06$ $\Omega_{\max} = 1,79$	$\mathcal{E}_1 = 9,3 \cdot 10^{-5}$ $\mathcal{E}_2 = 2,3 \cdot 10^{-4}$ $\bar{\Omega} = 0,863$ $\Omega_{\max} = 1,31$	$\mathcal{E}_1 = 5,3 \cdot 10^{-5}$ $\mathcal{E}_2 = 8,5 \cdot 10^{-5}$ $\bar{\Omega} = 0,680$ $\Omega_{\max} = 1,08$	$\mathcal{E}_1 = 3,2 \cdot 10^{-5}$ $\mathcal{E}_2 = 5,9 \cdot 10^{-5}$ $\bar{\Omega} = 0,589$ $\Omega_{\max} = 7,60$
nein	$\mathcal{E}_1 = 8,2 \cdot 10^{-5}$ $\mathcal{E}_2 = 1,9 \cdot 10^{-4}$ $\bar{\Omega} = 1,06$ $\Omega_{\max} = 1,78$	$\mathcal{E}_1 = 1,7 \cdot 10^{-4}$ $\mathcal{E}_2 = 2,8 \cdot 10^{-4}$ $\bar{\Omega} = 0,864$ $\Omega_{\max} = 1,31$	$\mathcal{E}_1 = 3,3 \cdot 10^{-4}$ $\mathcal{E}_2 = 6,7 \cdot 10^{-4}$ $\bar{\Omega} = 0,681$ $\Omega_{\max} = 1,08$	$\mathcal{E}_1 = 3,4 \cdot 10^{-4}$ $\mathcal{E}_2 = 7,0 \cdot 10^{-4}$ $\bar{\Omega} = 0,590$ $\Omega_{\max} = 0,762$

Tabelle 4.2: Ergebnisse für das $\sigma_z \otimes \sigma_z$ -Gatter mit einmaligem Spinecho ($\mathfrak{S} = \sigma_x \otimes \sigma_y$) für verschiedene Gatterzeiten $T_G = \Lambda \times T_{\text{RI}}$ (4.53), wobei T_{RI} in Einheiten des Fallenzyklus $\frac{2\pi}{\nu}$ angegeben ist. Als Offset-Verstimmung wurde $\gamma = 8\varepsilon$ ((3.13) & (4.50)) gewählt. Zu jeder Realisierung sind die Gatterfehler \mathcal{E}_1 und \mathcal{E}_2 (4.57) sowie die durchschnittliche und maximale Rabifrequenz $\bar{\Omega}$, Ω_{\max} angegeben.

Bei $T_{\text{RI}} = 80$ haben die Übergänge $|g, g, 0\rangle \rightarrow |e, e, 1\rangle$ und $|e, e, 0\rangle \rightarrow |g, g, 1\rangle$ ihre Rolle als dominante Fehler weitestgehend eingebüßt, so dass die Verbesserung durch das Offset kaum noch zu Buche schlägt.

Bisher haben wir nur ein einmaliges Spinecho betrachtet, d.h. wir haben den Impuls zur Erzeugung des Gatters in zwei Anteile aufgeteilt und dazwischen den Spinecho-Operator \mathfrak{S} ausgeführt. In (4.46) wird nun die Verwendung zweier verschiedener Spinecho-Operatoren und die Aufteilung des Impulses in vier Anteile vorgeschlagen. Obwohl dieses Verfahren beeindruckende theoretische Güten zu erzeugen vermag, lässt der erhöhte Aufwand die Umsetzung in die Praxis eher fraglich erscheinen. Die Komplexität dieses Verfahrens schlägt sich auch darin nieder, dass es keine vereinfachten Aussagen der Art „hohe Rabifrequenz = hohe Güte“ zulässt, sondern sehr stark vom Wechselspiel der verschiedenen Parameter abhängt. Bei günstiger Wahl kann man theoretische Güten von 0,999.999.9 erzielen. Als Beispiel seien die Kenngrößen für ein solches Gatter mit $T_G = 0,8875 \times 160 \cdot \frac{2\pi}{\nu}$ angegeben

$$\begin{aligned} \mathcal{E}_1 &= 1,1 \cdot 10^{-7} & \bar{\Omega} &= 0,638 \cdot \frac{\nu}{2\pi} \\ \mathcal{E}_2 &= 5,4 \cdot 10^{-8} & \Omega_{\max} &= 0,862 \cdot \frac{\nu}{2\pi}. \end{aligned} \quad (4.59)$$

4.5.2 Mølmer-Sørensen-Gatter

Wie wir sehen werden, lassen sich mit dem Mølmer-Sørensen-Gatter aberwitzige theoretische Güten erzielen. Daher muss die Fragestellung hier in erster Linie lauten, wie genau unser Hamilton-Operator das System beschreibt.

Die Güte des Mølmer-Sørensen-Gatters ist umso höher, je kleiner die Rabifrequenzen Ω ausfallen. Damit entspricht es viel mehr unserer Intuition als das $\sigma_z \otimes \sigma_z$ -Gatter. Es wurde daher versucht, Gatter-Realisierungen zu finden, die durch einen möglichst flachen Verlauf der Rabifrequenz Ω ausgezeichnet

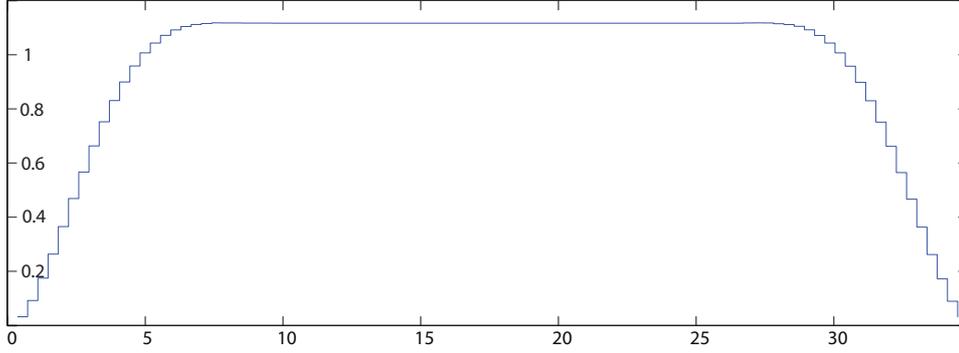


Abbildung 4.5: Mølmer-Sørensen-Gatter mit $T_G = \frac{93}{80} \times 30$ (4.53). Aufgetragen sind die Rabifrequenz Ω in Einheiten $\frac{\nu}{2\pi}$ über der Gatterzeit T_G in Einheiten des Fallenzyklus $\frac{2\pi}{\nu}$.

20	30	40	60	100
$\mathcal{E}_1 = 4,6 \cdot 10^{-5}$	$\mathcal{E}_1 = 1,7 \cdot 10^{-6}$	$\mathcal{E}_1 = 2,2 \cdot 10^{-7}$	$\mathcal{E}_1 = 1,5 \cdot 10^{-8}$	$\mathcal{E}_1 = 1,6 \cdot 10^{-9}$
$\mathcal{E}_2 = 1,7 \cdot 10^{-5}$	$\mathcal{E}_2 = 6,6 \cdot 10^{-7}$	$\mathcal{E}_2 = 8,5 \cdot 10^{-8}$	$\mathcal{E}_2 = 8,7 \cdot 10^{-9}$	$\mathcal{E}_2 = 5,8 \cdot 10^{-10}$
$\bar{\Omega} = 1,71$	$\bar{\Omega} = 1,00$	$\bar{\Omega} = 0,728$	$\bar{\Omega} = 0,478$	$\bar{\Omega} = 0,285$
$\Omega_{\max} = 1,95$	$\Omega_{\max} = 1,11$	$\Omega_{\max} = 0,813$	$\Omega_{\max} = 0,540$	$\Omega_{\max} = 0,318$

Tabelle 4.3: Ergebnisse für das Mølmer-Sørensen-Gatter mit $T_G = \frac{93}{80} \times T_{\text{RI}}$ (4.53) für verschiedene T_{RI} in Einheiten des Fallenzyklus $\frac{2\pi}{\nu}$. Zu jeder Realisierung sind die Gatterfehler \mathcal{E}_1 und \mathcal{E}_2 (4.57) sowie die durchschnittliche und maximale Rabifrequenz $\bar{\Omega}$, Ω_{\max} angegeben.

sind. Abbildung 4.5 zeigt ein Beispiel für das Ergebnis dieser Bestrebungen. Diese Formgebung – die an herkömmliche Impulsformung erinnert – konnte allen Gattern aufgeprägt werden. Hierfür wurden $\Lambda = \frac{93}{80}$ gesetzt, günstige Anfangswerte gewählt und die Nebenbedingungen angepasst, was in der Fußnote⁹ näher erläutert wird. In Tabelle 4.3 sind die mit dieser Form erzielten Ergebnisse zusammengefasst. Das schnellste Gatter kommt mit 23,25 Fallenzyklen aus. Bei noch kleineren Werten steigt die Rabifrequenz auf kritische Werte und beginnt, die Optimierung problematisch zu gestalten.

Bisher haben wir uns bei der Bewertung der Güte auf den Fall beschränkt, dass die Schwingungsmode in den Grundzustand $|n=0\rangle$ gekühlt wurde. Für den (unrealistischen) Fall eines gemischten Zustandes mit der gleichen Wahrscheinlichkeit für $|n=0\rangle$ und $|n=1\rangle$ verschlechtert sich die Güte F_1 aller Mølmer-Sørensen-Gatter um ungefähr den gleichen Wert im Bereich von $1 \cdot 10^{-6}$ bis $2 \cdot 10^{-6}$ (d.h. $\mathcal{E}_1 \rightarrow \mathcal{E}_1 + 1 \cdot 10^{-6}$).

⁹Die „M“-Form der Graphen ist bei den Mølmer-Sørensen-Gattern weniger ausgeprägt als bei den $\sigma_z \otimes \sigma_z$ -Gattern. Einer der Gründe dafür mag sein, dass ein großes Ω_{\max} sich bei den MS-Gattern im Gegensatz zu den $\sigma_z \otimes \sigma_z$ -Gattern eher nachteilig auswirkt und die Graphen deshalb die Tendenz zeigen, flacher zu verlaufen. Diese natürliche Tendenz konnte verstärkt werden, indem während der ungefähren Zeitspanne $0,2 \cdot T_G < t < 0,8 \cdot T_G$ die Straffunktion (4.1) viel stärker gewichtet wurde als am Anfang und am Ende.

Alle hier präsentierten Gatter-Realisierungen (Mølmer-Sørensen und $\sigma_z \otimes \sigma_z$) sind weitgehend unabhängig von dem Wert ϕ_{Start} der Phase beim Start des Experimentes (siehe Abschnitt 4.3.1). Einzige Ausnahme bildet hier das Mølmer-Sørensen-Gatter mit $T_{\text{RI}} = 40 \cdot \frac{2\pi}{\nu}$. Bei diesem Wert entspricht die Periode, mit der die Rabifrequenz Ω verändert wird, einem halben Fallenzklus. Nach dem Modell aus Abschnitt 4.3.1 sollte dies ideal sein, um unabhängig von der Phase ϕ_{Start} zu werden. Wie sich aber zeigt, ist der Wert \mathcal{E}_1 bei beliebiger Phase ϕ_{Start} im Durchschnitt um einen Faktor 2,5 größer als für $\phi_{\text{Start}} = 0$. Eine einleuchtende Erklärung dieses Effekts kann nicht gegeben werden.

4.6 Ausblick

Die vom Computer optimierten Gatter erreichen viel höhere theoretische Güten als die bisherigen Verfahren. Andererseits erfordern die neuen Lösungen häufige Veränderungen der Rabifrequenz, was sich als zusätzliche Fehlerquelle erweisen könnte. Zum Zeitpunkt der Niederschrift stehen experimentelle Erfahrungen in dieser Richtung noch aus. Könnte man vielleicht das zugrunde liegende Modell noch weiter verfeinern, um derartige Probleme zu antizipieren? Ein Computermodell, das sämtliche praktischen Unwägbarkeiten einkalkuliert, wird es vermutlich nie geben. Trotzdem ist die Frage nach möglichen Erweiterungen des betrachteten Modells sicherlich sinnvoll.

Das momentane Modell basiert auf einer stufenweisen Anpassung der Rabifrequenz und der Phase. Dies entspricht auch der digitalen Realisierung, wobei die Annahme des unendlich schnellen Sprunges natürlich nur eine Näherung ist. Wollte man hier einen fließenden Übergang einbauen, müsste das ganze Konzept zur Berechnung der Zeitentwicklungsoperatoren neu durchdacht werden. Da der Fehler der hier gemachten Näherung proportional zur Größe des Sprunges sein sollte, hilft die Erzwingung relativ glatter Kurvenverläufe durch die Nebenbedingungen, das Problem klein zu halten. Auch Probleme, die durch eine zu hohe Rabifrequenz verursacht werden, können durch eine entsprechende Wahl der Nebenbedingung abgemildert werden. Als zweites großes Feld für Verbesserungen könnte sich der Hamilton-Operator selbst erweisen. Die Ionen werden als Spinsysteme beschrieben, die über eine Schwingungsmode koppeln. Eine Verbesserungsmöglichkeit besteht nun darin, weitere Schwingungsmoden zu berücksichtigen. Die quantenmechanische Beschreibung sähe dann so aus

$$\text{Spin}_1 \otimes \text{Spin}_2 \otimes \text{Schwingungsmode}_1 \otimes \text{Schwingungsmode}_2 \otimes \dots \quad (4.60)$$

Die Umsetzung dieser Forderung erhöht den Ressourcenverbrauch allerdings enorm. Nimmt man nur drei Zustände für eine neue Schwingungsmode mit auf, so brauchen die Matrizen bereits neunmal soviel Speicher wie zuvor; eine

Matrixmultiplikation würde sogar 27 mal so lang dauern wie vorher. Wir können die Anzahl der zu verwendenden Zustände allerdings reduzieren. Mögen z.B. die Zustände $|\text{Mode1}, \text{Mode2}\rangle = |15, 0\rangle$ und $|\text{Mode1}, \text{Mode2}\rangle = |0, 3\rangle$ für eine genaue Berechnung gerade noch nötig sein, so ist es $|15, 3\rangle$ dann sicherlich nicht mehr. Wir müssen daher die Zustände neu anordnen und unnötigen Ballast streichen. Dies kann durch eine $m \times n$ -Matrix \mathfrak{P} besorgt werden. Wir können dann den Hamilton-Operator wie gewohnt in der Produktbasis aufstellen, um ihn anschließend mit $\mathfrak{P}H\mathfrak{P}^{-1}$ in die platzsparende Form zu transformieren¹⁰.

¹⁰Der Korrektheit halber sei erwähnt, dass \mathfrak{P} nur ein Pseudoinverses besitzt. \mathfrak{P}^{-1} transformiert alle noch vorhandenen Zustände in die alte Produktbasis zurück und setzt Nullen für die nun fehlenden, extrem unwahrscheinlichen Zustände.

Kapitel 5

Impulssequenzen

Im letzten Kapitel haben wir Wege gesehen, wie das $\sigma_z \otimes \sigma_z$ - und das Mølmer-Sørensen-Gatter mit hohen Güten auf zwei Ionen zu verwirklichen sind. Auch ohne die Verwendung der hier betriebenen Optimierung wurde das Mølmer-Sørensen-Gatter in unserer Gruppe bereits mit einer Güte von 99,3% nachgewiesen [10]. Es erscheint daher überaus angebracht, nach Möglichkeiten zu suchen, diesen Gattertyp zum Aufbau komplexerer Operationen zu nutzen. Hierfür benötigen wir noch weitere, ebenfalls mit hoher Güte erzeugbare Gatter. Die Aufgabe wird dann darin bestehen, diese elementaren Gatter in der richtigen Reihenfolge und Länge aneinanderzureihen. Auch dieses Problem werden wir wieder einem Optimierungsalgorithmus zuführen.

5.1 Grundbausteine

Als erstes haben wir zu klären, mit welchem Satz an elementaren Basisoperation wir künftig operieren wollen. Aufgrund der guten Erfahrung in unserer Gruppe fiel die Wahl auf das Mølmer-Sørensen-Gatter zur Erzeugung von Verschränkungen. Wir lassen das MS-Gatter in zwei Varianten zu

$$\text{MS}_x(\xi) = e^{+\frac{i}{4} \cdot \xi \cdot (S_x)^2} \quad S_x = \sum_{j=1}^{\text{alle Ionen}} \sigma_x^{(j)} \quad (5.1)$$

$$\text{MS}_y(\xi) = e^{+\frac{i}{4} \cdot \xi \cdot (S_y)^2} \quad S_y = \sum_{j=1}^{\text{alle Ionen}} \sigma_y^{(j)} \quad . \quad (5.2)$$

Der Faktor $\frac{1}{4}$ im Exponenten wurde eingeführt, um dem intuitiven Bloch-Vektor-Bild möglichst gut zu entsprechen. Man beachte auch das positive Vorzeichen im Exponenten. Dies hängt mit der Wahl des Vorzeichens in $\delta = \nu - \varepsilon$ (1.34) zusammen. Wählen wir stattdessen $\delta = \nu + \varepsilon$, so erhalten wir ein negatives Vorzeichen im Exponenten. Die maximale Verschränkung erhalten wir für $\xi = \frac{\pi}{2} + n \cdot \pi$. Für zwei Ionen gilt z.B.

$$\text{MS}_x\left(\frac{\pi}{2}\right) \cdot |0, 0\rangle \sim |0, 0\rangle + i \cdot |1, 1\rangle. \quad (5.3)$$

Wir wollen uns aber nicht auf zwei Ionen beschränken, sondern eine beliebige Anzahl zulassen. Die mathematische Beschreibung (5.1) und (5.2) beinhaltet bereits diese Verallgemeinerung. Das MS-Gatter soll immer mit allen Ionen gleichzeitig wechselwirken und wird dementsprechend alle Ionen miteinander verschränken. In diesem Sinne ist das MS-Gatter *global*.

Als weitere Basisoperationen lassen wir die beiden nicht verschränkenden, aber ebenfalls globalen, d.h. mit allen Ionen gleichzeitig wechselwirkenden Impulse $X(\xi)$ und $Y(\xi)$ zu

$$X(\xi) = e^{-\frac{i}{2}\xi \cdot S_x} \quad (5.4)$$

$$Y(\xi) = e^{-\frac{i}{2}\xi \cdot S_y} \quad , \quad (5.5)$$

wobei die Normierung wieder die Verbindung zum Bloch-Vektor-Bild herstellt. Bisher verwenden wir nur globale Operationen, die alle Ionen gleich behandeln. Dies allein reicht nicht aus, um den einzelnen Qubit-Ionen spezielle Aufgaben zuweisen zu können. Deshalb führen wir noch lokale σ_z -Impulse ein, die jeweils nur mit einem Ion (Nr. j) wechselwirken

$$Z_j(\xi) = e^{-\frac{i}{2}\xi \cdot \sigma_z^{(j)}} \quad (5.6)$$

Damit hätten wir all unsere Basisoperationen benannt. Zur Vereinfachung des Sprachgebrauchs werden wir diese Operationen von jetzt ab einfach als *Impulse* bezeichnen. Eine *Impulssequenz* ist die Aneinanderreihung verschiedener Impulse. Den Begriff *Gatter* verwenden wir zukünftig für die komplexe Zieloperation, die durch eine Impulssequenz verwirklicht werden soll. Als Beispiel geben wir hier eine Impulssequenz für das Toffoli-Gatter (1.10) auf drei Ionen an, die mit dem in diesem Kapitel beschriebenen Optimierungsverfahren gefunden wurde. Dazu benötigen wir kein MS_y und auch nur einen Z_j -Impuls auf dem dritten Ion (das Zielqubit – Ion 1 und 2 stellen die Kontrollqubits):

$$\begin{array}{c} \text{Toffoli} \\ \text{---} \bullet \text{---} \\ \text{---} \bullet \text{---} \\ \text{---} \oplus \text{---} \end{array} = Y\left(-\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right) MS_x\left(-\frac{\pi}{2}\right) Z_3\left(\frac{\pi}{2}\right) MS_x\left(-\frac{\pi}{4}\right) X\left(-\frac{\pi}{4}\right) Z_3\left(-\frac{\pi}{2}\right) X\left(-\frac{\pi}{2}\right) MS_x\left(-\frac{\pi}{2}\right) Z_3\left(\frac{\pi}{4}\right) Y\left(\frac{\pi}{2}\right). \quad (5.7)$$

Die Reihenfolge der Impulse entspricht der Produktdarstellung des Zeitentwicklungsoperators, d.h. der zeitlich erste Impuls steht ganz rechts und der letzte dementsprechend ganz links.

Aus mathematischer Sicht wären es sinnvoll, auch noch die zu Z_j analogen, lokalen X_j und Y_j in den Kanon der zulässigen Impulse mit aufzunehmen. So benötigt man z.B. für das Toffoli-Gatter unter Mitverwendung von X_j oder Y_j nur neun Impulse und nicht elf wie in (5.7). Technisch ist es aber schwierig, die nötige Phasentreue zwischen den global und den lokal agierenden X - und Y - Impulsen zu gewährleisten. Einen Z_j -Impuls hingegen kann

man über die AC-Stark-Verschiebung mit einem stark verstimmtten Laser erzeugen, wobei die Phase dann keine Rolle spielt. Ein weiterer Vorteil besteht in der quadratischen Abhängigkeit der AC-Stark-Verschiebung von der Rabi-Frequenz: Es bleibt nicht aus, dass der Laserstrahl eine kleine, ungewollte Überlappung mit den Nachbarionen hat. Die daraus resultierende AC-Stark-Verschiebung fällt nun ebenfalls quadratisch mit der dort noch vorherrschenden Rabi-Frequenz ab.

5.2 Optimierungsverfahren

Zur Optimierung der Impulssequenzen greifen wir auf den in den vorherigen Kapiteln beschriebenen Algorithmus zurück. Natürlich sind dabei einige Anpassungen vorzunehmen. Vieles wird dabei einfacher. Wir haben keinen zeitabhängigen Hamilton-Operator mehr und können das ganze Konzept der Dysonentwicklung über Bord werfen. Wir müssen nur noch Produkte aus

$$\exp(-i \cdot \xi_i \cdot \hat{O}_i) \quad \hat{O}_i = \text{konstanter Operator} \quad (5.8)$$

nach ξ_i optimieren. Des Weiteren wirken alle Operatoren auf die Schwingungszustände wie die Einheitsmatrix – wir können diesen Teil also vernachlässigen, wodurch sich unsere Matrizen erheblich verkleinern.

Aber es gibt auch neue Schwierigkeiten, wobei ein Problem alle anderen an Wichtigkeit überschattet: Wir müssen bei der Optimierung der Impulssequenzen nicht nur die Länge der einzelnen Impulse – sprich ξ_i – bestimmen, sondern auch deren Anordnung. Es gibt eine enorme Anzahl an Kombinationsmöglichkeiten für die verschiedenen Impulstypen. Wir können dabei noch nicht einmal vorhersagen, wie groß die Gesamtzahl der Impulse zu sein hat. Alle Impuls-Kombinationen mit der Brute-Force-Methode durchzuprobieren, wird aber durch den freien Parameter ξ_i aufs fast Unmögliche erschwert. Die hier präsentierte Lösung besteht nun darin, mit einer viel zu großen, regelmäßig angeordneten Impulssequenz zu starten. Die Idee dabei ist, dass die ideale Impulssequenz als Teilfolge in der großen Sequenz enthalten ist. Jetzt muss man es nur noch schaffen, die ungewollten Impulse loszuwerden. Dies mag sich ein wenig wie das Rezept zur Schaffung einer Statue anhören: „*Nimm einen großen Stein und hau alles weg, was nicht zur Statue gehört*“. Wie entledigen wir uns der überflüssigen Impulse? Wir müssen dafür sorgen, dass im Rahmen der ξ_i -Optimierung möglichst viele ξ_i den Wert null annehmen. Dies geschieht durch die Verwendung einer geeigneten Nebenbedingung in Form einer Straffunktion (Abschnitt 2.4). Im Rahmen der Optimierung wird der Computer bemüht sein, die Straffunktion möglichst klein werden zu lassen. Deshalb benötigen wir eine Straffunktion, die eine Impulssequenz aus vielen kurzen Impulsen mit einer höheren Strafe belegt, als eine Impulssequenz aus wenigen, aber dafür langen Impulsen. Die Gesamtlänge der

Impulse – also $\sum |\xi_i|$ – sei dabei in beiden Fällen als gleich groß angenommen. Eine Straffunktion der Art

$$\text{Strafe} = \sum_{i=1}^{\text{Anzahl Impulse}} |\xi_i|^\gamma \quad \text{mit} \quad 0 < \gamma < 1 \quad (5.9)$$

erfüllt diesen Zweck, wobei meist $\gamma = \frac{1}{2}$ also $\sqrt{|\xi_i|}$ verwendet wurde. Da die Nebenbedingung (5.9) der entscheidende Schlüssel zur Impulsoptimierung ist, wollen wir uns die Zeit nehmen, sie etwas näher zu betrachten. Die Straffunktion geht immer in Kombination mit der Gewinnfunktion (2.3) in den Optimierungsprozess ein. Wir müssen also sicherstellen, dass die Gewinnfunktion der Straffunktion nicht (zu stark) entgegenwirkt. Zur Einschätzung des Verhaltens der Gewinnfunktion betrachten wir die Auswirkungen einer einzelnen Veränderung $\xi_i^{\text{alt}} \rightarrow \xi_i^{\text{neu}}$ auf den Operator der Impulssequenz U_{Sequenz} . Letzteren stellen wir als Produkt der unitären Operationen U_i der einzelnen Impulse dar

$$\begin{aligned} U_{\text{Sequenz}} &= \underbrace{U_n \cdot U_{n-1} \cdot U_{n-2} \cdots U_{i+1}}_{U_{n \dots i+1}} \cdot \underbrace{U_i}_{\exp(-i \cdot \xi_i^{\text{alt}} \cdot \hat{O}_i)} \cdot \underbrace{U_{i-1} \cdots U_1}_{U_{i-1 \dots 1}} \quad (5.10) \\ &= U_{n \dots i+1} \cdot \exp(-i \cdot \xi_i^{\text{alt}} \cdot \hat{O}_i) \cdot U_{i-1 \dots 1} \\ &\rightarrow U_{n \dots i+1} \cdot \exp(-i \cdot \xi_i^{\text{neu}} \cdot \hat{O}_i) \cdot U_{i-1 \dots 1} \\ &= U_{n \dots i+1} \cdot \exp(-i \cdot \xi_i^{\text{alt}} \cdot \hat{O}_i) \cdot \exp(-i \cdot [\xi_i^{\text{neu}} - \xi_i^{\text{alt}}] \cdot \hat{O}_i) \cdot U_{i-1 \dots 1}. \end{aligned}$$

Wir können die Veränderung von U_{Sequenz} unter $\xi_i^{\text{alt}} \rightarrow \xi_i^{\text{neu}}$ also durch Einfügen des neuen Impulses

$$\exp(-i \cdot [\xi_i^{\text{neu}} - \xi_i^{\text{alt}}] \cdot \hat{O}_i) \quad (5.11)$$

beschreiben. Der entscheidende Parameter, von dem auch die Veränderung der Gewinnfunktion abhängt, ist damit

$$\Delta \xi_i = \xi_i^{\text{neu}} - \xi_i^{\text{alt}} \quad (5.12)$$

und nicht etwa $\frac{\xi_i^{\text{neu}}}{\xi_i^{\text{alt}}}$ oder Sonstiges. Nach dieser Feststellung berechnen wir die Auswirkung von $\xi_i \rightarrow \xi_i + \Delta \xi_i$ auf die Straffunktion (5.9) für $\gamma = \frac{1}{2}$

$$\begin{aligned} \Delta \text{Strafe} &= \sqrt{|\xi_i + \Delta \xi_i|} - \sqrt{|\xi_i|} \\ &\approx \text{sign}(\xi_i) \frac{d}{d|\xi_i|} \sqrt{|\xi_i|} \cdot \Delta \xi_i \\ &= \text{sign}(\xi_i) \frac{1}{2\sqrt{|\xi_i|}} \cdot \Delta \xi_i. \quad (5.13) \end{aligned}$$

Zusammenfassend stellen wir also fest, dass die Veränderung des Operators der Impulssequenz U_{Sequenz} und damit der Gewinnfunktion nur von $\Delta \xi_i$ abhängt und nicht von $|\xi_i|$. Der Einfluss, den die Nebenbedingung auf die ξ_i

ausübt, hängt hingegen sehr wohl von deren Größe ab. Je kleiner ein $|\xi_i|$ ohnehin schon ist, desto stärker der Druck, noch kleiner zu werden. Dies sorgt dafür, dass viele ξ_i zu null werden und die Impulse somit wie gewollt verschwinden. Mit der Einbindung der Straffunktion (5.9) haben wir also ein wirksames Verfahren, um eine große Anzahl an Impulsen auf eine kleine, brauchbare Anzahl schrumpfen zu lassen. Damit ist unser Hauptproblem bereits gelöst.

Jetzt wollen wir uns einem kleinen Schönheitsfehler zuwenden. Bisher waren wir immer in der angenehmen Position, mit Nebenbedingungen zu arbeiten, deren Straffunktionen Polynome zweiten Grades waren. Da wir diesen Luxus nicht missen wollen, schreiben wir (5.9) etwas um:

$$\text{Strafe} = \sum_{i=1}^{\text{Anzahl Impulse}} |\xi_i^{\text{alt}}|^{\gamma-2} \cdot |\xi_i|^2 \quad \text{mit} \quad \xi_i^{\text{alt}} = \text{const}, \quad (5.14)$$

wobei ξ_i^{alt} den Wert von ξ_i vom letzten Optimierungsschritt darstellt. Da abgesehen von der Anfangsphase die Veränderungen von ξ_i pro Optimierungsschritt eher klein ausfallen, wird sich die Größe der Variablen ξ_i nicht weit von ihrem Ursprungswert ξ_i^{alt} entfernen, womit (5.14) weitgehend identisch wie (5.9) wirkt. Für die Praxis ist noch die Substitution

$$|\xi_i^{\text{alt}}|^{\gamma-2} \rightarrow (|\xi_i^{\text{alt}}| + \varepsilon)^{\gamma-2} \quad (5.15)$$

zu empfehlen, was nicht nur das Abstürzen des Computerprogramms aufgrund einer Division durch null verhindert, sondern auch die „Wiederauferstehung“ von Impulsen mit $\xi_i = 0$ zulässt. Im Rahmen dieser Diplomarbeit wurde mit $\varepsilon = 10^{-10}$ gerechnet.

Natürlich kann das bisher beschriebene Verfahren noch weiter verbessert werden. Folgende drei Punkte sollten z.B. von Zeit zu Zeit abgearbeitet werden (z.B. alle 10 Optimierungsdurchläufe, wobei ein Durchlauf die einmalige Optimierung aller ξ_i beinhaltet) :

1. Entferne alle Impulse mit $\xi_i \approx 0$. Setze zwischen die verbliebenen Impulse neue, zufällig ausgewählte Impulse mit dem Startwert $\xi_i^{\text{Start}} = 0$. Damit muss die Lösung nicht von Anfang an als Teilfolge in der Startsequenz vorhanden sein. Sie kann jetzt im Laufe des Optimierungsprozesses erwachsen.
2. Überprüfe, ob einige Impuls-Kommutatoren so klein ausfallen, dass zwei Impulse gleichen Typs zusammengefasst werden können, z.B.

$$X(\xi_1) \text{MS}_x(\xi_2) X(\xi_3) = X(\xi_1 + \xi_3) \text{MS}_x(\xi_2).$$

3. Überprüfe, ob einige Impulse ersatzlos gestrichen werden können. Dies ist vor allem dann sinnvoll, wenn im Rahmen des Simulated Annealing (siehe unten und Abschnitt 2.5) die numerische Temperatur erhöht wird, was einen störenden Effekt auf das Löschen der Impulse via Nebenbedingung hat.

Es gibt unendlich viele Impulssequenzen, die ein und dasselbe Gatter erzeugen. Dies entspricht unendlich vielen Lösungen für unser Optimierungsproblem. Nun sind aber nicht alle Lösungen von der gleichen Qualität. Je kürzer eine Impulssequenz ist, desto bessere Resultate sind bei der praktischen Umsetzung zu erwarten. Da ein Mølmer-Sørensen-Impuls viel aufwendiger als die anderen Impulse ist, liegt ein besonderes Augenmerk auf dessen Reduzierung. Letzteres kann man auch in den Nebenbedingungen berücksichtigen.

Jede beliebige Impulssequenz, die das Gatter erzeugt, entspricht einem der unendlich vielen Maxima der Gewinnfunktion (2.3). Da wir aber nicht irgendeine, sondern die bestmögliche Impulssequenz suchen, ist ein besonderes Augenmerk auf einen Algorithmus zu legen, der aus einem lokalen Extremum zu entkommen vermag. Hierfür hat sich das im Abschnitt 2.5 beschriebene Simulated Annealing als überaus wertvolles Werkzeug erwiesen. Aber auch schon zum Auffinden der ersten, noch nicht optimalen Lösung ist das Simulated Annealing bei komplexeren Aufgaben fast unentbehrlich geworden.

Der gesamte Optimierungsprozess geht in der Regel einher mit vielfachem Aufheizen und Abkühlen der numerischen Temperatur und regelmäßigem Zwischenspeichern. Im Laufe der Zeit konnte der Autor eine recht gute Intuition für die notwendigen Operationen entwickeln, die auch direkte Manipulation der Impulssequenzen beinhalten. Dies in Form eines Computerprogramms zu automatisieren, bleibt aber Aufgabe zukünftiger Entwicklungen.

5.3 Resultate

Zuerst merken wir an, dass wir mit den von uns verwendeten Impulsen (Abschnitt 5.1) jede unitäre Transformation erzeugen können. Da bereits bekannt ist, dass jede unitäre Transformation aus lokalen Drehungen und dem CNOT-Gatter aufbaubar ist [11], reicht es zu zeigen, dass wir diese Operationen verwirklichen können. Die lokalen Drehungen können durch $\sigma_{x,y,z}$ -Rotationen erzeugt werden. Die lokalen σ_z -Rotationen gehören dem Kanon unserer Impulse an. Eine σ_x -Rotation um den Winkel ξ auf dem j ten Qubit wird erzeugt durch

$$\exp(-i\frac{\xi}{2}\sigma_x^{(j)}) = Z_j(\pi) X(-\frac{\xi}{2}) Z_j(\pi) X(\frac{\xi}{2}). \quad (5.16)$$

Entsprechendes gilt für σ_y . Hierbei entstammt das $\frac{1}{2}$ im Exponenten links der Normierung bzw. der Abbildung $SU(2) \rightarrow O(3)$. Das $\frac{1}{2}$ auf der rechten

Seite entspricht der Aufteilung von ξ auf die zwei Anteile $X(-\frac{\xi}{2})$ und $X(\frac{\xi}{2})$. Auf n Ionen kann ein CNOT z.B. durch folgende, elf Impulse umfassende Sequenz erzeugt werden

$$\begin{array}{c} \text{CNOT} \\ \text{---} \bullet \text{---} \\ \text{---} \oplus \text{---} \\ \vdots \end{array} = \begin{array}{l} Z_1(-\frac{\pi}{2}) X(-\frac{\pi}{4}) MS_x(\frac{\pi}{4}) Z_2(\pi) MS_x(-\frac{\pi}{4}) X(-\frac{\pi}{4}) \\ Z_1(-\frac{\pi}{2}) X(\frac{\pi}{2}) MS_x(\frac{\pi}{4}) Z_2(\pi) MS_x(-\frac{\pi}{4}), \end{array} \quad (5.17)$$

wobei o.B.d.A. das erste Ion als Kontrollqubit und das zweite als Zielqubit gewählt wurde. Man beachte, dass $\text{CNOT}^{-1} = \text{CNOT}$. Die Invertierung der obigen Impulssequenz ergibt also wieder ein CNOT. Hat man z.B. zwei aufeinander folgende CNOTs, die beide das gleiche Zielqubit besitzen, so heben sich durch die Anordnung

$$(\text{Impulssequenz}_{\text{cnot}}) \cdot (\text{Impulssequenz}_{\text{cnot}})^{-1} \quad (5.18)$$

die letzten vier Impulse des CNOTs und die ersten vier des CNOT^{-1} gegenseitig weg.

In der Praxis wird man die oben vorgestellte Realisierung des CNOTs vermutlich durch eine 13 Impulse umfassende Sequenz ersetzen wollen, bei der die MS-Impulse nur halb so lang sind

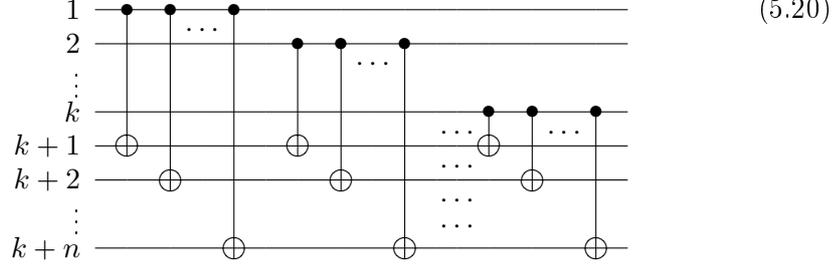
$$\begin{array}{l} X(-\frac{\pi}{2}) Z_1(\frac{\pi}{2}) MS_x(-\frac{\pi}{8}) Z_2(\pi) MS_x(\frac{\pi}{8}) X(-\frac{\pi}{4}) Z_1(\pi) \\ MS_x(-\frac{\pi}{8}) Z_2(\pi) MS_x(\frac{\pi}{8}) X(\frac{\pi}{4}) Z_1(\frac{\pi}{2}) X(\frac{\pi}{2}). \end{array} \quad (5.19)$$

5.3.1 Globale Strukturen

Bisher wurde von uns kein einziger MS_y -Impuls verwendet und es ist möglich, gänzlich auf diesen Impulstyp zu verzichten. Da es uns aber nicht darum geht, nur den minimal nötigen Satz an Impulsen zu verwenden, werden wir in sinnvollen Fällen den „Luxus“ eines MS_y in Anspruch nehmen.

Eine ähnliche Feststellung können wir auch eine Stufe höher treffen: Lokale Drehungen und das CNOT mögen ausreichend sein, um jede unitäre Transformation zu erzeugen, aber das heißt nicht, dass wir uns auf die Verwendung dieser Gatter beschränken müssen. Im Gegenteil: Da die von uns eingesetzten Mølmer-Sørensen-Impulse alle Ionen gleichzeitig verschränkt, können wir einige Gatter, die eine Vielzahl ähnlicher Elemente enthalten und hier als *global* bezeichnet seien, extrem einfach erzeugen. Als Beispiel für ein derartiges

globales Gattern schauen wir uns folgendes Schaltbild an



Jedes der ersten k Qubits kontrolliert n NOTs – eines auf jedem der verbleibenden Qubits. Sämtliche $k \cdot n$ CNOTs werden durch folgende Impulse auf einmal verwirklicht¹

$$\begin{aligned} & X\left(-\frac{\pi}{2}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) MS_x\left(-\frac{\pi}{4}\right) X\left(-\frac{k+n}{4}\pi\right) Z_{1\dots k}(\pi) \\ & MS_x\left(\frac{\pi}{4}\right) X\left(\frac{n-k}{4}\pi\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right). \end{aligned} \quad (5.21)$$

Hierbei haben wir die lokalen Z_j Impulse in einem Impuls zusammengefasst

$$Z_{1\dots k}(\xi) := Z_1(\xi) Z_2(\xi) \dots Z_k(\xi). \quad (5.22)$$

Da der globale $Z_{1\dots k+n}(\frac{\pi}{2})$ -Impuls unkompliziert (aber nicht trivial) mit den anderen Impulsen kommutiert, können wir über

$$Z_{1\dots k}\left(\frac{\pi}{2}\right) = Z_{1\dots k+n}\left(\frac{\pi}{2}\right) Z_{k+1\dots k+n}\left(-\frac{\pi}{2}\right) \quad (5.23)$$

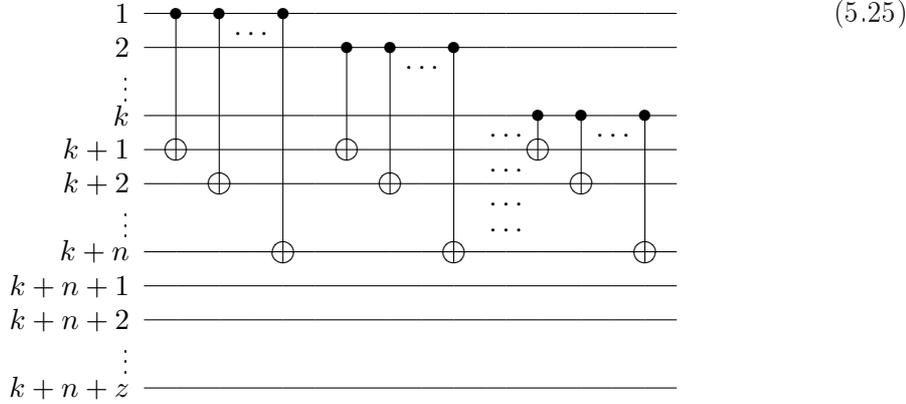
leicht eine Impulssequenz angeben, bei der die Zielqubits statt der Kontrollqubits mit einem $Z_{k+1\dots k+n}$ -Impuls wechselwirken.

$$\begin{aligned} & X\left(-\frac{\pi}{2}\right) Z_{k+1\dots k+n}\left(\frac{\pi}{2}\right) MS_y\left(-\frac{\pi}{4}\right) Y\left(\frac{k+n}{4}\pi\right) Z_{k+1\dots k+n}(\pi) \\ & MS_y\left(\frac{\pi}{4}\right) Y\left(-\frac{k-n}{4}\pi\right) Z_{k+1\dots k+n}\left(\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right). \end{aligned} \quad (5.24)$$

In den beiden obigen Impulssequenzen zeichnen wir entweder die Kontrollqubits oder die Zielqubits durch Z_j -Impulse aus. Dies ist nicht mehr ausreichend, wenn eine dritte Kategorie hinzu kommt: *Zuschauerqubits*. Darunter wollen wir unbeteiligte Qubits verstehen, die nicht mit den übrigen wechselwirken. Da unsere Qubit-Interaktionen aber über die global verschränkenden Mølmer-Sørensen-Impulse vermittelt werden, lassen sich Zuschauerqubits nicht trivial ergänzen. Dies führt zu einer Verkomplizierung der Impulssequenzen, was sich nicht nur in weiteren Z_j -Impulse niederschlägt.

¹Viele der hier vorgestellten Impulssequenzen beinhalten Verallgemeinerungen auf beliebige Qubit-Zahlen. Diese Verallgemeinerungen sind nicht streng mathematisch hergeleitet, sondern wurden aus einer hinreichend groß erscheinenden Menge an Einzelresultaten deduziert.

Ergänzen wir obiges Schaltbild also durch Zuschauerqubits



Die Impulssequenz hierfür lautet

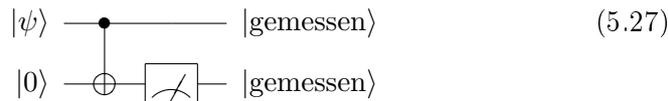
$$\begin{aligned}
 & X\left(-\frac{\pi}{2}\right) Z_{1\dots k}\left(\frac{\pi}{2}\right) X\left(\frac{k-n}{4}\pi\right) MS_x\left(-\frac{\pi}{8}\right) Z_{k+1\dots k+n}(\pi) X\left(-\frac{k}{4}\pi\right) MS_x\left(\frac{\pi}{8}\right) \\
 & Z_{1\dots k}(\pi) MS_x\left(-\frac{\pi}{8}\right) Z_{k+1\dots k+n}(\pi) MS_x\left(\frac{\pi}{8}\right) X\left(\frac{n}{4}\pi\right) Z_{1\dots k}\left(\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right).
 \end{aligned}
 \tag{5.26}$$

Auch diese Impulssequenz können wir mit dem Analogon zu (5.23) und anschließendem Durchkommutieren der globalen $Z_{1\dots k+n+z}$ -Impulse umschreiben und damit entweder $Z_{1\dots k}$ oder $Z_{k+1\dots k+n}$ durch $Z_{k+n+1\dots k+n+z}$ ersetzen.

5.3.2 CNOT in einer anderen Basis

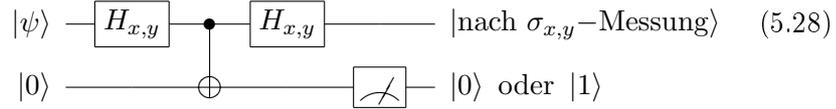
Nicht jedes beliebige Gatter lässt sich durch direkte Optimierung lösen. Bei zu hoher Komplexität wird es notwendig werden, das Gatter in kleinere, lösbare Untereinheiten zu zerlegen und das Gatter aus diesen Lösungen aufzubauen. Dabei ist es natürlich von Vorteil, wenn man auf einen breiten Fundus bereits gelöster Probleme zurückgreifen kann. In diesem Abschnitt wollen wir besagten Fundus um ein paar weitere Lösungen ergänzen.

Zur Motivation der folgenden Impulssequenzen schauen wir uns den Vorgang der indirekten Messung eines Qubits über ein Hilfsqubit² an, der z.B. bei der Quantenfehlerkorrektur Einsatz finden kann (6.4).



²In der Fachliteratur wird das Hilfsqubit als *ancilla* bezeichnet (Latein für Magd, Dienerin)

Dank des CNOTs entspricht die Messung am zweiten Qubit (dem Hilfsqubit) einer Messung von $|\psi\rangle$ (dem Datenqubit). Nun nehmen wir an, der Messapparat kann nur zwischen $|0\rangle$ und $|1\rangle$ unterscheiden. D.h. wir können mit dem obigen Aufbau nur noch σ_z -Messungen durchführen. Um durch eine σ_z -Messungen am Hilfsqubit eine σ_x - und σ_y -Messung am Datenqubit vornehmen zu können, erweitern wir das obige Schaltbild wie folgt



mit

$$\text{---} \boxed{H_x} \text{---} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{und} \quad \text{---} \boxed{H_y} \text{---} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \quad (5.29)$$

wobei H_x die Hadamard-Matrix darstellt. Das Element



bezeichnen wir im Folgenden als C_x NOT bzw. C_y NOT, weil es die jeweiligen Eigenvektoren von σ_x und σ_y sind, die das NOT kontrollieren. Dementsprechend bezeichnen wir das CNOT vorübergehend als C_z NOT.

Schauen wir uns kurz die mathematische Darstellung der $C_{x,y,z}$ NOT an. Die Eigenwerte der $\sigma_{x,y,z}$ sind $+1$ und -1 . Damit sind die Projektoren $P_{x,y,z}^\pm$ auf die jeweiligen Eigenvektoren leicht zu konstruieren³

$$P_{x,y,z}^\pm = \frac{1}{2}(\mathbb{1} \pm \sigma_{x,y,z}). \quad (5.32)$$

Der negative Eigenwert soll die NOT-Operation ($\equiv \sigma_x$) nach sich ziehen. Somit ergibt sich

$$C_{x,y,z}\text{NOT} = \left(\underbrace{P_{x,y,z}^+}_{\text{Kontrollqubit}} \otimes \underbrace{\mathbb{1}}_{\text{Zielqubit}} + \underbrace{P_{x,y,z}^-}_{\text{Kontrollqubit}} \otimes \underbrace{\sigma_x}_{\text{Zielqubit}} \right) \otimes \underbrace{\mathbb{1}}_{\text{Rest}}. \quad (5.33)$$

Analog zu den im vorherigen Abschnitt 5.3.1 angegebenen Impulssequenzen für größere C_z NOT-Strukturen präsentieren wir jetzt die entsprechenden Lösungen für die $C_{x,y}$ NOT. Wie sich zeigt, sind $C_{x,y}$ NOT-Operationen etwas leichter zu erzeugen als C_z NOT-Operationen. Somit ist von der Umsetzung

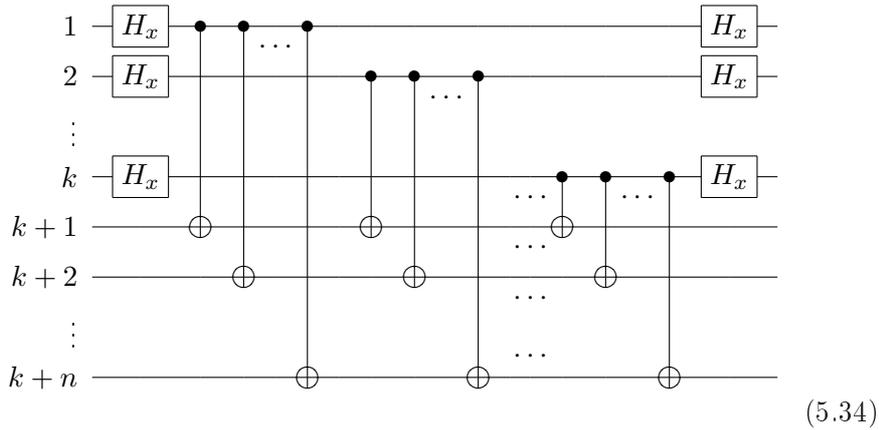
³Man überprüft leicht, dass

$$P_i^\pm \cdot P_i^\pm = \mathbb{1} \quad \text{und} \quad P_i^\pm \cdot P_i^\mp = 0 \quad (5.31)$$

über $H_{x,y} \cdot C_z\text{NOT} \cdot H_{x,y}$ abzuraten⁴.

$C_x\text{NOT}$

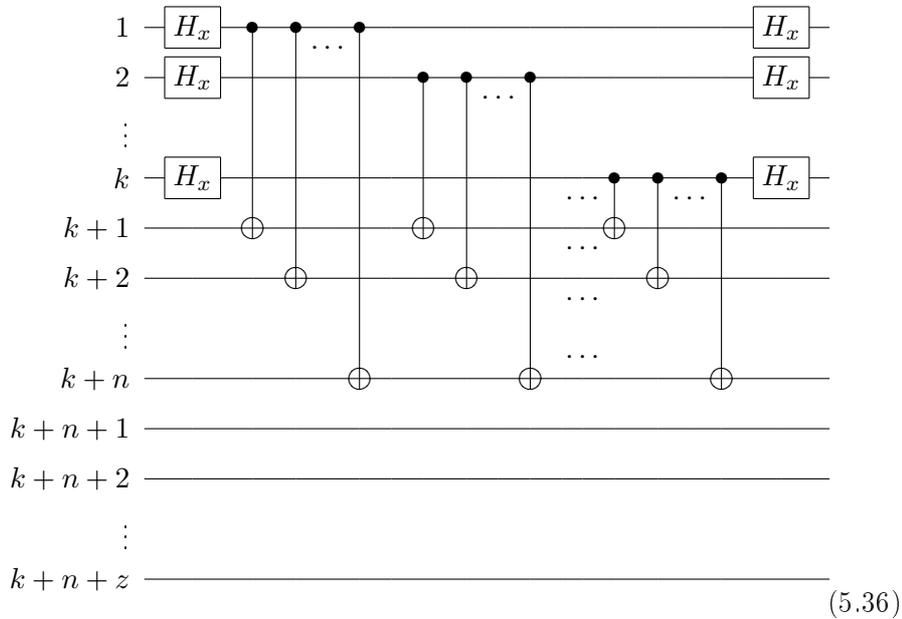
Betrachten wir das $C_x\text{NOT}$ Analogon zu (5.20)



Die entsprechende Impulssequenz ist relativ kurz und lautet

$$\text{MS}_x\left(-\frac{\pi}{4}\right) Z_{1\dots k}(\pi) X\left(\frac{n-k}{4}\pi\right) \text{MS}_x\left(\frac{\pi}{4}\right) Z_{1\dots k}(\pi) X\left(-\frac{k+n}{4}\pi\right) \quad (5.35)$$

Die zu (5.25) analoge Variante mit Zuschauerqubits sieht dann so aus



⁴Bei der experimentellen Umsetzung wird man ohnehin nicht auf die hier verwendeten $H_{x,y}$ zurückgreifen wollen. Das ändert aber nichts am Argument zugunsten der $C_{x,y}\text{NOT}$.

Erzeugt wird dies durch die Impulssequenz

$$\begin{aligned} & Z_{1\dots k}(\pi) X\left(\frac{k}{4}\pi\right) \text{MS}_x\left(-\frac{\pi}{8}\right) Z_{1+k\dots k+n}(\pi) X\left(-\frac{k+n}{4}\right) \text{MS}_x\left(\frac{\pi}{8}\right) \\ & Z_{1\dots k}(\pi) X\left(\frac{n}{4}\pi\right) \text{MS}_x\left(-\frac{\pi}{8}\right) Z_{1+k\dots k+n}(\pi) \text{MS}_x\left(\frac{\pi}{8}\right). \end{aligned} \quad (5.37)$$

C_y NOT

Das Analogon zu (5.34) – also k Kontrollqubits, n Zielqubits und keine Zuschauerqubits – wird erzeugt durch

$$\begin{aligned} & Z_{1\dots k}\left(-\frac{\pi}{2}\right) \text{MS}_x\left(\frac{\pi}{4}\right) X\left(\frac{n-k}{4}\pi\right) Z_{1\dots k}(\pi) \\ & X\left(-\frac{n+k}{4}\pi\right) \text{MS}_x\left(-\frac{\pi}{4}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right). \end{aligned} \quad (5.38)$$

Gibt es Zuschauerqubits, benötigen wir das Analogon zu (5.36). Erzeugt wird dies durch die Impulssequenz

$$\begin{aligned} & Z_{1\dots k}(\pi) X\left(\frac{k}{4}\pi\right) \text{MS}_x\left(-\frac{\pi}{8}\right) Z_{1+k\dots k+n}(\pi) X\left(-\frac{k+n}{4}\pi\right) \text{MS}_x\left(\frac{\pi}{8}\right) \\ & Z_{1\dots k}(\pi) X\left(\frac{n}{4}\pi\right) \text{MS}_x\left(-\frac{\pi}{8}\right) Z_{1+k\dots k+n}(\pi) \text{MS}_x\left(\frac{\pi}{8}\right). \end{aligned} \quad (5.39)$$

5.3.3 Variationen zum NOT

In Abschnitt 5.3.2 haben wir das CNOT in Form des $C_{x,y,z}$ NOT (5.33) verallgemeinert, was das Auslesen des Kontrollqubits in unterschiedlichen Basen gestattet. Nun wollen wir die Menge der Operationen auf dem Zielqubit erweitern. Das NOT ist identisch mit dem σ_x -Operator und eine nahe liegende Verallgemeinerung besteht darin, auch hier die beiden anderen Pauli-Matrizen zuzulassen. In der Literatur (z.B. [11]) ist vor allem das kontrollierte σ_z anzutreffen. Aber auch die kontrollierten $\sqrt{\sigma_z}$ -Gatter und $\sqrt[4]{\sigma_z}$ -Gatter sind gebräuchlich. Dies entspricht kontrollierten π , $\frac{\pi}{2}$ und $\frac{\pi}{4}$ Drehungen um die z -Achse der Blochkugel. Die Impulssequenzen für diese Operatoren sind weiter unten (5.47) & (5.48) angegeben. In Abschnitt 6.6 werden wir dem $C\sqrt{\text{NOT}}$ begegnen, dessen Impulssequenz ebenfalls aus der Erweiterung auf beliebige Drehwinkel folgt, die in diesem Abschnitt vorgestellt wird.

Die Erweiterung auf beliebige Drehwinkel wird uns aber auch ein tieferes Verständnis für die Wirkungsweise der vom Computer generierten Impulssequenzen bescheren. Als erstes wollen wir uns aber kurz die Zeit nehmen, um erstaunt auf die bisherigen Resultate zurückzublicken: Alle Impulse haben die Form

$$\exp\left(-i\frac{m}{2^n}\pi \cdot \text{Operator}\right) \quad \text{mit} \quad m, n \in \mathbb{N}_0. \quad (5.40)$$

Dies wird nicht vom Optimierungsalgorithmus gefordert – es muss also einen physikalischen bzw. mathematischen Grund dafür geben, dass die Impulslänge ξ_i immer als Vielfaches von $2^{-n} \cdot \pi$ auftritt⁵.

Werfen wir als nächstes einen Blick auf die Gesamtlänge der Mølmer-Sørensen-Impulse in den $C_{x,y,z}$ NOT-Schaltbildern

$$\text{Ohne Zuschauerqubits : } \quad \sum |\text{MS}| = 2 \cdot \frac{\pi}{4} = \frac{\pi}{2} \quad (5.41)$$

$$\text{Mit Zuschauerqubits : } \quad \sum |\text{MS}| = 4 \cdot \frac{\pi}{8} = \frac{\pi}{2}. \quad (5.42)$$

Die sich aus der Summe ergebende Gesamtlänge entspricht immer einem maximal verschränkenden $\frac{\pi}{2}$ -Impuls. Da die $C_{x,y,z}$ NOT auch maximal verschränken, darf es nicht überraschen, dass dies die Mindestlänge darstellt⁶. Sehen wir uns nun an, was passiert, wenn wir nicht länger das maximal verschränkende $C_{x,y,z}$ NOT verwenden. Das NOT lässt sich schreiben als

$$\text{NOT} \equiv \sigma_x = \exp\left(i\frac{\pi}{2}(\sigma_x - \mathbb{1})\right) = \exp\left(-i\frac{\pi}{2}(\sigma_x - \mathbb{1})\right). \quad (5.43)$$

Statt nur $\frac{\pi}{2}$ wollen wir jetzt allgemein $\frac{\pi}{\tau}$ mit beliebigem τ zulassen. Dabei ist zu beachten, dass wir das letzte Gleichheitszeichen in (5.43) einbüßen

$$\exp\left(i\frac{\pi}{\tau}(\sigma_x - \mathbb{1})\right) \neq \exp\left(-i\frac{\pi}{\tau}(\sigma_x - \mathbb{1})\right). \quad (5.44)$$

Die bisher präsentierten $C_{x,y,z}$ NOT-Impulssequenzen sind für die Erweiterung mit dem positiven Vorzeichen geeignet (Das negative Vorzeichen gilt dementsprechend für die hermitisch konjugierten Impulssequenzen). Folgende Anpassungen müssen bezüglich der Längen der Impulse (den Argumenten) vorgenommen werden, damit die Impulssequenzen für die Verallgemeinerung des NOTs gelten

1. Die Längen der MS-Impulse sind mit $\frac{2}{\tau}$ zu multiplizieren.
2. Die Längen der X- und Y-Impulse sind mit $\frac{2}{\tau}$ zu multiplizieren, sofern die Längen keine Konstante sind, sondern von der Anzahl der Kontrollqubits k , Zielqubits n (NOTs) oder Zuschauerqubits z abhängen.
3. Die Längen der Z_j -Impulse bleiben konstant.

⁵Erhält man bei der Optimierung eine Impulssequenz, deren Impulse teilweise nicht der Form (5.40) entsprechen, so hat die Praxis gezeigt, dass dies ein starkes Indiz für eine Impulssequenz ist, die noch überflüssige Impulse enthält und weiter verkürzt werden kann.

⁶Es gibt hingegen durchaus Sequenzen mit einem Mehr an Mølmer-Sørensen-Wechselwirkung. Diese kommen z.T. sogar mit einer kleineren Anzahl an Impulsen aus. Da ein MS-Impuls aber wesentlich aufwendiger ist als die anderen, sind diese kürzeren Sequenzen trotzdem ohne besonderen Reiz für uns.

Die Anpassung der Impulssequenz (5.21) für $\mathfrak{r} = 3$ sieht z.B. so aus

$$\begin{aligned} & X\left(-\frac{\pi}{2}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) MS_x\left(-\frac{\pi}{6}\right) X\left(-\frac{k+n}{6}\pi\right) Z_{1\dots k}(\pi) \\ & MS_x\left(\frac{\pi}{6}\right) X\left(\frac{n-k}{6}\pi\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right). \end{aligned} \quad (5.45)$$

Damit können wir folgendes Bild der Wirkungsweise der Impulse erstellen: Die Gesamtlänge der MS-Impulse ist $\frac{\pi}{\mathfrak{r}}$. Diese Gesamtlänge wird aus ein- oder mehrfach halbierten Impulsen aufgebaut. Zwischen diesen kleineren MS-Impulsen führen die lokalen $Z_j(\pi)$ -Impulse zu unterschiedlichen Vorzeichen der einzelnen Terme. Durch diesen Vorzeichenwechsel addieren sich bei einigen Qubits die Verschränkungen konstruktiv und bei anderen destruktiv. Die globalen X- und Y-Impulse können als Kitt zwischen den verschränkten und den Vorzeichen ändernden Impulsen aufgefasst werden, welche die Qubit-Vektoren in die richtige Position drehen. Damit können wir hier sowohl Abhängigkeit als auch Unabhängigkeit von \mathfrak{r} erwarten. So wurden z.B. der erste und der letzte X-Impuls in (5.45) nicht verändert, während die anderen – von k und n abhängigen Impulse – angepasst wurden.

Diese Skizze der Wirkungsweise der Impulse ist aber noch nicht der Weisheit letzter Schluss. Die $Z(-\frac{\pi}{2})$ -Impulse in (5.45) sowie der $Z(\frac{\pi}{4})$ -Impuls im Toffoli-Gatter (5.7) bewirken mehr als die hier aufgeführte Vorzeichenänderung.

Zum Schluss sollen noch die oben erwähnten Impulssequenzen für das kontrollierte σ_z -Gatter angegeben werden. Wir wählen gleich die verallgemeinerte Variante mit dem Operator

$$\exp\left(i\frac{\pi}{\mathfrak{r}}(\sigma_z - \mathbb{1})\right), \quad (5.46)$$

welcher das NOT in den Schaltbildern (5.20) und (5.25) ersetzt.

Betrachten wir zuerst das Analogon zu (5.20) mit k Kontrollqubits, n Zielqubits und keinen Zuschauerqubits

$$\begin{aligned} & Y\left(\frac{\pi}{2}\right) X\left(-\frac{k+n}{2\mathfrak{r}}\pi\right) Z_{1\dots k}(\pi) X\left(\frac{n-k}{2\mathfrak{r}}\pi\right) \\ & MS_x\left(\frac{\pi}{2\mathfrak{r}}\right) Z_{1\dots k}(\pi) MS_x\left(-\frac{\pi}{2\mathfrak{r}}\right) Y\left(-\frac{\pi}{2}\right). \end{aligned} \quad (5.47)$$

Bei Anwesenheit von Zuschauerqubits müssen wir auf das Analogon zu (5.25) zurückgreifen

$$\begin{aligned} & Z_{1\dots k}\left(-\frac{n}{\mathfrak{r}}\pi\right) Z_{1+k\dots k+n}\left(-\frac{k}{\mathfrak{r}}\pi\right) Y\left(-\frac{\pi}{2}\right) Z_{1+k\dots k+n}(\pi) MS_x\left(\frac{\pi}{4\mathfrak{r}}\right) Z_{1\dots k}(\pi) \\ & MS_x\left(-\frac{\pi}{4\mathfrak{r}}\right) Z_{1+k\dots k+n}(\pi) MS_x\left(\frac{\pi}{4\mathfrak{r}}\right) Z_{1\dots k}(\pi) MS_x\left(-\frac{\pi}{4\mathfrak{r}}\right) Y\left(\frac{\pi}{2}\right). \end{aligned} \quad (5.48)$$

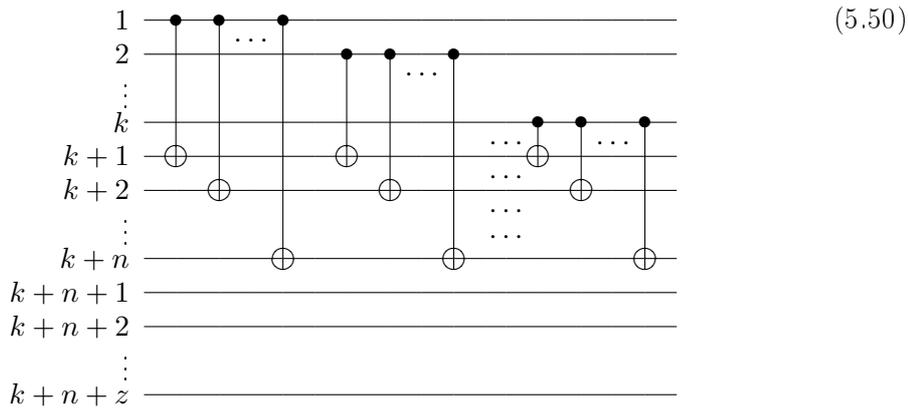
Die zeitlich letzten beiden Z-Impulse $Z_{1\dots k}(-\frac{n}{\tau}\pi)$ $Z_{1+k\dots k+n}(-\frac{k}{\tau}\pi)$ in (5.48) fallen aus unserem bisherigen Erklärungsschema heraus. Diese Z-Impulse erzeugen kein Vorzeichenwechsel zwischen MS-Impulsen, sondern können dahingehend interpretiert werden, dass sie die σ_z -Rotationsbilanz ins Lot bringen (was bei kontrollierten σ_z -Gattern nicht überraschen sollte). Man beachte, dass die Länge des Impulses auf die Kontrollqubits von der Anzahl der Zielqubits abhängt und umgekehrt.

5.4 Analyse der Resultate

Im letzten Abschnitt wurde bereits eine kurze Beschreibung der Wirkungsweise der Impulssequenzen gegeben. In diesem Abschnitt bemühen wir uns, diese noch besser zu verstehen. Dabei begnügen wir uns damit, die Verschränkungseigenschaften zu analysieren. Durch diese Beschränkung sind alle $C_{x,y,z}$ NOT sowie das kontrollierte σ_z zueinander äquivalent – sie unterscheiden sich nur durch lokale unitäre Transformationen. Da uns diese lokalen unitären Operationen hier nicht interessieren sollen, können wir auch die $X(\xi)$ - und $Y(\xi)$ -Impulse vernachlässigen, da sie nur dafür sorgen, dass die Ionen in die richtige Position gedreht werden. Auch ist eine Unterscheidung zwischen $MS_x(\xi)$ und $MS_y(\xi)$ nicht mehr notwendig. In diesem Abschnitt verwenden wir nur die Impulse

$$MS(\frac{\pi}{2}) \quad MS(\pm\frac{\pi}{4}) \quad MS(\pm\frac{\pi}{8}) \quad Z_j(\pi). \quad (5.49)$$

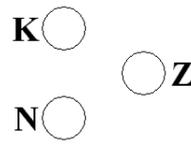
Wir wollen uns mit dem Problem beschäftigen, das exemplarisch durch das Schaltbild (5.25) gegeben ist und hier noch einmal gezeigt sei



k Kontrollqubits kontrollieren⁷ n Zielqubits (NOTs) unter Anwesenheit von z unbeteiligten Zuschauerqubits. Die Analyse des Problems soll graphisch

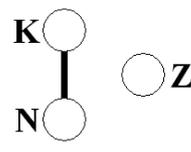
⁷Durch das Vernachlässigen der lokalen unitären Operationen ist es nicht eindeutig, wer hier wen kontrolliert. Die Zielqubits könnten auch die Kontrollqubits kontrollieren. Ein Sachverhalt bleibt aber bestehen: Jedes Kontrollqubit ist mit jedem Zielqubit verschränkt.

betrieben werden. Für den unverschränkten Ausgangszustand verwenden wir das Symbol



$$\begin{array}{c} \mathbf{K} \bigcirc \\ \mathbf{N} \bigcirc \end{array} \quad \bigcirc \mathbf{Z} \quad (5.51)$$

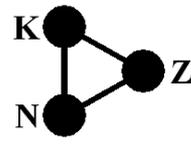
Die drei Kreise stehen entsprechend ihren Buchstaben für die Kontrollqubits (**K**), Zielqubits (**N**) und Zuschauerqubits (**Z**). Den Zustand nach der obigen Operation (5.50) stellen wir dar durch



$$\begin{array}{c} \mathbf{K} \bigcirc \\ | \\ \mathbf{N} \bigcirc \end{array} \quad \bigcirc \mathbf{Z} \quad (5.52)$$

Die verbindende Linie zwischen **K** und **N** zeigt an, dass die Kontrollqubits und Zielqubits miteinander verschränkt sind.

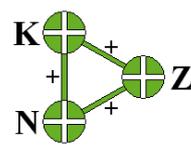
Im Folgenden interessieren wir uns aber weniger für die Zustände, als viel mehr für die Operatoren. Wir verwenden daher für die Operatoren das gleiche Symbol, wie für den Zustand, der sich nach Anwendung des entsprechenden Operators auf den unverschränkten Ausgangszustand (5.51) ergibt. In diesem Sinne beschreiben wir den Operator zum $MS(\frac{\pi}{2})$ -Impuls durch



$$\begin{array}{c} \mathbf{K} \bullet \\ | \\ \mathbf{N} \bullet \end{array} \quad \diagup \quad \bullet \quad \mathbf{Z} \quad (5.53)$$

Ein $MS(\frac{\pi}{2})$ -Impuls verschränkt jedes Ion mit jedem. Daher haben wir drei Linien, welche die Verschränkungen zwischen den drei Gruppen andeuten. Aber auch innerhalb der drei Gruppen sind die Ionen miteinander verschränkt. Dies wird durch die schwarze Füllung der Kreise angezeigt.

Jetzt müssen wir uns den Impulsen $MS(\pm\frac{\pi}{4})$ und $MS(\pm\frac{\pi}{8})$ zuwenden. Wir werden für die beiden Impulslängen $\frac{\pi}{4}$ und $\frac{\pi}{8}$ das gleiche Symbol verwenden, da die Längen aus dem Kontext folgen. Wir müssen aber zwischen den Vorzeichen unterscheiden. Für das positive Vorzeichen verwenden wir das Symbol



$$\begin{array}{c} \mathbf{K} \oplus \\ | \\ \mathbf{N} \oplus \end{array} \quad \diagup \quad \oplus \quad \mathbf{Z} \quad (5.54)$$

und für das negative Vorzeichen

$$(5.55)$$

Damit sind die folgende Impuls-Gleichung und die folgende Symbol-Gleichung äquivalent

$$MS(+\frac{\pi}{4}) MS(-\frac{\pi}{4}) = \mathbb{1} \quad (5.56)$$

Ebenso gilt

$$MS(+\frac{\pi}{4}) MS(+\frac{\pi}{4}) = MS(\frac{\pi}{2}) \quad (5.57)$$

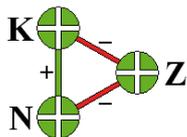
Hierbei wurden die beiden Symbole durch einen Stern * verknüpft, der sowohl als Multiplikation als auch als Addition interpretiert werden kann. Die Zeitentwicklungsoperatoren der Impulse sind Exponentialfunktionen, die multipliziert werden. Dabei addieren sich deren Exponenten gemäß der Baker-Campbell-Hausdorff-Formel. In diesem Sinne soll auch die Verschränkung als additiv betrachtet werden.

Jetzt müssen wir uns der Wirkungsweise der $Z_j(\pi)$ -Impulse zuwenden. Betrachten wir dazu folgende Impulssequenz

$$Z_j(\pi) MS(\xi) Z_j(\pi). \quad (5.58)$$

Der $Z(\pi)$ -Impuls auf das j te Ion hat zur Folge, dass $MS(\xi)$ auf diesem Ion wie ein $MS(-\xi)$ -Impuls wirkt. Alle Verschränkungen des j ten Ions mit den anderen Ionen tragen das umgekehrte Vorzeichen. Mit dieser Erkenntnis gewappnet, wollen wir folgende Impulssequenz und das dazugehörige Symbol betrachten

$$Z_Z(\pi) \text{MS}\left(+\frac{\pi}{4}\right) Z_Z(\pi) \quad \left\{ = Z_{\mathbf{KUN}}(\pi) \text{MS}\left(+\frac{\pi}{4}\right) Z_{\mathbf{KUN}}(\pi) \right\} \quad (5.59)$$



Die Verschränkung der Zuschauerqubits \mathbf{Z} mit den Kontrollqubits \mathbf{K} und den Zielqubit \mathbf{N} trägt jetzt das negative Vorzeichen. Es ist aber zu beachten, dass die Verschränkung der Zielqubits untereinander immer noch das positive Vorzeichen trägt. Hier fand ein doppelter Vorzeichenwechsel statt – das *relative* Vorzeichen zueinander hat sich nicht verändert. Dies ist auch der Grund, warum die globalen $X(\xi)$ - und $Y(\xi)$ -Impulse keinen Einfluss auf die Verschränkungseigenschaften nehmen können: Sie vermögen zwar, einen absoluten Vorzeichenwechsel herbeizuführen, aber keinen relativen.

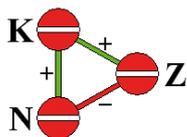
Damit können wir folgende Symbol-Gleichung aufstellen

$$\begin{array}{c} \mathbf{K} \oplus \\ + \\ \mathbf{N} \oplus \end{array} \oplus \mathbf{Z} \quad * \quad \begin{array}{c} \mathbf{K} \ominus \\ + \\ \mathbf{N} \ominus \end{array} \ominus \mathbf{Z} \quad = \quad \begin{array}{c} \mathbf{K} \bullet \\ | \\ \mathbf{N} \bullet \end{array} \bullet \mathbf{Z} \quad (5.60)$$

Was die Verschränkung der drei Gruppen miteinander anbelangt, entspricht dies bereits dem Operator (5.52), den wir zu generieren wünschen. Die Ionen innerhalb der jeweiligen Gruppen sind aber noch miteinander verschränkt (*schwarze Kreise*). Für den Fall, dass wir es nur mit drei Ionen zu tun haben, in jeder der drei Gruppen also nur ein einziges Ion ist, können wir diese Impulssequenz jedoch verwenden.

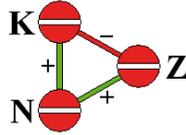
Bei mehr als drei Ionen müssen wir unseren „Wunschoperator“ (5.52) aus vier Graphen aufbauen. Zwei davon sind die beiden Graphen auf der linken Seite von (5.60). Während jeder der beiden Graphen dort noch einen $\text{MS}(\frac{\pi}{4})$ -Impuls als zentrales Element enthielt, handelt es sich im Folgenden nur noch um $\text{MS}(\pm\frac{\pi}{8})$ -Impulse. Die noch benötigten Graphen drei und vier repräsentieren die Impulssequenzen

$$Z_{\mathbf{K}}(\pi) \text{MS}\left(-\frac{\pi}{8}\right) Z_{\mathbf{K}}(\pi) \quad (5.61)$$



sowie

$$Z_{\mathbf{N}}(\pi) \text{MS}\left(-\frac{\pi}{8}\right) Z_{\mathbf{N}}(\pi) \tag{5.62}$$



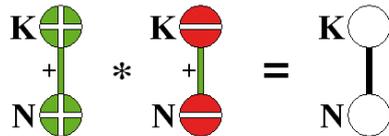
Damit können wir den gewünschten Operator wie folgt aufbauen

$$\tag{5.63}$$

Während die verschränkende Linie zwischen **K** und **N** viermal mit positivem Vorzeichen auftritt, kommen alle anderen Elemente zweimal mit positivem und zweimal mit negativem Vorzeichen vor, was zum gewünschten Resultat führt.

Für den Fall, dass es keine Zuschauerqubits **Z** gibt, wird die Aufgabe fast trivial

$$\text{MS}\left(+\frac{\pi}{4}\right) Z_{\mathbf{N}}(\pi) \text{MS}\left(-\frac{\pi}{4}\right) Z_{\mathbf{N}}(\pi) \tag{5.64}$$



Die ganze Trivialität verschwindet jedoch, wenn wir die lokalen unitären Operationen nicht einfach vernachlässigen können. Dann ist die Hilfe des Computers wieder herzlich willkommen.

Kapitel 6

Quantenfehlerkorrektur

Dieses Kapitel knüpft am vorhergehenden an und behandelt spezielle Impulssequenzen, die im Rahmen einfacher Quantenfehlerkorrekturen Einsatz finden können. Das notwendige Wissen zu deren Verständnis wird ebenfalls bereit gestellt. Zielsetzung ist dabei die Machbarkeit mit den heutigen technischen Möglichkeiten. Für weiterführende Ansätze zur Quantenfehlerkorrektur sei auf die Literatur verwiesen [11, 29].

Zur weiteren Vereinfachung der Impulssequenzen werden wir einen neuen Freiheitsgrad – U_{beliebig} genannt – in die Quantenschaltkreise einführen. Dies erfordert eine Anpassung unseres Algorithmus zum Auffinden der Impulssequenzen, worauf wir in Abschnitt 6.5 eingehen werden.

6.1 Grundlagen und ein erstes Modell

Die Grundlage der Fehlerkorrektur ist die redundante Kodierung der Daten. Betrachten wir folgendes Verfahren

$$|\bar{0}\rangle = |0, 0, 0\rangle \quad \text{und} \quad |\bar{1}\rangle = |1, 1, 1\rangle. \quad (6.1)$$

Hier wird ein logisches Qubit (mit Balken) mittels dreier physikalischer Qubits kodiert. Sollte nun auf einem der drei physikalischen Qubits ein Bitflip-Fehler auftreten, also

$$|0\rangle \leftrightarrow |1\rangle, \quad (6.2)$$

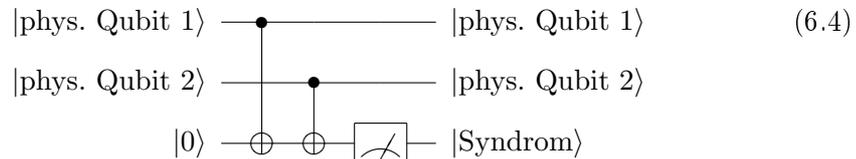
so ist das logische Qubit immer noch aus der Mehrheit der physikalischen Qubits rekonstruierbar. Bei einem Fehler auf dem zweiten physikalischen Qubit sähe dies wie folgt aus

$$\begin{array}{ccc} \alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle & \xrightarrow{\text{Fehler}} & \alpha |0, 1, 0\rangle + \beta |1, 0, 1\rangle \\ & \xrightarrow{\text{Korrektur}} & \alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle. \end{array}$$

In der Quantenmechanik hat die Geschichte allerdings einen Haken: Welchen Wert die Mehrheit der Qubits hat, erfahren wir normalerweise nur durch eine Messung, welche die Wellenfunktion kollabieren lässt und den Großteil der Quanteninformation zerstört. Der Trick besteht nun darin, so zu messen, dass man keine Information über den Wert des logischen Qubits erhält. Zwecks einer möglichst klaren Darstellung dieses Punkts kodieren wir nun ein logisches Qubit mit zwei physikalischen Qubits

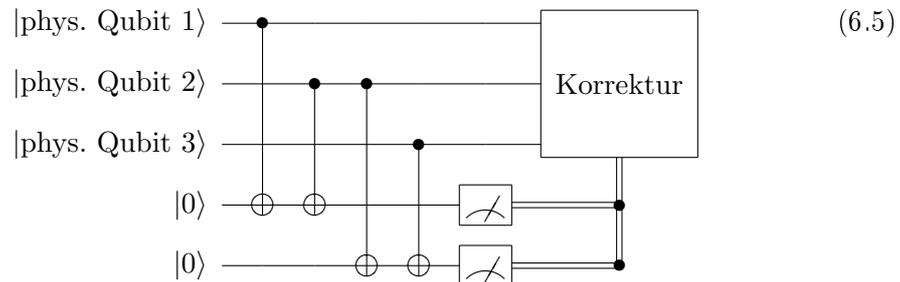
$$|\bar{0}\rangle = |0, 0\rangle \quad \text{und} \quad |\bar{1}\rangle = |1, 1\rangle \quad (6.3)$$

und betrachten eine kombinierte indirekte Messung der beiden physikalischen Qubits über ein Hilfsqubit



Für die beiden Basisvektoren des logischen Qubits $|0, 0\rangle$ und $|1, 1\rangle$ ist jeweils $|\text{Syndrom}\rangle = |0\rangle$. Wir gewinnen aus der Messung also keine Information über den Wert des logischen Qubits, womit der Superpositionszustand $\alpha|0, 0\rangle + \beta|1, 1\rangle$ nicht zerstört wird. Für die fehlerhaften Basen $|1, 0\rangle$ und $|0, 1\rangle$ hingegen ist $|\text{Syndrom}\rangle = |1\rangle$. Wir wissen also, dass ein Fehler aufgetreten ist. Welches physikalische Qubit betroffen ist, können wir nicht sagen (bei einem richtigen und einem falschen Qubit gibt es keine Mehrheit für das richtige). Für eine Fehlerunterdrückung mittels Quanten-Zeno-Effekt reicht (6.4) aber bereits aus.

Zur aktiven Korrektur eines aufgetretenen Fehlers greifen wir wieder auf die Kodierung mittels dreier physikalischer Qubits (6.1) zurück und verwenden folgende Schaltung



Unter der Annahme, dass maximal *ein* Bitflip-Fehler aufgetreten ist, können wir aus dem Messergebnis für $|\text{Hilfsqubit 1}, \text{Hilfsqubit 2}\rangle$ folgende Schlüsse ziehen

1. $|0, 0\rangle \rightarrow$ kein Bitflip-Fehler

2. $|1, 0\rangle \rightarrow$ Bitflip-Fehler auf dem ersten physikalischen Qubit
3. $|0, 1\rangle \rightarrow$ Bitflip-Fehler auf dem dritten physikalischen Qubit
4. $|1, 1\rangle \rightarrow$ Bitflip-Fehler auf dem zweiten physikalischen Qubit

Die vier CNOTs zur Erstellung des Fehlersyndroms können wir z.B. mittels folgender 17 Impulse verwirklichen

$$\begin{aligned}
& Z_2\left(-\frac{\pi}{2}\right) Y\left(-\frac{\pi}{2}\right) Z_3\left(\frac{\pi}{2}\right) Z_2\left(-\frac{\pi}{2}\right) Z_1\left(\frac{\pi}{2}\right) MS_y\left(\frac{\pi}{8}\right) \\
& Z_2(\pi) MS_y\left(-\frac{3}{8}\pi\right) Z_5(\pi) Z_3(\pi) MS_y\left(-\frac{\pi}{8}\right) Z_2(\pi) \\
& MS_y\left(-\frac{\pi}{8}\right) Z_5\left(-\frac{\pi}{2}\right) Z_4\left(\frac{\pi}{2}\right) Z_1(\pi) X\left(-\frac{\pi}{2}\right). \tag{6.6}
\end{aligned}$$

Eine alternative Sequenz aus ebenfalls 17 Impulsen soll auch noch genannt werden

$$\begin{aligned}
& Z_2\left(-\frac{\pi}{2}\right) Y\left(-\frac{\pi}{2}\right) Z_5\left(-\frac{\pi}{2}\right) Z_4\left(-\frac{\pi}{2}\right) MS_x\left(\frac{3}{8}\pi\right) Z_2(\pi) \\
& MS_x\left(-\frac{\pi}{8}\right) Z_5(\pi) Z_3(\pi) MS_x\left(\frac{\pi}{8}\right) Z_2(\pi) MS_x\left(\frac{\pi}{8}\right) \\
& Z_5\left(\frac{\pi}{2}\right) Z_4\left(-\frac{\pi}{2}\right) Z_3(\pi) Z_{1\dots 5}\left(-\frac{\pi}{2}\right) X\left(-\frac{\pi}{2}\right). \tag{6.7}
\end{aligned}$$

Auf den zeitlich zweiten Impuls $Z_{1\dots 5}\left(-\frac{\pi}{2}\right)$ in (6.7) ist besonderes Augenmerk zu legen. Es handelt sich um einen *globalen* Z-Impuls. Somit findet hier mehr Wechselwirkung statt als in der Impulssequenz (6.6). Die technische Umsetzung eines globalen Z-Impulses könnte aber mit weniger Fehlern behaftet sein, als es bei einem lokalen Z-Impuls der Fall ist.

6.2 Korrektur ohne Messung

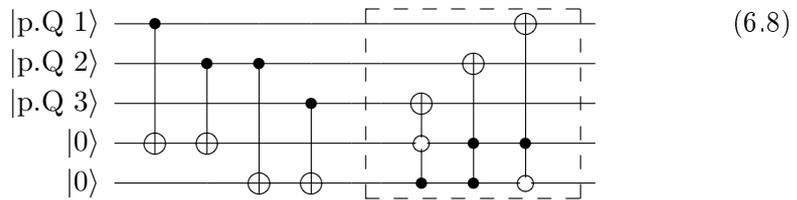
In Abschnitt 6.1 wurde die Grundidee der Quantenfehlerkorrektur präsentiert. Wir sollten uns aber über die Leistungsgrenzen des bisherigen Modells im Klaren sein:

1. Es darf bisher nur *ein* Fehler pro logischem Qubit auftreten.
2. Wir können bisher nur Bitflip-Fehler korrigieren.

Es gibt zahlreiche Kodierungen, für welche die obigen Einschränkungen nicht gelten. Dieses Mehr an Leistung ist aber nur über ein Mehr an Redundanz zu haben. Ist man bereit, diesen Preis zu zahlen, dann reicht es, wenn man jedes physikalische Qubit mittels zwei der drei Pauli-Matrizen $\sigma_{x,y,z}$ messen kann. Wie in Abschnitt 5.3.2 gezeigt, können dafür die eigens definierten $C_{x,y,z}$ NOT verwendet werden. In der Literatur wird oft die Messung mittels

σ_x und σ_z gewählt. Da aber das C_x NOT und C_y NOT etwas leichter zu verwirklichen sind als das klassische C_z NOT, könnten sich σ_x und σ_y als die günstigere Wahl erweisen.

Im Folgenden haben wir aber eine andere Erweiterung im Auge, welche die Anzahl der physikalischen Qubits nicht weiter erhöht. Wir möchten eine Fehlerkorrektur ohne den Messvorgang erreichen. Abgesehen vom ästhetischen Aspekt einer rein unitären Transformation ist der Messvorgang, für den die *Electron Shelving* Technik benutzt wird, mit einem erheblichen Aufheizen der Ionen verbunden und dadurch eher störend. Wir verändern das Schaltbild daher wie folgt

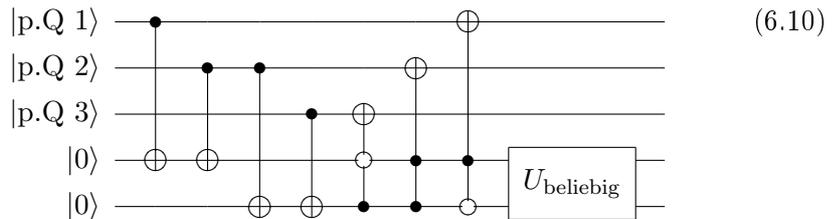


Ein offener „Kontrollpunkt“ $\text{---}\circ\text{---}$ bedeutet, dass dieses Kontrollqubit bei $|0\rangle$ schaltet, während $\text{---}\bullet\text{---}$ weiterhin den Schaltvorgang bei $|1\rangle$ symbolisiert. Wie man leicht überprüft, führt diese Schaltung die notwendigen Korrekturen durch.

Diese unitäre Fehlerkorrektur erlaubt auch ein besseres Verständnis für die Rolle der Hilfsqubits: Mögen sie bisher als Hilfskonstrukt zur Durchführung der Messung erschienen sein, so erkennt man nun ihre Notwendigkeit, um die Reversibilität der Entwicklung des Quantensystems zu garantieren. Beschränken wir uns auf den Raum der drei Datenqubits, so führt die Fehlerkorrektur folgende Abbildung durch

$$\begin{aligned} \{|1, 1, 1\rangle; |1, 1, 0\rangle; |1, 0, 1\rangle; |0, 1, 1\rangle\} &\rightarrow \{|1, 1, 1\rangle\} \\ \{|0, 0, 0\rangle; |0, 0, 1\rangle; |0, 1, 0\rangle; |1, 0, 0\rangle\} &\rightarrow \{|0, 0, 0\rangle\}. \end{aligned} \tag{6.9}$$

Dies allein wäre nicht reversibel. Eine andere Formulierung des Sachverhalts stellt fest, dass die Hilfsqubits die Entropie aufnehmen, die aus dem System der Datenqubits entfernt wird. Wie diese Entropie in den Hilfsqubits gespeichert wird, ist prinzipiell beliebig. Wir können das Schaltbild (6.8) auch ersetzen durch



Wie bereits aus einem Vergleich der Schaltbilder und Impulssequenzen des C_x NOT und des C_z NOT ersichtlich, kann das komplexere Schaltbild durchaus die einfachere Impulssequenz nach sich ziehen. Für (6.10) ist dies auch zu erwarten, denn das Einfügen von U_{beliebig} stellt einen neuen Freiheitsgrad dar, der die Anzahl der möglichen Lösungen und damit die Wahrscheinlichkeit, eine einfachere Impulssequenz zu finden erhöht. In Abschnitt 6.5 wird erklärt, wie man diesen Freiheitsgrad algorithmisch umsetzt.

Die beste im Rahmen dieser Arbeit gefundene Sequenz, um (6.10) zu verwirklichen, kommt mit 34 Impulsen aus

$$\begin{aligned}
& \text{MS}_x\left(\frac{\pi}{4}\right) \text{Z}_4\left(\frac{\pi}{2}\right) \text{MS}_x\left(-\frac{\pi}{8}\right) \text{Z}_2(\pi) \text{X}\left(\frac{\pi}{8}\right) \text{MS}_x\left(-\frac{\pi}{8}\right) \text{Z}_5(\pi) \text{MS}_x\left(\frac{\pi}{8}\right) \\
& \text{Z}_4(\pi) \text{X}\left(-\frac{3}{8}\pi\right) \text{Z}_2(\pi) \text{MS}_x\left(-\frac{\pi}{8}\right) \text{Z}_5(\pi) \text{Z}_4\left(-\frac{\pi}{2}\right) \text{Z}_1(\pi) \text{MS}_x\left(-\frac{\pi}{8}\right) \\
& \text{Z}_5(\pi) \text{Z}_4(\pi) \text{MS}_x\left(\frac{\pi}{8}\right) \text{Z}_3\left(-\frac{\pi}{2}\right) \text{Z}_1\left(-\frac{\pi}{2}\right) \text{X}\left(\frac{\pi}{2}\right) \text{MS}_y\left(\frac{3}{8}\pi\right) \\
& \text{Z}_2(\pi) \text{MS}_y\left(-\frac{\pi}{8}\right) \text{Z}_5(\pi) \text{Z}_3(\pi) \text{MS}_y\left(\frac{\pi}{8}\right) \text{Z}_2(\pi) \\
& \text{MS}_y\left(\frac{\pi}{8}\right) \text{Z}_5\left(\frac{\pi}{2}\right) \text{Z}_4\left(\frac{\pi}{2}\right) \text{Z}_3(\pi) \text{X}\left(-\frac{\pi}{2}\right). \tag{6.11}
\end{aligned}$$

6.3 Mehr durch weniger?

Es sind viele Schemata zur Fehlerkorrektur bekannt, die weitaus mächtiger sind, als das hier vorgestellte Verfahren. Der Preis dafür ist immer ein erheblicher Mehraufwand, der z.Z. nicht technisch umsetzbar ist. Es ist daher durchaus angebracht, die umgekehrte Richtung einzuschlagen und nach noch simpleren Verfahren der Fehlerkorrektur zu suchen, die leichter umzusetzen sind. Ein solches Verfahren soll jetzt vorgestellt werden.

Bisher benötigen wir fünf physikalische Qubits zur Fehlerkorrektur eines logischen Qubits. Von den vier zusätzlichen Qubits werden zwei für die redundante Kodierung benötigt und zwei für das Fehlersyndrom. Ein vereinfachter Ansatz besteht nun darin, die zwei redundanten Datenqubits auch für das Fehlersyndrom zu nutzen. Natürlich können die zwei zusätzlichen Qubits nicht beide Aufgaben zur gleichen Zeit erfüllen. Das müssen sie aber auch nicht, denn die Aufgaben sind zeitlich einigermaßen gut trennbar. Je nach der Aufgabe der beiden zusätzlichen Qubits wollen wir jetzt von Phase 1 und Phase 2 sprechen. Während der Phase 1 dienen die beiden zusätzlich Qubits der Redundanz. Ein Bitflip-Fehler, der in dieser Phase auftritt, ist dank der Redundanz feststellbar und korrigierbar. Diese Korrektur findet in Phase 2 statt. Die beiden zusätzlichen Qubits dienen nun zur Syndromberechnung. In dieser Phase ist das logische Qubit nur noch in einem physikalischen Qubit abgelegt. Fehler, die in Phase 2 auftreten, können daher nicht aufgespürt werden. Nach erfolgter Fehlerkorrektur können die beiden zusätzlichen Qubits wieder zur redundanten Kodierung genutzt werden.

Offensichtlich ist dieses Verfahren der Fehlerkorrektur nur dann sinnvoll, wenn eine Einteilung in Phase 1 & 2 derart erfolgen kann, dass die Wahrscheinlichkeit für einen Fehler größtenteils in Phase 1 fällt.

Zum besseren Verständnis des hier beschriebenen Schemas zur Fehlerkorrektur ist es hilfreich zu verstehen, wie die notwendige Redundanz erzeugt wird

$$(\alpha |0\rangle + \beta |1\rangle) \otimes |0, 0\rangle \rightarrow \alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle. \quad (6.12)$$

Dies wird bewerkstelligt durch einen Spezialfall von (5.20)

$$\left. \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right\} \alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle \quad (6.13)$$

Nochmalige Anwendung der gleichen Operation entkoppelt die Qubits wieder

$$\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle \left\{ \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right. \quad (6.14)$$

Nun fügen wir beide Operationen zusammen und nehmen an, dass dazwischen *ein* Bitflip-Fehler auftreten kann

$$\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle \left\{ \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right. \xrightarrow{\text{Fehler}} \left\{ \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right. \quad (6.15)$$

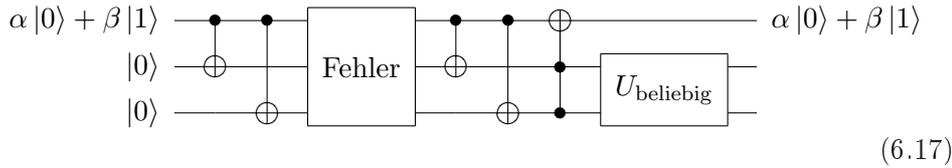
Folgende Resultate erhalten wir in Abhängigkeit des Fehlers

1. $(\alpha |0\rangle + \beta |1\rangle) \otimes |0, 0\rangle \leftrightarrow$ Kein Bitflip-Fehler
2. $(\alpha |1\rangle + \beta |0\rangle) \otimes |1, 1\rangle \leftrightarrow$ Bitflip-Fehler auf dem ersten Qubit
3. $(\alpha |0\rangle + \beta |1\rangle) \otimes |1, 0\rangle \leftrightarrow$ Bitflip-Fehler auf dem zweiten Qubit
4. $(\alpha |0\rangle + \beta |1\rangle) \otimes |0, 1\rangle \leftrightarrow$ Bitflip-Fehler auf dem dritten Qubit

In allen Fällen enthalten Qubit zwei und drei die Information über den Fehler, aber sie verraten nichts über den Wert des logischen Qubits, so dass wir sie problemlos messen können. Wir können den Fehler aber auch ohne Messung mittels eines Toffoli-Gatters korrigieren

$$\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle \left\{ \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right. \xrightarrow{\text{Fehler}} \left\{ \begin{array}{l} \alpha |0\rangle + \beta |1\rangle \\ |0\rangle \\ |0\rangle \end{array} \right. \quad (6.16)$$

Fehler auf dem zweiten oder dritten Qubit müssen nicht korrigiert werden, da deren Wert nicht mehr von Bedeutung ist. Von diesem Umstand machen wir auch Gebrauch, wenn wir analog zu (6.10) die abschließende Erweiterung des Schaltbildes vornehmen



Im Folgenden geben wir zwei Impulssequenzen für die Fehlerkorrektur (also die Operationen nach dem Fehler) an



Die erste Impulssequenz umfasst acht Impulse

$$\begin{aligned}
 & X\left(\frac{\pi}{4}\right) MS_y\left(\frac{\pi}{2}\right) MS_x\left(\frac{\pi}{2}\right) Z_1\left(-\frac{\pi}{2}\right) \\
 & Y\left(-\frac{\pi}{2}\right) Z_1\left(-\frac{\pi}{2}\right) X\left(-\frac{\pi}{4}\right) MS_y\left(\frac{\pi}{2}\right).
 \end{aligned}
 \tag{6.19}$$

Man beachte aber, dass die Gesamtlänge der Mølmer-Sørensen-Impulse mit $\frac{3}{2}\pi$ dem dreifachen Wert der $C_{x,y,z}$ -NOT-Gatter entspricht. Dies können wir auf den zweieinhalbfachen Wert reduzieren, brauchen dann aber insgesamt elf Impulse

$$\begin{aligned}
 & X\left(-\frac{\pi}{4}\right) MS_y\left(-\frac{\pi}{2}\right) Z_1\left(-\frac{\pi}{2}\right) MS_y\left(\frac{\pi}{4}\right) Y\left(\frac{\pi}{4}\right) X\left(\frac{\pi}{2}\right) \\
 & MS_y\left(-\frac{\pi}{4}\right) Z_1(\pi) MS_y\left(\frac{\pi}{4}\right) Z_1\left(\frac{\pi}{2}\right) Y\left(-\frac{\pi}{2}\right).
 \end{aligned}
 \tag{6.20}$$

6.4 Phasenfehler

In der Praxis sind die bisher beschriebenen Bitflip-Fehler weitaus seltener als Phasenfehler¹. Letztere können auf verschiedene Weisen beschrieben werden

$$\begin{aligned}
\alpha |0\rangle + \beta |1\rangle &\xrightarrow{\text{Fehler}} e^{i\phi_1}\alpha |0\rangle + e^{i\phi_2}\beta |1\rangle \\
&= \left(e^{\frac{i}{2}(\phi_1-\phi_2)}\alpha |0\rangle + e^{-\frac{i}{2}(\phi_1-\phi_2)}\beta |1\rangle \right) \cdot \underbrace{e^{\frac{i}{2}(\phi_1+\phi_2)}}_{\text{globale Phase}} \\
&\sim e^{i\cdot\varphi}\alpha |0\rangle + e^{-i\cdot\varphi}\beta |1\rangle \\
&= \cos(\varphi)(\alpha |0\rangle + \beta |1\rangle) + i \sin(\varphi)(\alpha |0\rangle - \beta |1\rangle) \\
&= (\cos(\varphi)\mathbb{1} + i \sin(\varphi)\sigma_z)(\alpha |0\rangle + \beta |1\rangle) \\
&= e^{i\varphi\sigma_z}(\alpha |0\rangle + \beta |1\rangle). \tag{6.21}
\end{aligned}$$

Für uns ist die Feststellung wichtig, dass ein *kontinuierlicher* Phasenfehler φ als *diskreter* Vorzeichenflip beschrieben werden kann, der mit der Wahrscheinlichkeit $\sin^2(\varphi)$ auftritt. Die Korrektur eines Phasenfehlers basiert auf der Suche nach einem solchen Vorzeichenflip. Dieser ist aufspürbar, wenn das Vorzeichen redundant kodiert ist. Dazu wechseln wir die Basis; statt $|0\rangle$ und $|1\rangle$ verwenden wir nun

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{und} \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{6.22}$$

Wir kodieren das logische Qubit wieder in drei physikalischen Qubits, aber mit den neuen Basisvektoren $|+\rangle$ und $|-\rangle$

$$|\bar{0}\rangle = |+, +, +\rangle \quad \text{und} \quad |\bar{1}\rangle = |-, -, -\rangle. \tag{6.23}$$

Da die Basisvektoren $|0\rangle$ und $|1\rangle$ zu gleichen Anteilen in $|+\rangle$ und $|-\rangle$ vorkommen, können wir nun keine Bitflip-Fehler mehr feststellen. (Wenn wir allerdings bereit wären, eine noch höhere Redundanz in Kauf zu nehmen, könnten wir beide Verfahren kombinieren.)

Die Korrektur eines Phasenfehlers verläuft nach dem gleichen Prinzip wie beim Bitflip-Fehler. Wir können beide Verfahren aufeinander abbilden mittels der Hadamard-Matrix

$$H = H^\dagger = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{6.24}$$

Sie vermittelt den Basiswechsel zwischen $|0\rangle, |1\rangle$ und $|+\rangle, |-\rangle$. Die zu (6.17) analoge Fehlerkorrektur für Phasenfehler lässt sich damit wie folgt als Schalt-

¹Unter der Voraussetzung, dass $|0\rangle$ und $|1\rangle$ durch die physikalischen Niveaus ausgedrückt werden, kann ein Phasenfehler z.B. durch äußere Felder entstehen, welche die Übergangsfrequenz zwischen den Qubit-Niveaus verändern oder durch Fluktuationen der Laserfrequenz, die als Phasenreferenz dient.

immer davon aus, dass $|0\rangle, |1\rangle$ die physikalischen Niveaus $|e\rangle, |g\rangle$ repräsentieren. Dies ist aber nicht zwingend. Wir können auch

$$|0\rangle_{\text{neu}} = \frac{1}{\sqrt{2}}(|e\rangle + |g\rangle) \quad \text{und} \quad |1\rangle_{\text{neu}} = \frac{1}{\sqrt{2}}(|e\rangle - |g\rangle) \quad (6.31)$$

wählen. Ausgehend von den bisher verwendeten Basisvektoren $|0\rangle_{\text{alt}} = |e\rangle$ und $|1\rangle_{\text{alt}} = |g\rangle$ können wir dies auch wie folgt beschreiben

$$|0\rangle_{\text{alt}}, |1\rangle_{\text{alt}} \xrightarrow{\text{Hadamard-Transformation}} |+\rangle, |-\rangle \xrightarrow{\text{Umbenennung}} |0\rangle_{\text{neu}}, |1\rangle_{\text{neu}}. \quad (6.32)$$

Wir wollen den Reigen der Impulssequenzen abschließen, indem wir zwei Realisierung des CNOTs in der neuen Basis angeben. Die Variante ohne Zuschauerqubits (5.20) gestaltet sich wie folgt

$$\begin{aligned} & X\left(-\frac{\pi}{2}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) \text{MS}_y\left(\frac{n-k}{4}\pi\right) Z_{1\dots k}(\pi) \\ & Y\left(\frac{n+k}{4}\pi\right) \text{MS}_y\left(-\frac{\pi}{4}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right). \end{aligned} \quad (6.33)$$

Mit Zuschauerqubits (5.25) lautet die Impulssequenz

$$\begin{aligned} & X\left(-\frac{\pi}{2}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) Y\left(-\frac{k}{4}\pi\right) \text{MS}_y\left(-\frac{\pi}{8}\right) Z_{k+1\dots k+n}(\pi) \\ & Y\left(\frac{k+n}{4}\pi\right) \text{MS}_y\left(\frac{\pi}{8}\right) Z_{1\dots k}(\pi) \text{MS}_y\left(-\frac{\pi}{8}\right) Y\left(-\frac{n}{4}\pi\right) \\ & Z_{k+1\dots k+n}(\pi) \text{MS}_y\left(\frac{\pi}{8}\right) Z_{1\dots k}\left(-\frac{\pi}{2}\right) X\left(\frac{\pi}{2}\right). \end{aligned} \quad (6.34)$$

6.5 Algorithmische Umsetzung

Bisher haben wir gesehen, wie eine simple Quantenfehlerkorrektur aussehen kann und wie man die dazu nötigen unitären Operatoren in einfache Impulssequenzen zerlegen kann. Jetzt wollen wir uns der Frage zuwenden, wie man diese Impulssequenzen auffindet, da unser altes Verfahren nicht mehr ausreichend ist. Rekapitulieren wir kurz den Optimierungsvorgang. Wir suchen das Maximum einer Funktion, die sich aus der *Gewinnfunktion* und der *Strafffunktion* ergibt. Die Strafffunktion beschreibt die Nebenbedingungen und wurde in Abschnitt 5.2 ausführlich besprochen. Die Gewinnfunktion hingegen verkörpert das primäre Optimierungsziel. Dies ist der Punkt, bei dem wir nachbessern müssen. Für sämtliche Impulssequenzen des Kapitel 5 reichte noch die einfache Gewinnfunktion (2.3)

$$|\langle U_{\text{Real}} | U_{\text{Ziel}} \rangle|^2 = |\text{tr}(U_{\text{Real}} \cdot U_{\text{Ziel}}^\dagger)|^2.$$

U_{Real} beschreibt die Gatteroperation, die sich für eine bestimmte Impulssequenz ergibt. U_{Ziel} beschreibt die Operation, die erreicht werden soll. Hier tut

sich nun unser Problem auf. U_{Ziel} ist nicht mehr fest vorgegeben. Die Schaltbilder (6.10) und (6.18) enthalten die nicht bestimmte Operation U_{beliebig} . Diese hatten wir eingeführt, da der Endzustand der Hilfsqubits für uns nicht von Bedeutung ist.

Bei dem letzten Satz muss man allerdings Vorsicht walten lassen. Der Endzustand der Hilfsqubits ist uns nämlich nur solange egal, wie er nichts über den Zustand des logischen Qubits verrät. Datenqubits und Hilfsqubits dürfen nicht miteinander verschränkt sein. Dies ist immer dann der Fall, wenn sich der Endzustand des Gesamtsystems als direktes Produkt aus Datenqubit(s) und Hilfsqubits schreiben lässt

$$|\text{Gesamtsystem}\rangle = |\text{Datenqubits}\rangle \otimes |\text{Hilfsqubits}\rangle_{\text{egal}}. \quad (6.35)$$

Aus diesem Kriterium wollen wir jetzt eine Gewinnfunktion für die Optimierung ableiten. Wir betrachten dazu die Fünf-Qubit-Variante mit drei Datenqubits und zwei Hilfsqubits (6.10). Wir können drei fehlerhafte Eingangszustände (Bitflip auf dem ersten, zweiten oder dritten Datenqubit) auf den fehlerfreien Zustand abbilden. Weiterhin wird der fehlerfreie Eingangszustand selbstverständlich auch wieder auf den fehlerfreien Zustand abgebildet. Damit können wir die Fehlerkorrektur durch die folgenden vier Zuweisungen beschreiben

1. $(\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle) \otimes |0, 0\rangle \rightarrow (\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle) \otimes |\text{H.Q. } 0\rangle$
2. $(\alpha |1, 0, 0\rangle + \beta |0, 1, 1\rangle) \otimes |0, 0\rangle \rightarrow (\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle) \otimes |\text{H.Q. } 1\rangle$
3. $(\alpha |0, 1, 0\rangle + \beta |1, 0, 1\rangle) \otimes |0, 0\rangle \rightarrow (\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle) \otimes |\text{H.Q. } 2\rangle$
4. $(\alpha |0, 0, 1\rangle + \beta |1, 1, 0\rangle) \otimes |0, 0\rangle \rightarrow (\alpha |0, 0, 0\rangle + \beta |1, 1, 1\rangle) \otimes |\text{H.Q. } 3\rangle$

wobei $|\text{H.Q. } 0 \dots 3\rangle$ die jeweiligen Endzustände der beiden Hilfsqubits beschreibt. Wir können das Gatter auch in Form einer Matrix beschreiben (Wir beschränken uns auf die erlaubten Eingangszustände)

$$\begin{aligned} U = & |0, 0, 0\rangle |\text{H.Q. } 0\rangle \langle 0, 0, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 0\rangle \langle 1, 1, 1| \langle 0, 0| \\ & + |0, 0, 0\rangle |\text{H.Q. } 1\rangle \langle 1, 0, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 1\rangle \langle 0, 1, 1| \langle 0, 0| \\ & + |0, 0, 0\rangle |\text{H.Q. } 2\rangle \langle 0, 1, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 2\rangle \langle 1, 0, 1| \langle 0, 0| \\ & + |0, 0, 0\rangle |\text{H.Q. } 3\rangle \langle 0, 0, 1| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 3\rangle \langle 1, 1, 0| \langle 0, 0| \end{aligned} \quad (6.36)$$

Dies beschreibt das *perfekte* Gatter zur Fehlerkorrektur. Während der Optimierungsphase werden wir es aber mit Gatteroperationen zu tun haben, die noch nicht perfekt sind. Hier sind Datenqubits und Hilfsqubits noch miteinander verschränkt. Diese Gatteroperationen lassen sich ebenfalls durch eine

Matrix beschreiben, aber wegen der Verschränkung müssen wir auf acht statt auf vier Endzustände für die Hilfsqubits zurückgreifen.

$$\begin{aligned}
U &= |0, 0, 0\rangle |\text{H.Q. } 0A\rangle \langle 0, 0, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 0B\rangle \langle 1, 1, 1| \langle 0, 0| \\
&+ |0, 0, 0\rangle |\text{H.Q. } 1A\rangle \langle 1, 0, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 1B\rangle \langle 0, 1, 1| \langle 0, 0| \\
&+ |0, 0, 0\rangle |\text{H.Q. } 2A\rangle \langle 0, 1, 0| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 2B\rangle \langle 1, 0, 1| \langle 0, 0| \\
&+ |0, 0, 0\rangle |\text{H.Q. } 3A\rangle \langle 0, 0, 1| \langle 0, 0| + |1, 1, 1\rangle |\text{H.Q. } 3B\rangle \langle 1, 1, 0| \langle 0, 0| \\
&+ \hat{Z}_{\text{Rest}}.
\end{aligned} \tag{6.37}$$

Der Ausdruck \hat{Z}_{Rest} in obiger Formel beschreibt Zuweisungen erlaubter Eingangszustände auf falsche Datenqubit-Endzustände.

Wir müssen im Folgenden zwei Optimierungsziele im Auge behalten: Zum Einen müssen alle mit \hat{Z}_{Rest} bezeichneten Zuweisungen verschwinden und zum Anderen müssen wir die Verschränkung zwischen den Datenqubits und den Hilfsqubits aufheben. Letzteres ist gleichbedeutend mit

$$\begin{aligned}
|\text{H.Q. } 0A\rangle &= |\text{H.Q. } 0B\rangle & |\text{H.Q. } 1A\rangle &= |\text{H.Q. } 1B\rangle \\
|\text{H.Q. } 2A\rangle &= |\text{H.Q. } 2B\rangle & |\text{H.Q. } 3A\rangle &= |\text{H.Q. } 3B\rangle.
\end{aligned} \tag{6.38}$$

Aber auch für das Verschwinden von \hat{Z}_{Rest} gibt es ein einfaches Kriterium. Für die acht oben aufgeführten Endzustände der Hilfsqubits muss gelten

$$\begin{aligned}
\| |\text{H.Q. } k\rangle \| &= 1 \quad \text{mit} \quad k = 0, 1, 2, 3 \\
\Rightarrow \| |0, 0, 0\rangle |\text{H.Q. } kA\rangle \| &= \| |1, 1, 1\rangle |\text{H.Q. } kB\rangle \| = 1.
\end{aligned} \tag{6.39}$$

Da jede Impulssequenz eine unitäre Transformation darstellt und als solche normerhaltend wirkt, bleibt einfach kein Platz mehr für fehlerhafte Zuweisungen, wenn (6.39) erfüllt ist. Die Bedingungen (6.38) und (6.39) sind äquivalent zur Forderung

$$\begin{aligned}
\mathfrak{G} &:= \langle \text{H.Q. } 0A | \text{H.Q. } 0B \rangle + \langle \text{H.Q. } 1A | \text{H.Q. } 1B \rangle \\
&+ \langle \text{H.Q. } 2A | \text{H.Q. } 2B \rangle + \langle \text{H.Q. } 3A | \text{H.Q. } 3B \rangle \\
&= 4.
\end{aligned} \tag{6.40}$$

Der maximale Realwert jedes der vier Skalarprodukte ist eins. Dieser wird nur dann erreicht, wenn Bra- und Ketvektor jeweils vom Betrag her eins sind und durch hermitische Konjugation auseinander hervorgehen. Vier ist daher der maximale Realwert von \mathfrak{G} . Somit können wir $\text{real}(\mathfrak{G})$ als Gewinnfunktion verwenden und nach einem Maximum dieser Funktion suchen.

Auf den ersten Blick mag das Fehlen von U_{Ziel} oder eines entsprechenden Ausdrucks in der Gewinnfunktion (6.40) verwundern. Dies ist aber nur scheinbar der Fall. Das Gatter wird durch eine 32×32 -Matrix beschrieben. Die $|\text{H.Q. } k\rangle$ sind 1×4 -Untermatrizen. Nur die Kombination der richtigen Untermatrizen ergibt eine brauchbare Gewinnfunktion. Es ist diese Kombination der richtigen Untermatrizen, über welche die zu U_{Ziel} analoge Information in die Bewertung eingeht.

6.6 Optimieren in der Praxis

In diesem, dem vorletzten Abschnitt der Diplomarbeit geht es nicht mehr um Fragen der Fehlerkorrektur, sondern um das Auffinden und Optimieren von Impulssequenzen im Allgemeinen.

Aus der Impulssequenz für das CNOT und den Sequenzen für lokale Drehungen können wir zu jedem Gatter eine Impulssequenz konstruieren, die dieses Gatter erzeugt. Die so konstruierte Impulssequenz wird aber fast immer weit komplizierter ausfallen als die optimale Impulssequenz. Das Auffinden der optimalen Impulssequenz wird hingegen mit zunehmender Komplexität der Gatter immer schwieriger und ab einem gewissen Grad praktisch unmöglich. Wie schon in Abschnitt 5.3.2 erwähnt, wird man in der Praxis ein komplexes Gatter in kleinere, optimal lösbare Untereinheiten zerlegen und die Gesamtlösung aus diesen Untereinheiten aufbauen. Die so erhaltene Impulssequenz kann man anschließend nochmals einem Optimierungsprozess zuführen, um weitere Impulse einzusparen. Die Einsparungen können bei diesem letzten Schritt überraschend groß ausfallen. Bei der Fünf-Qubit-Fehlerkorrektur konnte die Anzahl der Impulse auf ein Viertel (!) des ursprünglichen Wertes reduzieren (6.11) werden. Dies ist auch der einzige Fall, bei dem es nötig war, den Umweg über eine zusammengesetzte Lösung zu gehen. Das Fehlersyndrom der Fünf-Qubit-Fehlerkorrektur (6.5) musste dabei anfangs noch aus den CNOTs aufgebaut werden. Mit zunehmender Verfeinerung des Algorithmus konnte auch dieses ohne Vorgabe einer zusammengesetzten Lösung errechnet werden. Diese Konstruktion „aus dem Nichts“ – wo möglich – erweist sich als vorteilhaft. Eine vorgegebene, zusammengesetzte Lösung entspricht nämlich einem Maximum der Gewinnfunktion und somit meist auch annähernd einem lokalen Maximum der Kombination aus Gewinnfunktion und Straffunktion (Abschnitt 2.4). Einmal in einem lokalen Maximum gefangen, sind viele Optimierungsalgorithmen unfähig, dieses wieder zu verlassen. Erst durch das in Abschnitt 2.5 beschriebenen Verfahrens des Simulated Annealing wurde es möglich, lokalen Maxima zu entkommen. Der vorgeprägten Struktur der Lösung aber gänzlich zu entrinnen, gelingt äußerst selten.

Als Beispiel für eine zusammengesetzte Lösung wollen wir uns die drei Toffoli-Gatter aus der Fünf-Qubit-Fehlerkorrektur (6.8) vornehmen



Das in (5.7) gegebene Toffoli-Gatter funktioniert nur, wenn es keine Zuschauerqubits gibt - also auf genau drei Qubits. Das Toffoli-Gatter auf fünf

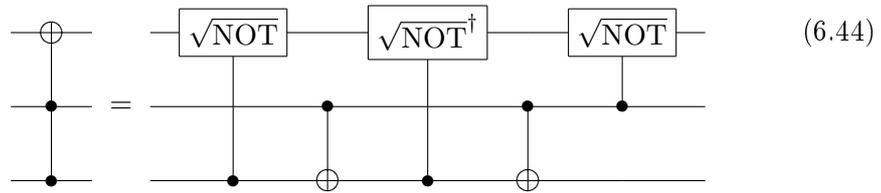
Qubits werden wir mittels des $C\sqrt{\text{NOT}}$ konstruieren. Bei der Definition des $C\sqrt{\text{NOT}}$ gibt es zwei Möglichkeiten für das Vorzeichen des Exponenten. Wir wählen³

$$\sqrt{\text{NOT}} := \exp\left(+i\frac{\pi}{4}(\sigma_x - \mathbb{1})\right). \quad (6.42)$$

Es gilt

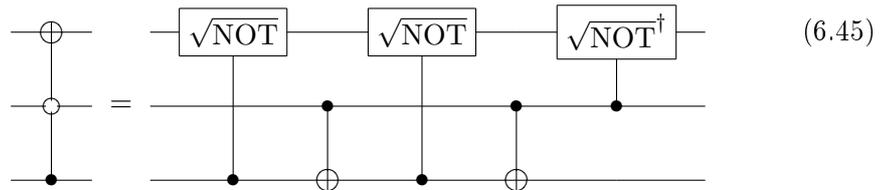
$$\begin{aligned} \sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} &= \sqrt{\text{NOT}}^\dagger \cdot \sqrt{\text{NOT}}^\dagger = \text{NOT} \\ \sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}}^\dagger &= \sqrt{\text{NOT}}^\dagger \cdot \sqrt{\text{NOT}} = \mathbb{1}. \end{aligned} \quad (6.43)$$

Damit können wir das Toffoli-Gatter wie folgt zu konstruieren [11]

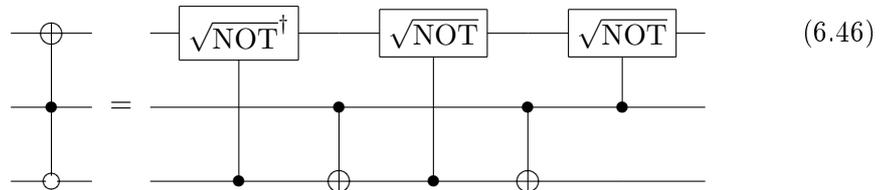


wie man leicht überprüft, indem man für die Kontrollqubits die Basisvektoren $|0, 0\rangle$, $|0, 1\rangle$, $|1, 0\rangle$ und $|1, 1\rangle$ einsetzt.

Die bei $|0\rangle$ schaltenden Toffoli-Gatter (mit $\text{---}\circ\text{---}$ statt $\text{---}\bullet\text{---}$) werden ähnlich erzeugt

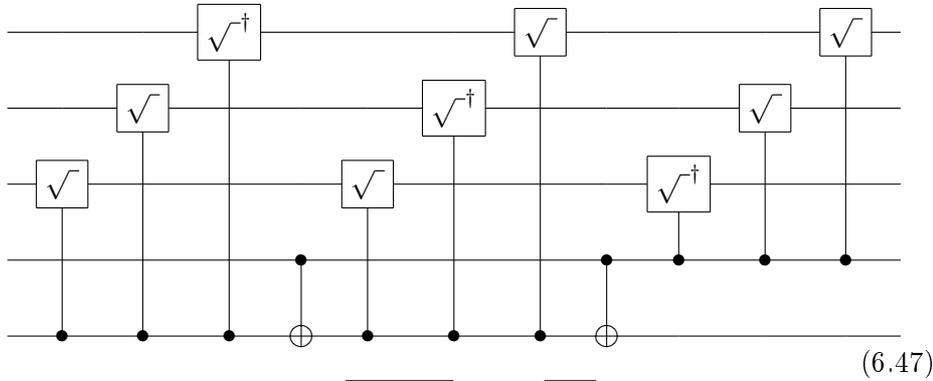


und



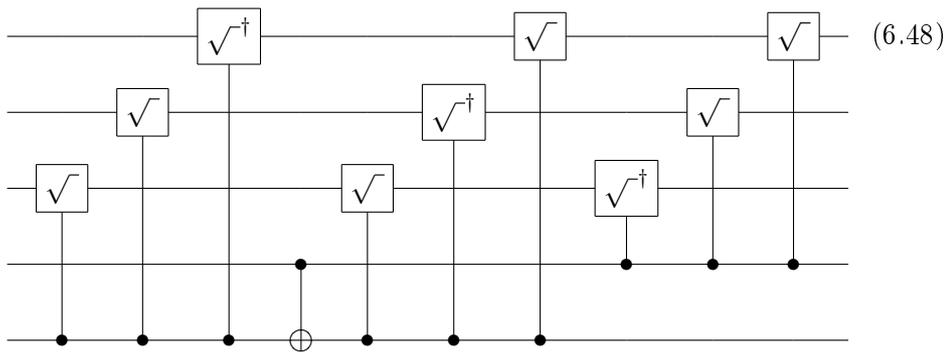
³Für die Gültigkeit der folgenden Schaltbilder ist die Wahl des Vorzeichens unerheblich. Sie wird erst wieder wichtig, wenn wir die dazugehörigen Impulssequenzen konstruieren.

Damit können wir die drei Toffoli-Gatter in (6.41) wie folgt aufbauen

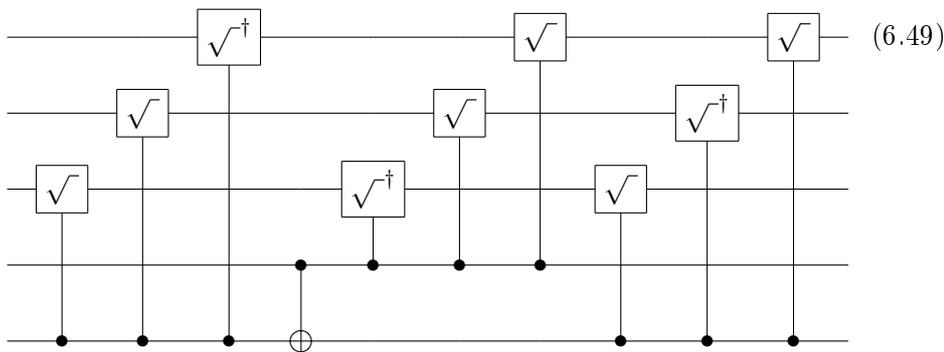


wobei wir zur Platzersparnis $\boxed{\sqrt{\text{NOT}}}$ durch $\boxed{\sqrt{}}$ ersetzt haben.

Die Aufgabe des rechten CNOT in (6.47) besteht nur darin, das unterste Qubit wieder in den Ausgangszustand zu versetzen. Da uns bei der Fehlerkorrektur der Endzustand der Hilfsqubits vollkommen egal ist (6.10), können wir dieses CNOT auch weglassen.



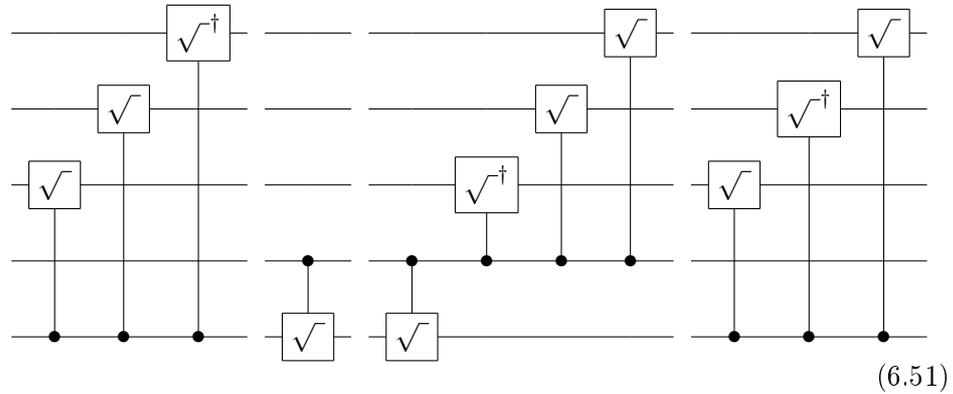
Nach dem Wegfall des CNOTs können die letzten sechs $C\sqrt{\text{NOT}}$ s beliebig vertauscht werden, was wir nutzen wollen



Als nächstes ersetzen wir das verbliebene CNOT gemäß

$$\text{CNOT} = C\sqrt{\text{NOT}} \cdot C\sqrt{\text{NOT}}. \quad (6.50)$$

Jetzt besteht unser Gatter nur noch aus elf $C\sqrt{\text{NOT}}$, die wir in vier funktionalen Einheiten zusammenfassen können (alles, was auf einer ununterbrochenen Linie liegt, gehört zu einer Einheit)



In jedem der vier Einheiten ist jedes Qubit entweder Kontroll-, Ziel- oder Zuschauerqubit, wobei die dritte Einheit von links sogar ohne Zuschauerqubits auskommt. Dies ähnelt den CNOT-Strukturen (5.20) und (5.25). In Abschnitt 5.3.3 wurde beschrieben, wie man aus einer CNOT-Impulssequenz die entsprechende $C\sqrt{\text{NOT}}$ -Impulssequenz erstellt. Wir haben es hier allerdings mit einer kleinen zusätzlichen Komplikation zu tun: Wir haben eine Durchmischung von $C\sqrt{\text{NOT}}$ s und $C\sqrt{\text{NOT}}^\dagger$ s. Die Anpassung der $C\sqrt{\text{NOT}}$ -Impulssequenzen sieht so aus, dass hinter den beiden $Z_{\text{Kontrollqubit}}(\pm\frac{\pi}{2})$ -Impulsen in (5.21) bzw. (5.24) ein $Z(\pi)$ -Impuls auf dasjenige Qubit eingefügt wird, das Ziel des $C\sqrt{\text{NOT}}^\dagger$ s ist.

Dies beschreibt die geschickteste zusammengesetzte Lösung, die im Rahmen dieser Diplomarbeit gefunden wurde. Interessanterweise ist die beste Impulssequenz (6.11) zur Fünf-Qubit-Fehlerkorrektur aber aus der Optimierung einer weitaus schlechteren zusammengesetzten Lösung als der soeben beschriebenen erwachsen. Der beste Ansatz muss nicht unbedingt zur besten Lösung führen. Dies ist verständlich, wenn wir uns daran erinnern, dass eine zusammengesetzte Lösung einem lokalen Maximum des Optimierungsprozesses entspricht. Ist dieses lokale Maximum zu ausgeprägt, so wird der Optimierungsprozess immer innerhalb der vorgegebenen Strukturen agieren und nur kleinere Verbesserungen schaffen. Aus einem schwachen Maximum mag die „Befreiung“ sehr viel erfolgreicher gelingen. Die nahe liegende Vermutung, man könnte durch eine gezielte Verschlechterung einer zusammengesetzten Lösung dem Computer unter die Arme greifen, kann hier leider nicht bestätigt werden.

Kapitel 7

Zusammenfassung und Ausblick

In der hier vorliegenden Arbeit wurden Methoden der optimal control theory vorgestellt und auf zwei spezielle Fragestellungen der Ionenfallen-Physik angepasst bzw. erweitert. Zum Einen ging es um die Verbesserung der Güten des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters auf zwei Ionen, zum Anderen um das Auffinden komplexer Impulssequenzen.

Optimierung des Mølmer-Sørensen- und des $\sigma_z \otimes \sigma_z$ -Gatters

Es wurde erfolgreich demonstriert, wie allein durch Anpassung der Intensität des Lasers eine signifikante Verbesserung der Gattergüten erreicht werden kann. Da beim aktuellen Verfahren der Impulsformung die Aufweichung der Impulsflanken ebenfalls durch ein stufenweises Anpassen der Intensität erzeugt wird, sind zur Umsetzung der hier gemachten Vorschläge keine neuen Elemente notwendig. Zum Zeitpunkt der Niederschrift gibt es leider noch keine experimentellen Erfahrungen, inwieweit sich die hier angestellten Berechnungen in der Praxis bewähren.

Für zukünftige Projekte könnten zwei einfache, aber ressourcenintensive Erweiterungen von Interesse sein: Zum Einen könnten weitere Schwingungsmoden berücksichtigt werden, um eine akkuratere Beschreibung des Systems zu erreichen. Zum Anderen wäre eine Erweiterung auf mehr als zwei Ionen angebracht, da derartige Impulse z.B. im zweiten Teil der Diplomarbeit benötigt werden. Diesbezügliche ist die Hoffnung zu hegen, dass der zeitliche Verlauf für die Intensitäten der Laserstrahlen, den man aus der Optimierung eines Gatters auf wenigen Ionen gewinnt, auch zu vorteilhaften Güten bei vielen Ionen führt. Sollte dies nicht der Fall sein, könnte der mit der Anzahl der Ionen exponentiell ansteigende Ressourcenbedarf zu einem Problem werden.

Das Vorgehen bei der hier betriebene Gatteroptimierung entspricht weitgehend dem Standard. Neu entwickelt wurde in dieser Arbeit hingegen das Verfahren zur Berechnung der Zeitentwicklung. Die Übertragung dieses An-

satzes auf andere Probleme könnte sich allerdings problematisch gestalten, da die Durchführung des am Ende von Abschnitt 3.3 eingeführten Zeitshifts an eine bestimmte Form der Zeitabhängigkeit des Hamilton-Operators gebunden ist. Sehr viel leichter sollte es hingegen fallen, die ebenfalls in dieser Arbeit entwickelte Variante des Simulated Annealing (Abschnitt 2.5) zu übernehmen. Hierbei handelt es sich um einen leicht zu programmierenden, wenig spezifischen Algorithmus, der ein wirksames Mittel gegen ein häufig auftretendes Problem zu sein verspricht.

Impulssequenzen

Im zweiten Teil dieser Diplomarbeit wurde ein erfolgversprechendes Verfahren vorgestellt, das beliebige Quantengatter durch Sequenzen verschiedener elementarer Impulse zu verwirklichen vermag. Da dieses Verfahren es uns gestattet, das CNOT und beliebige lokale Rotationen zu erzeugen, können wir stets dem konventionellen Ansatz über Quantenschaltkreise folgen. Darüber hinaus eröffnet sich uns aber auch die Möglichkeit, Operationen geschickt zusammenzufassen und dadurch zu sehr viel eleganteren Lösungen zu gelangen. Dies wird in beeindruckender Weise anhand der Impulssequenzen zur Quantenfehlerkorrektur demonstriert. Hier wird insbesondere von der Möglichkeit profitiert, den unbestimmten Operator U_{beliebig} einzufügen ((6.10) und (6.17)).

Zum Zeitpunkt der Schriftsetzung laufen in unserer Gruppe die Vorbereitungen für eine experimentelle Überprüfung dieses Verfahrens. Damit sollten bald genauere Kenntnisse über das Fehler-Budget verfügbar sein, die in die Bewertung der Impulssequenzen einfließen können. Sollte sich das hier beschriebene Verfahren als zukunftsfruchtig erweisen, bietet sich z.B. die Erstellung einer „Impulsbibliothek“ für häufig vorkommende Operatorkombinationen an. Ein weiterer interessanter Punkt ist das mögliche Zusammenspiel dieses Verfahrens mit segmentierten Fallen [30]. Wie mehrfach demonstriert, können Zuschauerqubits (siehe Abschnitt 5.3) die Impulssequenzen erheblich verkomplizieren. Durch das gezielte Verschieben von Ionen in segmentierten Fallen bestünde die Möglichkeit, störende Zuschauerqubits zu entfernen. Weiterhin muss sich noch erweisen, mit wie vielen Ionen gleichzeitig ein Mølmer-Sørensen-Gatter sich in hinreichender Güte verwirklichen lässt. In zukünftigen Anwendungen käme den segmentierten Fallen dann die Aufgabe der richtigen Portionierung der Ionen zu.

Auf den ersten Blick scheint das Problem, diskrete Impulssequenzen auffinden zu wollen, nur wenig Gemeinsamkeiten mit der optimal control theory zu haben. Umso erstaunlicher darf es daher anmuten, mit wie wenig Programmieraufwand der hier entwickelte Algorithmus diese Elemente erfolgreich zu verknüpfen vermochte. Dies gibt Anlass zur Hoffnung, dass auch andere diskrete Probleme so einer Lösung zugeführt werden können.

Literaturverzeichnis

- [1] R. P. Feynman, *Simulating Physics with Computers*, Int. J. Theor. Phys. **21**, 467 (1982).
- [2] D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. R. Soc. London A **400**, 97–117 (1985).
- [3] P. W. Shore, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, arXiv:quant-ph/9508027v2 (1995).
- [4] J. I. Cirac, P. Zoller, *Quantum Computations with Cold Trapped Ions*, Phys. Rev. Lett. **74**, 4091 (1995).
- [5] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, D. J. Wineland, *Demonstration of a Fundamental Quantum Logic Gate*, Phys. Rev. Lett. **75**, 4714 - 4717 (1995).
- [6] F. Schmidt-Kaler, H. Häffner, M. Riebe, S. Gulde, G. P. T. Lancaster, T. Deuschle, C. Becher, C. F. Roos, J. Eschner, R. Blatt, *Realization of the Cirac-Zoller controlled-NOT quantum gate*, Nature **422**, 408 (2003).
- [7] E. Knill, *Quantum computing with realistically noisy devices*, Nature **434**, 39–44 (2005).
- [8] G. J. Milburn, *Simulating nonlinear spin models in an ion trap*, arXiv:quant-ph/9908037v1 (1999).
- [9] K. Mølmer, A. Sørensen, *Entanglement and quantum computation with ions in thermal motion*, Phys. Rev. A **62**, 022311 (2000).
- [10] J. Benhelm, G. Kirchmair, C. F. Roos, R. Blatt, *Towards fault-tolerant quantum computing with trapped ions*, Nature Physics **4**, 463 - 466 (2008).
- [11] M. A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000).

- [12] D. E. Browne, H. J. Briegel, *One-way Quantum Computation - a tutorial introduction*, arXiv:quant-ph/0603226v2, (2006).
- [13] D. Leibfried, R. Blatt, C. Monroe, D. J. Wineland, *Quantum dynamics of single trapped ions*, Rev. Mod. Phys. **75**, 281 (2003).
- [14] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, D. M. Meekhof, *Experimental Issues in Coherent Quantum-State Manipulation of Trapped Atomic Ions*, J. Res. Natl. Inst. Stand. Technol. **103**, 259 (1998).
- [15] C. Cohen-Tannoudji, *Atom-Photon-Interactions*, Wiley (1992).
- [16] R. Loudon, *The Quantum Theory of Light*, Oxford Science Publications (1983).
- [17] D. F. Walls, G. J. Milburn, *Quantum Optics*, Springer-Verlag (2008).
- [18] C. F. Roos, *Ion trap quantum gates with amplitude-modulated laser beams*, New J. Phys. **10**, 013002 (2008).
- [19] P. J. Lee, K.-A. Brickman, L. Deslauriers, P. C. Haljan, L.-M. Duan, C. Monroe, *Phase control of trapped ion quantum gates*, J. Opt. B: Quantum Semiclass. Opt. **7**, S371 (2005).
- [20] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, S. J. Glaser, *Optimal Control of Coupled Spin Dynamics: Design of NMR Pulse Sequences by Gradient Ascent Algorithms*, J. Magn. Reson. **172**, 296-305 (2005).
- [21] J. Werschnik, E. K. U. Gross, *Quantum optimal control theory*, J. Phys. B: Mol. Opt. Phys. **40** R175-R211 (2007).
- [22] J. P. Palao, R. Kosloff, *Optimal control theory for unitary transformations*, Phys. Rev. A **68** 062308 (2003).
- [23] W. H. Pree, W. T. Vetterling, S. A. Teukolsky, B. P. Flannery, *Numerical Recipes in C++*, Cambridge University Press (2002).
- [24] P. Venkataraman, *Applied Optimization with MATLAB Programming*, John Wiley & Sons (2002).
- [25] W. Magnus, *On the exponential solution of differential equations for a linear operator*, Commun. Pure Appl. Math. **7** 649 (1954).
- [26] S. Klarsfeld, J. A. Oteo, *Recursive generation of higher-order terms in the Magnus expansion*, Phys. Rev. A **39**, 3270 - 3273 (1989).
- [27] D. P. Burum, *Magnus expansion generator*, Phys. Rev. B. **24** 3684 (1981).

- [28] D. Leibfried *et al*, *Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate*, Nature **422** 412 (2003).
- [29] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, arXiv:quant-ph/9705052v1 (1997).
- [30] R. Reichle, D. Leibfried, R. B. Blakestad, J. Britton, J. D. Jost, E. Knill, C. Langer, R. Ozeri, S. Seidelin, D. J. Wineland, *Transport dynamics of single ions in segmented microstructured paul trap arrays*, Fortschr. Phys. **54** 666 (2006).

Danksagung

Ich bin vielen Menschen zu Dank verpflichtet, möchte mich hier aber auf das physikalische Umfeld beschränken. Namentlich gilt mein Dank an erster Stelle Professor Heyszenau und Professor Blatt – nicht nur, weil sie diese Arbeit bewerten werden. Professor Heyszenau hatte immer ein offenes Ohr für mich und ermutigte mich wiederholt, eine externe Diplomarbeit zu schreiben. Professor Blatt möchte ich insbesondere für das Vertrauen danken, das er in mich gesetzt hat, als er mich in seine Arbeitsgruppe aufnahm und mir damit die Chance bot, an einem interessanten und aktuellen Thema zu arbeiten. Meinem Betreuer Christian Roos danke ich für die wertvollen physikalischen Erklärungen, die mir ein Verständnis der Materie wesentlich vereinfacht haben. Hartmut Häffner – obwohl offiziell nicht mit meiner Betreuung beauftragt – musste in der Abwesenheit von Christian Roos oft als Ersatz herhalten, wofür ich ihm danken möchte. Zum Schluss möchte ich noch Kihwan Kim danken, der sich immer sehr für meine Arbeit interessiert hat und mich auch auf das Drei-Qubit-Schema zur Quantenfehlerkorrektur aufmerksam gemacht hat.

Index

- U_{beliebig} , 106, 109
 - algorithmische Umsetzung, 112–114
- $\sigma_z \otimes \sigma_z$ -Gatter
 - Interpretation, 16
 - Optimierungsziel, 57
 - phys. Erklärung, 14
 - Resultate, 71
- $C_{x,y,z}\text{NOT}$
 - Definition, 91
 - Impulssequenzen, 93–94
- Adiabatheorem, 61
- Ancilla, 91
- bichromatisches Laserfeld, 9
- Bitflip-Fehler, 103–109
- Bloch-Vektor-Bild
 - Impulssequenzen, 83
- Boltzmannverteilung, 33
- Burum, 47
- $C\sqrt{\text{NOT}}$, 116
- CNOT
 - ‘globales’ CNOT, 90
 - aus $\sigma_z \otimes \sigma_z$ -Gatter, 18
 - $C\sqrt{\text{NOT}}$, 116
 - Definition, 4
 - Impulssequenzen, 89–91
 - in $(|+\rangle, |-\rangle)$ -Basis, 112
- Datenqubit, 92
- Drehwellennäherung, 9
- Dysonentwicklung
 - allgemeine Form, 39
 - angepasste Form, 39
 - numerische Integration, 43
- Rekursionsformel, 45, 46
- verkürzte Summation, 50
- Vorteile der D., 39
- Entropie
 - bei Quantenfehlerkorrektur, 106
- EVO-Algorithmus, 25
- Extrema
 - lokal und global, 32
 - numerische Suche, 23
- Fallenfrequenz, 7
- Fehlertoleranz, 59–66
- Güte, 70
- Gatteroptimierung, 55
- Gatterzeiten
 - $\sigma_z \otimes \sigma_z$ -Gatter, 16
 - Mølmer-Sørensen, 18
- geometrische Phase, 17
- Gewinnfunktion, 23
 - Definition, 22
 - Gatteroptimierung, 57
 - Impulssequenzen, 112–114
- Gleichverteilung, 35
- Gradientenverfahren, 24
- Hadamard-Matrix, 18
- Hilfsqubit, 91
- Impulse
 - globales X,Y, 84
 - lokales Z, 84
 - MS, 83
- Impulsformung, 55
- Impulssequenzen
 - $C_{x,y}\text{NOT}$, 93–94

- Analyse, 97–101
- CNOT, 89–91
 - in $(|+\rangle, |-\rangle)$ -Basis, 112
- Definition, 83
- Gewinnfunktion, 112–114
- kontrolliertes σ_z -Gatter, 96
- lokale Drehungen, 88
- Optimierung
 - U_{beliebig} , 112–114
 - Ansatz, 85, 88
- Quantenfehlerkorrektur
 - Bitflip 3 Ionen, 109
 - Kodierung 3 Ionen, 111
 - Korrektur 3+2 Ionen, 107
 - Phasenfehler 3 Ionen, 111
 - Syndrom 3+2 Ionen, 105
- Reihenfolge der Impulse, 84
- Toffoli-Gatter (3 Ionen) , 84
- universelle Gatter, 88
- Verschränkung mit P., 97
- verwendete Impulse, 83
- Ion-Laser-Wechselwirkung, 7
- Ionenfalle, 6
- Kodierung, 108
- konjugierter Gradient, 25
- kontrolliertes σ_z -Gatter, 18, 96
 - Impulssequenzen, 96
- Kostenfunktion, 22
- Lagrangescher Multiplikator, 28
- Lamb-Dicke-Faktor
 - Definition, 8
- Lamb-Dicke-Näherung, 11
- logisches Qubit, 103
- lokale Drehungen
 - mit Impulssequenzen, 88
- Mølmer-Sørensen-Gatter, 83
 - Optimierungsziel, 57
 - phys. Erklärung, 12
 - Resultate, 78
- Magnusentwicklung, 39
 - aus Dysonentwicklung, 47
- Matrixdarstellung, 5
- Näherungsparabel, 27, 30, 33
- Nebenbedingungen
 - allgemein, 28
 - Impulssequenzen, 86
 - MS- & $\sigma_z \otimes \sigma_z$ -Gatter, 56
- NOT
 - Variationen zum NOT, 94
- numerische Integration, 47
- OCT, 21
- Offset, 69
 - Resultate, 77
- Operator, 3
 - Darstellung als
 - Matrix, 5
 - Quantenschaltkreis, 4
- optimal control theory, 21
- Optimierung
 - allgemein, 21
 - Impulssequenzen
 - U_{beliebig} , 112–114
 - Ansatz, 85, 88
 - Ergebnisse, *siehe* Impulssequenzen
 - MS- & $\sigma_z \otimes \sigma_z$ -Gatter
 - Ansatz, 56–59
 - Ergebnisse, 69–80
 - Fehlertoleranz, 59–66
 - Rahmenbedingungen, 55–56
 - Nebenbedingungen, 28
 - Simulated Annealing, 32
- Phasenabhängigkeit
 - von Gattern, 60
- Phasenfehler, 110–112
- Phasenraum, 17
- physikalisches Qubit, 103
- Quantencomputer
 - Einführung, 1
- Quantenfehlerkorrektur, 103–114
 - U_{beliebig} , 106, 109
 - algorithmische Umsetzung, 112–114
 - Bitflip-Fehler, 103–109

- Entropie, 106
- Grundlagen, 103
- Kodierung, 108
- Korrektur ohne Messung
 - 3 Ionen, 109, 111
 - 3+2 Ionen, 105
- Phasenfehler, 110–112
- Quanten-Zeno-Effekt, 104
- Syndrom, 104
 - 3+2 Ionen, 104
- Quantenschaltkreise, 4
- Qubit
 - Datenqubit, 92
 - Definition, 2
 - Hilfsqubit, 91
 - Kontrollqubit, 4
 - logisches, 103
 - physikalisches, 103
 - Zielqubit, 4
 - Zuschauerqubit, 90
- Rabifrequenz
 - Definition, 7
 - Fluktuationen, 64
- Recycling von Berechnungen, 38
- Redundanz, 103
- Ressourcenverbrauch, 80
- rotating-wave approximation, 9
- RWA, 9
- Schwingungsmoden, 6, 7
- segmentierte Fallen, 120
- Seitenband, 11
- Simulated Annealing, 32
 - neue Variante, 33
- Spinecho
 - Funktionsweise, 66–69
 - plus Offset, 69
 - Resultate, 77
- Straffunktion
 - allgemein, 29
 - für Impulssequenzen, 86
 - für MS- & $\sigma_z \otimes \sigma_z$ -Gatter, 56
- Syndrom
 - für Quantenfehlerkorrektur, 104
- Toffoli-Gatter
 - aus $C\sqrt{\text{NOT}}$, 116
 - Definition, 4
 - Impulssequenz
 - drei Ionen, 84
 - in Quantenfehlerkorrektur, 106, 108
 - Matrixdarstellung, 5
- Träger-Übergang, 11
- Tschebyschow-Polynome, 51
- universelle Gatter, 3
 - mit Impulssequenzen, 88
- Verschränkung
 - mit Impulssequenzen, 97
- Verstimmung, 9
- Vorzeichenflip, 110
- Wahrscheinlichkeitsverteilung, 34
- Wechselwirkungsbild, 8
 - Vorteile, 40
- Zeiteinheit, 69
- Zeitentwicklungsoperator, 9, 37
 - allgemeiner Ansatz, 25
- Zeitshift, 43
- Zielqubit, 4
- Zuschauerqubit
 - bei Impulssequenzen, 90
 - Definition, 90